Spanair

Transportista oficial de las Jornadas

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
SERVICIO DE PUBLICACIONES

IV Jornadas de Ingeniería Telemática

IV Jornadas de Ingeniería Telemática

Gran Canaria
15 al 17 de septiembre de 2003

Editores:
Álvaro Suárez Sarmiento
Elsa María Macías López
Carmen Nieves Ojeda Guerra

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
VICERRECTORADO DE ESTUDIOS Y CALIDAD DOCENTE

Sesión 1A

Modelado y control de tráfico (I)

Análisis de las Prestaciones del Acondicionador de Tráfico CBM en un Dominio DiffServ

María-Dolores Cano, Fernando Cerdán, Joan García-Haro, Josemaría Malgosa-Sanahuja
Departamento de Tecnología de la Información y las Comunicaciones
Universidad Politécnica de Cartagena
Campus Muralla del Mar s/n
30202 Cartagena
Teléfono: 968 325953 Fax: 968 32 59 73
E-mail: {mdolores.cano, fernando.cerdan, joang.haro, josem.malgosa}@upct.es

Abstract*. The Counters-Based Modified (CBM) traffic conditioner was introduced in a previous work as a feasible option to implement the Assured Forwarding (AF) service in DiffServ. In this paper we present an end-to-end performance analysis of TCP Reno sources that employ the CBM in a DiffServ domain. We present simulation results in a three-RIO-node topology under miscellaneous characteristics: different contract rates, heterogeneous RTT, co-existance of best-effort and AF sources, and eficience of CBM when some network node does not implement service differentiation. As shown in simulation results, it is possible to guarantee an AF service that ensures contracted target rates and performs a fair share of the excess bandwidth.*

1 Introducción

Los Servicios Diferenciados (*Differentiated Services*, DiffServ) han sido estandarizados como una de las soluciones más prometedoras a la hora de ofrecer Calidad de Servicio (*Quality of Service*, QoS) en las redes IP [1]. La arquitectura DiffServ hace uso de un esquema sencillo para proporcionar diferentes niveles de QoS en el que la complejidad permanece en los bordes de la red, intentando que los mecanismos empleados en el interior de la misma sean lo más sencillos posible.

La implementación de DiffServ se basa en el uso del byte DSCP (DiffServ *Code Point*) de la cabecera IP. En los nodos frontera o en la propia fuente de tráfico, los paquetes se marcarán, clasificarán y acondicionarán antes de entrar en la red con el fin de recibir un tratamiento particular en los nodos que atraviesen a lo largo de su camino. Este tratamiento que reciben los paquetes en los nodos interiores se conoce como *Per-Hop Behavior* (PHB). Actualmente existen dos PHB estandarizados por el IETF: el *Expedited Forwarding* (EF) PHB [2] y el *Assured Forwarding* (AF) PHB [3].

Los objetivos del servicio AF son asegurar un caudal (*throughput*) mínimo a cada fuente, que normalmente es la tasa contratada, también denominada CIR (*Committed Information Rate*); y además, permitir a las fuentes consumir más ancho de banda del contratado si la carga de la red es baja. El reparto del ancho de banda en exceso entre las diferentes fuentes se ha de realizar de modo justo, encontrándose dos definiciones para el término *justicia*. La primera define *justicia* como el reparto equitativo del ancho de banda en exceso entre todas las fuentes que componen el agregado. Mientras que la segunda definición, determina que un reparto justo del ancho de banda será aquel proporcional al CIR de cada fuente. En este trabajo, así como en la mayor parte de la literatura relacionada, se utiliza la primera definición, pues se asume que si se consigue un reparto equitativo del ancho de banda en exceso, pasar a un reparto proporcional dependerá únicamente del uso de un sistema de ponderación.

Para alcanzar los objetivos del servicio AF, los paquetes de cada flujo individual de tráfico se marcan como pertenecientes a una de las cuatro clases de tráfico AF. Como se detalla en [3], dentro de cada clase de tráfico AF un paquete puede pertenecer a tres niveles distintos de precedencia. En caso de congestión, el nivel de precedencia de un paquete determinará la importancia del mismo dentro de la clase AF a la que pertenece. Un nodo DiffServ que presente congestión descartará preferiblemente paquetes con un nivel de precedencia más alto, protegiendo así a los paquetes con un nivel de precedencia más bajo. A la hora de implementar un servicio AF dentro de la arquitectura DiffServ, se tendrá que definir por tanto qué tipo de funciones se van a utilizar para acondicionar el tráfico en los nodos frontera o en las fuentes de tráfico (marcar, clasificar, aplicar funciones policía, etc.) y cómo construir el AF PHB.

La introducción de RIO (RED (*Random Early Detection*) In y Out) [4] supuso un paso importante en el desarrollo de DiffServ. Este mecanismo que se usa para implementar el AF PHB, utiliza sólo dos niveles de precedencia dentro de cada clase AF. Los paquetes que se consideran dentro del perfil de tráfico de una fuente se marcarán como *in* y los que están fuera del perfil como *out*. Una vez un paquete queda marcado, el agregado de tráfico llega al *router* donde se aplica RIO. RIO es la combinación de dos algoritmos RED [5] con diferentes curvas de probabilidad de descarte, de tal manera que los paquetes *out* tienen más probabilidad de ser

eliminados. RIO utiliza una única cola FIFO (*First In First Out*) para servir ambos tipos de paquetes. La probabilidad de descartar un paquete *out* depende del número total de paquetes de la cola, mientras que la probabilidad de eliminar un paquete *in* depende exclusivamente de la ocupación de la cola con paquetes *in*.

Durante los últimos años se han presentado diferentes propuestas de acondicionadores de tráfico en la literatura especializada [4, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18]. No obstante, ha quedado demostrado que es difícil encontrar un acondicionador de tráfico cuya interacción con los mecanismos de gestión de colas para implementar los PHB permita lograr los dos objetivos del servicio AF. Algunos de los acondicionadores de tráfico propuestos no consiguen garantizar los CIR de modo estricto debido a la gran dependencia que existe con parámetros de la red como por ejemplo el tiempo de ida y vuelta (*Round Trip Time*, RTT). Otros, aún en condiciones favorables donde no hay diversidad en los parámetros de la red, presentan una configuración demasiado compleja que hace que cualquier pequeña variación en los valores de ésta no garantice los contratos. A su vez, existen propuestas que son capaces de asegurar los contratos de los usuarios pero que a la hora de distribuir el ancho de banda en exceso no lo hacen de modo justo (en ninguna de las dos definiciones contempladas para el término *justicia*). Las últimas tendencias en cuanto al desarrollo de acondicionadores de tráfico o bien requieren el uso de señalización, o necesitan una monitorización por flujos en el *router* con los consecuentes problemas de escalabilidad. Además, incluso en estas últimas propuestas que consiguen garantizar los CIR, existen claras deficiencias en cuanto al reparto del ancho de banda sobrante entre las distintas fuentes TCP que componen el agregado.

El acondicionador de tráfico *Counters-Based Modified* (CBM) introducido en [19] se presenta como un enfoque alternativo para conseguir un reparto *justo* del ancho de banda en exceso entre las diferentes fuentes TCP que componen un agregado dentro del servicio asegurado AF. Una vez quedan garantizados los CIR de cada una de las fuentes gracias al marcado de tráfico mediante el algoritmo CB [18], es posible lograr una distribución equitativa del ancho de banda en exceso utilizando una función policía que descarta de manera probabilística paquetes que están calificados como *out*. Empleando este mecanismo, conseguimos mantener la complejidad en los nodos frontera, utilizando exclusivamente RIO para implementar el PHB. La probabilidad de descarte de un paquete *out* se determina asumiendo que el acondicionador de tráfico conoce la cantidad de ancho de banda sobrante y una aproximación del RTT medio de las conexiones. Aunque este hecho implica que sea necesario algún tipo de señalización, ésta es más sencilla que la empleada en otras propuestas de acondicionadores de tráfico. Las primeras simulaciones realizadas en una topología sencilla de un solo nodo con características variadas (diferentes contratos, diferentes RTT y uso compartido de recursos con fuentes *best-effort*) mostraron que CBM consigue garantizar los CIR de cada fuente de manera estricta y repartir el ancho de banda no contratado de modo justo [19].

En este artículo se estudian las prestaciones de CBM cuando se utiliza en conjunción con RIO en un dominio DiffServ más realista formado por varios nodos. El análisis se centra en examinar el funcionamiento de TCP Reno extremo a extremo cuando nos encontramos en una red con características heterogéneas (diferentes contratos, diferentes RTT y coexistencia con tráfico *best-effort*); en concreto, en términos de qué garantías existen de asegurar los CIR de cada fuente TCP y de cómo se realiza el reparto del ancho de banda en exceso entre las distintas fuentes que componen el agregado. Nótese que en este estudio consideramos un reparto justo del ancho de banda en exceso como la distribución equitativa del mismo. Como se muestra en los resultados, es posible lograr justicia en la distribución del ancho de banda sobrante utilizando el acondicionador de tráfico CBM sin perder exactitud a la hora de garantizar los contratos de cada fuente en dominios DiffServ compuestos de tres nodos.

El resto del artículo queda organizado como sigue. En la sección 2 se describe el algoritmo CBM. En la sección 3 presentamos la herramienta de simulación, así como la topología y escenarios de simulación. A continuación, en la sección 4 se muestran y discuten los resultados obtenidos. Finalmente, la sección 5 resume los puntos más importantes de este trabajo.

2 El acondicionador de tráfico Counters-Based Modified

Partiendo de la suposición de que todos los paquetes que se inyectan en la red tienen un tamaño similar, se puede afirmar que si las fuentes introducen el mismo número de paquetes entonces cada fuente obtiene la misma porción de ancho de banda. Extendiendo este hecho a los paquetes fuera de perfil, podemos afirmar que si todas las fuentes introducen el mismo número de paquetes *out* entonces se consigue un reparto equitativo del ancho de banda sobrante.

Este comportamiento ideal se ve afectado por el diferente funcionamiento de cada fuente TCP, que se ve influenciada por el efecto de diferentes RTT o diferentes contratos de tráfico. Además, se ha de tener en cuenta la interacción con el mecanismo RIO empleado para la gestión de las colas en los *routers*. Con el objetivo de hacer frente a estos efectos de interacción, CBM penaliza a aquellas fuentes que envían paquetes fuera del perfil por encima del valor ideal: Una penalización basada en el descarte probabilístico de paquetes *out* en el propio acondicionador de tráfico.

En [19] se muestra que las conexiones con contratos pequeños y RTT reducidos generan más paquetes *out* entre paquetes *in* consecutivos, que las conexiones con mayores tasas contratadas y RTT más elevados. En consecuencia, obteniendo las primeras más recursos de la red. A partir de estas observaciones, el algoritmo CBM se desarrolla para funcionar como sigue (véase Fig. 1). Cada acondicionador de tráfico, situado junto a la fuente TCP, fuera del alcance del usuario final, dispone de una variable que cuenta el número de paquetes *out* entre dos paquetes *in* consecutivos. Cada vez que un paquete se marca como *out*, el acondicionador de tráfico CBM comprueba esta variable. Si la variable no sobrepasa un umbral mínimo al que llamaremos *min*, entonces el paquete *out* se inyecta en la red. Si la variable excede un umbral máximo al que denotaremos como *max*, entonces el paquete *out* se elimina. Por último, si la variable permanece entre estos dos umbrales *min* y *max*, el paquete se descarta con una probabilidad a la que llamaremos *p*.

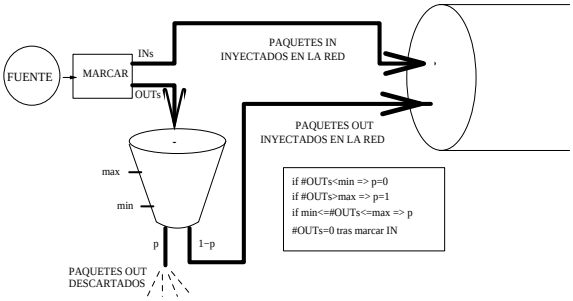


Figura 1. Descarte de paquetes out con CBM.

El uso de este algoritmo requiere por tanto la configuración de los límites *min* y *max* y el cálculo de la probabilidad de descarte *p* que pasamos a describir a continuación. Como se explica en [19], para obtener los valores de *min* y *max* se utilizan las ecuaciones (1) y (2), donde MSS es el acrónimo de tamaño máximo de segmento (*Maximum Segment Size*). El ancho de banda en exceso se puede imaginar como el correspondiente a otra fuente TCP cuya ventana máxima de transmisión fuese el producto ancho de banda en exceso por el RTT medio. De este modo, el umbral máximo *max* quedaría establecido a dicho valor. Una fuente que inyectase un número de paquetes *out* cercano a este límite consumiría casi por completo el ancho de banda sobrante. Además, en el caso en el que una conexión superase dicho límite significaría que no sólo consume todo el ancho de banda en exceso sino que además podría estar *robando* parte del ancho de banda correspondiente al contrato garantizado de otra conexión. En consecuencia, este umbral no debe sobrepasarse nunca. Es conocido el hecho de que en la arquitectura TCP/IP, un algoritmo de crecimiento aditivo y de disminución multiplicativo satisface las condiciones necesarias de convergencia para alcanzar un estado eficiente en la red, siendo utilizado para implementar mecanismos de prevención de la congestión. Por esta razón, se ha optado por un valor mínimo *min* como la mitad del umbral máximo.

$$max = \left\lceil \frac{Ancho\_de\_banda_{exceso} \cdot RTT_{medio}}{MSS} \right\rceil \quad (1)$$

$$min = \left\lceil \frac{max}{2} \right\rceil \quad (2)$$

La estimación del RTT se puede obtener a partir de una señalización enviada de manera periódica entre el *router* frontera y las fuentes TCP. El protocolo TCP implementa un algoritmo de estimación del RTT de la conexión actual. Esta estima puede ser enviada al *router*, el cual se encarga de calcular el RTT medio de las conexiones. Este valor es entonces devuelto a los acondicionadores de tráfico, donde los paquetes son marcados y descartados, si procede. Obsérvese que no se requiere una monitorización de los flujos de tráfico en el *router*, función conocida como *per-flow state monitoring*, en el sentido de que el *router* no mantiene información de cada flujo de paquetes activo. Sólo se encarga de determinar el RTT medio con la información que recibe de las fuentes TCP y una vez calculado, estos valores no se almacenan, a diferencia de otros acondicionadores de tráfico como [7, 8, 14, 15].

Finalmente, la probabilidad de descarte *p* se calcula mediante la ecuación (3). Cada fuente tiene un valor diferente de *p* entre 0 y 1, basándose en la tasa contratada. Por simplicidad denotaremos *x* al cociente tasa contratada entre capacidad del enlace. De las primeras observaciones, donde se advierte que fuentes de contrato pequeño y RTT reducidos generan más paquetes *out* entre dos paquetes *in* consecutivos que el resto de fuentes, sería intuitivo aplicar una ecuación de la forma *p=1-x* (véase Fig. 2). Así, las conexiones con contratos pequeños tendrían una mayor probabilidad de eliminar paquetes *out*. Sin embargo, no hay que olvidar que una vez establecido el umbral máximo, el acondicionador de tráfico elimina aquellos paquetes fuera del perfil que hacen que se supere dicho umbral. El hecho de eliminar paquetes se refleja en las fuentes TCP, que reducen su tasa de transmisión, dejando libres más recursos y permitiendo de este modo que el resto de fuentes introduzcan más tráfico en la red (más paquetes fuera del perfil).

En consecuencia, si utilizamos una ecuación para la probabilidad de descarte que penaliza en mayor medida a las fuentes con contratos pequeños, éstas se ven perjudicadas sobremanera. Así, cuando consiguen recuperarse de las sucesivas pérdidas de paquetes, los recursos están siendo utilizados por las fuentes con contratos mayores. Esta situación provocaría nuevas pérdidas haciendo que las fuentes de menor contrato volvieran a reducir sus tasas de transmisión y se originaría el efecto opuesto: Las conexiones con mayores contratos y RTT más elevados obtendrían más recursos de la red, lo que tampoco es deseable. En consecuencia, se debe utilizar una ecuación para la probabilidad de descarte *p* que tienda a favorecer ligeramente las fuentes con contratos pequeños.

$$p = 2 \cdot \frac{tasa\_contratada / capacidad\_enlace}{1 + tasa\_cot\ ratada / capacidad\_enlace} \quad (3)$$

En [19] se evaluó en principio una ecuación de la forma $p=x$ (véase Fig. 2). Las simulaciones mostraron que el mecanismo CBM realizaba un reparto más equitativo del ancho de banda sobrante que el original CB aunque todavía lejos del comportamiento ideal. Con el objetivo de observar el efecto que tendría sobre el reparto del ancho de banda sobrante una ecuación que, aún favoreciendo a las fuentes de menor tasa contratada lo hiciera de manera no lineal, se realizaron simulaciones con las ecuaciones $p=2 \cdot x/(1+x)$ y $p=x/(2-x)$. Estas dos ecuaciones se incluyen en la Fig. 2. Es importante resaltar que pequeñas variaciones en el valor de *p* pueden generar grandes diferencias a la hora de aplicar el algoritmo debido a la reacción de las fuentes TCP ante la pérdida de paquetes. De estos resultados, se concluyó que la ecuación (3) es la más apropiada para el mecanismo de descarte en CBM. Nótese que esta ecuación se aplica únicamente cuando el número de paquetes *out* entre dos paquetes *in* consecutivos está en el intervalo (*min, max*).

3 Escenarios de simulación

El acondicionador de tráfico CBM se evalúa mediante simulaciones en la topología de tres nodos de la Fig. 3 (los cuellos de botella son los propios nodos de la red; T≡Acondicionador de tráfico). Ocho fuentes generan tráfico TCP Reno, transmitiendo a la velocidad máxima del enlace, que ha sido establecida a 33 Mbps. Para comprobar el impacto de diferentes contratos y la influencia de RTT variables, se utilizarán distintos valores en las simulaciones.

La herramienta de simulación empleada para el protocolo de ventana deslizante TCP Reno fue desarrollada en [20] y ha sido ampliamente utilizada en [21, 22]. Además, se usó como herramienta de validación del estudio analítico desarrollado en [23]. Algunas de las características de esta herramienta son las siguientes: todas las fuentes TCP son codiciosas (*greedy sources*) con el fin de tener un peor caso en el que se consigue un estado de congestión de la red elevado; los destinos sólo envían reconocimientos que no presentan pérdidas ni retardos, y el tamaño máximo de la ventana es igual al producto ancho de banda por retardo como es habitual en redes de área amplia (*Wide Area Network*, WAN).

Para las simulaciones se utiliza un tamaño de paquete de 9.188 bytes que corresponde a IP sobre ATM (*Asynchronous Transfer Mode*) y puede representar DiffServ sobre MPLS (*Multi Protocol Label Switching*), donde se ha impuesto el uso de la tecnología ATM entre los fabricantes.

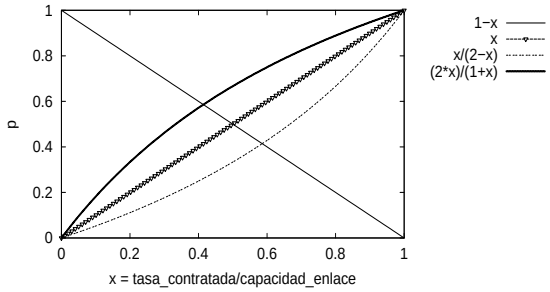


Figura 2. Funciones para la probabilidad de descarte p.

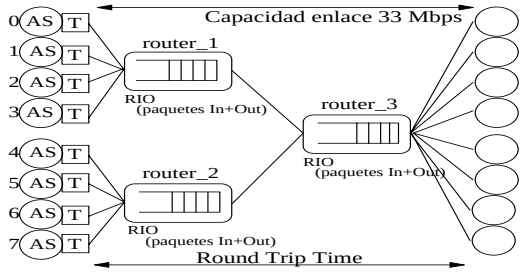


Figura 3. Topología general de tres nodos.

Se van a estudiar tres casos que quedan descritos en las Fig. 4, 5 y 7. Los *routers* almacenan y envían los paquetes de los agregados de tráfico. La gestión de las colas de estos dispositivos se realiza con el mecanismo RIO o con RED según quede indicado. El nodo etiquetado como *router_1* emplea el mecanismo RIO con parámetros ([minth, maxth, maxp]) [40/70/0.02] para los paquetes *in* y [10/40/0.2] para los paquetes *out*. El nodo etiquetado como *router_2* implementa el algoritmo RED con parámetros [10/40/0.2] o el RIO con los mismos valores que los utilizados en *router_1*. El último nodo, *router_3*, recibe el tráfico procedente de los dos nodos anteriores y ejecuta RIO con los mismos valores que los empleados en *router_1*. En cuanto a los valores de los parámetros de configuración que se usan en RED para calcular el tamaño medio de las colas, *weight_in* y *weight_out*, se han establecido a 0.002 siguiendo las recomendaciones de [5].

Obtendremos resultados para cinco escenarios diferentes. Trabajaremos en una situación donde la carga de la red está alrededor del 60%. Esta situación es más interesante para nuestro estudio ya que el ancho de banda en exceso representa una porción importante del ancho de banda total disponible. El intervalo de confianza de los resultados es del 95%, que ha sido calculado con una distribución normal y usando 30 muestras que proporcionan un valor aproximado de ±0.002 en los valores de justicia y de ±0.01 en los *throughputs* alcanzados. El término *throughput* hará referencia al *goodput*, es decir, no se tendrán en cuenta los paquetes retransmitidos. A continuación resumimos las características de los diferentes escenarios de simulación en la Tabla 1.

El escenario A es al que más se ha recurrido en estudios similares sobre prestaciones de acondicionadores de tráfico. Dadas sus características

es presumible obtener en él los mejores resultados. Con la introducción de contratos diferentes en el escenario B pretendemos acercarnos a un ambiente más realista con QoS [24]. En el caso del escenario C, opuesto al escenario B, podemos analizar el efecto producido en las prestaciones del mecanismo CBM por el hecho de tener fuentes con diferente RTT. El escenario D es el más complejo debido a que las conexiones TCP con contratos más bajos y RTT menores se ven claramente favorecidas como se demuestra en [25]. Finalmente, el escenario E por el hecho de asignar RTT mayores a las fuentes con contratos más pequeños evita parcialmente favoritismos en el reparto de los recursos de la red a diferencia de lo que ocurre en D.

Tabla 1. Escenarios de simulación.

Escenario	Contrato (Mbps)	RTT (ms)
A	2.5	50
B	1-1-2-2-3-3-4-4-	50
C	2.5	10 to 80 a intervalos de 10
D	1-1-2-2-3-3-4-4	10 to 80 a intervalos de 10
E	4-4-3-3-2-2-1-1	10 to 80 a intervalos de 10

4 Resultados

En esta sección se presentan y discuten los resultados obtenidos para la topología y escenarios descritos anteriormente. Se evalúan las prestaciones de TCP extremo a extremo atravesando una red de tres *routers*. El estudio se realiza en términos de garantías de asegurar los contratos, reparto justo del ancho de banda sobrante y robustez del mecanismo cuando tráfico *best-effort* (BE) comparte recursos con el AF. Esta topología es notablemente más compleja y heterogénea que las empleadas normalmente en la literatura especializada. Los trabajos realizados en esta misma dirección concluyeron que no era asequible garantizar de modo estricto un servicio cuantificable al tráfico TCP [6, 27]. Aunque los últimos estudios presentan resultados más favorables [8], no parece del todo obvia una implementación factible. En todos los casos que se van a estudiar, los acondicionadores de tráfico CBM están situados junto a las fuentes TCP cuando se requiera un servicio asegurado AF. De otro modo, las fuentes pertenecen al servicio *best-effort* y sus paquetes son tratados como fuera del perfil (paquetes *out*).

Para evaluar la justicia utilizamos el índice *f* que se obtiene a partir de la ecuación (4). En esta ecuación, x_i es el exceso en *throughput* de la fuente *i*, y *n* es el número de fuentes que componen el agregado [26]. Conforme más se aproxime a 1 el valor del índice *f* más justicia habrá en el sistema en el reparto del ancho de banda sobrante. Para calcular el índice de justicia *f* utilizamos el término *throughput* en el sentido de *goodput* comentado anteriormente.

$$f = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \cdot \sum_{i=1}^{n} x_i^2}; \; f \le 1 \qquad (4)$$

4.1 Servicio asegurado AF en una red de tres nodos

Este primer caso de estudio está compuesto por tres routers RIO y ocho fuentes TCP Reno con servicio asegurado como se ilustra en la Fig. 4. Las fuentes generan tráfico a la velocidad del enlace, establecida a 33 Mbps. Las características de los diferentes escenarios A, B, C, D y E fueron descritas en la sección 3. Los valores de los umbrales *min* y *max* utilizados en el mecanismo CBM se incluyen en la Tabla 2. Calculamos estos límites para los *routers router_1* y *router_2*, asumiendo el hecho de que *router_1* no sabe de la existencia de *router_2* y viceversa.

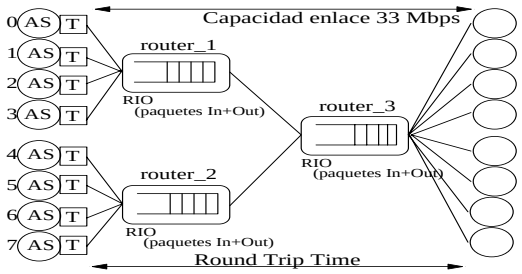


Figura 4. Topología con ocho fuentes TCP Reno con servicio AF y tres nodos RIO.

Tabla 2. Umbrales min y max de cada escenario de simulación con CBM en la topología de tres nodos.

	Escenario A	Escenario B	Escenario C	Escenario D	Escenario E
Capacidad del enlace (Mbps)	33	33	33	33	33
Valores de fuentes nº 0 a 3					
Σ CIR (Mbps)	10	6	10	6	14
Ancho de banda $_{exceso}$ (Mbps)	23	27	23	27	19
RTT_{medio} (ms)	50	50	25	25	25
max router_1 (nº paquetes *out*)	16	19	8	10	7
min router_1 (nº paquetes *out*)	8	9	4	5	4
Valores de fuentes nº 4 a 7					
Σ CIR (Mbps)	10	14	10	14	6
Ancho de banda $_{exceso}$ (Mbps)	23	19	23	19	27
RTT_{medio} (ms)	50	50	65	65	65
max router_2 (nº paquetes *out*)	16	13	21	17	24
min router_2 (nº paquetes *out*)	8	7	12	9	12

Tabla 3. Throughput (Mbps) de paquetes *in* de cada fuente obtenido en las simulaciones con la topología de la Fig. 4.

Fuente	A	B	C	D	E
0	2.50	0.99	2.50	0.99	3.85
1	2.49	1.00	2.50	0.99	3.99
2	2.49	1.99	2.49	2.00	3.00
3	2.49	1.99	2.49	2.00	2.99
4	2.49	2.99	2.50	2.99	1.99
5	2.49	2.99	2.50	2.99	1.99
6	2.49	3.99	2.49	3.95	1.00
7	2.50	3.99	2.48	3.70	1.00

Como se observa de los resultados de las simulaciones (véase Tabla 3), el uso combinado de CBM y RIO permite que los usuarios obtengan sus contratos a pesar de la heterogeneidad de la red. El descarte probabilístico de paquetes fuera del perfil es causante de la adaptación por parte de las fuentes TCP a las características de la red. Una vez se ha conseguido garantizar los contratos, cada fuente obtiene una fracción similar del ancho de banda en exceso como queda plasmado en la Fig. 8, en la que el índice de justicia está por encima de 0,8 para todos los escenarios excepto el escenario D.

La distribución poco pareja del ancho de banda sobrante en este escenario D se puede explicar como sigue. *Router_1* recibe el tráfico de las conexiones con contratos pequeños y RTT bajos, mientras que *router_2* se ocupa de los contratos más elevados y los RTT más altos. En una topología de un solo nodo con características misceláneas como éstas no supone ningún problema por la buena interacción entre CBM y RIO. No obstante, en este caso la tarea de distribución el ancho de banda en exceso recae sobre *router_3*. Un nodo que exclusivamente hace uso del gestor de colas RIO y en consecuencia apenas es capaz de proporcionar un reparto equitativo del ancho de banda no contratado (f=0,623).

4.2 Robustez de CBM frente a fallos en la red

En este segundo caso de estudio, *router_2* implementa RED en lugar de RIO (véase Fig. 5). Esta situación podría ser interesante para un proveedor de servicios de Internet (*Internet Service Provider, ISP*), principalmente porque sería una ventaja poder ofrecer un servicio asegurado con una implementación más sencilla, es decir utilizando RED que básicamente es una cola FIFO capaz de evitar el problema de la sincronización global. Incluso, podría resultarle interesante desde el punto de vista de la reconfiguración de los recursos de la red, siendo capaz de hacer frente a un posible fallo en algún nodo que deba ser reemplazado de modo temporal por otro que sólo sea capaz de implementar una gestión de colas sencilla como RED. De la misma forma que en el caso de estudio anterior, el tráfico se genera con ocho fuentes TCP Reno que contratan un servicio asegurado AF. Los escenarios de simulación A-B-C-D-E son los descritos en la sección 3 y quedan resumidos en la Tabla 2 junto con los valores de los límites *min* y *max*.

Los resultados muestran que se mantienen las garantías de asegurar a cada conexión la tasa contratada, cumpliéndose incluso para el peor caso, el escenario D, donde se alcanzan los contratos tras un intervalo transitorio (véase la Fig. 6 donde se incluyen los primeros 180 segundos de simulación). Nótese que el transitorio en el *throughput* no es relevante para las prestaciones finales y además, se puede considerar despreciable para el resto de escenarios. El índice de justicia se mantiene de nuevo por encima de 0,8 para todos los escenarios excepto para el D (véase Fig. 8). El escenario D es la peor situación en la que nos podemos encontrar, al igual que ocurría en el primer caso de estudio, con la añadidura de que en este segundo caso *router_2* recibe la mayor carga de paquetes *in* (recibe tráfico de las fuentes con mayores contratos) y no implementa diferenciación de servicios (sólo emplea RED). Aún así, el descarte de paquetes *out* en el acondicionador de tráfico CBM origina un equilibrio en el uso compartido del ancho de banda sobrante sin interferir en las garantías de asegurar los contratos de cada fuente.

4.3 Robustez de CBM ante el servicio *best-effort*

En este último caso de estudio, estamos interesados en conocer el efecto de la coexistencia de dos tipos de tráficos, servicio asegurado AF y *best-effort*, que además compiten en este caso por los recursos de la red. Normalmente, las implementaciones reales de DiffServ no mezclan diferentes tipos de tráfico como pueden ser el AF y el BE en una misma cola, sino que separan los paquetes correspondientes a uno y otro tipo y se almacenan en colas diferentes. Por este motivo, el objetivo de este tercer caso de estudio no es plantear la unión de tráfico AF y BE en una misma cola como configuración a emplear en redes reales, sino analizar si sería factible para un ISP reaccionar ante un fallo temporal en la red reconfigurando sus recursos de tal manera que ambos tipos de tráfico pudieran compartir la misma cola dentro del *router* sin afectar a las prestaciones de la red. Es decir, garantizando los contratos de las fuentes AF y haciendo un reparto justo del ancho de banda no contratado entre fuentes AF y BE.

Para las simulaciones se han utilizado doce fuentes TCP Reno transmitiendo a la velocidad del enlace (33 Mbps). Las fuentes numeradas del 0 a 3 y las numeradas del 6 al 9 contratan un servicio AF. Las fuentes restantes, 4 y 5 del *router_1* y 10 y 11 del *router_2* son *best-effort* (véase la Fig. 7). Las características de los diferentes escenarios de simulación, que han sido modificadas respecto a los casos anteriores, quedan resumidas en la Tabla 4. Los paquetes procedentes de fuentes *best-effort* se marcan como *out* y por ser *best-effort* estas fuentes no realizan contratos, por lo tanto, tratando de obtener tanto ancho de banda como les sea posible.

A pesar de no realizar diferenciación de servicios en *router_2*, nótese que sólo implementa RED, los resultados indican que los contratos de cada fuente con servicio AF están garantizados. Véase la Tabla 5 donde lógicamente las fuentes *best-effort* no están incluidas. Nuevamente, experimentamos algunos problemas en el escenario D, ya que las fuentes 8 y 9 quedan por debajo de la tasa garantizada. En el escenario D, estas dos conexiones presentan unos RTT de 90 y 100 ms respectivamente, junto con los contratos más elevados (4 Mbps), y ambas confluyen en el router RED (*router_2*). Este hecho provoca que sea *router_3* quien deba dar precedencia a los paquetes dentro del perfil procedentes de las conexiones con mayores contratos, que junto con la presencia de tráfico *best-effort* hace que las fuentes 8 y 9 no alcancen al cien por cien sus contratos.

Debido a las diferencias substanciales que existen entre las distintas conexiones en cuanto a retardos y contratos, no es posible garantizar de modo estricto las tasas contratadas. Sin embargo, no hay que olvidar que el objetivo de mezclar en una misma cola tráfico asegurado AF y tráfico *best-effort* tiene como única finalidad hacer frente a fallos en la red en los que el ISP se vea obligado a utilizar este tipo de configuración. En consecuencia, el hecho de asegurar los contratos prácticamente en su totalidad, donde en el caso peor quedan garantizados al 70%, puede entenderse como un avance a la hora de ofrecer diferenciación de servicios con el servicio asegurado AF cuando coexiste con tráfico BE, incluso cuando no es posible implementar diferenciación de paquetes en alguno de los *routers* de la red.

En cuanto al reparto del ancho de banda en exceso, nuevamente el algoritmo CBM controla el número de paquetes *out* que entran en la red, obligando a las fuentes *best-effort* a adaptarse a las condiciones de la red y generar menos paquetes *out*. Por lo que el índice de justicia *f* se mantiene por encima de 0,75 en todos los escenarios excepto el D (véase la Fig. 8).

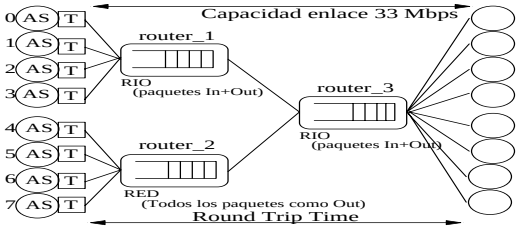


Figura 5. Topología con ocho fuentes TCP Reno con servicio AF y un nodo RED.

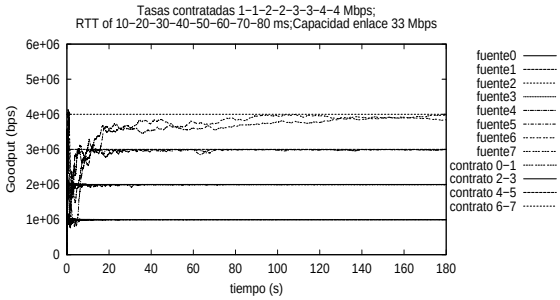


Figura 6. Los contratos de todas las fuentes quedan garantizados con CBM en el peor escenario (D).

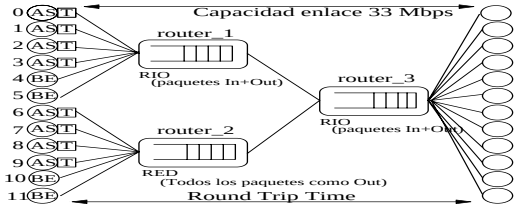


Figura 7. Topología con ocho fuentes TCP Reno con servicio AF, cuatro fuentes TCP Reno con servicio BE y un nodo RED.

Tabla 5. Throughput (Mbps) de paquetes in de cada fuente obtenido en las simulaciones con la topología de la Fig. 7.

Fuente	A	B	C	D	E
0	2.50	1.00	2.49	1.00	3.30
1	2.49	0.99	2.50	1.00	3.99
2	2.49	1.99	2.49	1.99	2.99
3	2.49	1.99	2.50	2.00	2.99
6	2.49	2.99	2.49	2.90	1.99
7	2.49	2.99	2.49	2.70	1.99
8	2.50	3.97	2.50	2.70	1.00
9	2.49	3.99	2.49	2.60	1.00

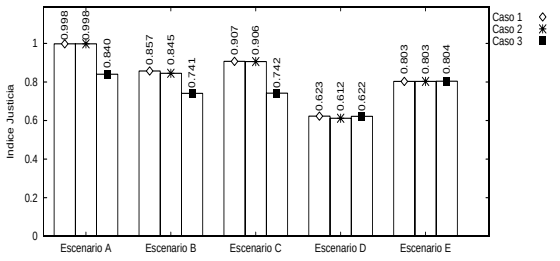


Figura 8. Índice de justicia obtenido en los tres casos de estudio.

Tabla 4. Contratos, RTT y umbrales min y max de las fuentes TCP Reno en la topología de la Fig. 7.

	CIR (Mbps)	RTT (ms)	*Fuentes 0 a 3*		*Fuentes 6 a 9*	
			max	*min*	*max*	*Min*
Escenario A	2.5-2.5-2.5-2.5-0-0-2.5-2.5-2.5-2.5-0-0	50	16	8	16	8
Escenario B	1-1-2-2-0-0-3-3-4-4-0-0	50	19	9	13	7
Escenario C	2.5-2.5-2.5-2.5-0-0-2.5-2.5-2.5-2.5-0-0	10 a120 a intervalos de 10	11	6	30	15
Escenario D	1-1-2-2-0-0-3-3-4-4-0-0	10 a120 a intervalos de 10	13	7	25	13
Escenario E	4-4-3-3-0-0-2-2-1-1-0-0	10 a 120 a intervalos de 10	10	5	35	18

5 Conclusiones

En este artículo hemos desarrollado un análisis de las prestaciones del acondicionador de tráfico *Counters-Based Modified* (CBM), una nueva propuesta que consigue distribuir de manera justa el ancho de banda en exceso de la red entre las fuentes TCP que componen el agregado, garantizando además de modo estricto los contratos de las fuentes. CBM logra este objetivo gracias al descarte probabilístico de paquetes fuera de perfil que realiza en función del contrato de cada fuente, del ancho de banda en exceso y de una estimación del RTT medio.

Cuando se utiliza CBM en combinación con RIO, se mitiga parcialmente el efecto que tiene sobre las fuentes TCP la diversidad en contratos, en RTT o la coexistencia con tráfico *best-effort*. Así se ha comprobado mediante simulaciones en una topología de tres nodos con características heterogéneas: contratos variables, RTT variables, reacción ante nodos que no implementan diferenciación de paquetes (RED) y coexistencia con tráfico BE. Concluyendo de los resultados de las simulaciones que los contratos quedan garantizados prácticamente en todos los casos independientemente de las particularidades del escenario de simulación. Además el ancho de banda en exceso se distribuye con un índice de justicia que en la mayoría de los casos está por encima de 0,8. En consecuencia, alcanzando los objetivos del servicio Assured Forwarding.

Agradecimientos

Este trabajo se enmarca dentro del proyecto CICYT FAR-IP (TIC2000-1734-C03-03).

Referencias

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, “An Architecture for Differentiated Services”, RFC 2475, December 1998.

[2] B. Davie, A.Charny, J. C. R. Bennet, K. Benson, J. Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu, D. Stiliadis, “An expedited forwarding PHB (Per-Hop Behavior)”, RFC 3246, March 2002.

[3] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, “Assured Forwarding PHB Group”, RFC 2597, June 1999.

[4] D. Clark y W. Fang, “Explicit Allocation of Best-Effort Packet Delivery Service”, IEEE/ACM Transactions on Networking, Vol. 6, No. 4, pp. 362-373, August 1998.

[5] S. Floyd and V. Jacobson, “Random Early Detection Gateways for Congestion Avoidance”, IEEE/ACM Transactions on Networking, Vol. 1, No.4, pp. 397-413, August 1993.

[6] J. Ibañez, K. Nichols, “Preliminary simulation evaluation of an assured service”, Internet draft, work in progress, draft-ibanez-diffserv-assured-eval-00.txt, August 1998.

[7] W. Lin, R. Zheng, J. Hou, “How to make assured service more assured”, Proceedings of the 7th International Conference on Network Protocols (ICNP'99), pp. 182-191, Toronto, Canada, October 1999.

[8] B. Nandy, N. Seddigh, P. Pieda, J. Ethridge, “Intelligent Traffic Conditioners for Assured Forwarding Based Differentiated Services Networks”, Proceedings of Networking 2000, LNCS 1815, Paris, France, pp.540-554, May 2000.

[9] Elloumi O, De Cnodder S, Pauwels K, “Usefulness of the three drop precedences in Assured Forwarding Service”, Internet draft, work in progress, July 1999.

[10] J. Heinanen, R. Guerin, “A single rate three color marker”, RFC 2698, septiembre 1999.

[11] J. Heinanen, R. Guerin, “A two rate three color marker”, RFC 2698, septiembre 1999.

[12] M. Goyal, A. Durresi, P. Misra, C. Liu, R. Jain, “Effect of number of drop precedences in assured forwarding”, Proceedings of Globecom 1999, Rio de Janeiro, Brazil, Vol. 1(A), pp. 188-193, December 1999.

[13] H. Kim, “A Fair Marker”, Internet draft, work in progress, April 1999.

[14] I. Alves, J. De Rezende, L. De Moraes, “Evaluating Fairness in Aggregated Traffic Marking”, Proceedings of IEEE Globecom’2000, San Francisco, USA, pp. 445-449, November 2000.

[15] I. Andrikopoulos, L. Wood, G. Pavlou, “A fair traffic conditioner for the assured service in a differentiated services internet”, Proceedings of IEEE International Conference on Communications ICC2000, New Orleans, LA, Vol. 2, pp. 806-810, June 2000.

[16] Mohamed A. El-Gendy, Kang G. Shin, “Assured forwarding fairness using equation-based packet marking and packet separation”, Computer Networks, Vol. 41 Issue 4, pp. 435-450, 2003.

[17] S. Tartarelli, A. Banchs, “Random Early Marking: Improving TCP Performance in DiffServ Assured Forwarding”, Proceedings of ICC 2002, New York, USA, May 2002.

[18] Maria-Dolores Cano, Fernando Cerdan, Joan Garcia-Haro, Josemaria Malgosa-Sanahuja, “Performance Evaluation of Traffic conditioner Mechanisms for the Internet Assured Service”, in Quality of Service over Next-Generation Data Networks, Proceedings of SPIE Vol. 4524, pp. 182-193, 2001.

[19] Maria-Dolores Cano, Fernando Cerdan, Joan Garcia-Haro, Josemaria Malgosa-Sanahuja, “Counters-Based Modified Traffic Conditioner”, Lecture Notes in Computer Science (QofIS 2002), Vol. 2511, pp. 57-67, Springer-Verlag, 2002.

[20] F. Cerdan, O.Casals, “Performance of Different TCP Implementations over the GFR Service Category”, ICON Journal, Special Issue on QoS Management in Wired &Wireless Multimedia Communications Network, Vol.2, pp.273-286, Baltzer Science, January 2000.

[21] F. Cerdan, O.Casals, “Mapping an Internet Assured Service on the GFR ATM Service”, Lecture Notes in Computer Science (Networking 2000), Vol. 1815, pp. 398-409, Springer-Verlag, 2000.

[22] V. Bonin, F. Cerdan, O.Casals, “A simulation study of Differential Buffer Allocation”, Proceedings of 3rd International Conference on ATM, ICATM’2000, pp. 365-372, Germany, June 2000.

[23] V. Bonin, O.Casals, B. Van Houdt, C. Blondia, “Performance Modeling of Differentiated Fair Buffer Allocation”, Proceedings of the 9th International Conference on Telecommunications Systems, Dallas, USA, 2001.

[24] F. Cerdan, J. Malgosa-Sanahuja, J. Garcia-Haro, F. Burrull, F. Monzo-Sanchez, “Quality of Service for TCP/IP Traffic: An overview”, Proceedings of PROMS’00, pp.91-99, Cracow 2000.

[25] N. Seddigh, B. Nandy, P. Pieda, “Bandwidth Assurance Issues for TCP flows in a Differentiated Services Network”, Proceedings of IEEE Globecom’99, Rio de Janeiro, Brazil, Vol. 3, pp. 1792-1798, December 1999.

[26] R. Jain, “The Art of Computer Systems Performance Analysis”, John Wiley and Sons Inc., 1991.

[27] J. F. De Rezende, “Assured Service Evaluation”, Proceedings of IEEE Globecom’99, Rio de Janerio, Brazil, December 1999.

Modelos Markovianos para la Resolución de Sistemas con Reintentos. Evaluación de diferentes Metodologías

Mª José Doménech Benlloch José Manuel Giménez Guzmán Vicente Casares Giner
Departamento de Communicaciones. Universidad Politécnica de Valencia (UPV)
{mdoben,jogiguz}@doctor.upv.es, vcasares@dcom.upv.es

Abstract *This paper deals with the phenomenon of retrials in telecommunication systems. A novel approximate technique is generalized for the estimation of parameters such as the probability of retrial and the probability of no service or abandon. The approximate analysis technique is based on Markovian models with a finite population and a number of state space proportional to the maximum number of users that can be simultaneously in progress. The extension allows to fit the desired precision of the mentioned probabilities, in a gradual manner. We conclude that excellent accuracy can be obtained with very small state spaces of the Markovian models. The second aspect that has been covered in the paper is the computational effort. Several algorithms have been studied and compared in order to solve the $QBD's$ process that appear in all previous Markovian models.*

1. Introducción

En la teoría de colas clásica, habitualmente se supone que los usuarios que no consiguen servicio inmediato tras su petición, abandonan el sistema sin intentarlo de nuevo (sistema de pérdidas) o permanencen en una cola de espera hasta que son atendidos (sistema de espera) [1]. En este caso, los ususarios impacientes abandonan el sistema, sin la intención de reintentar el acceso al servicio. Por contra hay casos en donde los usuarios reintentan la solicitud del servicio tras cierto tiempo de espera. Tal es el caso del servicio telefónico, en donde abonados que reciben el tono de ocupado, reintentan el acceso tras un tiempo de espera relativamente corto. Otro ejemplo es el de los protocolos de acceso aleatorio, en donde el envío simultáneo de dos o más paquetes de datos por otros tantos usuarios, provoca a una colisión destruyendose los paquetes en cuestión (ignorando el efecto captura, [2]). Las estaciones involucradas en la colisión, aleatorizan el tiempo de espera tras el cual intentan de nuevo la transmisión de los respectivos paquetes con la esperanza de que en algún reintento puedan tener éxito en la transmisión. Son dos clásicos ejemplos en donde parámetros de prestaciones como la probabilidad de servicio inmediato, la probabilidad de demora, tiempos de demora, etc. influyen en la calidad de servicio que perciben los usuarios.

La necesidad de considerar los reintentos en los sistemas de telecomunicación se ha traducido en una amplia gama de modelos en donde se distinguen dos bloques funcionales. Un primer bloque que alberga el conjunto de servidores más una posible cola de espera, y un segundo bloque en donde se ubican los usuarios que reintentan la petición de servicio. La suposición habitual es considerar que el régimen de llegadas, el de servicio y el de reintentos obedecen a leyes Markovianas. Estos modelos tienen la ventaja de ofrecer fórmulas cerradas para los parámetros de prestaciones de interés. Por contra, uno de los inconvenientes es la explosión del espacio de estados en sistemas de relativa complejidad o envergadura. Por ejemplo, en sistemas celulares con distinción entre llamadas nuevas y traspasadas (de handover), (Fig. 1 de [3]), el elevado número de estados puede hacer prohibitivo el cálculo de los parámetros de interés en un tiempo de cómputo razonable. Por consiguiente, surge la atractiva idea de utilizar modelos aproximados y algoritmos asociados que, sin apenas pérdida de precisión en el cómputo de los parámetros de QoS, agilicen los citados tiempos de cómputo. La reducción del número de estados se consigue al agruparlos según características comunes. Ejemplos ya clásicos en donde aplica esta filosofía los encontramos en el modelo de teoría del azar equivalente (ERT) [4] y en el proceso IPP [5], ambos como caracterización aproximada del tráfico telefónico que desborda a una ruta alternativa de segunda elección.

Sin pérdida de generalidad, el presente trabajo presta especial atención a la aproximación propuesta en [3]. En [3], la reducción de estados se efectúa acorde con la presencia o no de usuarios en la órbita (o bloque) de reintentos. Es decir, una variable booleana indica presencia ('1') o ausencia ('0') de usuarios bloqueados con opción a reintentos. Tal planteamiento, en esencia es paralelo a la caracterización de un proceso de desbordamiento mediante un proceso IIP, [5]. En nuestro trabajo hemos planteado un modelo *QBD* que posibilita una transición gradual desde el modelo propuesto en [3] al modelo exacto, lo que permite el estudio del grado de precisión alcanzado en las sucesivas aproximaciones. Por otra parte se efectuado un estudio comparativo entre diversos algoritmos que para la resolución de sistemas *QBD* se han venido proponiendo en la literatura abierta, tales como los de [7]-[10].

El artículo se estructura en la siguientes secciones. En

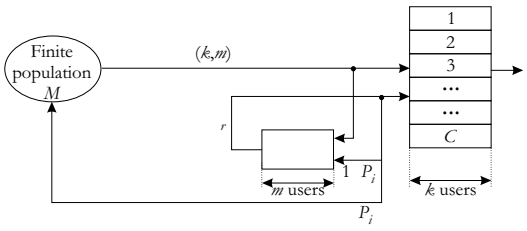


Figura 1: Sistema. Modelo de red

la sección 2 se presenta el modelo markoviano objeto de estudio con la obtención de los parámetros de interés. En la sección 3 se plantea un nuevo modelo Markoviano aproximado. En la sección 4 se discuten diversos algoritmos para la resolución de los procesos QBD. Finalmente en la sección 5 se aportan algunos resultados comparativos, finalizando con las conclusiones del trabajo.

2. Modelo

El modelo a estudiar queda reflejado en la Fig. 1. Hay un colectivo de M usuarios cuyas peticiones son atentidas por C servidores según ley exponencial de tasa μ. Cuando un usuario solicita servicio por vez primera y encuentra todos los servidores ocupados, pasa a la órbita de los reintentos, consistente en una espera aleatoria (exponencial) de tasa μ_r. Al expirar el tiempo de espera el usuario efectua un reintento, pudiendo ser exitoso en caso de hallar un servidor libre. De lo contrario volverá a la órbita de reintentos -con probabilidad $(1 - P_i)$ - o abandonará el sistema -con probabilidad P_i -. La suposición implícita de distribución geométrica para el número de reintentos, resulta una primera aproximación a modelos más exactos como los propuestos en [11]. El régimen de llegadas es dependiente del estado del sistema, $\lambda\,(k, m) = \lambda(M-k-m)$, siendo λ la tasa de transición por usuario, de inactivo a activo, y k (m) el número de usuarios en servicio (en la órbita de reintentos).

Parámetros de interés o de mérito son la probabilidad de obtener servicio inmediato, p_{si}, la probabilidad de servicio retardado o demorado, p_{sr}, y la probablidad de no obtener servicio, p_{ns}). Obviamente se deberá cumplir que

$$p_{si} + p_{sr} + p_{ns} = 1 \tag{1}$$

Las expresiones de p_{si}, p_{sr} y p_{ns} pueden darse en términos de las siguientes tasas. En primer lugar tenemos la tasa ofrecida, R_o, que a su vez se descompone en la tasa de primer intento exitoso, $R_{1,s}$, y la tasa de primer intento fallido, $R_{1,f}$; $R_o = R_{1,s} + R_{1,f}$. La tasa de reintentos, R_r, que según el modelo de la Fig. 1, es coincidente con $R_{1,f}$, $R_r = R_{1,f}$, también puede descomponerse en la tasa de reintentos exitosos, $R_{r,s}$, y la tasa de reintentos fallidos, $R_{r,f}$; $R_r = R_{r,s} + R_{r,f}$. Finalmente la tasa de abandonos, R_{ab}, y que según la Fig. 1 resultará ser $R_{ab} = P_i R_{r,f}$. En función de estas tasas tendremos que:

$$p_{si} = \frac{R_{1,s}}{R_o}; p_{sr} = \frac{R_{r,s}}{R_o}; p_{ns} = \frac{R_{ab}}{R_o} \tag{2}$$

En las subsecciones siguientes planteamos los modelos Markovianos exacto y aproximado, así como las correspondientes las expresiones probabilisticas que de ellos se derivan.

2.1. Modelo Markoviano Exacto

En la Fig. 2 se muestra el diagrama de estados con sus respectivas transiciones. El estado (i, j) indica que hay i servidores ocupados y j en la órbita (cola) de reintentos. El generador infinitesimal (3) presenta una estructura tridiagonal en la que los elementos de la matriz serán matrices a su vez, es decir, un proceso QBD [6].

$$\mathbf{Q} = \begin{bmatrix} \mathbf{D}_0 & \mathbf{L}_0 & \dots & 0 & 0 \\ \mathbf{M}_1 & \mathbf{D}_1 & \dots & 0 & 0 \\ 0 & \mathbf{M}_2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \mathbf{L}_{M-C-2} & 0 \\ 0 & 0 & \dots & \mathbf{D}_{M-C-1} & \mathbf{L}_{M-C-1} \\ 0 & 0 & \dots & \mathbf{M}_{M-C} & \mathbf{D}_{M-C} \end{bmatrix} \tag{3}$$

Donde $\mathbf{M}_j$, $\mathbf{D}_j$ and $\mathbf{L}_j$, son matrices cuadradas de dimensión $(C + 1) \cdot (C + 1)$

Para el caso $C = 5$:

$$\mathbf{M}_m = \begin{bmatrix} 0 & m\mu_r & 0 & 0 & 0 & 0 \\ 0 & 0 & m\mu_r & 0 & 0 & 0 \\ 0 & 0 & 0 & m\mu_r & 0 & 0 \\ 0 & 0 & 0 & 0 & m\mu_r & 0 \\ 0 & 0 & 0 & 0 & 0 & m\mu_r \\ 0 & 0 & 0 & 0 & 0 & m\mu_r P_i \end{bmatrix}$$

para $m = 1, 2, .., M - C$

$$\mathbf{L}_m = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda(5, m) \end{bmatrix}$$

para $m = 0, 1, 2, .., M - C - 1$

$$\mathbf{D}_m = \begin{bmatrix} * & \lambda(0, m) & 0 & 0 & 0 & 0 \\ \mu & * & \lambda(1, m) & 0 & 0 & 0 \\ 0 & 2\mu & * & \lambda(2, m) & 0 & 0 \\ 0 & 0 & 3\mu & * & \lambda(3, m) & 0 \\ 0 & 0 & 0 & 4\mu & * & \lambda(4, m) \\ 0 & 0 & 0 & 0 & 5\mu & * \end{bmatrix}$$

para $m = 0, 1, 2, .., M - C$

La identificación de $\lambda(k, m) = \lambda(M - k - m)$ ya se ha comentado anteriormente. Los asteriscos que aparecen en $\mathbf{D}_m$ son los valores negativos que hacen que la suma de elementos de una fila de $\mathbf{Q}$ sea cero.

Resolviendo $\boldsymbol{\pi}\mathbf{Q} = \mathbf{0}$ junto con la condición de normalización $\boldsymbol{\pi}\mathbf{e} = 1$ se obtiene el vector $\boldsymbol{\pi}$ de probabilidades de estado, $\{\pi(k, m)\}$. A partir de $\pi(k, m)$ se

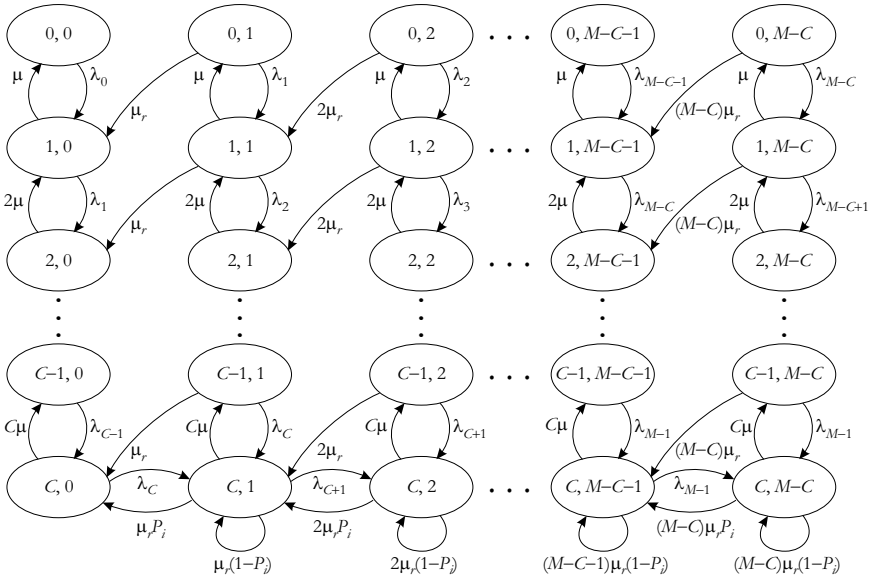


Figura 2: Modelo Markoviano exacto, con $\lambda_i = \lambda(k,m)$; $i = k + m$

calculan las tasas que aparecen en (2) y las respectivas probabilidades. Claramente, las citadas tasas vienen dadas por:

Tasa original (primer intento exitoso o fallido):

$$R_o = \sum_{k=0}^{C} \sum_{m=0}^{M-C} \lambda(k,m)\pi(k,m) \quad (4)$$

$$R_{1,s} = \sum_{k=0}^{C-1} \sum_{m=0}^{M-C} \lambda(k,m)\pi(k,m) \quad (5)$$

$$R_{1,f} = \sum_{m=0}^{M-C} \lambda(C,m)\pi(C,m) \quad (6)$$

Tasas de reintentos (exitosos, fallidos y de abandonos):

$$R_r = \sum_{k=0}^{C} \sum_{m=0}^{M-C} m\mu_r\pi(k,m) \quad (7)$$

$$R_{r,s} = \sum_{k=0}^{C-1} \sum_{m=0}^{M-C} m\mu_r\pi(k,m) \quad (8)$$

$$R_{r,f} = \sum_{m=0}^{M-C} m\mu_r\pi(C,m) \quad (9)$$

$$R_{ab} = P_i \sum_{m=0}^{M-C} m\mu_r\pi(C,m) = P_i R_{r,f} \quad (10)$$

2.2. Modelo Markoviano Aproximado

El objetivo de este modelo es reducir el número de estados y con ello el coste computacional sin que los resultados sufran una pérdidad de precisión significativa. La Fig. 3, refleja el modelo "ON-OFF"de dos columnas de estados. La primera columna es exactamente

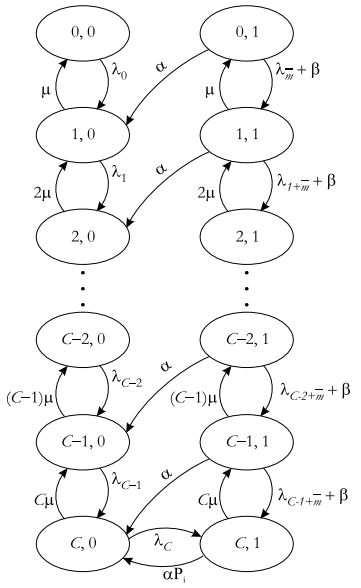


Figura 3: Modelo Markoviano aproximado de dos columnas

igual a la de la Fig. 2. El estado genérico $(k,0)$, corresponde a la situación de k usuarios en servicio sin que hayan usuarios en la órbita (cola) de reintentos. La tasa de nacimientos resulta ser $\lambda(k,0) = \lambda(M-k)$. La segunda columna viene a reemplazar al colectivo de columnas restantes del modelo de la Fig. 2. El estado genérico $(k,1)$, corresponde a la situación de k usuarios en servicio habiendo algún usuario en la órbita (cola) de reintentos. Supondremos que la media de los usuarios en la cola de reintentos es $\overline{m} = E[m]$. La tasa de nacimientos resulta ser $\lambda(k,\overline{m}) = \lambda(M-k-\overline{m}) + \overline{m}\mu_r$.

Cuando un usuario de la órbita (cola) de reintentos logra el acceso a un servidor, puede dejar la citada cola vacía -con probabilidad p- o no vacía -con probabilidad $(1-p)$-. Por lo tanto, la tasa de reintentos se desglosa

en dos contribuciones, tasas α y β. La primera correspondiente a transiciones $(k,1) \rightarrow (k+1,0)$, dada por $\alpha = \overline{m}\mu_r(1-p)$. La segunda a transiciones $(k,1) \rightarrow (k+1,1)$, dada por $\beta = \overline{m}\mu_r p$.

El cálculo de las probabilidades p_{si}, p_{sr} y p_{ns} se haría acorde con las expresiones (2), si bien en las expresiones (4) - (6), habría que sustituir cada sumatorio desde $m = 0$ hasta $m = M - C$ por un sumatorio desde $b = 0$ hasta $b = 1$ y $\pi(k,b)$ por $\pi_a(k,b)$. Las expesiones de (7) - (10) quedarían acorde con:

$$R_r = \sum_{k=0}^{C} \overline{m}\mu_r \pi_a(k,1) \quad (11)$$

$$R_{r,s} = \sum_{k=0}^{C-1} \overline{m}\mu_r \pi_a(k,1) \quad (12)$$

$$R_{r,f} = \overline{m}\mu_r \pi_a(C,1) \quad (13)$$

$$R_{ab} = P_i \overline{m}\mu_r \pi_a(C,1) = P_i R_{r,f} \quad (14)$$

Los nuevos parámetros $\overline{m}$ y p han de estimarse convenientemente. La probabilidad p ha de ser consistente con el balance de flujos en la órbita (cola) de reintentos -ésta no es ni fuente ni sumidero-. Así pues, la tasa entrante a dicha cola ha de igualar la tasa saliente de la misma, ecuación (15). Por otra parte, haremos uso de la propiedad de sistemas de colas con incrementos y decrementos unitarios en el tamaño de la población (crossing up-down argument). Una porción, $(1-p)$, de usuarios que abandonan la citada órbita (cola) la dejan tras de sí vacía, y que expresado en términos de tasa resulta ser $(1-p)(R_{r,s} + R_{ab})$. Dicha tasa ha de igualar a la tasa de peticiones que acceden a la órbita (cola) de reintentos y la encuentran vacía, esto es, a $\lambda(C,0)\pi_a(C,0)$, ecuación (16).

$$\sum_{b=0}^{1} \lambda(C,b)\pi_a(C,b) = R_{r,s} + R_{ab} \quad (15)$$

$$(1-p)(R_{r,s} + R_{ab}) = \lambda(C,0)\pi_a(C,0) \quad (16)$$

Por tanto, igualando (15) y (16) se tiene

$$p = \frac{\lambda(C,1)\pi_a(C,1)}{\lambda(C,0)\pi_a(C,0) + \lambda(C,1)\pi_a(C,1)} \quad (17)$$

Una expresión para $\overline{m}$ puede obtenerse al considerar la ecuación de flujo que se obtiene del corte longitudinal que separa las dos columnas de la Fig. 3.

$$\overline{m} = \frac{\lambda(C,0)\pi_a(C,0) + \lambda(C,1)\pi(C,1)}{\mu_r[\sum_{k=0}^{k=C-1} \pi(k,1) + P_i\pi_a(C,1)]} \quad (18)$$

Obvia resaltar la dependencia entre los valores de p y $\overline{m}$, los cuales no son conocidos a priori. Es preciso arbitrar un procedimiento iterativo para la obtención de los mismos. Iniciando con $\overline{m} = 1$ y $p = 0$ se calculan los $\{\pi_a(k,b)\}$, y se sustituyen en (17) y (18) obteniendo los siguientes valores de p y $\overline{m}$. El procedimiento se repite hasta que la precisión relativa sea menor que, por ejemplo, 10^{-5}.

3. Generalización del Modelo Markoviano aproximado

El modelo Markoviano exacto puede requerir un alto coste computacional si la población es elevada. Por otra parte el modelo de dos columnas puede no ser lo suficientemente preciso. Por lo tanto, se ha propuesto un modelo que permite, mediante un parámetro Q, ir desde el modelo de dos columnas hasta el modelo exacto; permitiendo así situaciones intermedias en cuanto a precisión y coste computacional se refiere. Casos particulares serían $Q = 1$, modelo de dos columnas, y $Q = M - C$ el modelo exacto. El generador infinitesimal, $\mathbf{Q}$, presenta la misma estructura tridiagonal a bloques dada por (3), excepción hecha para la última fila de matrices, $\mathbf{M}_Q$ y $\mathbf{D}_Q$, que vienen dadas por (caso de $C = 5$)

$$\mathbf{M}_Q = \begin{bmatrix} 0 & \alpha & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha \\ 0 & 0 & 0 & 0 & 0 & \alpha P_i \end{bmatrix} \quad (19)$$

$$\mathbf{D}_Q = \begin{bmatrix} * & \lambda(0,\overline{m}) + \beta & \dots & 0 \\ \mu & * & \dots & 0 \\ 0 & 2\mu & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & \lambda(4,\overline{m}) + \beta \\ 0 & 0 & \dots & * \end{bmatrix} \quad (20)$$

El proceso computacional es paralelo a los casos anteriores. Únicamente alertamos del cambio en los límites de los sumatorios, sustituyendo los $(M - C + 1)$ niveles por los $(Q + 1)$ niveles. Los parámetros p y $\overline{m}$ se obtienen mediante razonamientos similares al caso de la Fig. 3 (los detalles pueden encontrarse en [12]).

$$p = \frac{\lambda(C,Q)\pi_g(C,Q)}{\lambda(C,Q-1)\pi_g(C,Q-1) + \lambda(C,Q)\pi_g(C,Q)} \quad (21)$$

$$\overline{m} = \frac{\lambda(C,Q-1)\pi_g(C,Q-1) + \lambda(C,Q)\pi_g(C,Q)}{\mu_r[\sum_{k=0}^{k=C-1} \pi_g(k,Q) + P_i\pi_g(C,Q)]} \quad (22)$$

Similar al caso anterior, p y $\overline{m}$ se obtienen por procedicimiento iterativo, con valores iniciales $\overline{m} = Q$ y $p = 0$.

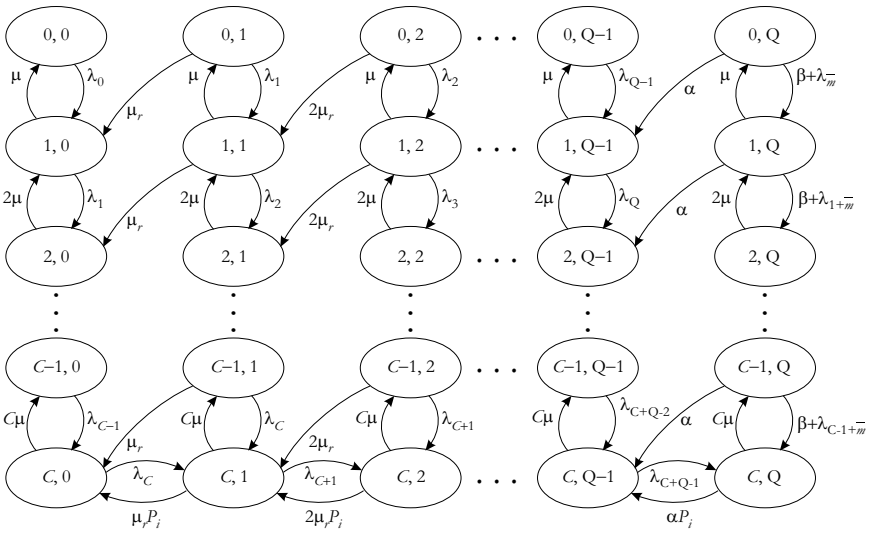


Figura 4: Modelo Markoviano de Q columnas

4. Métodos de resolución

En esta sección se presta atención a la algorítmica para la resolución de los sistemas QBD que surgen en la sección anterior. Con el fin de comparar su eficiencia computacional se han programado tres métodos. El propuesto por Gaver et al. [7], el propuesto por Servi [8] y el Folding Algorithm [9].

4.1. Algoritmo de Gaver et al. [7]

Fijándonos, por ejemplo, en el diagrama de estados de la Fig. 2, el algoritmo opera según sigue:

Paso 1) Obtención de las matrices $\overline{\mathbf{C}}_k$ mediante la recursión inversa

$$\overline{\mathbf{C}}_k = \mathbf{D}_k + \mathbf{L}_k(-\overline{\mathbf{C}}_{k+1}^{-1})\mathbf{M}_{k+1};$$
$$\text{para} \quad k = M - C - 1, .., 1, 0.$$

Con la condición inicial $\overline{\mathbf{C}}_{M-C} = \mathbf{D}_{M-C}$

Paso 2) Evaluar el vector de probabilidades $\boldsymbol{\pi}_0$ ($\overline{\mathbf{C}}_0$ es un generador infinitesimal)

$$\boldsymbol{\pi}_0\overline{\mathbf{C}}_0 = \mathbf{0}$$

Paso 3) Cálculo del resto de vectores de probabilidad mediante recursión directa (y posterior normalización)

$$\boldsymbol{\pi}_k = \boldsymbol{\pi}_{k-1}\mathbf{L}_{k-1}(-\overline{\mathbf{C}}_k^{-1});$$
$$\text{para} \quad k = 1, 2, ..., M - C.$$
$$\sum_{k=0}^{M-C} \boldsymbol{\pi}_k \mathbf{e} = 1$$

Para el modelo de Q columnas, los pasos a seguir son iguales salvo los límites de las recursiones, cambiando $(M - C)$ por Q.

4.2. Algoritmo Propuesto por Servi [8]

Se define un vector de probabilidades de estado, $\boldsymbol{e}_j = (e_{j0}, \ldots, e_{jn})^T$ que representa la probabilidad de encontrarnos en los estados $(j, 0)$, $\ldots$, (j, n). El generador infinitesimal del sistema vendrá dado por tres tipos de submatrices, las matrices $[v_j^-]_{i,k}$ que definen las transiciones desde (j, i) hasta $(j - 1, k)$, las matrices $[v_j^+]_{i,k}$ que definen las transiciones desde (j, i) hasta $(j + 1, k)$ y las matrices $[v_j^0]_{i,k}$ que definen las transiciones desde (j, i) hasta (j, k), donde los elementos de la diagonal $(i = k)$ vendrán dados de forma que la suma de cada fila de la matriz Q sea cero. En el caso que nos ocupa, no son posibles transiciones de (j, i) a (j', i') cuando $|j - j'| > 1$ o $|i - i'| > 1$. Esto hace que se tenga un matriz Q con estructura matricial tridiagonal a bloques, a su vez compuesta por matrices tridiagonales. En [8] se proponen dos algoritmos diferentes para resolver procesos QBD definidos de este modo. En este trabajo se ha escogido el llamado Algoritmo 0.

4.3. Folding Algorithm [9],[10]

El método consiste en reducir de forma progresiva el sistema, resolviendolo para un caso elemental y posteriormente ampliamdo la solución a la del sistema final.

Partimos del siguiente generador infinitesimal QBD:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{v}_0^0 & \mathbf{v}_0^+ & 0 & \ldots & 0 & 0 & 0 \\ \mathbf{v}_1^- & \mathbf{v}_1^0 & \mathbf{v}_1^+ & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & \mathbf{v}_{C-1}^- & \mathbf{v}_{C-1}^0 & \mathbf{v}_{C-1}^+ \\ 0 & 0 & 0 & \ldots & 0 & \mathbf{v}_C^- & \mathbf{v}_C^0 \end{bmatrix}$$

Se define $K = C + 1$ como el número de columnas de la matriz Q. En primera aproximación se parte de $K = 2^n$, para posteriormente ampliar el resultado a cuando K no sea potencia de 2. A título ilustrativo, supongamos $K = 8$:

Paso 1) Reducción directa.
Se realiza una permutación, primero de filas y luego de columnas, o al revés, de la matriz Q. Si el orden original es $0,1,2,3,4,5,6,7$ al permutar quedará $0,2,4,6,1,3,5,7$, esto es, primero pares y luego impares. Tras la permutación de filas (columnas) sobre el resutado obtenido se efectua la segunda permutación de columnas (filas).

Siendo $\mathbf{S}$ la matriz de permutación definida por:

$$\mathbf{S} = \begin{bmatrix} \mathbf{I} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I} & 0 & 0 & 0 \\ 0 & \mathbf{I} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{I} & 0 & 0 \\ 0 & 0 & \mathbf{I} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & \mathbf{I} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{I} \end{bmatrix} \tag{23}$$

$$\begin{aligned} \mathbf{Q_P} &= \mathbf{S}\mathbf{Q}\mathbf{S}^T = \\ &= \left[\begin{array}{cccc|cccc} \mathbf{v}_0^0 & 0 & 0 & 0 & \mathbf{v}_0^+ & 0 & 0 & 0 \\ 0 & \mathbf{v}_2^0 & 0 & 0 & \mathbf{v}_2^- & \mathbf{v}_2^+ & 0 & 0 \\ 0 & 0 & \mathbf{v}_4^0 & 0 & 0 & \mathbf{v}_4^- & \mathbf{v}_4^+ & 0 \\ 0 & 0 & 0 & \mathbf{v}_6^0 & 0 & 0 & \mathbf{v}_6^- & \mathbf{v}_6^+ \\ \hline \mathbf{v}_1^- & \mathbf{v}_1^+ & 0 & 0 & \mathbf{v}_1^0 & 0 & 0 & 0 \\ 0 & \mathbf{v}_3^- & \mathbf{v}_3^+ & 0 & 0 & \mathbf{v}_3^0 & 0 & 0 \\ 0 & 0 & \mathbf{v}_5^- & \mathbf{v}_5^+ & 0 & 0 & \mathbf{v}_5^0 & 0 \\ 0 & 0 & 0 & \mathbf{v}_7^- & 0 & 0 & 0 & \mathbf{v}_7^0 \end{array}\right] \\ &= \left[\begin{array}{c|c} \mathbf{P}_s & \mathbf{P}_{st} \\ \hline \mathbf{P}_{ts} & \mathbf{P}_t \end{array}\right] \end{aligned} \tag{24}$$

Mediante la expresión $\boldsymbol{Q}_1 = \boldsymbol{P}_t - \boldsymbol{P}_{ts}\boldsymbol{P}_s^{-1}\boldsymbol{P}_{st}$ reducimos a una matriz cuadrada de tamaño $K/2$ como:

$$\mathbf{Q}_1 = \begin{bmatrix} \mathbf{v}_0^{0(1)} & \mathbf{v}_0^{+(1)} & 0 & 0 \\ \mathbf{v}_1^{-(1)} & \mathbf{v}_1^{0(1)} & \mathbf{v}_1^{+(1)} & 0 \\ 0 & \mathbf{v}_2^{-(1)} & \mathbf{v}_2^{0(1)} & \mathbf{v}_2^{+(1)} \\ 0 & 0 & \mathbf{v}_3^{-(1)} & \mathbf{v}_3^{0(1)} \end{bmatrix} \tag{25}$$

Los elementos $v^{(i+1)}$ de $\mathbf{Q}_{i+1}$ se obtienen a partir de $v^{(i)}$ de $\mathbf{Q}_i$ mediante las siguientes expresiones:

$$\begin{aligned} \mathbf{v}_j^{0(i+1)} = \quad & \mathbf{v}_{2j+1}^{0(i)} - \mathbf{v}_{2j+1}^{-(i)}[\mathbf{v}_{2j}^{0(i)}]^{-1}\mathbf{v}_{2j}^{+(i)} \\ & -\mathbf{v}_{2j+1}^{+(i)}[\mathbf{v}_{2j+2}^{0(i)}]^{-1}\mathbf{v}_{2j+2}^{-(i)}; \\ & \text{para} \quad j = 0,\ldots,K/2-2. \\ \mathbf{v}_{\frac{K}{2}-1}^{0(i+1)} = \quad & \mathbf{v}_{K-1}^{0(i)} - \mathbf{v}_{K-1}^{-(i)}[\mathbf{v}_{K-2}^{0(i)}]^{-1}\mathbf{v}_{K-2}^{+(i)} \end{aligned}$$

$$\begin{aligned} \mathbf{v}_j^{+(i+1)} &= -\mathbf{v}_{2j+1}^{+(i)}[\mathbf{v}_{2j+2}^{0(i)}]^{-1}\mathbf{v}_{2j+2}^{+(i)}; \\ &\text{para} \quad j = 0,\ldots,K/2-2. \end{aligned} \tag{26}$$

$$\begin{aligned} \mathbf{v}_j^{-(i+1)} &= -\mathbf{v}_{2j+1}^{-(i)}[\mathbf{v}_{2j}^{0(i)}]^{-1}\mathbf{v}_{2j}^{-(i)}; \\ &\text{para} \quad j = 1,\ldots,K/2-1. \end{aligned} \tag{27}$$

Para el último paso

$$\mathbf{Q}_{n-1} = \begin{bmatrix} \mathbf{v}_0^{0(n-1)} & \mathbf{v}_0^{+(n-1)} \\ \mathbf{v}_1^{-(n-1)} & \mathbf{v}_1^{0(n-1)} \end{bmatrix} \tag{28}$$

$$\text{se reduce a } \mathbf{Q}_n = [\mathbf{v}_1^{0(n)}] \tag{29}$$

mediante

$$\mathbf{v}_1^{0(n)} = \mathbf{v}_1^{0(n-1)} - \mathbf{v}_1^{-(n-1)}[\mathbf{v}_0^{0(n-1)}]^{-1}\mathbf{v}_0^{+(n-1)}$$

Paso 2) Resolución sistema.
Resolución de $\boldsymbol{\pi}^{(n)}\mathbf{Q}_n = 0$ con la habitual condición de normalización.

Paso 3) Expansión de la resolución del sistema anterior. A partir de la $\boldsymbol{\pi}^{(n)}$ calculada, obtenemos las probabilidades de estado del sistema original, $\boldsymbol{\pi}$ como:

$$\boldsymbol{\pi} = \boldsymbol{\pi}^{(n)}\mathbf{E}^{(n-1)}\ldots\mathbf{E}^{(1)}\mathbf{E}^{(0)}$$

donde $\mathbf{E}$ se ha calculado en cada reducción como

$$\mathbf{E}^{(i)} = [-\mathbf{P}_{ts}^{(i)}[\mathbf{P}_s^{(i)}]^{-1}, \mathbf{I}]\mathbf{S}^{(i)}; \text{para} \quad i = n,\ldots,1.$$

Cuando K no es potencia de 2, este método requiere una pequeña modificación. Se sigue reduciendo del mismo modo mientras K sea par y cuando K sea impar se resuelve sin tener en cuenta las matrices que forman la última fila y columna. Se sigue reduciendo hasta que vuelve a ser impar, momento en que se vuelven a introducir las matrices eliminadas anteriormente. En el paso 3, $\mathbf{E}$ deberá tener en cuenta esta eliminación.

5. Resultados

La presente sección presenta resultados relativos al grado de aproximación obtenido al variar el parámetro Q y los relativos a la comparativa de los tres algoritmos descritos en la sección 4. En todos los estudios se ha supuesto que la carga ofrecida por usuario es igual a $\rho = \lambda/(\lambda+\mu) = 0.22$ con un tiempo medio de servicio de $1/\mu = 180 segundos$. También se ha supuesto una tasa de reintentos de $1/\mu_r = 10 segundos$ y una probabilidad de reintento -tras un intento fallido- igual a $P_i = 0.5$.

5.1. Resultados del sistema. Convergencia con Q

Fijado el número de canales $C = 30$, hemos analizado las probabilidades p_{si}, p_{sr} y p_{ns} (o de abandono), variando el tamaño de la población, M. Los resultados para el modelo exacto y el aproximado de Q columnas, con $Q = (M-C)/10$ aproximadamente, se reportan

Tabla 1: Resultados en las probabilidades de servicio

$(M, C, M-C, Q)$	P_{si}(exacto)	$P_{si}Q$	P_{sr}(exacto)	$P_{sr}Q$	P_{ns}(exacto)	$P_{ns}Q$
$(61, 30, 31, 2)$	0.999998	0.999998	$1.445 \cdot 10^{-6}$	$1.434 \cdot 10^{-6}$	$5.875 \cdot 10^{-7}$	$5.992 \cdot 10^{-7}$
$(77, 30, 47, 3)$	0.999575	0.999566	$2.870 \cdot 10^{-4}$	$2.851 \cdot 10^{-4}$	$1.380 \cdot 10^{-4}$	$1.396 \cdot 10^{-4}$
$(91, 30, 61, 5)$	0.992881	0.992862	0.004569	0.004564	0.002550	0.002553
$(107, 30, 77, 6)$	0.949919	0.949817	0.030148	0.030121	0.019933	0.019941
$(121, 30, 91, 8)$	0.863292	0.863248	0.077471	0.077457	0.059238	0.059237

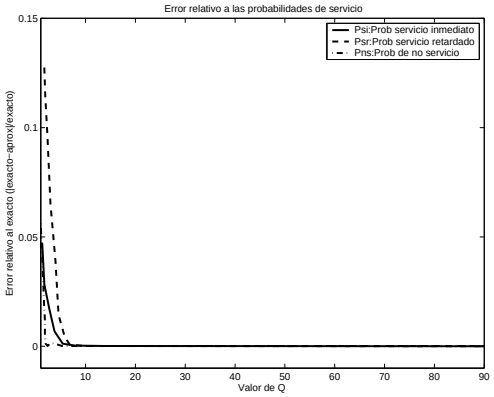


Figura 5: Error relativo en las probabilidades de servicio.

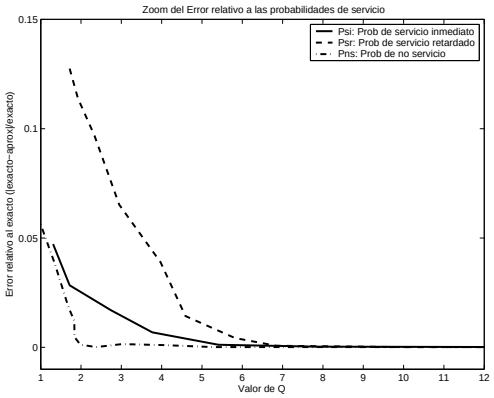


Figura 6: Zoom de Error relativo en las probabilidades de servicio.

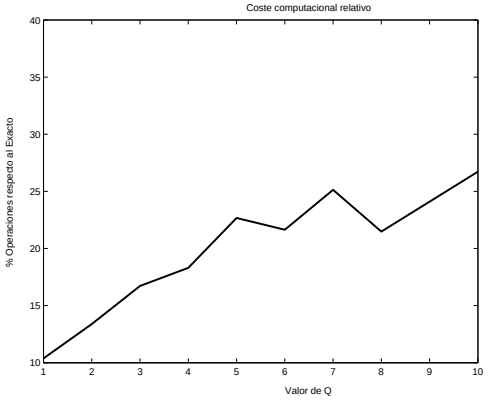


Figura 7: Coste computacional relativo para el método de Gaver.

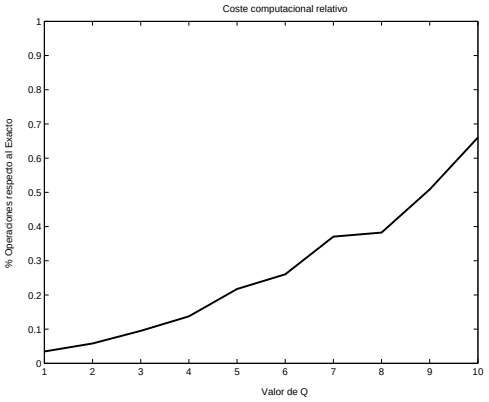


Figura 8: Coste computacional relativo para el algoritmo 0 de Servi.

en la Tabla 1. Obvia resaltar que la QoS empeora a medida que aumenta M. Por otra parte, las diferencias entre ambos modelos apenas son significativas, poniendo de relieve la bondad del modelo aproximado.

Otros resultados han sido los errores relativos en p_{si}, p_{sr} y p_{ns}, para $M = 120$ y $C = 30$, variando Q ($Q = 1, 2, .., M - C + 1$). La Fig. 5 refleja que con $Q = 10$, sobre un total de $Q = M - C + 1 = 91$ se alcanzan precisiones de 6 cifras decimales. La Fig. 6 es un zoom para los valores de interés de Q, $(Q = 1, 2, .., 10)$.

5.2. Costes de cómputo. Comparación entre métodos

Para $M = 120$ y $C = 30$, variando Q ($Q = 1, 2, .., 10$), en la Fig. 7 se observa el gran alivio computacional, pues no se llega al 35 % de lo que supondría el método exacto, ambos implementados con el algoritmo de Gaver et al. [7] (con una precisión de 6 cifras). Tal diferencia es mucho más acentuada cuando se utiliza el algoritmo 0 de Servi. En este caso nos quedamos por debajo del 1 % del coste respecto al modelo exacto, según se aprecia en la Fig. 8.

Comparando el método de Gaver et al. [7] con el de Servi [8], para $Q = M - C + 1$, observamos una mejora considerable del primero respecto al segundo. Así para el de Gaver se precisan $31958 \cdot 10^3$ operaciones[1] mientras que en Servi se realizan $184081 \cdot 10^3$ operaciones. Este resultado puede extenderse para los casos de Q elevada para el modelo aproximado,variando Q, según se observa en la Fig. 9. Sin embargo Servi resulta muy

[1] Se ha trabajado con Matlab, por lo que se entiende por operación el número de operaciones en coma flotante. Sumas y restas consumen 1flop si son reales y 2flops en caso de ser complejas. Multiplicaciones y divisiones tienen un coste de 1flop si el resultado es real y de 6flops en caso de no serlo.

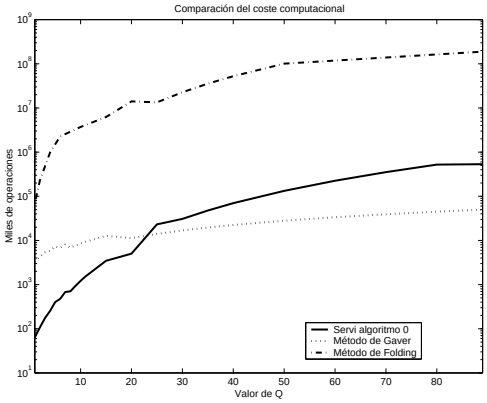


Figura 9: Comparación costes relativos.

eficiente para valores de Q pequeños, los cuales nos dan una aproximación muy buena al resultado exacto.

Otro detalle a destacar de la Fig. 9 es la diferencia en la evolución con Q de los costes. Mientras en Gaver sigue una tendencia lineal, en Servi se produce un crecimiento muy superior al lineal al aumentar Q.

Finalmente subrayar el alto coste computacional (tiempo y memoria) del algoritmo Folding [9], pues se precisa una gran cantidad de cálculos y la necesidad de almacenar matrices de diferentes tamaños en todos los pasos. Así en la Fig. 9 se observa cómo el número de operaciones realizadas con este método es muy superior al que se da en los restantes métodos aquí analizados.

6. Conclusiones

El presente trabajo plantea el estudio de un sistema con reintentos, modelado mediante procesos de Markov. El objetivo principal es el estudio de modelos aproximados que reducen la dimensión del diagrama de estados correspondiente, valorando el grado o bondad de la aproximación que ello comporta. Al mismo tiempo se han contrastado distintos algoritmos de resolución para los procesos $QBD's$ implícitos en las formulaciones, habiendo considerado su carga computacional. Entre ellos, hemos considerado el de Gaver et al. [7], el "folding algorithm" [9], y el recientemente propuesto por Servi, algoritmo 0 de [8].

Agradecimientos

El presente trabajo ha sido financiado por el *Ministerio de Ciencia y Tecnología* a través de los proyectos TIC2000-1041-C03-02 y TIC2001-0956-C04-04.

Referencias

[1] J. Artalejo, G. Falin. "Standard and retrial queueing system: A comparative analysis", Revista Matemática Complutense, vol 15, no. 1, pág. 101 -129. 2002.

[2] D. J. Goodman, A. A. M. Saleh, "The near/far effect in local ALOHA radio communications", IEEE Trans. on Vehicular Technology, vol 36, no 1, pág. 19-27. Febrero 1987.

[3] M.A. Marsan, G. De Carolis, E. Leonardi, R. Lo Cigno, M. Meo. "Efficient Estimation of Call Blocking Probabilities in Cellular Mobile Telephony Networks with Customer Retrials", IEEE Journal on Selected Areas in Communications, vol 19, no. 2, pág. 332-346. Febrero 2001.

[4] R. J. Wilkinson,"Theories for toll traffic engineering in the USA", Bell System Technical Journal, Vol. 35, no 2. pág. 421-513, 1956.

[5] A. Kuczura, "The interrupted Poisson process as an overflow process", Bell System Technical Journal, Vol. 52, no 3. pág. 437-448. Marzo 1973.

[6] M. F. Neuts. "Matrix Geometric Solutions in Stochastic Models: An Algorithmic Approach". The John Hopkins University Press, Baltimore, 1981.

[7] D. P. Gaver, P. A. Jacobs, G. Latouche. "Finite birth-and-death models in randomly changing environments". Adv. Appl. Prob. vol. 16, pág. 715–731. 1984.

[8] L. D. Servi. "Algorithmic Solutions to Two-Dimensional Birth-Death Processes with Application to Capacity Planning". Telecommunication Systems, vol. 21, nos. 2-4, pág. 205-212. Octubre - Diciembre 2002.

[9] J. Ye, S. Li. "Folding Algorithm: A Computational Method for Finite QBD Processes with Level-Dependent Transitions". IEEE Trans. on Communications, vol. 42 no. 2/3/4, pág. 625-639. Febrero/Marzo/Abril 1994.

[10] S. Q. Li, H. Sheng. "Generalized Folding Algorithm for Sojourn Time Analysis of Finite QBD Processes and its Queueing Applications". 2nd Int. Workshop on the numerical solution of Markov chains. Edited by W. J. Stewart. 1995.

[11] E. Onur, H. Deliç, C. Ersoy, M. U. Çaglayan, "Measurement-based replanning of cell capacities in GSM networks". Compuer Networks Vol. 39, pág. 749-767, 2002.

[12] M. J. Doménech, J.M. Giménez, V. Casares. "Modelos Markovianos para la resolución de sistemas con reintentos. Evaluación de diferentes metodologías". Informe Interno. Marzo 2003.

Identificación de Tráfico Anómalo mediante Modelado Estadístico de Protocolos. Aplicación a la Detección de Intrusiones en Redes[†]

Juan M. Estévez-Tapiador, Pedro García-Teodoro, Jesús E. Díaz-Verdejo
Área de Ingeniería Telemática. Depto. Electrónica y Tecnología de Computadores.
Universidad de Granada
E.T.S. Ingeniería Informática. C/ Daniel Saucedo Aranda, s/n, 18071, Granada
Teléfono: +34 958 24 23 05 Fax: +34 958 24 08 31
E-mail: {tapiador, pgteodor, jedv}@ugr.es

***Abstract**. This paper presents a new method for detecting anomalies in the usage of protocols in computer networks. The proposed approach is illustrated through its application to TCP and disposed in two steps. First, a quantization of the protocol header space is accomplished, so that a unique symbol is associated with each protocol instance. Network traffic is thus captured and represented by a sequence of symbols. The modeling of these temporal sequences by means of a Markov chain constitutes the second step in our approach. Once the model is built it is possible to use it as a representation of the "normal" usage of the protocol, so that deviations from the behavior provided by the model can be considered as a sign of protocol misusage. Preliminary experimental results reveal that several protocol misusages used in certain network attacks are detected through the introduced scheme. Additionally, anomaly-based protocol modeling can be used in conjunction with other intrusion detection techniques for improving the performance of current detection technology.*

1 Introducción

La investigación en Sistemas de Detección de Intrusiones (IDS en adelante) ha sido un campo extremadamente activo durante los últimos veinte años. No obstante, la tecnología actual de detección aún padece limitaciones de rendimiento en lo concerniente a aspectos tales como la alta probabilidad de falsas alarmas generadas, la baja precisión de detección obtenida o las elevadas necesidades de monitorización de eventos y carga computacional asociadas a las técnicas empleadas.

Actualmente son dos los principales enfoques establecidos frente al problema de la detección de intrusiones: detección de abusos (*misuse detection*) y detección de anomalías (*anomaly detection*). En los sistemas basados en detección de abusos, cada ataque conocido es modelado mediante un patrón referido comúnmente como *firma*. Las actividades que coinciden con alguno de los patrones en la base de datos de firmas de ataques provocan el disparo de una alarma. El porcentaje de falsas alarmas en estos sistemas depende, entre otros factores, de si el proceso de emparejamiento admite únicamente emparejamientos estrictos o permite algún tipo de desviación. Por el contrario, en los sistemas basados en detección de anomalías el objetivo principal es el modelado del perfil *normal* de comportamiento del sistema, de tal forma que cualquier desviación sustancial con respecto al mismo puede ser etiquetada como intrusiva o, al menos, como sospechosa. Diversas herramientas estadísticas han sido las técnicas más empleadas para la construcción de los perfiles de funcionamiento normal del sistema. Los lectores interesados pueden encontrar excelentes introducciones al campo de la detección de intrusiones en [1] y [2].

Con independencia del método utilizado para los procesos de detección, un IDS puede ser clasificado como basado en *host* o basado en red en función de cuál sea su fuente de datos de análisis. Un IDS basado en *host* (HIDS) intenta identificar intrusiones analizando las actividades que se producen en cada nodo final de la red, principalmente aquéllas producidas por usuarios y programas. En un trabajo pionero, Denning propuso un esquema en el que se construían patrones de comportamiento relacionados con la duración de las sesiones de los usuarios y los recursos consumidos por las aplicaciones [3]. En cambio, los sistemas de detección basados en red (NIDS) centran su atención en el tráfico que es transportado a través de los distintos enlaces de comunicaciones [4]. Snort [5], Bro [6] y NetSTAT [20] son ejemplos representativos de sistemas de estas características.

La necesidad de definir el estado normal de un sistema monitorizado es una cuestión crucial para

[†] Este trabajo ha sido parcialmente financiado por el Ministerio Español de Educación, Cultura y Deporte dentro del Programa Nacional de Formación de Profesorado Universitario (PNFPU) con referencia AP2001-3805.

cualquier IDS basado en la detección de anomalías. Varios autores señalan que, con toda seguridad, el reto más importante para este tipo de métodos radica en la elección de las características que modelan el comportamiento [7], [8]. Tales rasgos deben caracterizar con precisión los patrones de uso de los distintos servicios monitorizados, con objeto de obtener posteriormente modelos precisos de la operación normal de los mismos; pero al mismo tiempo deben contener elementos característicos que permitan una adecuada discriminación posterior entre actividades intrusivas y no intrusivas. Medir la normalidad de un sistema se convierte así en uno de los problemas más importantes en lo concerniente a la mejora de prestaciones de la tecnología actual de detección de anomalías.

En el caso de IDS basados en *host*, los primeros esfuerzos en esta línea dejaron patente la dificultad de obtener perfiles correctos del comportamiento *normal* de usuarios. En efecto, se trata de elementos sujetos a una elevada variabilidad debido a la diversidad y constante evolución de su conducta frente al sistema. Algunos trabajos posteriores, centrados en el modelado del comportamiento de las aplicaciones, han evidenciado que las secuencias de llamadas al sistema ejecutadas por un programa son excelentes características para establecer perfiles de comportamiento normal [7], [9]. Una vez que una aplicación es muestreada mediante un conjunto extenso de las secuencias de llamadas al sistema que éste ejecuta durante su funcionamiento normal, es posible extraer ciertas propiedades estadísticas con la finalidad de modelar su comportamiento correcto. Cadenas de Markov, sistemas de aprendizaje de reglas y otros paradigmas similares han sido ampliamente utilizadas para este propósito (véanse, por ejemplo, [10] y [11]).

En el contexto de IDS basados en red se ha argumentado recientemente que determinadas características asociadas con el modelado de tráfico, como el ancho de banda consumido en ventanas de tiempo de diferente escala o ciertos estadísticos relativos a la operación de los protocolos, son especialmente adecuados para la detección de algunos ataques [12], [13]. Otros enfoques propuestos definen el estado normal de la red mediante autómatas finitos, asociando de este modo las secuencias de acciones correctas con transiciones permitidas entre los estados del autómata [8], [14]. Algunas de estas soluciones son enfoques basados en la detección de abusos, y las máquinas de estados obtenidas son utilizadas fundamentalmente como contexto general para derivar las firmas de ataques.

1.1 Sinopsis de la Contribución

En este trabajo se presenta un método basado en anomalías para la detección de abusos en la utilización de protocolos de red. Para ello se diseña un detector de anomalías para supervisar un protocolo dado, monitorizando constantemente la actividad en la red en busca de desviaciones con respecto a su uso "normal". La justificación para este enfoque proviene del hecho de que gran parte de los ataques de red se sustentan en usos de los protocolos que caen fuera de la descripción oficial de los mismos. La construcción de tales detectores precisa un análisis de la implementación específica del protocolo existente en el sistema, así como del uso concreto que los servicios están haciendo de él.

El método propuesto por los autores se inspira en el utilizado en IDS basados en *host*. La idea básica consiste en la definición de un conjunto de características para un protocolo dado, de tal forma que puedan ser concebidas como el equivalente al papel que las llamadas al sistema representan para las aplicaciones (es decir, como una rúbrica de su operación). Estas características son posteriormente utilizadas para modelar los flujos de tráfico que utilizan el protocolo. El uso *normal* de un protocolo como fenómeno temporal se modela así a través de una cadena de Markov, utilizando secuencias de tráfico observado *in situ* como conjuntos de entrenamiento. Por último, se propone el empleo de una medida de reconocimiento específica, denominada *MAP*, como mecanismo básico para la evaluación y clasificación del tráfico a partir del modelo obtenido.

1.2 Organización del Resto del Artículo

El contenido restante de este trabajo está organizado en torno al método IDS presentado como sigue. La Sección 2 introduce brevemente los fundamentos de las cadenas de Markov y su utilización en el reconocimiento de secuencias. El enfoque propuesto para el modelado de protocolos dentro del contexto de la detección de anomalías es descrito en la Sección 3. En la Sección 4 se exponen detalladamente la aplicación del esquema propuesto al caso de TCP, el entorno de pruebas utilizado durante la experimentación y los resultados obtenidos. Finalmente, la Sección 5 resume el trabajo desarrollado presentando las principales conclusiones así como futuros objetivos de investigación.

2 Cadenas de Markov y su Uso

Supongamos un sistema que evoluciona a través de un conjunto de estados numerados de acuerdo con leyes probabilísticas y satisfaciendo la hipótesis de Markov (es decir, el estado del sistema en el instante $t+1$ depende únicamente de su estado en el instante t). Cada uno de los posibles estados $\mathbf{\Gamma}=\{S_1, S_2, ..., S_N\}$ representa una situación diferente y específica en la que el sistema puede estar. Sea q_t la variable que representa el estado del sistema en el instante t. Entonces, si $\mathbf{P}[q_t=i] > 0$, definimos a_{ij} como:

$$a_{ij} = P[q_{t+1} = j \mid q_t = i] = \frac{P[q_t = i, q_{t+1} = j]}{P[q_t = i]} \qquad (1)$$

y sea $\boldsymbol{A}$ la matriz $[a_{ij}]$. Entonces, si $\mathbf{P}[q_t=i] > 0$,

$$a_{ij} \geq 0, \qquad \sum_j a_{ij} = 1 \tag{2}$$

De esta forma, la matriz de probabilidades de transición $\boldsymbol{A}=[a_{ij}]$ representa la probabilidad de, estando en el estado i en un instante de tiempo t, alcanzar el estado j en el instante $t+1$. Asimismo, se requiere la definición de un vector de probabilidades iniciales $\boldsymbol{\Pi}=\{\pi_i\}$, siendo $\pi_i=\mathbf{P}[q_1=i]$, para establecer la probabilidad del estado inicial. Tal vector debe satisfacer:

$$\pi_i \geq 0, \qquad \sum_i \pi_i = 1 \tag{3}$$

De acuerdo con las definiciones anteriores, cualquier matriz $\boldsymbol{A}=[a_{ij}]$ satisfaciendo (2) puede ser utilizada, junto con el vector de probabilidades iniciales $\boldsymbol{\Pi}=\{\pi_i\}$ satisfaciendo (3) para definir una cadena de Markov con probabilidades de transición estacionarias. La probabilidad $p_j^{(n)}$ de que el sistema esté en el estado j en el instante de tiempo n está dada recursivamente por:

$$\begin{aligned} p_j^{(1)} &= \pi_j \\ p_j^{(n)} &= \sum_i p_i^{(n-1)} a_{ij}, \qquad n > 1 \end{aligned} \tag{4}$$

Una buena introducción a las cadenas de Markov puede encontrarse en [15] y [16].

2.1 Estimación de Parámetros

A lo largo de esta exposición asumiremos siempre que el conocimiento concerniente a los distintos estados alcanzados por el sistema es adquirido mediante la observación de las salidas proporcionadas por el mismo. Estas *salidas* son elementos de un conjunto finito $\boldsymbol{\Theta}=\{O_i\}$, de forma tal que cada una de ellas se corresponde exclusivamente con uno de los posibles estados del sistema.

Supongamos una secuencia temporal ordenada O_1, O_2, ..., O_T de salidas observadas para un sistema. En la teoría de cadenas de Markov se considera la generalización más simple posible, consistente en asumir que la observación en un instante de tiempo depende exclusivamente de la salida obtenida en el instante de tiempo inmediatamente anterior [15]. Así, la matriz de probabilidades de transición puede ser estimada por

$$a_{ij} = \frac{P[q_t = O_j, q_{t-1} = O_i]}{P[q_{t-1} = O_i]} \tag{5}$$

Obsérvese que los dos términos de probabilidad en la expresión anterior pueden calcularse mediante un simple proceso de recuento de ocurrencias a partir de las secuencias de observaciones.

Por otro lado, el vector de probabilidades iniciales $\boldsymbol{\Pi}$ puede ser estimado de manera similar. La probabilidad inicial de cada símbolo puede ser fácilmente computada contando el número de veces que dicho símbolo aparece como primer elemento de las secuencias observadas.

2.2 Reconocimiento de Secuencias

Supongamos una cadena de Markov $\boldsymbol{\lambda}=(\boldsymbol{A}, \boldsymbol{\Pi})$, donde $\boldsymbol{A} = [a_{ij}]$ es la matriz de probabilidades de transición y $\boldsymbol{\Pi} = (\pi_i)$ el vector de probabilidades iniciales, y sea $\boldsymbol{O} = \{O_1, O_2, \ldots O_T\}$ una secuencia observada de símbolos. El problema del reconocimiento con cadenas de Markov se puede formular como la estimación de $\mathbf{P}[\boldsymbol{O} \mid \boldsymbol{\lambda}]$, esto es, la probabilidad de la secuencia observada evaluada por la cadena. Una medida útil para este propósito es la conocida como *Probabilidad Máxima A-posteriori* (MAP), definida como:

$$MAP(O, \lambda) = \pi_{O_1} \cdot \prod_{t=1}^{T-1} a_{O_t O_{t+1}} \tag{6}$$

Un problema relacionado con esta medida es su rápida convergencia a cero. Consecuentemente, en ocasiones resulta más útil su representación en una escala logarítmica, esto es:

$$LogMAP(O, \lambda) = \log(\pi_{O_1}) + \sum_{i=1}^{T-1} \log(a_{O_i O_{i+1}}) \tag{7}$$

El uso de probabilidades acumuladas presenta el inconveniente de que ninguna probabilidad puede ser cero. Para la aplicación práctica de cadenas de Markov, esta cuestión se resuelve mediante un proceso previo de *suavizado* de la cadena. Aunque son muchas las técnicas existentes para su consecución, la más simple consiste en fijar aquellas probabilidades menores que un valor dado "**ε**" a dicho valor ([15]).

3 Modelado Estocástico para la Detección de Anomalías

La información concerniente a la señalización y dinámica de los protocolos de red se ubica en los encabezados de cada PDU (*Protocol Data Unit*). Por consiguiente, es de esperar que las variables convenientes para el modelado del comportamiento "normal" del protocolo sean los valores de los campos del encabezado o combinaciones de los mismos.

La Fig. 1 ilustra esquemáticamente la arquitectura general del modelo propuesto en este trabajo. La idea subyacente es la de modelar el comportamiento de un protocolo dado como una secuencia estocástica de eventos que ocurren mientras las partes involucradas en la comunicación intercambian mensajes. Por su naturaleza, este modelado puede realizarse de una

forma sencilla haciendo uso de cualquier formalismo matemático para el estudio de procesos estocásticos. Las cadenas de Markov son, seguramente, una de las técnicas más elementales para ello.

Sin embargo, la aplicación de esta técnica de modelado a la problemática planteada no es inmediata. Para alcanzar el propósito anterior es necesario llevar a cabo una etapa previa de cuantización de los encabezados del protocolo. El objetivo de tal cuantización es realizar un mapeado del espacio Ω_p de posibles encabezados del protocolo a modelar en un conjunto finito de símbolos Σ. La principal propiedad que debe verificar el esquema de cuantización elegido es que conserve el discernimiento entre instancias del protocolo que son significativamente distintas, esto es, que dos encabezados "suficientemente" diferentes sean representados como dos símbolos distintos, y viceversa.

Tras la etapa de cuantización se dispone de una representación del tráfico de red en una determinada capa como una secuencia de observaciones escalares (símbolos). Una vez que esta transformación ha sido realizada, el siguiente paso consistirá en el modelado de tal secuencia. En el modelo propuesto son las cadenas de Markov las herramientas utilizadas para capturar la dinámica del protocolo. Dado un conjunto de secuencias de observaciones (instancias de uso del protocolo) es posible, haciendo uso de la expresión (5), estimar todos los parámetros que definen el modelo.

Finalmente, durante la etapa de evaluación, una vez disponible el modelo construido en la fase anterior, se procederá a estimar las expresiones (6) o (7) a fin de clasificar los eventos observados como normales o no.

4 Aplicación al Caso TCP

En esta sección ilustraremos la aplicación práctica del modelo expuesto al caso particular del protocolo TCP. Los apartados siguientes detallan la construcción de un detector simple para la identificación de anomalías en el uso de este protocolo, con especial énfasis en aquéllas concernientes a problemas de seguridad.

4.1 Parametrización y Cuantización

En el caso de TCP, la mayor parte de la información de señalización está localizada en los campos conocidos como *flags* [17]. Un enfoque simple (aunque eficaz, como se verá posteriormente) es considerar la configuración de *flags* de cada segmento TCP como su representación particular. De esta forma, es posible asociar un único símbolo S_p con cada segmento dado por:

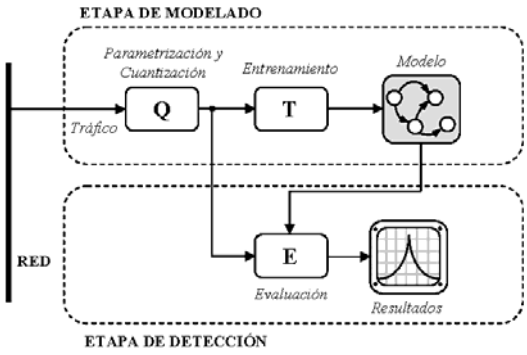


Figura 1: Arquitectura general del sistema de modelado de protocolos propuesto para la detección de anomalías.

$$S_p = \sum_{i=1}^{6} w_i \cdot b_i = S + 2A + 4P + 8R + 16U + 32F \qquad (8)$$

donde *S*, *A*, *P*, *R*, *U* y *F* se corresponden, respectivamente, con el valor de los *flags* SYN, ACK, PSH, RST, URG y FIN. La idea subyacente a este esquema de cuantización es la de asociar con cada segmento TCP el símbolo S_n, donde "*n*" es el número resultante de considerar la disposición de *flags* del segmento como un número binario de 6 bits (n = 0, 1, … 63).

4.2 Entorno de Pruebas

El principal requisito para efectuar una evaluación del modelo propuesto es la obtención de tráfico de las dos clases necesarias: normal y de naturaleza anómala. Como primera aproximación han sido utilizadas conexiones TCP filtradas por el puerto de destino (es decir, por la aplicación destinataria del tráfico). Las aplicaciones monitorizadas para los experimentos han sido HTTP, SSH y FTP.

La Tabla 1 resume algunas estadísticas sobre los ficheros de tráfico utilizados. Los paquetes han sido capturados monitorizando conexiones normales destinadas a un *host* que ejecuta un servidor HTTP, un servidor FTP y un servidor SSH. La captura, filtrado y extracción de los segmentos TCP puede ser llevada a cabo fácilmente haciendo uso de la herramienta *tcpdump* [18]. Cada fichero contiene varias sesiones compuestas por secuencias ordenadas de las cabeceras TCP intercambiadas a lo largo de toda la sesión.

Por otro lado, las conexiones anómalas empleadas durante la experimentación han sido obtenidas utilizando diversas herramientas que explotan ciertas debilidades y ambigüedades en TCP con fines diversos. Por ejemplo, *nmap* [19] y otras herramientas similares de reconocimiento utilizan ciertos segmentos TCP para llevar a cabo sus objetivos. La mayor parte de estas técnicas se basan en un uso anómalo de la señalización transportada en los *flags*. Cabe citar, por ejemplo, segmentos con ningún *flag* activado (*null scan*), segmentos con todos

Tabla 1: Conjuntos de datos de tráfico normal utilizados para la construcción de los modelos TCP. El tamaño de cada traza indica el número de cabeceras TCP registradas.

Aplicación FTP			Aplicación HTTP			Aplicación SSH		
Traza	No. de sesiones	Tamaño Total	Traza	No. de sesiones	Tamaño Total	Traza	No. de sesiones	Tamaño Total
ftp.1	14	5207	http.1	29	8975	ssh.1	11	3349
ftp.2	9	3762	http.2	41	13862	ssh.2	9	3294
ftp.3	18	6862	http.3	102	28107	ssh.3	12	3766
ftp.4	32	18101	http.4	57	19343	ssh.4	24	7069
ftp.5	69	27753	http.5	98	50462	ssh.5	143	63252
ftp.6	78	51345	http.6	62	21310	ssh.6	218	122355
ftp.7	156	133615	http.7	117	41329	ssh.7	241	151142

los *flags* activados (*Xmas scan*), o segmentos con el *flag* FIN activado destinados a un puerto sin una conexión previa establecida (una de las variantes de *stealth scan*). A pesar de que éstas y otras técnicas son bien conocidas, y es posible instalar filtros correspondientes para evitarlas, resulta obvio que las capacidades de detección siempre serán dependientes de las firmas contenidas en la base de datos de firmas, de forma que nuevos ataques requerirán nuevas firmas.

4.3 Estimación del Modelo

La etapa de cuantización produce una representación de cada conexión como una secuencia ordenada de símbolos. Estas trazas son utilizadas a continuación como entradas para la fase de estimación del modelo, siguiendo las expresiones brevemente comentadas en la Sección 2.1.

En una aproximación preliminar, la construcción del modelo se ha realizado filtrando previamente las conexiones por el puerto de destino. El resultado es la obtención de un modelo distinto para el tráfico TCP destinado a cada aplicación. La Fig. 2 muestra los modelos obtenidos para las tres aplicaciones que componen los conjuntos de datos comentados anteriormente: FTP, SSH y HTTP. Nótese que la ilustración corresponde a los modelos estimados *antes* de llevar a cabo el proceso de suavizado

En concreto, el modelo obtenido con secuencias procedentes de tráfico FTP presenta cuatro estados con probabilidades de transición no nulas: S_1, S_2, S_6 y S_{34}. El estado S_1 corresponde a un segmento TCP con el *flag* SYN puesto a 1, representando la solicitud para el establecimiento de una conexión. Los estados S_2 y S_6 son conceptualmente idénticos: el reconocimiento de un paquete recibido. No obstante, mientras que S_2 únicamente tiene el *flag* ACK puesto a 1, el estado S_6 corresponde a un segmento con los *flags* ACK y PSH activados simultáneamente. Esta diferencia podría estar originada por diferentes estados de carga de la red, de forma que en ciertos paquetes se activa el *flag* PSH para su entrega inmediata. Finalmente, el estado S_{34} corresponde a un paquete con los *flags* ACK y FIN activados, y

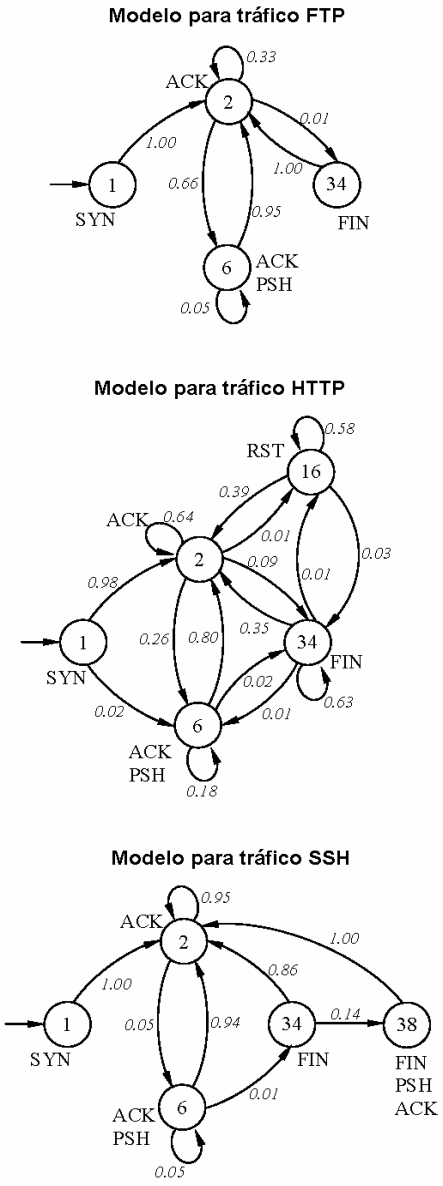


Figura 2: Modelos estimados para diferentes servicios sobre TCP. Los valores de las probabilidades de transición entre estados están representados sobre cada arco del autómata. Cada transición queda así definida por el estado actual S_i junto con el siguiente S_{i+1}. Las transiciones de probabilidad nula no se encuentran representadas en la figura.

representa el cierre de una conexión junto con el reconocimiento de la recepción de un paquete anterior.

El análisis de las transiciones presentes en el modelo revela que se ha capturado correctamente la dinámica especificada para el protocolo TCP [17]. Más específicamente, es inmediato establecer una analogía entre este autómata y un cierto subconjunto de la conocida máquina de estados TCP.

La discusión anterior se puede extender fácilmente al caso de los servicios HTTP y SSH. Aunque los modelos obtenidos son esencialmente equivalentes, las diferencias observadas (como la aparición de estados con el *flag* RST activado) son originadas por el uso concreto que cada aplicación hace del protocolo. Sin embargo, en ellos es posible identificar la misma semántica de uso.

4.4 Evaluación y Análisis de Resultados

Tras el periodo de entrenamiento se dispone de una cadena de Markov para el tráfico TCP entrante destinado a cada servicio específico. La evaluación posterior es realizada siguiendo el procedimiento descrito en la Sección 2.2. El método de suavizado implementado para solventar el problema de las transiciones nulas es el que fue comentado brevemente en dicha Sección: aquellas transiciones con probabilidad menor que un umbral fijo ($\varepsilon=10^{-6}$ en nuestro caso) son fijadas a este valor.

La curva mostrada en la gráfica superior izquierda de la Fig. 3 corresponde a la función *LogMAP* para una sesión HTTP "normal". Esta función posee siempre una forma similar a la allí mostrada. Mientras los símbolos entrantes se adecuen correctamente a los esperados por el modelo en términos probabilísticos, las transiciones entre estados serán las esperadas y, por lo tanto, la suma acumulada proporcionada por *LogMAP* no presentará cambios bruscos de pendiente. Por el contrario, la aparición de patrones de tráfico inesperados producirá una ráfaga de bajas probabilidades. Este fenómeno se manifestará en un cambio abrupto de la pendiente de la curva, como queda patente en el gráfica inferior izquierda de la Fig. 3.

Un método útil para detectar estos cambios y, por tanto, la presencia de tráfico anómalo, es observar cuándo la derivada de *LogMAP* supera cierto umbral. Para este propósito se ha utilizado la familia de funciones:

$$D_{W_m}(t) = \left| LogMAP(t) - \frac{1}{W_m}\sum_{i=1}^{W_m} LogMAP(t-i) \right| \qquad (9)$$

para valores del parámetro W_m = 1, 2, 3, … Nótese que el segundo término en (9) es la media de las últimas W_m salidas. Un incremento en este parámetro producirá una amplificación de $D_{Wm}(t)$,

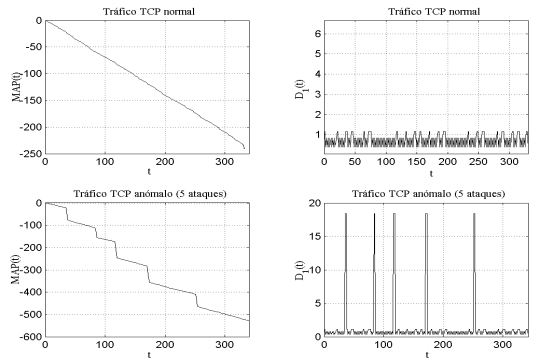


Figura 3: Gráficas comparativas de la salida producida por el detector (cadena HTTP) en presencia de sesiones con tráfico normal y anómalo. En las curvas de la parte inferior los ataques están localizados en los instantes de tiempo t=37, t=85, t=118, t=172, y t=235.

desempeñando así un papel equivalente al del parámetro de suavizado ε.

La curva inferior derecha de la Fig. 3 muestra la salida proporcionada por el detector (función $D_1(t)$) durante la monitorización de varias sesiones con presencia de tráfico intrusivo del tipo del descrito en la Sección 4.2 (concretamente, cinco variantes distintas de paquetes sonda para el escaneo de puertos). Resulta evidente cómo el detector ha capturado en este caso la aparición de anomalías en las conexiones. El rango de salida de la función $D_1(t)$ para los patrones normales oscila en el intervalo [0'0,3'0], mientras que las secuencias anómalas se manifiestan con valores superiores a 15'0 en todos los casos.

4.5 Discusión

De acuerdo con la metodología expuesta hasta ahora, el resultado de la etapa de entrenamiento es la obtención de un modelo de uso de TCP por parte de cada uno de los servicios considerados. El despliegue de detectores basados en este esquema se realizaría como ha sido comentado previamente: cada modelo aislado monitoriza el tráfico entrante destinado a la aplicación concreta.

No obstante, resulta razonable concebir un único modelo para el uso global del protocolo (TCP en el caso anterior), con independencia de la aplicación que lo utiliza. Tal modelo puede ser fácilmente construido siguiendo las mismas pautas de modelado, pero considerando todos los conjuntos de tráfico de entrenamiento sin discernir en función del puerto de destino de cada segmento.

La Fig. 4 muestra el modelo resultante obtenido tras la aplicación de este enfoque. Como era de esperar, el modelo obtenido corresponde a una "media" de los modelos individuales mostrados en la Fig. 2. Efectivamente, el conjunto de estados obtenido es la unión de los estados de los tres modelos individuales. Por otro lado, las nuevas probabilidades de transición

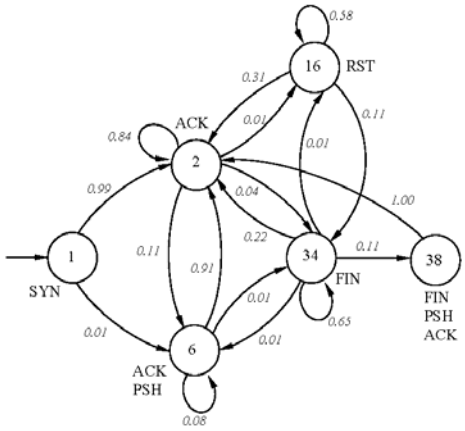


Figura 4: Modelo TCP obtenido con tráfico procedente de diferentes fuentes. Nótese cómo el modelo puede ser visto como una "media" de los modelos individuales mostrados en la Fig. 2.

pueden ser vistas como una media ponderada de las presentes en los originales. Este hecho se puede ilustrar con un ejemplo sencillo si consideramos la transición $S_2 \rightarrow S_6$. La probabilidad de esta transición es 0'66 en el caso de la cadena para FTP, 0'26 en el caso de HTTP y 0'05 para el caso SSH (véase Fig. 2). La correspondiente transición en el modelo global presenta una probabilidad de ocurrencia de 0'11 (véase Fig. 4). Es posible realizar un análisis análogo para el resto de transiciones.

La Fig. 5 muestra algunos resultados de evaluación del modelo global obtenido para la detección de usos anómalos del protocolo. En este caso, el valor del parámetro de suavizado ε ha sido de 10^{-9}. Aunque se observa claramente cómo el modelo detecta anomalías de igual forma que lo hacían los modelos individuales, es preciso hacer notar que los nuevos rangos de la señal de salida son distintos. Por ejemplo, el modelo correspondiente a HTTP produce valores inferiores a 2'0 para el tráfico normal y superiores a 17'0 para el anómalo. La evaluación del mismo tráfico con el nuevo modelo proporciona una salida inferior a 6'5 para las conexiones normales y superior a 8'5 para los usos anómalos.

Este fenómeno se encuentra directamente relacionado con la pérdida de especialización del modelo anteriormente discutida. No obstante, la precisión de la detección puede ser controlada tanto a través del parámetro de suavizado ε como de W_m. Por ejemplo, en el caso de modelos individuales, valores en torno a $\varepsilon = 10^{-6}$ eran suficientes para una separación correcta entre conexiones normales y anómalas. Sin embargo, para el caso del modelo global se requiere un valor de $\varepsilon = 10^{-9}$ o inferior para una discriminación precisa.

5 Conclusiones y Trabajo Futuro

En este trabajo han sido presentados los resultados de un nuevo esquema para la detección de anomalías en el uso de los protocolos de red. La aplicación de cadenas de Markov como modelo subyacente a la dinámica de un protocolo, en conjunción con el empleo de medidas de reconocimiento del tipo de las presentadas (*MAP* y *LogMAP*), proporcionan un marco de trabajo sólido y prometedor de cara a una investigación más extensa en esta línea.

El método expuesto ha probado su efectividad a lo largo de todos los experimentos preliminares llevados a cabo. En el caso de TCP, los resultados obtenidos tras el modelado son similares a los que pudieran ser derivados directamente de la especificación formal del protocolo (véase [17]). No obstante, la elaboración "manual" de tal modelo no es siempre factible por, al menos, dos razones. En primer lugar, aunque la especificación abstracta de cada protocolo esté siempre disponible y sea respetada como estándar, determinados aspectos concernientes al diseño suelen ser ambiguos y su uso concreto se delega en la implementación que cada fabricante efectúa del protocolo. Este hecho se manifiesta claramente en el comportamiento heterogéneo que las pilas de protocolos de diferentes sistemas operativos exhiben en ciertas circunstancias y que, entre otros hechos, permiten la identificación remota de la plataforma ([21], [19]). En segundo lugar, muchos de los protocolos comúnmente empleados no se sustentan sobre una máquina de estados similar a la existente para TCP o ARP, por citar algunos ejemplos. La construcción de un modelo obtenido directamente de la utilización específica que las aplicaciones hacen de los mismos es, en tales casos, aún más útil si cabe.

Adicionalmente a todo lo expuesto a lo largo de esta discusión, el despliegue de sensores basados en el modelado propuesto no debe ser concebido como una solución completa para las tareas de detección. Por el contrario, su uso en conjunción tanto con otras técnicas de detección de anomalías, como con métodos basados en firmas, es especialmente recomendado. Recuérdese que la utilización anómala de protocolos constituye únicamente una pieza dentro

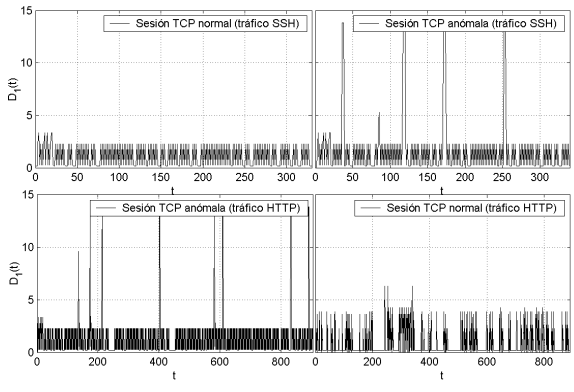


Figura 5: Salida proporcionada por el detector global durante la monitorización de dos sesiones SSH y dos HTTP. Aunque la precisión de la detección no ha cambiado, se puede observar cómo los rangos de la señal de salida son distintos.

del vasto arsenal de herramientas de intrusión existentes.

La evaluación de la viabilidad en entornos reales, la extensión a otros protocolos y la investigación de técnicas de cuantización de paquetes más generales constituyen las principales líneas de trabajo futuro. La particularización al caso de protocolos en la capa de aplicación (allí donde sea aplicable), resulta especialmente relevante dado el papel que esta capa desempeña en los problemas de seguridad existentes. Finalmente, la obtención de mecanismos eficientes de correlación entre los flujos de tráfico entrantes y salientes promete suponer un avance significativo en la mejora de la tecnología de detección existente.

Referencias

[1] J. Allen, A. Christie, W. Fithen, J. McHugh, J. Pickel, E. Stoner. "State of the practice of intrusion detection technologies". Technical Report CMU/SEI-99-TR-028. Software Engineering Institute, Carnegie Mellon, January 2000.

[2] S. Axelsson. "Intrusion Detection Systems: A Survey and Taxonomy". *http://citeseer.nj.nec.com/axelsson00intrusion.html*, 2000.

[3] D. Denning. "An intrusion-detection model". IEEE Transactions on Software Engineering, Vol. SE-13, No.2, pp. 222-232, February 1987.

[4] B. Mukherjee, L. T.Heberlein, K. N. Levitt. "Network Intrusion Detection". IEEE Network, Vol. 8, No. 3, May/June, pp. 26-41, 1994.

[5] M. Roesch. "Snort – lightweight intrusion detection for networks". Proceedings of the 1999 USENIX LISA conference, November 1999.

[6] V. Paxon. "Bro: A System for detecting network intruders in real-time". Proceedings of the 7th USENIX Security Symposium, San Antonio, Texas, 1998.

[7] C. Warrender, S. Forrest, B. Pearlmutter. "Detecting Intrusions Using System Calls: Alternative Data Models". Proceedings of 1999 IEEE Symposium on Security and Privacy, pp. 133-145, 1999.

[8] K. Llgun, R. A. Kemmerer, P. A. Porras. "State Transitions Analysis: A Rule-based Intrusion Detection Approach", 1995.

[9] S. Forrest, S. A. Hofmeyr, A. Somayaji, T. A. Logstaff. "A sense of Self for Unix process". Proceedings of 1996 IEEE Symposium on Computer Security and Privacy, pp. 120-128, 1996.

[10] S. Jha, K. Tan, R. A. Maxion. "Markov Chains, Classifiers, and Intrusion Detection". Proceedings of the 14th IEEE Computer Security Foundations Workshop, pp. 206-219, 2001.

[11] T. Lunt, A. Tamaru, F. Gilham, R.Jagannathan, P. Neumann, H. Javitz, A. Valdes, T. Garvey. *A real-time intrusion detection expert system (IDES) – final technical report*. Technical Report, Computer Science Laboratory, SRI International, Menlo Park, California, February 1992.

[12] J.B.D. Cabrera, B. Ravichandran, R. K. Mehra. "Statistical Traffic Modeling for Network Intrusión Detection". Proceedings of the 8th IEEE International Symposium on Modeling, Analysis and Simulation of Computer Telecommunication Systems, pp. 466-473, 2000.

[13] M. Bykoba, S. Ostermann, B. Tjaden. "Detecting Network Intrusions via a Statistical Analysis of Network Packet Characteristics". Proceedings of the 33rd IEEE Southeastern Symposium on System Theory, pp. 309-314, 2001.

[14] S. Zheng, C. Peng, X. Ying, X. Ke. "A Network State Based Intrusion Detection Model". Proceedings of the International IEEE Conference on Computer Networks and Mobile Computing, pp. 481-486, 2001.

[15] J. L. Doob. *Stochastic Processes*, John Wiley & Sons, 1953

[16] W. Feller. *An Introduction to Probability Theory and Its Applications, Vol. I*, 3rd Edition, John Wiley & Sons, 1968.

[17] J. Postel. "Transmission Control Protocol", RFC-793, September 1981.

[18] V. Jacobson, C. Leres, S. McCanne. *tcpdump*, *http://www.tcpdump.org*, June 1994.

[19] Fyodor. *Nmap – Free Stealth Port Scanner for Network Exploration & Security Audits*. *http://www.insecure.org/nmap/index.html*

[20] G. Vigna, R. A. Kemmerer. "NetSTAT: A Network-based Intrusion Detection System". *Journal of Computer Security*, Vol. 7, No. 1, pp. 37-71, 1999.

[21] Ofir Arkin. ICMP Usage in Scanning: The Complete Know-How, Version 3.0, Junio 2001. *http://www.sys-security.com*

Sistema de medida de tráfico IP en un *switch* Ethernet

Julián Fernández Navajas, Mª Jesús Clemente Clemente, Ángela Hernández Solana.
GTC (Grupo de Tecnologías de las Comunicaciones), I3A (Instituto de Investigación en Ingeniería de Aragón)
Universidad de Zaragoza
Teléfono: 976 761963 Fax: 976 762111
E-mail: navajas@posta.unizar.es anhersol@posta.unizar.es

Nowadays, one of the most used LAN technologies is Ethernet, but it has gradually invaded from multimedia traffic. This Kind of traffic demands certain Quality of Service (QoS), thus we will make the network designs in order to fulfil these requirements. Then we must evaluate the performance of networks already implemented. In this document we introduce a traffic test system based on the use of computers that incorporate a certain software analysis . This system is capable to make measurements with an acceptable degree or precision. In the second part of the document, we present the obtained results when the traffic test system is tried on an Ethernet network. These results let us deduce the network operation and foresee its behaviour with various traffic models.

1 Introducción

El diseño de redes es considerado un procedimiento previo indispensable para conseguir la correcta implementación de cualquier red. Como se expone en [1], este proceso de diseño sigue un orden lógico: primero, determinar el ancho de banda disponible en la red, después analizar el tipo y volumen de información que va ha ser transferida, finalmente diseñar las diferentes aplicaciones que la red va a soportar y decidir como se van a gestionar. Esto introduce una pregunta fundamental que deberá responder el estudio teórico: "¿Qué ancho de banda es necesario?" [2]. Los administradores de redes están preocupados porque la tecnología multimedia va a exigir demasiado ancho de banda y no se ha evaluado adecuadamente el posible impacto de las redes multimedia corporativas, cuyo tráfico necesita que se satisfagan unos requerimientos mínimos dada la naturaleza de su tráfico (imagen, audio, vídeo). Pero más importante que el diseño teórico, es el sistema de medida para evaluar el rendimiento de las redes que han sido implementadas considerando los requerimientos que se han mencionado.

Se debe tener en cuenta una cierta Calidad de Servicio (*QoS*) para el transporte de los datos generados por las aplicaciones [3]. Para ofrecer *QoS*, existen básicamente dos procedimientos: reserva de recursos y asignación de prioridades. La reserva de recursos implica que los dispositivos no podrán transmitir mientras no tengan asegurados los requerimientos de calidad de servicio preestablecidos. Ejemplos de redes o sistemas con reserva de recursos pueden ser *Frame Relay* (*FR*) y *ATM* [4, 5, 6]. La asignación de prioridades consiste en la reorganización que los nodos de la red hacen en sus colas de paquetes internas, basada en etiquetas de prioridad, que estarán relacionadas directamente con el tipo de tráfico y sus correspondientes requerimientos de QoS. En todo caso, un sistema basado en la asignación de prioridades está siempre restringido por la capacidad del medio físico. Por ello si se supera dicha capacidad, la asignación de prioridades no garantiza la QoS para cada usuario en términos absolutos, como mucho un reparto adecuado de la capacidad disponible en función de las prioridades. Por este motivo resulta aconsejable utilizar ambas posibilidades de forma complementaria, es decir, establecer prioridades sobre redes con reserva de recursos.

La red *Ethernet* conmutada es un ejemplo de red que no admite reserva de recursos, pero sí la asignación de prioridades. No obstante, en este caso el incremento del ancho de banda obtenido gracias a la evolución de esta clase de redes [7, 8, 9,10] hace que la reserva de recursos no sea tan importante. Por otra parte un coste y complejidad bajos la convierten en una alternativa de red interesante siempre que nuestros requerimientos de QoS se ajusten a las garantías de tráfico obtenidas. Así pues, uno de los objetivos fundamentales de este trabajo será demostrar que a pesar de que en las redes LAN conmutadas no se admite reserva de recursos, sí que es posible garantizar un ancho de banda mínimo y un retardo máximo, para determinadas configuraciones de conexión de sus puertos. Dado que el cálculo de los recursos garantizados depende del algoritmo de reparto seguido por el fabricante de cada dispositivo, y que dicho algoritmo no aparece comúnmente en las especificaciones del producto, un aspecto importante del trabajo será la extracción del mismo a partir del estudio del comportamiento del tráfico.

Para evaluar el comportamiento y rendimiento de estas redes, se propone medir parámetros como el retardo y la tasa de tráfico enviada. El retardo está presente en todas las transmisiones, por lo que es un factor a considerar para analizar la calidad de las comunicaciones, siendo especialmente importante para las realizadas en tiempo real. El cálculo del retardo, obtenido a partir de la diferencia entre el instante (especificado en una marca temporal) en que un paquete se recibe y el instante en el que se transmite, requiere la utilización de un sistema de medida preciso, donde receptor y transmisor estén perfectamente sincronizados.

El IETF (*Internet Engineering Task Force*) tiene un grupo cuya tarea es definir un sistema de medida de tráfico IP (IPPM, *IP Performance Metric*

Workgroup). Este grupo ha publicado el RFC 2330 [11], que incluye el desarrollo de un sistema de medida estándar que puede ser utilizado por operadores de red, usuarios o grupos independientes de prueba. Algunos conceptos introducidos por IPPM han sido utilizados como base de este trabajo.

Una primera clasificación de las medidas permite distinguir entre medidas activas (inyectando tráfico de prueba en la red cuyas características queremos analizar) y, medidas pasivas (utilizando tráfico que ya existe en la red). Las pruebas activos ofrecen una predicción de como será tratado el tráfico en la parte de la red examinada, pero se necesita generar modelos de tráfico adecuados para simular las condiciones de la red.

Un segundo punto de vista distingue entre medidas de un solo sentido (para tráfico UDP) y de dos sentidos (usadas cuando es necesaria una respuesta, como en TCP o ICMP).

En una tercera división, los sistemas de medida pueden ser clasificados considerando la manera en la cual se consigue el sincronismo entre los equipos de medida. De esta manera, se puede distinguir entre aquellos en los que existe una red dedicada para este propósito (la red puede ser un cable o una red complicada como es GPS [12], [13]) y los que utilizan las ventajas de los protocolos de sincronismo que pueden estar atravesando la red (como NTP en la red IP) [14].

Finalmente, es posible clasificar estos sistemas según su precisión y coste. Aunque existen dispositivos específicos de los principales fabricantes (*Acterna* [15], *Agilent* [16], *Anritsu* [17], etc.) capaces de medir el retraso con una gran precisión, este trabajo muestra que es posible implementar una solución más económica basada en el uso de ordenadores que incorporan un cierto software de análisis capaz de realizar medidas con un grado de precisión aceptable.

El análisis de dichas medidas, bajo determinadas condiciones de prueba, servirá para deducir el comportamiento genérico de los dispositivos utilizados en la construcción de la red LAN conmutada. Deducido el modo de operación real de dichos dispositivos, con respecto a la asignación de prioridades o tratamiento diferenciado de flujos de datos, será posible establecer a priori si bajo determinadas condiciones de tráfico es posible mantener o no un grado de QoS específico.

El resto del artículo se organiza como sigue. En la sección II y III se presentan el método de medida propuesto y los dispositivos utilizados. En la sección IV se presentan algunos resultados obtenidos con la aplicación de dicho sistema. Resultados que han permitido determinar el funcionamiento real de los dispositivos 3COM utilizados en la implementación de la red LAN conmutada así como las garantías de QoS proporcionadas por la configuración de red propuesta. Finalmente en la sección V se presenta un sumario de las conclusiones más relevantes.

2 Descripción general del sistema de medida

Se propone un sistema de medida de *QoS* mediante software. Para ello se captura un conjunto de paquetes a la entrada de la red que se va a analizar y se almacenan en un fichero junto al instante de tiempo en el que han sido capturados (*marca de tiempo)*. Por otro lado se capturan los mismos paquetes tras haber atravesado la red y se almacenan en otro fichero junto con su correspondiente *marca de tiempo.* La información temporal almacenada, permite calcular el retardo sufrido por los paquetes al viajar por la red. Este retardo nos permite deducir el funcionamiento de la misma, con lo que podremos prever su comportamiento con diferentes tipos de tráfico.

El sistema de medida se puede observar en Fig. 1. Se requieren 3 generadores de tráfico (dos de los cuales sólo se utilizarán para congestionar el sistema a analizar), y dos receptores (uno opcional según el escenario). El generador 1 (Gen1) y el receptor 1 (Rec1) se conectan a través del sistema bajo medida.

Todos los ordenadores tienen tarjetas de red 3Com Tornado PCI 3c905CTX 10/100 y usan el sistema operativo *Linux,* con diferentes versiones: 2.2.17 en generadores y receptores; 2.4.18-3 en el analizador, el cual incluye la versión 3.6 de *tcpdump* que se ejecuta junto con la versión 0.6 de *libpcap* para la captura de tráfico.

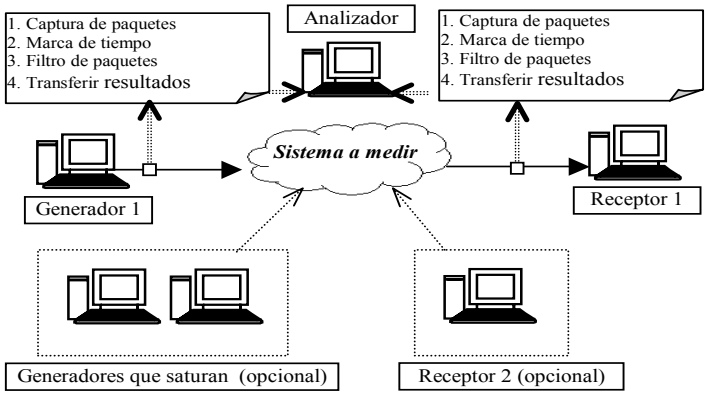


Fig. 1. Descripción del sistema de medida

Los generadores, que disponen de una sola tarjeta de red, son capaces de enviar tráfico siguiendo diferentes modelos para simular aplicaciones reales, incluso pueden elegir el *socket* por el cual quieren enviar. El analizador dispone de dos tarjetas de red, una de ellas se utiliza para capturar el tráfico que envía el Gen1 y la otra para capturar el tráfico que recibe Rec1 una vez que ha atravesado la red.

Este equipo, siguiendo las recomendaciones de RFC 2330, implementa un proceso de medida dividido en cuatro pasos principales (ver Fig. 1):

1. Captura de paquetes: El peor o mejor comportamiento del analizador durante la captura de paquetes está limitado por:

- **El número de interrupciones generadas por la tarjeta de red**. Esto depende del número de paquetes enviados desde el hardware al *Kernel*. Este límite puede ser minimizado almacenando más paquetes en el buffer de la tarjeta de red, aún cuando esto pudiese disminuir la precisión de la marca de tiempo (generalmente, esta función la realiza el *Kernel* y no la tarjeta de red).
- **Número de paquetes procesados por el *Kernel*** y pasados de su espacio al espacio de usuario: esto implica diversos cambios de proceso.
- **Número de bytes transferidos al espacio de usuario.**
- **Carga al sistema generada por otros procesos.**

2. *Marca de tiempo:* hay que tener en cuenta muchos factores para calcular el retardo, principalmente el problema de la sincronización, ya que se necesitan dos puntos de medida distintos. La existencia de colas internas en el hardware y en el *Kernel*, generan retardos adicionales en la medida. Aunque todos los dispositivos implicados tengan el mismo hardware y el mismo sistema operativo, los paquetes pueden sufrir diferentes retardos (p.e. debido a la carga de CPU o a los niveles de los *buffers*). Para reducir al máximo estos retardos se debe asignar la marca de tiempo lo antes posible.

3. *Filtrado de paquetes:* se utiliza en caso de que no todos los paquetes capturados sean necesarios.

4 Transferencia de resultados: se transfieren los datos para calcular el retardo. Para disminuir el número de datos transferidos, éste cálculo lo debería hacer el analizador, el cual selecciona la información que puede interesarnos para analizar el comportamiento del sistema medido.

3 Materiales y métodos

A continuación se va a describir las herramientas y la metodología de uso de las mismas, que permiten la obtención de resultados en el presente trabajo.

3.1 Herramientas para el test

Los principales elementos del sistema de medida son el analizador y los cables UTP utilizados para *"absorber"* (esnifar) el tráfico.

El analizador desempeña dos funciones fundamentales: capturar el tráfico IP y actuar como referencia de tiempo. Ésta última es especialmente importante al calcular parámetros diferenciales.

Para capturar tráfico ejecuta el programa *tcpdump*. Éste utiliza la librería *libpcap*, que se vale del sistema *Berkeley Packet Filter* (BPF) para capturar paquetes siguiendo los siguientes pasos:

- *tcpdump* permite a BPF escuchar el interfaz de red. La tarjeta de red llama a BPF al llegar un paquete.
- Si el paquete es aceptado, BPF lo copia en el buffer asociado con el filtro y añade la marca de tiempo. Esto se lleva a cabo en el espacio del *Kernel*.
- Cuando *tcpdump* consigue la CPU y lee el buffer, libera uno o más paquetes y procesa sus contenidos. Esto se lleva a cabo en el espacio de usuario.

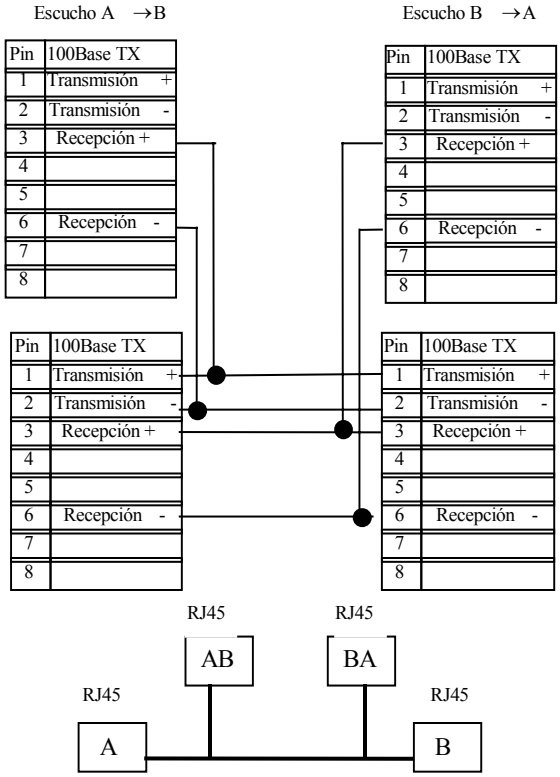


Fig. 2. Cable UTP modificado

Los cables UTP han sido modificados, lo que permite escuchar tanto la entrada como la salida del sistema, pero en ningún momento permiten inyectar tráfico. Estos cables pueden ser utilizados para recibir tráfico en ambas direcciones, pero al necesitar solo una de ellas, la otra quedará sin conectar. En Fig. 2 se puede ver la apariencia física del cable UTP utilizado. Si deseáramos analizar el tráfico en ambas direcciones a la entrada y a la salida del sistema a medir (p.e. TCP), necesitaríamos 4 tarjetas de red en el analizador.

3.2 Metodología

Con este sistema se pueden hacer dos tipos de medidas, en una o en dos direcciones. En principio es interesante caracterizar cada dirección por separado, para después analizarlas al mismo tiempo.

El analizador ejecutará dos procesos *tcpdump* simultáneos para escuchar la transmisión y la recepción en la misma máquina, lo cual permite la sincronización necesaria para calcular el retardo. Cada uno de los pasos del proceso genérico de prueba descrito en la sección II se han optimizado:

1. *Captura de paquetes*: la carga en el analizador de tráfico se minimiza ejecutando únicamente los dos procesos *tcpdump*. Otro parámetro que disminuimos es el número de bytes transferidos mediante al opción "-s" de *tcpdump*, con la que definimos el número exacto de estos bytes. Además la opción "-w" guarda la información en un fichero y no la muestra por pantalla.
2. *Marca de tiempo*: la sincronización entre los dos procesos *tcpdump* se garantiza porque ambos utilizan el mismo reloj (el de la máquina que actúa como analizador)
3. *Filtrado de paquetes:* el tráfico innecesario se filtra mediante las opciones de *tcpdump*
4. *Transferencia de resultados:* el tráfico capturado no necesita ser transferido, ya que el analizador tiene capacidad suficiente para almacenarlo, procesarlo y analizarlo.

El primer paso es lanzar dos procesos *tcpdump* en el analizador. Una tarjeta escucha el tráfico que genera Gen1, mientras la otra escucha el tráfico que recibe Rec1. Una vez lanzados estos procesos, los generadores comenzarán la transmisión durante un tiempo determinado. Los paquetes transmitidos utilizan el protocolo UDP (por lo que se estudia el sistema en un solo sentido) y tienen un número de secuencia que varía progresivamente desde 0 hasta 65535. En el momento en que la transmisión ha finalizado, se detienen los procesos *tcpdump*, los cuales habrán almacenado la información capturada en dos ficheros diferentes.

Estos ficheros serán definidos como entrada en un *script* desde donde se lanzan varios programas en C. Estos programas proporcionan diferentes parámetros, por ejemplo: retardo de los paquetes de Gen1, número de paquetes que hay dentro de la cola de salida del sistema cuando llega un paquete de Gen1, reparto de ancho de banda de salida del sistema, pérdidas de Gen1 etc. Esta información es representada gráficamente para obtener una visión general de lo ocurrido durante toda la transmisión.

4 Resultados

El sistema de medida descrito anteriormente se ha probado en diversos escenarios cuyo elemento común es el *Switch 4400 de 3COM*. Éste introduce una novedad respecto a los *switch* que 3COM comercializaba hasta el momento: "dar prioridad al tráfico". En todos los escenarios, el sistema de medida se ha conectado a Gen1 y a Rec1.

4.1 Escenario de análisis de las colas de salida

El primer escenario se muestra en la Fig. 3. Éste nos permite conocer tanto el tiempo de conmutación del *switch*, como el comportamiento de éste en situación de saturación.

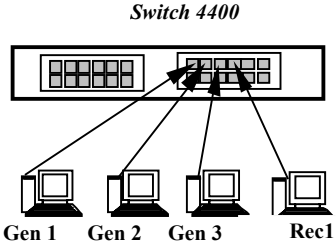


Fig.3 Escenario de análisis de las colas de salida

A. Modelo de retardo del tráfico

Como se puede observar en la Fig. 4, existen únicamente dos factores que influyen en el Retardo Total (T_R) : el *Tiempo de transmisión* y el *Tiempo de switch*. Sin embargo, sólo uno de ellos, *Tiempo de Switch*, será objeto de medida por el sistema diseñado.

- **Tiempo de transmisión (T_{trx}):** Este valor depende directamente de la longitud de la trama y de la velocidad del enlace. (Switch 4400 permite velocidades de 10 ó 100 Mbps).

Para calcularlo habrá que añadir 26 bytes, correspondientes a la cabecera *Ethernet*, a la longitud del datagrama IP. Por ejemplo, para un datagrama IP de tamaño 500B (bytes) y utilizando una velocidad de transmisión de 10Mbps, el valor teórico de T_{trx} se obtiene:

$$T_{trx} = (500 + 26) * 8 / 10\text{Mbps} = 0.420\text{ms}$$

- **Tiempo de switch (T_{SW}):** Este valor corresponde al tiempo que permanece la trama dentro del switch antes de ser retransmitida por el puerto de salida correspondiente. T_{SW} se puede dividir en dos partes:

- **Tiempo de conmutación** (T_{con}), es decir, el tiempo de encaminamiento de la trama desde el puerto de entrada hacia el puerto de salida.
- **Tiempo de espera** (T_{esp}), es decir, el tiempo que la trama permanece en la cola de salida hasta que le toca ser retransmitida. Este valor depende de la cantidad de tramas almacenadas en dicha cola, así como del tamaño de las mismas. Por tanto, es un tiempo variable, que puede llegar a tomar un valor nulo en situaciones de bajo tráfico.

El diagrama de tiempo representado en Fig. 4 no ha sido construido a escala, con el fin de poder observar los distintos parámetros que intervienen en las medidas.

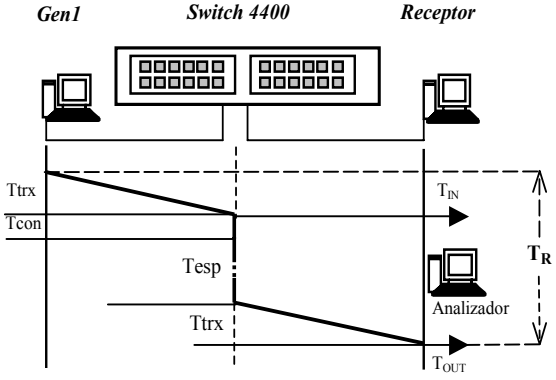


Fig. 4 Modelo de retardo

El analizador nos proporciona en todos los casos el valor de T_{OUT} y el de T_{IN} (mediante las marcas de tiempo asignadas a un paquete capturado a la entrada y a la salida del switch). Como se puede ver en Fig.4, T_{OUT} - T_{IN} está formado por la suma del tiempo que le cuesta al paquete atravesar el switch (T_{SW}) (que a su vez es la suma de T_{con} y de T_{esp}) y T_{trx} hasta el receptor. Por tanto:

$$T_{OUT} - T_{IN} \text{ (valor proporcionado por el analizador)} = T_{SW} + T_{trx} = T_{con} + T_{esp} + T_{trx.}$$

B Análisis del switch sin dar prioridad al tráfico

En primer lugar estudiamos diferentes características de la conmutación del *switch* en una **situación de no saturación** del enlace de salida. Para ello, se realizan dos pruebas distintas. En ambas pruebas se envía tráfico durante 10sg únicamente desde Gen1 hacia Rec1. En la prueba 1 se envía un tráfico con las características que muestra la tabla 1, mientras que en la segunda el tráfico se caracteriza según la tabla 2.

	Gen1	**Gen2**	**Gen3**
BW	2 Mbps	0	0
Lon. datos UDP	500 B	0	0

Tabla1. Parámetros prueba 1

	Gen1	**Gen2**	**Gen3**
BW	2 Mbps	0	0
Lon. datos UDP	1000 B	0	0

Tabla 2.Parámetros prueba 2

Para cada uno de los paquetes se calcula T_{OUT} - T_{IN}. Posteriormente, se halla la media y la desviación típica del conjunto de valores obtenidos, resultando una desviación típica nula en ambas pruebas y una media:

$T_{OUT} - T_{IN} = 0.454$ ms prueba 1

$T_{OUT} - T_{IN} = 0.853$ ms prueba 2

Al aumentar el tamaño de las tramas se ha incrementado en la misma proporción el retardo sufrido por el paquete, esto nos indica que el método de conmutación es *Store and Forward.*

Conocido el método de transmisión, podemos hallar el tiempo de conmutación introducido por el *switch*. Como sólo Gen 1 transmite hacia el receptor, la cola de salida no tendrá almacenado ningún paquete. Esto nos permite obtener T_{con} al medir T_{SW} ya que T_{esp} tomará, bajo estas condiciones, un valor nulo.

Utilizando los resultados obtenidos de la prueba1:

$T_{OUT} - T_{IN} = 0.4532$ ms

donde

$$T_{OUT} - T_{IN} = T_{trx} + T_{con} + T_{esp}$$

Para calcular T_{trx}, hay que tener en cuenta, que la longitud que se observa en la tabla, se refiere a datos UDP. Por tanto, se añade la cabecera UDP (8 B), la cabecera IP (20 B), y la cabecera *Ethernet* (26 B).

$T_{trx} = (500 + 8 + 20 + 26) * 8 /10$ Mbps $= 0.4432$ ms

$T_{esp} = 0$

$T_{con} = 0.4532 – T_{trx} - T_{esp} = 0.4532 - 0.4432 = 10$ μsg

Con lo que concluimos, que T_{con} es prácticamente nulo.

A continuación se analiza el comportamiento del *switch* bajo una **situación de saturación** del enlace salida en la que ningún tráfico se trata como prioritario. Para ello se han almacenado diversas medidas en el analizador. En cada una de ellas se ha transmitido tráfico durante 10sg desde los tres generadores hacia Rec1, con tasas de 300Kbps, 1Mbps, 4Mbps, 6Mbps y 10Mbps, así como con paquetes de longitud (datos UDP) 18B, 20B, 30B, 33B, 34B, 35B, 36B, 37B, 38B, 39B, 40B, 80B, 100B, 300B, 500B, 700B, 1000B, 1400B Y 1472B.

El primer resultado a destacar es la ausencia de control de flujo. Las medidas anteriormente citadas se han realizado configurando el *switch* de dos maneras: sin control de flujo y con control de flujo. En ambos casos se ha obtenido el mismo resultado: existen paquetes recibidos por el *switch* que no son retransmitidos hacia el puerto de salida, es decir se pierden, lo cual indica que el *switch* no hace control de flujo, aunque permita su configuración.

Antes de calcular cual es el valor de T_{esp}, es interesante averiguar la política de encolamiento que sigue el *switch*. Esto nos ayudará a comprender por qué existen pérdidas y qué valor pueden llegar a tomar. Para ello utilizamos las gráficas de "número de paquetes en cola" que disponemos con el sistema de medida diseñado, así como los valores de retardo obtenidos para cada paquete transmitido.

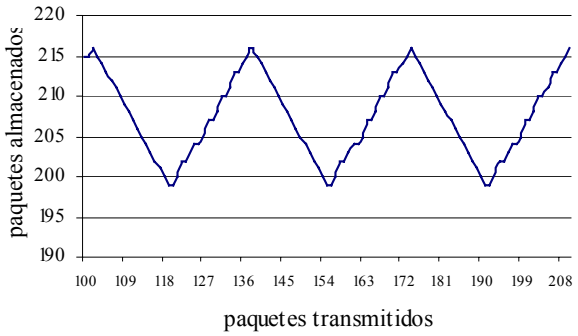


Fig. 5. Capacidad de las colas de salida

Tras analizar los valores que toman los retardo de cada uno de los paquetes, observamos que en una situación de saturación del enlace de salida, el *switch* es capaz de almacenar hasta 216 paquetes. Una vez ha llegado a esta situación, desecha todos los paquetes que van hacia ese puerto hasta tener almacenados solamente 199. Este comportamiento se puede observar en Fig. 5, donde se representa la ocupación de la cola de salida en función del número de paquetes retransmitidos.

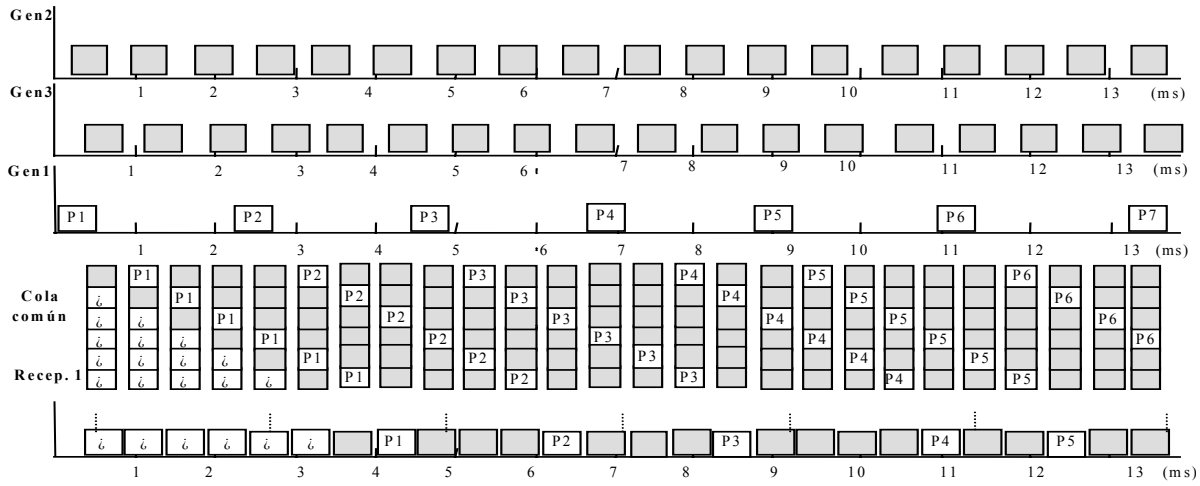


Fig. 6. Modo de trabajo del switch con la introducción de prioridades

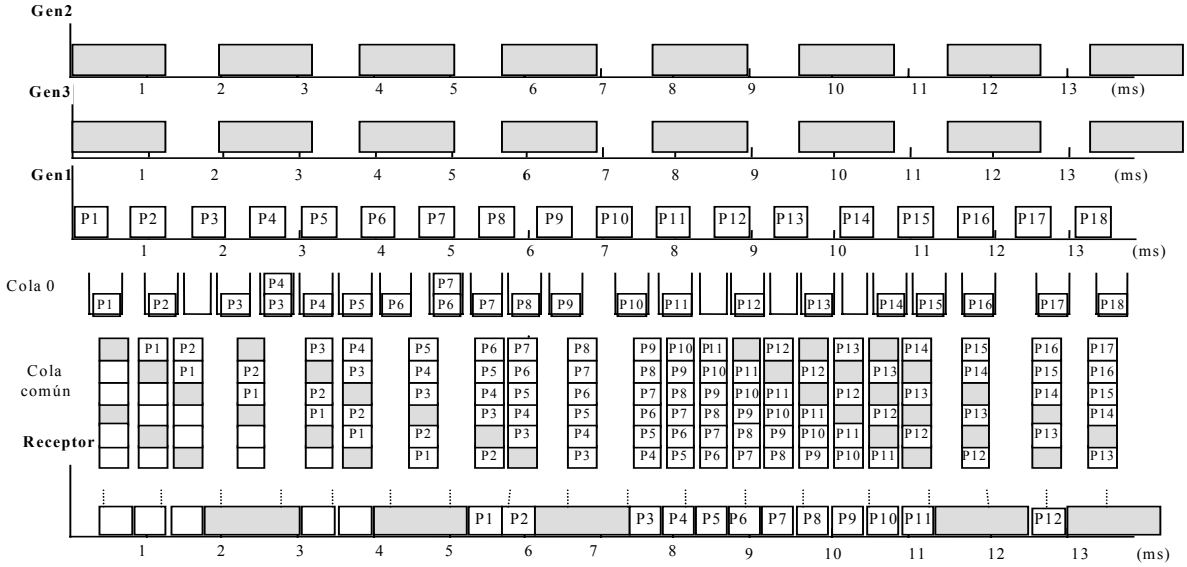


Fig. 7. Modo de trabajo del switch con prioridad y tamaño grande de paquetes

A continuación se estudia el reparto del ancho de banda del enlace de salida entre los 3 generadores, así como el tiempo que los paquetes esperan en la cola de salida para ser retransmitidos.

T_{esp} está constituido por el tiempo que dura la transmisión de los paquetes que contiene la cola, el cual depende a su vez del tamaño de los mismos.

$$T_{esp} = \sum_{i=1}^{i=N} (L_i + 8 + 20 + 26) * 8 / BW_{enlace\ salida}$$

Donde:
L_i = Longitud del paquete i
N= número de paquetes que tienen la cola en ese instante
Normalmente, en un escenario real, no se conoce el tamaño de los paquetes que ocupan la cola de salida, por lo que no podremos calcular T_{esp}. Sin embargo, podemos asegurar que:

$T_{esp}|_{MAX} = 216 * (1472 + 8 + 20 + 26) * 8 / BW_{enlace\ salida}$

Si $BW_{enlace\ salida} = 10Mbps \rightarrow T_{esp}|_{MAX} = 263.69ms$.

Bajo esta forma de trabajo, se ha observado que el ancho de banda de salida se reparte entre los distintos generadores según el ancho de banda solicitado:

$$BW_{Ai} = BW_{enlace\ salida} * \frac{BW_{Ai}}{\sum_{i=1}^{i=N} BW_{Si}}$$

Donde:
BW_{Ai} = Ancho de banda asignado al Gen_i
BW_{Si} = Ancho de banda solicitado por Gen_i
N = número de generadores que transmiten hacia el puerto de salida saturado.

C Análisis del *Switch* priorizando el tráfico

Bajo el mismo escenario damos prioridad, mediante la configuración de las tablas internas del *switch*, al tráfico procedente de Gen1. Como en el caso anterior, se envía tráfico durante 10sg desde los tres generadores hacia Rec1. Esta transmisión se repite en diversas ocasiones utilizando tráfico con tasas comprendidas entre 1Mbps y 10Mbps incrementadas en pasos de 1Mbps. El tamaño de las tramas (datos UDP) utilizadas toma los siguientes valores: 46B, 146B, 500B y 1146B.

El manual del *switch 4400* [19] nos informa sobre la existencia de cuatro colas en cada puerto de salida, en las que se almacenará el tráfico según su prioridad y el cual será retransmitido según una política *Weighted Round Robin (WRR):*

- Cola 0 : niveles de prioridad 0, 1 y 2
- Cola 1 : niveles de prioridad 3 y 4
- Cola 2 : nivel de prioridad 5
- Cola 3 : niveles de prioridad 6 y 7

El resultado de una de las pruebas realizadas, se muestra en Fig.6. Los parámetros utilizados son los que se muestran en la tabla.3. Al observar el orden de salida de los paquetes, nos percatamos de que entre la recepción de un paquete prioritario y su retransmisión, son transmitidos 6 paquetes que pueden o no tener prioridad. De esto deducimos la existencia de una cola de salida con capacidad de 6 paquetes, común para todo el tráfico. Cuando un paquete de dicha cola es transmitido, deja un hueco que será rellenado según la política WRR anteriormente mencionada.

	Gen1	**Gen2**	**Gen3**
BW	2 Mbps	6 Mbps	6 Mbps
Lon. datos UDP	500 B	500 B	500 B

Tabla.3 Parámetros en Fig.6. y escenarios 2 y3

Si aumentamos el tamaño de los paquetes no prioritarios (ver Fig.7), observamos como algunos paquetes con prioridad (por ejemplo P4) esperan la retransmisión de 7 paquetes en lugar de 6. Este efecto se produce porque durante la retransmisión de un paquete de longitud grande, llegan dos paquetes prioritarios (P3 y P4), por lo que P4 debe esperar en su cola de almacenamiento correspondiente a que P3 pase a la cola común de salida antes de pasar él. Esto conlleva la espera de la transmisión de un paquete más.

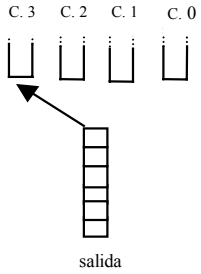


Fig. 8. Modelo de colas de salida

De todos los resultados analizados, deducimos la existencia de un sistema de colas tal y como se muestra en Fig.8. Existe una cola común para todo el tráfico, la cual se va llenando con los paquetes almacenados en cuatro colas según una política WRR. El tráfico es almacenado en una u otra cola según su nivel de prioridad (según dice el manual y según se ha comprobado).

Salvo en los casos en que el tráfico prioritario excede la capacidad del enlace de salida, se le concede todo el ancho de banda solicitado, pero el ancho de banda no es el único requerimiento de un tráfico prioritario, también lo es el retardo, el cual debe mantenerse por debajo de un cierto valor máximo para asegurar una determinada QoS.

Para calcular $T_{esp|MAX}$ hay que tener en cuenta un T_{guarda} desde que el paquete llega al *switch*, hasta que éste lo considera dentro de la cola prioritaria. Por tanto, si un paquete no llega T_{guarda} antes del comienzo de la retransmisión de un paquete, deberá esperarse a la retransmisión del siguiente paquete de la cola de salida para ser introducido en ella. La peor situación, es decir $T_{esp|MAX}$ se obtiene cuando tanto la cola de salida, como la cola de almacenamiento no prioritaria están ocupadas por paquetes de longitud máxima (1526 *bytes*) y el paquete prioritario no llega T_{guarda} antes del comienzo de la retransmisión de un paquete de la cola de salida. En esta situación deberá esperar durante la retransmisión de 7 paquetes de 1526 *bytes*.

$T_{esp|MAX} = T_{guarda} + 7 * 1526*8/ BW_{enlace\ salida}$

Si $BW_{enlace\ salida}$ =10Mbps $T_{esp|MAX}$ =8.55ms + T_{guarda}

Si $BW_{enlace\ salida}$ =100Mbps $T_{esp|MAX}$= 0.855ms+T_{guarda}

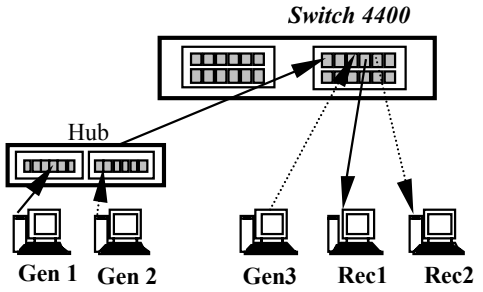


Fig. 9 Escenario de análisis de colas de entrada sin dar prioridad al tráfico

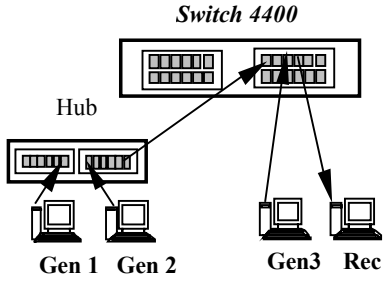


Fig. 10 Escenario de análisis de entrada dando prioridad al tráfico

4.2 Escenarios de análisis de las colas de entrada

Una vez estudiado el comportamiento de las colas de salida, es interesante conocer como actúan las colas de entrada. Para conseguir este objetivo se ha utilizado un escenario de análisis de las colas de entrada en una situación en la que no se da prioridad al tráfico (Fig. 9) y un escenario de análisis de las colas de entrada en una situación en la que sí se da prioridad al tráfico (Fig. 10).

A Análisis de colas de entrada sin priorizar tráfico

Durante 10sg se envía tráfico con las características que muestra la tabla.3. Mientras Gen1 transmite hacia Rec1, Gen2 y Gen3 lo hacen hacia Rec2 saturando el enlace de salida. Aunque en principio se pueda pensar que esto perjudicaría la transmisión desde Gen1 hacia Rec1, esto no ocurre así. Los paquetes de Gen2 que no son admitidos por la cola de almacenamiento del puerto de salida hacia Rec2, son eliminados. Es decir, no permanecen en la cola de entrada, lo cual haría que también los paquetes de Gen 1 se perdiesen. De esta manera se perjudica únicamente el tráfico que genera la congestión.
Por tanto, la única función que desempeña la cola de entrada es recibir el paquete. En el momento en que se recibe, se manda a la cola de salida correspondiente. Si ésta no admite nuevos paquetes, es eliminado. De esta forma la congestión en un puerto de salida no se propaga hacia los demás.

B Análisis de colas de entrada priorizando tráfico

Durante 10sg se envía tráfico con las mismas características que en el escenario 2, pero priorizando el tráfico transmitido por Gen1. En este caso todos los generadores transmiten hacia Rec1. Los resultados obtenidos son equivalentes a los del escenario 2. La saturación producida por el tráfico no prioritario, no perjudica al tráfico que si lo es. Los paquetes de Gen2 que no son admitidos por la cola 0 de almacenamiento son eliminados, por lo que no colapsan la entrada que comparte con Gen1. Cuando Gen1 transmite, la cola de entrada no está ocupada y su cola de almacenamiento (C. 4.) no está saturada, por lo que no percibirá la saturación generada por el tráfico no prioritario.

5 Conclusiones

En este trabajo se ha demostrado que es posible construir un sistema que permite medir retardos, tasas de transferencia, pérdidas, etc. ocurridos al atravesar diferentes tráficos IP una red *Ethernet* (compuesta por un *switch*). Este sistema, basado principalmente en un software ejecutado en una máquina analizadora, ha resultado ser bastante sencillo y económico. Esta cualidad le permite competir ventajosamente con otros sistemas de medida presentes en el mercado.

Además, el sistema ha sido probado con el fin de obtener información acerca de un dispositivo particular, el *switch* 4400 de 3com. Esta información obtenida sobre un equipo determinado no está habitualmente disponible para los usuarios. En nuestro caso particular nos ha permitido completar la información proporcionada por el fabricante y así poder hacer un mejor uso del producto.

En definitiva, el sistema de medida presentado, junto con la correcta interpretación de la información obtenida, puede resultar muy útil para una mejor utilización de múltiples equipos de red.

Agradecimientos

El presente trabajo ha sido posible gracias a la financiación de los proyectos CICYT (TIC2001-2481 y TIC2002-04495-C02), financiados por FEDER y el Ministerio de Ciencia y Tecnología.

Referencias

[1] Soble JS, Yurow G, Brar R, Stamos T, Neumann A, García MJ, Soddard MF, Chrian PK, Bhamb B, y Thomas JD, "Comparison of MPEG digital video with super VHS tape, for diagnostic echocardiographic readings", *Journal of American Society of echocardiography*, vol. 11, no. 8, pp. 819-25, Aug 1998.

[2] Tolly K, "Networked multimedia: How much bandwidth is enough?", *Data* Communications, no. 23, pp. 44-51,Sep 1994.

[3] Stephenson A., "DIFFSERV and MPLS: A quality choice (Tech Tutorial)", *Data* Communications, vol. 27, no. 17, pp. 73-77, Nov 1998.

[4] Greenfield D., "ATM, the services are real. What about the savings?", *Data* Communications, vol. 27, no. 2, pp. 48c-48j, Feb 1998.

[5] Weiser J Chacko AK Radvany M Gac RJ, Harding D y Romlein JR, "Traffic and trend analysis of local and wide area networks for a distributed PACS implementation", en *Proceedings of SPIE – The International Society for Optical Engineering 3980,* 2000, pp. 447-57.

[6] Zhang L. Wang W. Subramanian and He JH, "Medical Imaging Transmission for Telemedicine over ATM Networks", in *Proceedings of Applied Telecommunications Symposium (ATS´00). (Bodnar B and Sharon A. eds.) SCS, San Diego CA, 2002, pp. 267-72.*

[7] Deval Sh. Mandeville R., "GIGABIT Ethernet. Get It Done (Lab Test)", *Data* Communications, vol. 27, no. 2, pp. 66-81, Feb 1998.

[8] Dornan A., "Ethernet makes a run for the WAN", *Network Magazine,* pp. 74-81, Oct 200

[9] Newmqn D. Kumar S., "Ethernet switches: Quantity, not comodity (Lab Test)", *Data* Communications, Nov 1996.

[10] Mandeville R.Newman D., "Traffic tuners: Striking the right note? (Lab Test)", *Data* Communications, vol. 27, no. 17, pp.51-60, Nov 1998.

[11] Paxson, V., Almes, G., Mahdavi, J. y Mathis, M., "Framework for IP Performance Metrics", RFC 2330, USC/ Information Sciences Institute, mayo de 1998. http://www.ietf.org/rfc/rfc2330.txt.

[12] Langely Rb., "Time, Clocks, and GPS", GPS World, vol. 2, no. 10, pp. 38-42, Nov-Dic 1991.

[13] SchildKnecht T. And Dudle G., "Time and Frequency transfer: High precision using GPS phase measurements" GPS World, vo. 11, no. 2, pp. 48-52, Feb 200.

[14] D. Mills, "Network time synchronization project", *http://www.eecis.udel.edu/mills/ntp.htm.*

[15] Acterna LLC. http://www.acterna.com/products/dominio/index.html

[16] Agilent Technologies. http://advanced.comms.agilient.com/routertester/index.htm

[17] Anritsu. http://www.anritsu.com

[18] RFC 1483. http://www.anritsu.com

[19] Manual de usuario SuperStack 3 Switch4400 3C17203, 3C17204 de 3com

Contribución a la caracterización de variables de teletráfico en redes FCA urbanas mediante simulación

Israel Martín-Escalona, Francisco Barceló, Enrica Zola

Dept. d'Enginyeria Telemàtica de la Universitat Politècnica de Catalunya

Av. Canal Olímpic s/n, Barcelona 08860, Spain.

imartin@entel.upc.es ; barcelo@mat.upc.es ; enrica@entel.upc.es

Abstract— *In this paper, a software simulator is used to achieve a complete modeling of cellular networks using Fixed Channel Allocation. This simulation tool has been adapted to gather information about most of teletraffic variables that are involved in cellular-system evaluations: channel holding time, time between two consecutive handoff arrivals and the time until handoff completion (handoff delay). In addition, information about the Quality of Service offered by the system is also provided. A large set of scenarios has been simulated and the results obtained from their study have been presented and analyzed.*

1 Introducción

El estudio de variables de teletráfico en redes celulares se presenta como un proceso muy distinto al llevado a cabo para el caso de redes fijas. La división en celdas de este nuevo tipo de redes altera la manera en que se producen las llegadas al sistema, así como la duración del tiempo de ocupación del canal. De esta forma, en redes celulares el tiempo de ocupación del canal pasa a ser una fracción del tiempo de servicio, produciéndose un proceso de handoff cada vez que una estación móvil (MS) supera la zona de cobertura ofrecida por una estación base (BS). Estas características propias de las redes celulares motivan la aparición de nuevas variables de teletráfico, tales como el proceso de llegadas de handoff y la duración de un handoff. De esta forma, es necesario llevar a cabo una caracterización de este nuevo conjunto de variables, si se pretende obtener modelos precisos que permitan el dimensionado de redes celulares.

Recientemente se han publicado muchos estudios que analizan algunas de las variables mencionadas anteriormente, mediante procedimientos analíticos [1], simulaciones [2] y estudios de campo [3]. Una de las variables que más estudios aglutina es el tiempo de ocupación del canal [3, 4, 5]. Sin embargo, variables como el proceso de llegadas de handoff [6, 7] o la duración de los mismos [8, 9] no han sido estudiados en igual medida. Muchos de estos estudios consideran una única variable, sin proporcionar datos sobre el resto. Normalmente, esta información puede ser encontrada en diversos estudios basados en distintos escenarios, suposiciones y metodologías. Otro tipo de información como puede ser la calidad de servicio (QoS) ofrecida por el sistema, no acostumbra a ofrecerse en este tipo de estudios.

El principal objetivo de este artículo es la simulación de una red celular enmarcada en un entorno urbano, para de esta forma caracterizar un conjunto representativo de variables de teletráfico bajo las mismas suposiciones e idénticos escenarios. También se proporciona la información referente a la QoS también es obtenida: tráfico cursado, probabilidad de bloqueo e interrupción, movilidad, etc. Debido a las restricciones impuestas por el documento, tan sólo se han incluido resultados referentes al proceso de llegadas de handoff y la duración de dichos procesos (tiempo que la MS pasa en la zona de handoff), para el caso de escenarios urbanos bajo un modelo Manhattan. El lector puede obtener más información referente al resto de variables y distintos escenarios (Ej. hexagonal) en [10, 11].

Este artículo queda estructurado de la siguiente forma. En la sección 2 se definen cada una de las variables que van a ser estudiadas. La sección 3 describe las principales características del simulador utilizado, así como cada uno de los entornos que serán simulados. La sección 4 muestra la metodología seguida para llevar a cabo el estudio de las variables de teletráfico y los resultados obtenidos tras su caracterización. Finalmente, en la sección 5 se presentan las conclusiones alcanzadas por los autores.

2 Variables consideradas

Los siguientes párrafos definen el conjunto de variables estudiadas en este documento.

2.1 Variables de teletráfico

1) Tiempo entre dos llegadas de handoff consecutivas. Esta variable mide el tiempo transcurrido entre dos llegadas de handoff consecutivas a la misma celda. Esta variable no tendrá en cuenta los reintentos asociados al proceso de handoff, en caso de que éste no se conceda de manera inmediata.

2) Tiempo en la zona de handoff. Esta variable aleatoria contiene la duración del proceso de handoff, es decir, el tiempo que una MS está en la zona de degradación que motiva la petición de traspaso. De acuerdo con esto, esta variable tomará por valor el tiempo transcurrido desde que se produce una petición de handoff, hasta que ésta se concede o la llamada finaliza. Nótese que la finalización de la llamada puede ocurrir tanto si no se logra el handoff, como si expira el tiempo de servicio.

2.2 Variables de QoS

1) Probabilidad de bloqueo (BCP). Esta variable mide la probabilidad de que una nueva llegada (no se consideran las llegadas de handoff) no pueda ser aceptada, debido a la falta de recursos disponibles en la celda. Esta variable se calcula mediante la Ecuación (1), en la que *BC* hace referencia al número de llamadas de nueva generación que han sido bloqueadas:

$$BCP = \frac{BC}{Número\ \ de\ llamadas\ \ recibidas}. \qquad (1)$$

2) Probabilidad de interrupción (ICP). Esta variable es la probabilidad de finalización forzada, es decir, la probabilidad de que una llamada que se encuentra inmersa en un proceso de handoff sea interrumpida, debido a que no se ha materializado el traspaso. El simulador utilizado emplea la Ecuación (2) para estimar esta variable. En dicha ecuación *IC* representa el número de llamadas interrumpidas.

$$ICP = \frac{IC}{Número\ \ de\ llamadas\ \ cursadas}. \qquad (2)$$

3) Grado de Servicio (GoS). Esta es una función de coste que penaliza el hecho de que las interrupciones son mucho más molestas para el usuario que los bloqueos [11, 12]. La función de *GoS* se utiliza como referencia para medir el grado de servicio ofrecido por el sistema. La expresión utilizada para calcular esta variable es la siguiente:

$$GoS = \frac{BC + 10\ x\ IC}{Número\ \ de\ llamadas\ \ recibidas}. \qquad (3)$$

3. El simulador

El estudio de las variables presentadas en la sección 2 se lleva a cabo mediante el uso de una herramienta de simulación. Estudios anteriores ya optaron por esta vía para realizar sendos análisis [2, 5, 13, 14 ,15]. El tiempo de ocupación del canal ha sido simulado en la gran mayoría de estos trabajos, prestando poca atención a otras variables, cuya caracterización se hace imprescindible en procesos de dimensionado: llegadas de handoff y tiempo en la zona de handoff. El simulador propuesto en este estudio realiza un análisis completo de todas estas variables en un escenario altamente parametrizable, al tiempo que se proporciona cierta información asociada con la QoS ofrecida por el sistema.

A continuación se detallan las hipótesis más relevantes establecidas durante el desarrollo del simulador:

1) El área de simulación es cuadrada y comprende NxN celdas.

2) La técnica utilizada para evitar el efecto de bordes es el *cell wrapping*: la estación móvil rebota hacia adentro cuando alcanza los límites del área de simulación.

3) Las llamadas de nueva generación siguen un proceso de Poisson [1, 2, 8, 16].

4) Densidad de usuarios uniforme, es decir, una llegada de nueva generación puede producirse en cualquier punto del área de simulación, con idéntica probabilidad.

5) Se permite que un proceso de handoff genere reintentos siempre y cuando su duración no alcance un umbral máximo (parámetro del simulador). En el momento en que esto se produzca, el handoff será abortado y la llamada interrumpida. Los reintentos asociados a los procesos de handoff se distribuyen según una ley determinista. Esta es la forma en la que operan muchos de los sistemas actuales (como GSM) y futuros.

6) No se realiza reserva de recursos para llevar a cabo los procesos de handoff.

7) La dirección inicial seguida por las estaciones móviles (MSs) sigue una distribución uniforme. Se permite de esta forma que las MSs sigan cualquier camino, teniendo en cuenta siempre las limitaciones impuestas por las zonas de cobertura (el simulador es capaz de gestionar celdas bajo modelo hexagonal y Manhattan).

8) La velocidad de las MS está distribuida según una ley Gaussina, en la que los dos primeros momentos son parámetros de entrada al simulador.

9) Los MSs pueden cambiar su velocidad y dirección mientras la comunicación está activa. Los tiempos transcurridos entre dos cambios consecutivos de velocidad y dirección siguen sendas leyes exponenciales.

El simulador funciona muestreando el nivel de potencia de todas las estaciones (base y móviles), así como nivel de relación señal a interferente (C/I) para ambos canales, ascendente y descendente. De esta forma, se iniciará un proceso de handoff siempre que alguno de estos valores caiga por debajo de un cierto umbral. Estos umbrales que controlan las zonas de cobertura son parámetros del simulador. No se ha

establecido ningún otro tipo de consideración con respecto al tiempo de ocupación del canal o el proceso de handoff. De esta forma, estas variables son función del tráfico, el tipo de entorno, las dimensiones de las celdas que componen la zona a simular, el patrón de movilidad de las MS, el nivel de señal recibido por cada estación, etc.

Tal y como se ha comentado anteriormente, la herramienta software utilizada en el presente estudio es capaz de simular escenarios suburbanos con una disposición de celdas hexagonal o urbanos bajo un modelo Manhattan. Ambos tipos de escenarios pueden utilizar un mecanismo de asignación de canal fijo (FCA) o dinámico (DCA). En este estudio tan sólo se consideran escenarios urbanos bajo modelo Manhattan, donde la asignación de canal es FCA. En [11] puede encontrarse información acerca de las variables aquí planteadas para el caso de redes FCA y celdas distribuidas de forma hexagonal. La simulación de redes DCA se ha propuesto para trabajos posteriores. La Ecuación (4) es la utilizada por el simulador para calcular el nivel de señal recibido en la MS, en entornos Manhattan como los considerados en este estudio.

$$P_{MS}\,(dB) = P_{BS}\,(dB) - A_S\,(dB) - A_{PL}\,(dB) - A_{NLOS}\,(dB). \quad (4)$$

En la Ecuación (4), P_{BS} y P_{MS} representan el nivel de potencia emitido por las BSs y recibido por las MSs respectivamente; A_S indica las pérdidas debidas al shadow logonormal, A_{NLOS} es la atenuación extra producida cuando se pierde visibilidad directa con la BS, mientras que A_{PL} representa las pérdidas debidas a la propagación por el canal radio. Este último parámetro se calcula mediante la Ecuación (5), donde A_{OM} hace referencia a las pérdidas sufridas por la señal transmitida transcurrido un metro, P_{LC} indica la pendiente de pérdidas y d representa la distancia entre la BS y la MS.

$$A_{PL}\,(dB) = A_{OM}\,(dB) + PL_C \text{ x } 10\,Log(d). \quad (5)$$

Con respecto a la C/I, el nivel de interferencia se obtiene sumando la interferencia cocanal, la producida por el primer canal adyacente y la potencia del ruido blanco asociado al canal.

La Fig. 1 muestra la disposición del cluster compuesto por 5 celdas, utilizado en los escenarios simulados. Debe notarse que la razón para que una MS pierda cobertura en una celda específica, no radica únicamente en la distancia desde dicha BS hasta la MS. Esto es debido al efecto aleatorio asociado al shadow logonormal, así como al hecho de que la interferencia recibida es también aleatoria puesto que depende de la situación de las estaciones móviles y de los canales utilizados por cada una de ellas. Pese a que la disposición de las BS es regular, el área de cobertura de la celda varía constantemente. De esta forma, un proceso de handoff no ocurre cuando la MS cruza los bordes teóricos que separan

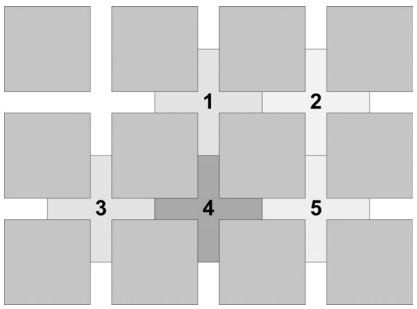


Figura 1: Disposición del cluster de celdas

dos celdas colindantes, sino que existe un área de cobertura en la que dos estaciones base pueden dar servicio a una misma MS: el área de handoff. Estas condiciones reflejan el comportamiento real de las redes celulares FCA, comportamiento que resulta difícil modelar mediante estudios analíticos. Los estudios de campo por su parte, limitarían el estudio ya que no permiten analizar el sistema con un tráfico en concreto.

La Tabla 1 muestra los parámetros que todos los escenarios simulados tienen en común. Algunos de estos valores han sido tomados de otros estudios [11, 13, 14]. Debido a la complejidad del simulador, tan sólo se muestran los más relevantes. Todos los escenarios han sido simulados utilizando antenas omni-direccionales y una portadora por BS.

Los parámetros referentes a medidas temporales son mostrados en la Tabla 2. Aunque no se consideran los intervalos de confianza, se ha escogido un tiempo de simulación suficientemente alto como para asegurar que las estadísticas sean estables y por tanto útiles para una primera aproximación. Las estadísticas tomadas durante el periodo de transitorio no son tenidas en cuenta en los resultados finales. En cuanto a la duración del servicio, se ha seguido una ley logonormal para su generación, con una desviación estándar acorde a los datos expuestos en [17].

El patrón de movilidad utilizado en la MS presenta como características destacables una alta velocidad media y una alta varianza. De esta forma se pretende la simulación de escenarios ricos en movilidad. Los valores específicos utilizados en este estudio se muestran en la Tabla 3. Este tipo de estaciones móviles han sido introducidas en otros trabajos [5, 11] con el objetivo de determinar los efectos de la movilidad sobre las variables de teletráfico.

El número de escenarios presentados en este estudio es de 4. La forma de obtener dichos escenarios es la siguiente. Se genera un escenario básico mediante los parámetros listados en las Tablas 1, 2 y 3. Los 4 escenarios aparecen de aplicar a cada canal una carga del 40%, 50%, 60% y 95%. Los tres primeros niveles de carga pueden considerarse como medios/altos, con probabilidades de bloqueo del 1% al 6%, según Erlang-B. El cuarto escenario simula un entorno de

Tabla 1: Parámetros establecidos en el simulador

Parámetro	*Valor*
Número de celdas simualdas (*N* x *N*)	36 (6 x 6)
Número de portadoras por celda	1
Número de slots por portadora	8
Distancia entre dos BS	300 metros
Ancho de la calle en al BS	30 metros
Nivel de señal emitido por la BS	23.98 dBm
Desviación standard del shadow logonormal	8 dB
Potencia de ruido en la MS	-120 dBm
Mínima C/I en la MS y la BS	14 dB
Sensibilidad en la BS y la MS	-113 dBm
Selectividad al primer canal adyacente en la MS	28 dB
Pérdidas a 1 metro	23 dB
Path losses slope	4

congestión. La Tabla 4 presenta el tráfico ofrecido a cada celda, así como la tasa de llegadas y la carga ofrecida a cada canal.

4 Resultados de simulación

4.1 Metodología

En primer lugar, para realizar el estudio de las variables propuestas anteriormente, el simulador es alimentado con los parámetros que modelan el entorno a simular. Una vez hecho esto, se lleva a cabo la simulación, recogiéndose todos los datos necesarios para efectuar el posterior análisis. Con dichos datos se efectúan los siguientes cálculos:

1) Se obtienen datos estadísticos, es decir, se calcula la media y el coeficiente de variación al cuadrado (SCV). Estos parámetros estadísticos se obtienen con el objetivo de decidir qué distribuciones teóricas son susceptibles de ser utilizadas para ajustar las muestras obtenidas.

2) Los parámetros de cada distribución teórica seleccionada en el proceso anterior son estimados mediante el método de máxima verosimilitud (MLE). Se emplea este método ya que se obtienen valores de significancia superiores a los alcanzados con cualquier otro, siempre que los resultados del ajuste sean evaluados mediante tests de bondad de ajuste (GOF) [3]. La notación empleada en este estudio ha sido tomada de [3, 11].

Tabla 2: Parámetros temporales

Parametro	*Valor*
Distribución del tiempo de servicio	Logonormal
Duración media de las llamadas	100 sec.
Desviación típica de las llamadas	132 sec.
Tiempo máximo en la zona de handoff	15 sec.

Tabla 3: Parámetros de movilidad

Parámetro	*Valor*
Velocidad media de las MS	20 m/s
Desviación típica de la velocidad de las MS	6 m/s
Tiempo medio entre cambios de velocidad	15 sec.
Probabilidad de girar en los cruces	30 %

Tabla 4: Tasa de llegadas

Carga ofrecida	*Tráfico ofrecido a la celda*	*Tasa de llegadas en el area de simulación* $\left(Llamadas/\text{sec}\right)$
40 %	3.2 Er.	1.152
50 %	4.0 Er.	1.440
60 %	4.8 Er.	1.728
95 %	7.6 Er.	2.736

3) Las funciones de distribución con los parámetros ya estimados, son evaluadas mediante los GOF de Kolmogorov-Smirnov y Anderson-Darling. La razón de utilizar dos GOFs radica en dotar de una mayor fiabilidad a la valoración de los ajustes realizados.

4.2 Resultados de QoS

La Tabla 5 muestra los valores de *GoS* e *ICP* obtenidos tras simular los escenarios propuestos, así como la carga ofrecida a cada canal y el número medio de handoffs generados por cada llamada recibida en el sistema.

La Fig. 2 muestra el *BCP* asociado a los escenarios propuestos para simulación. La evolución de las variables estudiadas se muestra afín a los resultados aparecidos en [5, 11]. Los resultados teóricos para el *BCP* calculados mediante los modelos de Erlang-B y Molina han sido incluidos como referencia. Esta figura muestra una separación clara entre los valores obtenidos mediante simulación y los calculados previamente utilizando la expresión de Erlang-B. Esta diferencia es conocida como "coste de movilidad" y es debida principalmente a la aparición de handoffs. Por tanto su efecto se hace más relevante cuanto más aumenta la tasa de llegadas de handoff. Debe notarse que las llegadas de handoff gozan de prioridad frente a las llamadas de nueva generación (las llegadas de handoff pueden reintentar la petición de recursos hasta la consecución del canal). Además, el simulador asocia un conjunto de canales, que

Tabla 5: Resultados de variables de tráfico

Carga ofrecida	Carga cursada	*Handoffs por llamada*	*ICP*	*GoS*
40 %	37,50 %	2,85	0,00 %	5,62 %
50 %	47,25 %	3,09	0,12 %	8,62 %
60 %	51,62 %	3,2	0,26 %	12,40 %
95 %	69,62 %	3,25	1,04 %	31,84 %

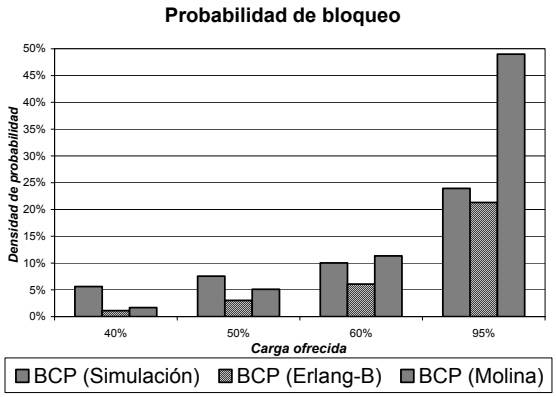


Figura 2: Probabilidad de bloqueo

pueden pertenecer a una o varias estaciones base, a la comunicación que pretende establecerse. La manera de proceder es por tanto intentar el acceso a cada uno de estos canales, de forma secuencial. Así, una llamada de nueva generación puede solicitar servicio a varias estaciones base. Este comportamiento difiere en gran medida de las suposiciones establecidas en el modelo de Erlang-B.

De acuerdo con los datos mostrados en la Fig. 2 y la Tabla 5, se puede observar como al aumentar la carga ofrecida, las diferencias entre *BCP* e *ICP* se incrementan. Este fenómeno también es explicado por la mayor prioridad de que gozan las llegadas de handoff. De esta forma, cuanto mayor es la carga ofrecida, peor es el valor del *BCP* simulado, puesto que el número de llegadas de handoffs también se incrementa con la carga. Estos resultados contrastan con los observados en [18], donde el autor muestra que el modelo Erlang-B ofrece una estimación correcta de la probabilidad de bloqueo en sistemas con un alto número de canales, mientras que sobredimensiona aquellos sistemas que poseen un número bajo de canales.

4.3 Tráfico de handoff

La Tabla 6 muestra el tiempo medio entre dos handoffs consecutivos dentro de una misma celda, así como el SCV asociado a esta variable. Este último parámetro presenta valores muy cercanos a la unidad en todos los escenarios simulados, a excepción del cargado al 60%. Este hecho no puede ser explicado a priori, ya que la teoría de teletráfico demuestra que el tráfico asociado a las llegadas de handoff está suavizado si se compara con el tráfico de nuevas llegadas. Además, también según dicha teoría debería

Tabla 6: Estadísticas del tráfico de handoff

Carga ofrecida	*Media (sec.)*	*SCV*
40 %	10.41	1.03
50 %	8.18	1.03
60 %	7.15	1.12
95 %	4.58	1.03

Tabla 7: Parámetros de ajuste con 50% de carga

Ajuste			*Prob.*	*D*	*A*
Exponencial	β : 8.18			5.55	21.35
Erlang–JK	β : 2.13	J : 2 K : 10	0.77	5.09	21.34
Hiper Erlang-KK	β_1 : 5.94 β_2 : 1.33	K : 2	0.59	4.38	17.60
Hiper Erlang–JK	β_1 : 1.33 β_2 : 5.94	J : 2 K : 2	0.40	4.38	17.60
Gamma	β : 8.42	C : 0.97		5.82	22.45

esperarse que al producirse un aumento en el tráfico cursado, disminuyese el SCV. Sin embargo, este parámetro no parece evolucionar según estas reglas, permaneciendo constante en todos los escenarios excepto en el cargado al 60%. Los autores suponen que este hecho responde a que la cobertura de la celda varía constantemente debido a la posición de las MSs y la interferencia que se generan entre ellas. La influencia de este factor en el SCV no puede ser cuantificado y podría cambiar según el patrón de movilidad. En los escenarios simulados, la influencia del canal radio parece ser considerable ya que hace que, pese a aumentar la carga, el SCV no disminuya (tal y como se podría esperar).

La Tabla 7 muestra los resultados obtenidos en el escenario cargado al 50%. No se han incluido los resultados del resto de escenarios debido a que, las conclusiones alcanzadas en todos ellos coinciden con las mostradas para el caso de carga igual al 50%. Uno de los datos mostrados en la Tabla 7 es la distancia estadística de los tests de bondad de ajuste (D para el caso del GOF de Kolmogorov-Smirnov y A para el caso de Anderson-Darling). Estas distancias serán empleadas como criterio para seleccionar el ajuste óptimo de la variable bajo consideración.

De acuerdo con los valores de SCV mostrados en la Tabla 6, la Tabla 7 demuestra que el ajuste exponencial es el peor de entre todos los evaluados. Esto corrobora resultados anteriores, los cuales mostraban que el tráfico de handoff es un tráfico suavizado [6, 7] (es decir, no de Poisson). Las funciones de distribución que mejor ajustan esta variable son las pertenecientes a la familia Hiper Erlang. Pese a que el valor de la distancia estadística es idéntico para ambas distribuciones, la función Hiper Erlang-KK es propuesta como ajuste óptimo del tráfico de handoffs en lugar de la Hiper Erlang-JK. La razón de esta decisión estriba en el hecho de que la distribución Hiper Erlang-KK requiere de un parámetro menos a estimar, por lo que su ajuste resulta más beneficioso en términos de coste computacional.

4.4 Tiempo en la zona de handoff

Aunque el estudio del tiempo de permanencia en la zona de handoff es una pieza clave en el análisis y

dimensionado de redes celulares, tan sólo unos cuantos estudios [8, 9, 11] han tenido en cuenta esta variable. El presente documento caracteriza esta variable en los escenarios propuestos en la sección 3.

Debe notarse que en sistemas bien dimensionados, el tiempo hasta la concesión de handoff tiende a ser cero, ya que cualquier petición de traspaso es servida inmediatamente. Esto implica que la mayor parte de la densidad de probabilidad se concentra en el valor temporal asociado al retardo 0. Este hecho motiva que el ajuste de esta variable sea particularmente costoso. Para poder superar este impedimento, los autores han optado por no considerar estos valores 0 a la hora de caracterizar esta variable, proporcionando por separado la probabilidad de que el handoff sea servido de manera inmediata (P_0). De esta forma, la caracterización completa del tiempo asociado al proceso de handoff se consigue mediante la probabilidad $P_0\delta(t)$ y la distribución que mejor ajusta a la variable estudiada (sin los valores 0).

La Tabla 8 muestra los datos obtenidos tras simular los entornos propuestos en la sección 3. En dicha tabla, $Media_1$ y $Media_2$ hacen referencia al valor medio del tiempo hasta la consecución del handoff, no teniendo y teniendo en cuenta los servidos inmediatamente, respectivamente. Tal y como muestra esta tabla, un incremento en la carga ofrecida resulta en un descenso en la probabilidad P_0, ya que cuanto mayor sea la carga, mayor es también el número de handoffs que la celda debe gestionar. Esto provoca un incremento en la $Media_2$, tal y como podría esperarse. Los valores de SVC por otro lado, se muestran inferiores a la unidad, lo cual implica que un ajuste exponencial no será óptimo.

De igual forma que ya ocurriera para el caso del tiempo entre dos llegadas de handoff consecutivas, la Tabla 9 tan sólo muestra los valores de ajuste obtenidos en el escenario cargado al 50%. El motivo de esta decisión es nuevamente el hecho de que las conclusiones mostradas para este escenario son compartidas por el resto. Según los datos incluidos en la Tabla 9, las distribuciones que mejor ajustan la variable a caracterizar son las pertenecientes a la familia Hiper Erlang. Aunque la distancia estadística lograda por la Hiper Erlang-JK es la menor de todas las distribuciones evaluadas, los autores proponen el uso de la distribución Hiper Erlang-KK como distribución óptima para caracterizar la duración del handoff, puesto que las distancias estadísticas

Tabla 8: Estadísticas del tiempo de handoff

Carga ofrecida	*$Media_1$ (s)*	*SCV*	*P_0*	*$Media_2$ (s)*
40 %	2.98	0.72	0.97	0.06
50 %	3.15	0.68	0.95	0.13
60 %	3.28	0.68	0.92	0.24
95 %	3.46	0.68	0.80	0.68

Tabla 9: Parámetros de ajuste con 50% de carga

Ajuste			*Prob.*	*D*	*A*
Exponencial	β: 3.15			12.92	117.42
Erlang–JK	β: 1.14	J: 2 K: 6	0.81	8.12	66.71
Hiper Erlang-KK	β_1: 0.99 β_2: 2.22	K: 2	0.52	8.17	65.81
Hiper Erlang-JK	β_1: 1.13 β_2: 1.35	J: 2 K: 5	0.80	8.10	65.81
Logonormal	μ: 0.84	σ: 0.79		9.72	74.40
Weibull	β: 3.45	C: 1.31		8.10	78.67
Gamma	β: 2.16	C: 1.45		8.87	71.67

obtenidas por esta función son muy cercanas a las obtenidas mediante la función de Hiper Erlang-JK, si bien requieren de la estimación de un parámetro menos.

Debe notarse que distribuciones relativamente simples como la Gamma o la Weibull proporcionan ajustes muy aceptables, si los comparamos con los alcanzados mediante funciones de tipo Hiper Erlang, para tráficos aplicados medios o bajos (inferiores al 60%). Estas funciones podrían utilizarse como una primera aproximación debido a que requieren de la estimación de un número de parámetros relativamente reducido.

5 Conclusiones

Una completa herramienta de simulación ha sido utilizada para caracterizar el tráfico de handoff y la duración del proceso de handoff, en redes celulares. Esas variables han sido analizadas en diversas condiciones de carga con el objetivo de presentar un conjunto significativo de resultados. En los escenarios propuestos, el tráfico de handoff no parece seguir un comportamiento poissoniano, obteniéndose el mejor ajuste mediante las distribuciones de la familia Hiper Erlang. De entre ellas, se propone el uso de la distribución Hiper Erlang – KK puesto que ésta requiere de un parámetro menos a estimar que la función Hiper Erlang – JK. Estas dos distribuciones son las que ofrecen también un mejor ajuste para el caso de la duración del handoff. De igual forma que en el caso del tráfico de handoff, el ajuste óptimo propuesto para esta variable es el obtenido mediante la distribución Hiper Erlang – KK, debido a que requiere de la estimación de un número menor de parámetros que la función Hiper Erlang – JK.

Además de estas caracterizaciones, se han estudiado algunos parámetros relacionados con la QoS ofrecida por el sistema. Se han percibido algunas diferencias entre los resultados obtenidos y métodos de dimensionado clásicos como el M/M/C/C (Erlang-B) o Molina. El proceso de llegadas para el caso de nuevas llamadas es de Poisson en todos los escenarios simulados. Sin embargo, el tráfico de handoff está suavizado, lo cual debería contribuir a obtener probabilidades de bloqueo inferiores a las

estimadas mediante las fórmulas de Erlang-B o Molina. Sin embargo, puesto que el modelo de Erlang-B no tiene en cuenta la prioridad de que gozan las llegadas de handoff, el efecto global es una calidad de servicio peor que la estimada mediante la expresión de Erlang-B. Debe tenerse en cuenta que en general, los modelos basados en mantener las llamadas bloqueadas, como puede ser Molina, presentan probabilidades de bloqueo, más cercanas a las obtenidas mediante simulación.

Agradecimientos

Este trabajo de investigación ha sido financiado por el gobierno español a través del proyecto CICYT TIC2000-1041-C03-01.

Referencias

[1] D. Hong, S. S. Rappaport, "Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and nonprioritized handoff procedures", *IEEE Trans. Veh. Tech*. 35(3), pp. 77-92, August 1986.

[2] E. Chlebus, T. Zbiezek, "A novel approach to simulation of mobile networks", *12th ITC Specialist Seminar on Mobile Systems and Mobility*, March 2000.

[3] F. Barceló, J. Jordán, "Channel Holding Time Distribution in Public Telephony Systems (PAMR and PCS)", *IEEE Trans. Veh. Tech.*, Vol. 29, No. 5, pp. 1615-1625, September 2000.

[4] D. Lara-Rodríguez, "Influence of the Handoff Process on the Channel Holding Time Distribution for Cellular Systems", *IEEE Int. Conf. on Personal Wireless Communications*, pp. 149-152, 1996.

[5] K. Saitoh, H. Hidaka, N. Shinagawa, T. Kobayashi, "Vehicle Motion in Large and Small Cities and Teletraffic Characterization in Cellular Communication Systems", *IEICE Trans. Commun*. Vol. E84-B, No 4 pp. 805-812, April 2001.

[6] F. Barceló, J. I. Sánchez, "Probability distribution of the Inter-Arrival time to Cellular Telephony Channels", *IEEE Veh. Tech. Conf.*, pp. 762-766, 1999.

[7] M. Rajaratnam, F. Takawira, "Nonclassical Traffic Modeling and Performance Analysis of Cellular Mobile Networks with and Without Channel Reservation", *IEEE Trans. Veh. Tech*. 2000, pp. 817-834.

[8] M. Ruggieri, F. Graziosi, F. Santucci, "Modeling of the Handoff Dwell Time in Cellular Mobile Communications Systems", *IEEE Trans. Veh. Tech*., Vol. 47 No 2 pp. 489-498, May 1998.

[9] V. Pla and V. Casares, "Analytical-Numerical Study of the Handoff Area", *Technical Internal Report in UPV*.

[10] I. Martin-Escalona, "Modelado estadístico mediante simulación de variables de tráfico en redes de telefonía móvil celular", *UPC PFC ETSETB*, Sept. 2001.

[11] I. Martin-Escalona, F. Barcelo, J. Casademont, "Teletraffic simulation of cellular networks: modeling the handoff arrivals and the handoff delay", *Proc. of 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 2209 – 2213, Sept. 2002.

[12] F. Barcelo, "A scheme to handle fresh and handoff traffic based on state-dependent rejection", *Proc. of IEEE Global Telecommunications Conference 2000*, Vol. 3, pp. 1522 –1527, 2000.

[13] P. R. C. Gomez, M. D. Yacoub, A. F. De Toledo, "Performance of a Microcellular Network in a More Realistic Condition", *Proc. of 11th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1594-1598, 2000.

[14] H. S. Cho, D. K. Sung, "Channel Holding Time Distribution in Cross- and Cigar-Shaped Urban Microcells", *IEEE Trans. Commun,* Vol. E81-B, No. 6, pp. 1275-1279, June 1998.

[15] K. Saitoh, H. Hidaka, N. Shinagawa, T. Kobayashi, "Teletraffic and its Self-Similarities in Cellular Communication Networks Based on Vehicle Motion Measurements", *Specialist Seminar on Access Networks and Systems 14th ITC Specialist Seminar on Access Networks and Systems*, pp. 43-54, April 2001.

[16] Y. Fang, I. Chlamtac, "A new mobility model and its application in the channel holding time characterization in PCS networks", *IEEE INFOCOM '99*, Vol. 1, pp. 20-27, 1999.

[17] V. Bolotin., "Telephone Circuit Holding Time Distributions", *Proc. 14th International Teletraffic Congress, Elsevier Science B.V.*, pp. 125-134, 1994.

[18] P. J. Smith, A. Sathyendran, A. R. Murch, "Analysis of Traffic Distribution in Cellular Networks", *Proceedings of the 49th IEEE Veh. Tech. Conf.*, No. 3, pp. 2075-2079, 1999.

Servicio Simultáneo de Flujos Semi-Elásticos en Internet. Primera Aproximación: Caso Homogéneo

Marcos Postigo Boix[1], Joan García Haro[2], Jose L. Melús Moreno[1]
[1]Departamento de Ingeniería Telemática, Universitat Politècnica de Catalunya
C/ Jordi Girona 1-3, Mòd. C3, Campus Nord, 08034 Barcelona.
[2]Departamento de Tecnologías de la Información y las Comunicaciones, Universidad Politécnica de Cartagena
Campus Muralla de Mar s/n, 30202 Cartagena.
E-mail: marcos.postigo@entel.upc.es; joang.haro@upct.es, teljmm@entel.upc.es

Abstract*. Service providers need to minimize reserved resources from the network to achieve a reduction on semi-elastic flows transmission cost when using an Internet empowered with some type of end-to-end quality of service facilities. Their benefits depend on how efficiently resources are utilized, because network operators must apply higher rates for using resource reservation. Cost reduction is accomplished by using classical best-effort service when resource reservation is not required since enough network resources are available. When serving multiple simultaneous clients, the server must appropriately manage reservations on its network access link to prevent reservation rejections. In this paper, it is proposed a methodology to control this bandwidth to serve the maximum possible number of clients. This is achieved without increasing excessively extra signalling involved to manage reservations required by clients.*

1 Introducción

La transmisión eficiente de flujos semi-elásticos en una Internet con garantías de servicio extremo a extremo [1], permite minimizar el coste que debe pagar un proveedor de servicios (y en consecuencia el cliente) a su proveedor u operador de red por el uso de reserva de recursos [2], necesaria para realizar una transmisión rápida y fiable. El cliente controla el modo de transmisión, indicando al servidor si debe enviar los datos mediante reserva de recursos (ReR) o mediante el modo *best-effort* (BE). Para minimizar el coste, el cliente debe maximizar el uso del modo BE, teniendo en cuenta la carga existente en la red en este modo. Para evaluar esta carga y determinar cuándo reservar recursos, el cliente observa la ocupación de su memoria, asegurando que se mantiene entre dos umbrales. Estos umbrales, tal y como se estudia en [3], se escogen convenientemente para que el cliente disponga de datos durante toda la transmisión y para que no se reserven más recursos de los necesarios. Cuando la ocupación de la memoria del cliente llega a uno de estos umbrales, se pide al servidor un cambio de modo de transferencia. Así, si se alcanza el umbral mínimo (*Min*), el cliente pedirá un cambio a modo ReR para garantizar la llegada de datos, mientras que si se alcanza el umbral máximo (*Max*), se solicitará la transmisión en modo BE para reducir el coste de la transmisión. Seleccionando de forma conveniente, el umbral *Max* permite minimizar el coste de transmisión debido a reservas de recursos. El servidor, por su parte, envía los datos a la tasa adecuada según se utilice el modo ReR o BE. Además, debe intentar servir el máximo número de clientes de forma simultánea y eficiente, es decir, controlando que la minimización del coste siga siendo efectiva, y que no se incremente en exceso la señalización extra necesaria para controlar el servicio simultáneo, con respecto a la situación de dar servicio a un único cliente. Suponiendo que los clientes son independientes entre sí, es posible que varios de ellos realicen una petición de reserva de recursos al mismo tiempo, pudiendo llegar a reservar todo el ancho de banda de acceso del servidor, si éste no se gestiona correctamente, y por tanto, producir bloqueo a posibles reservas de otros clientes, afectando seriamente la calidad de servicio de la transmisión.

En este artículo, se analiza un mecanismo de gestión de ancho de banda de acceso del servidor a la red. Asimismo, se examinarán las implicaciones que supone esta gestión en los protocolos de señalización de reserva de recursos, particularizando para el protocolo RSVP (*resource ReSerVation Protocol*) [4], que es el utilizado en la implementación del sistema mediante el simulador ns-2 [5] [6]. En la sección 2, se investigarán tres posibles modelos de servicio simultáneo a varios clientes, uno de ellos genérico, que se analizará con mayor profundidad en la sección 3. La sección 4, presenta y reflexiona sobre los resultados de simulación obtenidos mediante el simulador ns-2. Finalmente, en la sección 5, se muestran las conclusiones más relevantes del estudio.

2 Servicio simultáneo de flujos semi-elásticos

El servicio de flujos semi-elásticos en redes con calidad de servicio extremo a extremo, está basado en dos modos de servicio: *best-effort* (BE) y reserva de recursos (ReR). El servidor utiliza los dos modos de transmisión según sean las necesidades del cliente. Así, si el cliente tiene suficientes datos almacenados en su memoria, se podrá utilizar el modo BE, mientras que si la memoria alcanza un determinado umbral mínimo de ocupación, se utilizará el modo ReR. Cuando se debe servir a varios clientes que demandan simultáneamente la transmisión de flujos semi-

elásticos (que no tienen porque responder a la misma información pre-almacenada), es necesario analizar cómo el servidor debe gestionar su ancho de banda de acceso a la red, ya que el servicio es simultáneo puede implicar que fácilmente se agote el ancho de banda disponible, impactando seriamente en la calidad de servicio que perciben los clientes.

En esta sección, se investigarán tres modelos de gestión de ancho de banda para dar servicio simultáneo: reserva de recursos por cliente, reserva de recursos para un cliente y reserva de recursos para *m* clientes. De estos tres modelos, el tercero es el caso genérico, mientras que el primero y el segundo se pueden interpretar como casos particulares de él. El modelo se basa en dividir el ancho de banda de acceso existente en dos partes: una para garantizar el servicio en modo BE y otra para el modo ReR. La parte que se gestiona es la reservada para el modo ReR de forma que se garantice el servicio para todos los clientes de forma simultánea. Así, si se da servicio a $\mathcal{C}$ clientes, se supone que $\mathcal{C}_{ReR}$ se servirán en modo ReR y $\mathcal{C}_{BE}$ en modo BE (1).

$$\mathcal{C} = \mathcal{C}_{ReR} + \mathcal{C}_{BE} \tag{1}$$

Este estudio es una primera aproximación a la gestión del ancho de banda de acceso que debe realizar el servidor, por lo que se asumirá que todos los clientes son homogéneos, es decir, todos leen la información a la misma velocidad, reservan el mismo ancho de banda, y la oferta de ancho de banda disponible en el camino entre cliente y servidor es la misma y sin restricciones (aunque la información de los distintos flujos puede ser diferente). Este no es un caso realista, pero permitirá analizar los distintos modos de servicio en condiciones similares para abordar en el futuro el caso heterogéneo.

2.1 Número máximo de clientes

Primeramente, cabe examinar cuál es el número máximo de clientes ($\mathcal{C}_{\max}$) a los que podrá dar servicio un servidor en la situación supuesta. Para ello, en primera instancia se descartan los efectos que puede tener la señalización de reserva de recursos en el sistema[1]. En [3] se define la carga apreciada por el cliente, de tal forma que una carga $\rho = 1$ indica que la tasa de llegada de paquetes a un cliente en modo BE (α_{BE}) es nula, mientras que $\rho = 0$ indica que dicha tasa es capaz de servir a un cliente con tasa de lectura (α_r).

$$\rho = \begin{cases} 1 - \dfrac{\alpha_{BE}}{\alpha_r} & \alpha_{BE} \le \alpha_r \\ 0 & \alpha_{BE} \ge \alpha_r \end{cases} \tag{2}$$

En el caso en que la carga apreciada por los clientes sea nula durante toda la transmisión, se dará servicio en modo de transmisión BE, a una tasa igual a la tasa de lectura del cliente (α_r). Por tanto, el valor que limitará el número de clientes es el ancho de banda disponible para el modo BE, que en este caso será igual al ancho de banda de acceso del servidor (B_s), ya que se podrá utilizar todo para este modo, al no ser necesario el empleo del modo ReR. Así, el número máximo de clientes a los que se podrá dar servicio será,

$$\mathcal{C}_{\max} = \frac{B_s}{\alpha_r} \tag{3}$$

El otro caso extremo, es aquél en que todos los clientes precisan recibir toda la información mediante el modo ReR. En esta situación, si se supone que el servidor dispone de un ancho de banda de acceso $B_s = B_{BE} + B_{ReR}$ (*bits/s*), donde B_{BE} es el ancho de banda mínimo disponible para tráfico BE, y B_{ReR} es el ancho de banda máximo disponible para tráfico ReR, el servicio estará limitado por B_{ReR}.

$$\mathcal{C}_{\max} = \frac{B_{ReR}}{\alpha_r} \tag{4}$$

Tal y como se deduce de lo anterior, en casos intermedios, el mayor número de clientes al que se puede dar servicio estará entre los valores extremos (3) y (4) anteriores, por tanto, para evitar que este número dependa de la carga de la red, se supondrá el número mayor de clientes a los que se puede dar servicio al valor de la parte entera de (4), ya que es el más restrictivo.

2.2 Reserva de recursos por cliente

En este modo de servicio, el servidor debe garantizar que todos los clientes puedan reservar recursos en el momento que así lo requieran, controlando los recursos máximos que puede reservar cada cliente. En [3], se demuestra que el coste de la transmisión depende de la tasa de llegada de paquetes en modo BE (α_{BE}) y en modo ReR (α_{ReR}). Cuanto mayores son estas tasas, menor es el coste[2], por lo que se deberá utilizar siempre la mayor cantidad posible de ancho de banda de B_{ReR} para cada cliente en modo ReR. Por tanto, si se sirve a un solo cliente, parece lógico pensar que se podrá utilizar todo B_{ReR}, mientras que si son dos clientes se podrá utilizar $B_{ReR}/2$ para cada uno, y en general para un número de clientes $\mathcal{C} = k$, se podrá

[1] La señalización de reservas de recursos se puede descartar bajo el supuesto razonable de que sea pequeña respecto a la cantidad de datos que envía el servidor. Debe tenerse en cuenta si esta señalización es tan grande como para ocupar el ancho de banda necesario para el servicio de un cliente. Como se verá más adelante, la cantidad de señalización necesaria dependerá de la forma en que se dé servicio a los clientes.

[2] Si la tasa $\alpha_{ReR} = \alpha_r$, no hay reducción del coste, ya que todos los datos se envían usando modo ReR; y si la tasa en modo BE es nula, sucede lo mismo.

reservar, para cada uno, el ancho de banda expresado en (5)[3].

$$\alpha_{Res}(k) = \frac{B_{ReR}}{k} \quad (5)$$

Cabe resaltar, que k no puede ser cualquier valor, ya que siempre se debe cumplir que α_{Res} sea mayor que la tasa de lectura del cliente α_r, ya que en caso contrario es imposible realizar la transmisión correctamente, es decir, no se debe superar el número de clientes máximo ($\mathcal{C}_{\max}$) al que se puede dar servicio. Por otro lado, cuanto mayor sea α_{Res} respecto de α_r, mayor será la eficiencia media conseguida ($\overline{\eta}$) (7), ya que se incrementarán las eficiencias de transmisión de cada cliente (η_i). La eficiencia se define como la reducción del coste conseguida con el método respecto del coste máximo que se obtiene reservando recursos durante toda la duración de la comunicación (6). Así una eficiencia igual a 1 indica que el coste (C) ha sido 0, mientras que una eficiencia igual a 0 implica que todos los datos se han enviado usando el modo ReR $C = C_{MAX}$.

$$\eta = 1 - \frac{C}{C_{MAX}} \quad (6)$$

$$\overline{\eta} = \frac{\sum_{i=1}^{\mathcal{C}} \eta_i}{\mathcal{C}} \quad (7)$$

Analizando la señalización relacionada con las reservas de recursos, se advierte la necesidad de cambiar el ancho de banda que reserva cada cliente a medida que varía el número de clientes que se está sirviendo en cada momento (varía k). Por otra parte, se precisa que el protocolo de señalización de reserva de recursos pueda estar controlado por el servidor, con el fin de controlar de forma inmediata el ancho de banda reservado para cada cliente que demande el uso del modo ReR en un determinado momento. Por consiguiente, si se utiliza el protocolo RSVP será necesario que la interfaz con el servidor le informe de la llegada de mensajes RSVP antes de realizar alguna acción (rechazo de reserva, actualización de reservas, etc.) para poder modificar el estado de las reservas actuales, si ello es factible, a fin de evitar rechazos innecesarios[4].

2.3 Reserva de recursos para un cliente

En este modelo sólo reserva recursos un cliente y el resto recibe datos en modo BE. En este caso se simplifica al máximo el control de cuáles son los recursos disponibles por cliente, ya que todos los recursos reservables se asignan a un único cliente. El cliente al que se asignan estos recursos es el que necesita con mayor urgencia que su memoria se llene, para así evitar quedarse sin datos que leer[5]. En este caso el ancho de banda a reservar es siempre el mismo (8).

$$\alpha_{Res}(k) = B_{ReR} \quad (8)$$

En cuanto a la eficiencia media para este modo de servicio, será la máxima posible, ya que se reserva la mayor cantidad de recursos posible. Por otro lado, esta eficiencia se degradará progresivamente a medida que aumente el número de clientes a los que se da servicio, ya que también aumentará el tráfico en modo BE, pudiendo llegar a ser el enlace de acceso el más restrictivo, limitando la tasa de BE que se ofrece a los clientes.

2.4 Reserva de recursos para *m* clientes

En este caso se permite dar servicio simultáneo en modo ReR a m clientes, repartiéndose los recursos disponibles para reservar. Así, el modelo de reserva de recursos por cliente es igual a este modelo pero con $m = \mathcal{C}_{\max}$, mientras que la reserva de recursos para un cliente ocurre para $m = 1$.

El funcionamiento general de esta metodología es idéntico al del primer caso (apartado 2.2) para $k \le m$ (el número de transmisiones permitidas en modo ReR es menor o igual que m), mientras que para $k > m$ es igual que el segundo (apartado 2.3), teniendo en cuenta que una nueva petición de paso a modo ReR, cuando el número de clientes que se sirven en modo ReR en un momento concreto ($\mathcal{C}_{ReR}$) es igual a m, implica que el cliente que lleva más tiempo en modo ReR pase a modo BE.

3 Servicio mediante reserva de recursos para *m* clientes

En esta sección, se analizará el servicio genérico basado en la reserva de recursos para m clientes. En la primera parte, se describe el funcionamiento de la gestión del ancho de banda de acceso del servidor. Esta gestión permite repartir de forma apropiada el ancho de banda entre todos los clientes, y controlar cuáles se sirven en modo ReR y cuáles en modo BE. En la segunda parte, se realiza un análisis de la señalización necesaria para este tipo de servicio. A modo de resumen, se muestra en la Tabla 1 las variables relacionadas con el servicio simultáneo de flujos semi-elásticos presentadas en apartados anteriores.

3.1 Gestión del ancho de banda de acceso del servidor

El ancho de banda que se reservará en el modo ReR

[3] El servidor debe conocer cuál es el ancho de banda máximo que se puede reservar en el camino hacia el cliente, para no modificar reservas de forma errónea y evitar que se rechacen las modificaciones. Este ancho de banda se puede deducir controlando las características del camino mediante objetos ADSPEC (*ADvertisement SPECification*) utilizados en el protocolo RSVP.

[4] En el modelo implementado en el simulador ns-2, se propone un interfaz con estas características, de forma que el protocolo RSVP indica la llegada de una nueva reserva al servidor para que actúe antes de continuar con el proceso de reserva.

[5] Será el último cliente cuya memoria haya alcanzado el umbral mínimo.

Tabla 1: Variables relacionadas con el servicio simultáneo de flujos semi-elásticos.

Variable	Descripción
B_s	Ancho de banda de acceso del servidor (*bits/s*)
B_{BE}	Ancho de banda mínimo disponible para modo BE (*bits/s*)
B_{ReR}	Ancho de banda máximo disponible para modo ReR (*bits/s*)
α_{Res}	Tasa de reserva de recursos (*bits/s*)
α_r	Tasa de lectura del cliente (*bits/s*)
C	Número de clientes (sesiones) a los que se está dando servicio
C_{ReR}	Número de clientes sirviéndose en modo ReR
C_{BE}	Número de clientes sirviéndose en modo BE
$C_{\max}$	Número máximo de clientes (de ambas clases) a los que se puede dar servicio
k	Número de sesiones que se pueden dar en modo ReR
m	Número de sesiones máximo que se permiten en modo ReR

depende del número de clientes a servir. Su gestión varía según el número de clientes que tienen iniciada una sesión en el servidor[6], por lo que se analizarán las acciones a realizar en el inicio y al final de una sesión. Por otra parte, se verá cuál es la gestión de este ancho de banda a medida que se recibe por parte de los clientes, nuevas peticiones de paso a modo ReR conforme sus memorias alcanzan los umbrales mínimos permitidos.

Inicio de sesión

Inicialmente, el servidor no da servicio a ningún cliente hasta que llega la primera petición. Se aceptarán nuevas peticiones siempre que no se supere el número máximo de clientes que acepta el servidor. El número de peticiones recibidas determina el ancho de banda disponible para el uso de cada cliente en el modo ReR (Fig. 1). Así, si el número de clientes a los que se puede dar servicio en modo ReR en ese momento[7] (k) es menor que m, se incrementará en uno ese número para controlar el valor instantáneo, se cambiará el ancho de banda a utilizar por cada cliente en modo ReR (α_{Res}), y se informará a todos los clientes de éste nuevo valor[8]. En caso de que k sea igual a m (hay un número de clientes con sesión abierta mayor o igual al máximo permitido en modo ReR), sólo se informará al cliente del valor del ancho de banda que deberá reservar en modo ReR.

Fin de sesión

Cuando se finaliza una sesión, el ancho de banda disponible se deberá repartir entre el resto de clientes si es posible (Fig. 2). De esta forma, si el número de clientes a los que se da servicio (sin contar el cliente que acaba de finalizar) C, es menor a m, se reduce k y se aumenta el ancho de banda máximo a reservar por cada cliente en modo ReR, además de informar al resto de clientes del nuevo ancho de banda disponible. Si no es así, no se realiza ninguna acción.

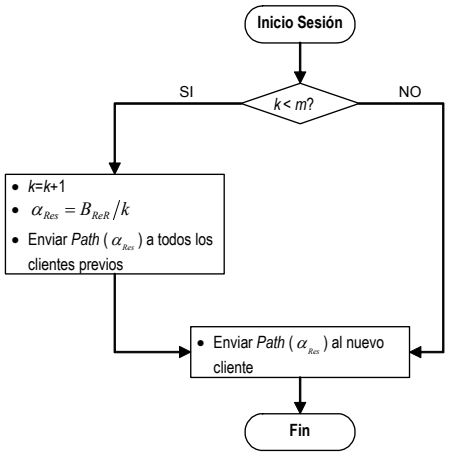


Figura 1: Diagrama de flujo para la gestión del ancho de banda de acceso del servidor (llega una nueva petición de servicio).

Peticiones de paso a modo ReR

Por otro lado, el servidor recibe peticiones de paso a modo de transferencia ReR. Según el número de clientes que se sirvan en ese momento, el servidor actuará de una u otra manera (Fig. 3). Si llega una petición de paso a modo ReR y no se está sirviendo a ningún cliente en ese modo ($C_{ReR} = 0$), no se realiza ninguna acción, ya que todo el ancho de banda para modo ReR está disponible y se supone que la petición de reserva de recursos es correcta. Si se está sirviendo a algún cliente en modo ReR, pero $C_{ReR} < m$,[9] se ajustarán las tasas reservadas para todos los clientes que todavía no hayan ajustado su tasa a la máxima especificada y se aceptará la petición de paso a modo ReR para ese cliente. Esto implica que el servidor puede requerir un cambio en las reservas existentes. El protocolo RSVP no permite realizar este cambio, ya que debe ser el cliente quien lo realice. Para implementar esta nueva habilidad, se propone el uso de un nuevo mensaje RESV_SENDER similar al mensaje RESV de RSVP, pero enviado por el servidor. En el caso de que $C_{ReR} = m$,[10] se pasará a modo BE al cliente que lleve en modo ReR más tiempo, para permitir que el cliente que ha realizado la petición (y que se supone con una ocupación de memoria más crítica) pase a modo ReR. Tal y como se observa, el servidor puede decidir el paso a modo BE de una de las transmisiones. Esto puede hacer que el cliente pase a modo de transmisión BE, cuando su memoria

[6] Se entiende por sesión el proceso de transmisión de la información requerida al servidor hacia un cliente. Se asume que cada cliente sólo puede abrir una sesión.

[7] Este número depende del número de sesiones abiertas y nunca puede superar m.

[8] El protocolo RSVP puede informar a los clientes de este valor, mediante el uso de mensajes PATH que indican el camino que siguen los mensajes, así como la tasa en modo ReR que permite utilizar el servidor.

[9] Se sirve en modo ReR a un número menor al número máximo de clientes permitidos en ese modo.

[10] Se da servicio en modo ReR a un número igual al número permitido en ese modo.

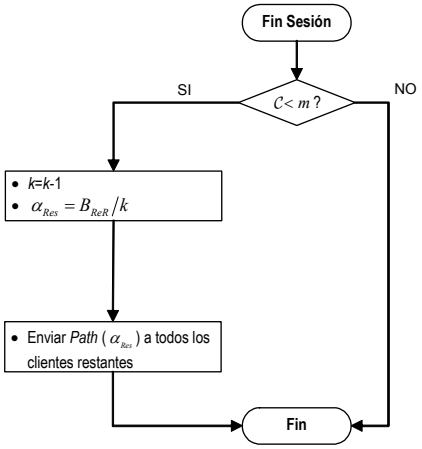


Figura 2: Diagrama de flujo para la gestión del ancho de banda de acceso del servidor (finaliza una sesión).

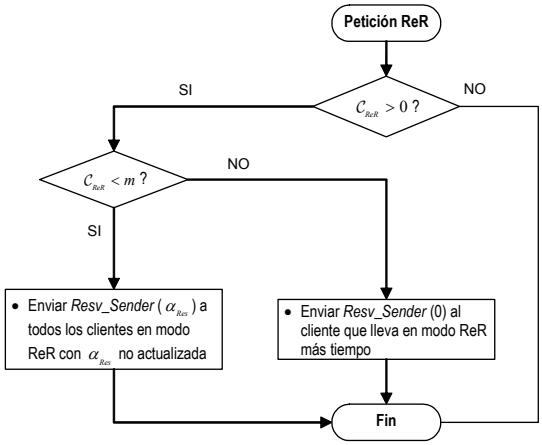


Figura 3: Diagrama de flujo para la gestión del ancho de banda de acceso del servidor (peticiones de paso a modo ReR)

está cerca del umbral mínimo[11] o incluso estando por debajo si es al inicio de la recepción de datos (Fig. 4), provocando en el primer caso que se repita la petición de reserva de recursos rápidamente, o que no se pueda iniciar correctamente la lectura de información en el cliente, en el segundo. Para evitar que se cambie el modo de transmisión a modo BE antes de que la memoria del cliente llegue al nivel mínimo, es necesario bloquear la conexión, de forma que no se permita que esa conexión pase a modo BE hasta que haya alcanzado el nivel mínimo de ocupación (este nivel se aconseja que sea el nivel inicial utilizado para obtener una primera estimación de α_{BE} [3]). En caso de que el servidor esté totalmente bloqueado[12], la petición inicial se almacena en una cola hasta que sea posible iniciar la transmisión. Por otro lado, también se debe evitar que se pase a modo BE si la ocupación de la memoria del cliente está cerca del nivel mínimo. Para evitar esto, es posible aumentar el nivel mínimo del cliente.

[11] En este caso, en un breve instante de tiempo el cliente realizará otra petición de paso a modo ReR.

[12] El servidor estará totalmente bloqueado si se está dando servicio en modo ReR, por primera vez, al número máximo de clientes en ese modo.

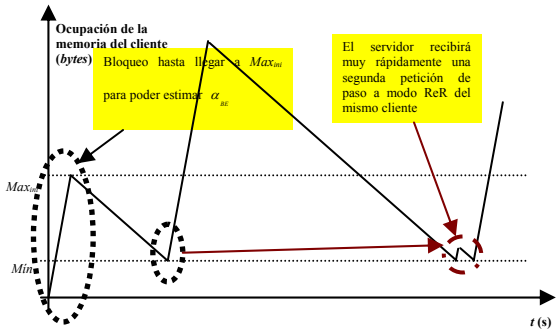


Figura 4: Zonas de ocupación de la memoria del cliente en las que no se debe pasar a modo BE.

3.2 Análisis preliminar de señalización

Es importante estimar la cantidad necesaria de señalización en el servidor para poder gestionar el ancho de banda de forma adecuada, ya que es un factor muy importante que permitirá comparar las distintas formas propuestas de servicio. Por tanto, es posible contabilizar el número de mensajes PATH, necesarios en cada caso (M_{Path}), utilizados para indicar a los clientes la nueva reserva que deberán utilizar, así como el de mensajes RESV_SENDER (M_{Resv_Sender}) (véase la nota 10). El número de mensajes que se precisa depende de las distintas fases descritas en las Figuras 1, 2 y 3, así como del número de clientes que se sirva en ese momento $\mathcal{C}$ y del valor máximo de clientes en modo ReR aceptado (m).

Inicio de sesión

En este caso y en base a la Fig. 1, el número de mensajes de señalización extra necesarios[13] en esta fase cada vez que se inicie una sesión, será:

$$M_{inicio} = M_{Path} = \begin{cases} \mathcal{C} & \mathcal{C} < m \\ 0 & \mathcal{C} \geq m \end{cases} \qquad (9)$$

Para el caso de reserva de recursos para cada cliente, el número de mensajes será más crítico a medida que se incremente $\mathcal{C}$. Esto implica que el número de mensajes necesario dependa del número de clientes a los que se esté dando servicio, lo que puede suponer un problema de escalabilidad en el sistema cuando $\mathcal{C}$ es grande. Por otro lado, disminuyendo m es posible controlar este problema. En el caso de reserva de recursos para un cliente, se cumple $m = 1$, por lo que no se necesitan mensajes de señalización extra al inicio de la sesión, independientemente del número de clientes en el sistema.

Fin de sesión

De la Fig. 2, se puede deducir el número de mensajes necesarios en esta fase si $\mathcal{C}$ es el número de clientes que restan sin contar el que abandona la sesión:

$$M_{final} = M_{Path} = \begin{cases} \mathcal{C} & \mathcal{C} < m \\ 0 & \mathcal{C} \geq m \end{cases} \qquad (10)$$

[13] Mensajes de señalización extra en relación al número de mensajes necesario para el caso de un solo cliente.

Tal y como se advierte, para el servicio con reserva de recursos para cada cliente, la señalización necesaria será $\mathcal{C}$ y para el caso de reserva de recursos por cliente no será necesario ningún mensaje extra.

Peticiones de paso a modo ReR

Observando la Fig. 3, se deducen los mensajes necesarios para esta fase, teniendo en cuenta los que se envían para cambiar las tasas reservadas no actualizadas (9) y los necesarios para liberar la reserva al cliente con mayor tiempo continuo reservando recursos, cuando al llegar la petición se están sirviendo en modo ReR $\mathcal{C}_{ReR}$ clientes (10).

$$M_{ReR\_cambio\_tasa} = M_{Resv\_Sender} = \begin{cases} 0; & \mathcal{C}_{ReR} = 0 \\ \mathcal{C}_{ReR}; & 0 < \mathcal{C}_{ReR} < m \\ 0; & \mathcal{C}_{ReR} = m \end{cases} \quad (11)$$

$$M_{ReR\_liberar\_reserva} = M_{Resv\_Sender} = \begin{cases} 0 & \mathcal{C}_{ReR} < m \\ 1 & \mathcal{C}_{ReR} = m \end{cases} \quad (12)$$

En el caso de $\mathcal{C}_{ReR} = 0$, no es necesaria señalización extra, ya que no hay ningún problema en realizar la reserva. Cuando se cumple $0 < \mathcal{C}_{ReR} < m$ se supone el peor caso, es decir, se considera necesario enviar a todos los clientes un cambio de reserva de recursos. Esto será en general relativamente habitual para $\mathcal{C} < m$, ya que en este caso, se deberán ajustar las tasas inmediatamente. Para el servicio mediante reserva de recursos por cliente, esta situación se produce durante el máximo tiempo posible ya que depende de *m* que en este caso es máxima. En el caso extremo con reserva de recursos para un cliente, no se requieren los mensajes para cambiar la reserva ya que se utiliza todo el ancho de banda para dicha reserva. Por otro lado, la señalización precisa para liberar la reserva que lleva más tiempo iniciada, es únicamente necesaria si $\mathcal{C}_{ReR} < m$. Asimismo, la probabilidad de que una nueva petición se encuentre con $\mathcal{C}_{ReR} = m$ clientes, será mayor a medida que *m* sea menor y $\mathcal{C}$ mayor. Otro factor importante a tener en cuenta, es el número de peticiones de paso a modo ReR que realizarán los clientes, que será dependiente de los diversos parámetros controlados por su memoria. En particular, para una transmisión concreta, el parámetro más crítico es el tamaño máximo de la memoria del cliente, ya que es el que incide directamente en el umbral Max_n que se utiliza y por tanto, en el número de periodos ReR necesarios [3].

4. Resultados de simulación

El sistema servidor se ha implementado en el simulador ns-2 y se describe en [5]. En este apartado, se presentan los distintos experimentos realizados, para observar las prestaciones del sistema propuesto. En el primer experimento, se muestra la evolución de la eficiencia media del sistema en función de *m* (valor máximo de clientes en modo ReR aceptado) y de $\mathcal{C}$ (número de clientes que se sirve). En el segundo, se ilustra la evolución de la cantidad de señalización extra necesaria en función de los mismos parámetros.

El sistema que se simula es un sistema cliente servidor, donde se presta servicio a $\mathcal{C}$ clientes de forma simultánea. Los clientes son todos homogéneos, es decir, leen los datos a la misma tasa (250 *kbits/s*) y el tamaño de su memoria es de 1 *Mbyte*. Además, inician las sesiones de forma desincronizada entre ellos. El tráfico de fondo (*background*) es de tasa constante (*Constant Bit Rate*, CBR), como el utilizado en algunas simulaciones desarrolladas en [3] y [5], utilizando todo el ancho de banda disponible en los enlaces hacia los clientes, menos aproximadamente unos 200 *kbits/s*, para obtener una carga apreciada por el cliente de aproximadamente un 25%. El servidor dispone de un ancho de banda de acceso de 5 *Mbits/s*, de los cuales 4 *Mbits/s* se utilizan para reservas de recursos. En esta situación el número máximo de clientes a los que se puede dar servicio es de 16, pero ello supone no poder reducir el coste, debido a que en ese caso se reserva una tasa igual a la de lectura del cliente. Por otra parte, la señalización extra también afecta al número máximo de clientes real, ya que a su vez consume ancho de banda reservable, y como se verá en algunas simulaciones puede ser considerable. Por todo esto, se supone un número máximo de clientes de 14.

4.1 Evolución de la eficiencia media del sistema en función de *m* y $\mathcal{C}$

En la Fig. 5, se muestra la evolución de la eficiencia media del sistema en función de *m* y de $\mathcal{C}$. Como se observa, la eficiencia disminuye a medida que aumentan *m* y $\mathcal{C}$. Esto depende de forma directa de la cantidad de ancho de banda que se reserva en modo ReR. A medida que este ancho de banda es menor, también es menor la reducción del coste que se obtiene. La gráfica está dividida en tres zonas destacables para *m* creciente. Para $m < \mathcal{C}$, a medida que *m* crece, disminuye la eficiencia debido a que α_{Res} es cada vez menor. $\mathcal{C}$ no afecta a la eficiencia en este área ya que al ser mayor que *m*, α_{Res} no varía. Únicamente se aprecia un ligero decrecimiento con $\mathcal{C}$ alta debido a que el servidor está muy cargado y el tráfico que circula en BE es elevado. Para $m = \mathcal{C}$, se observa un punto de inflexión a partir del cual α_{Res} se mantiene constante para $m > \mathcal{C}$. Si $m > \mathcal{C}$, la eficiencia se mantiene aproximadamente constante, disminuyendo

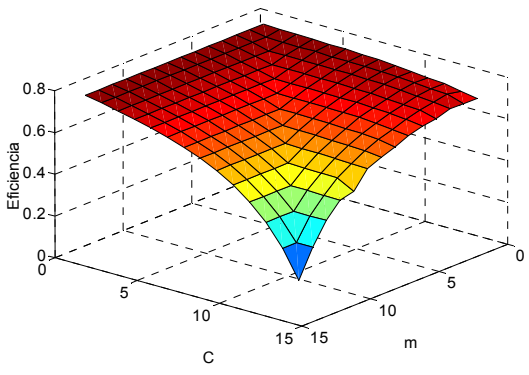


Figura 5: Eficiencia media en función de *m* y $\mathcal{C}$.

a medida que $\mathcal{C}$ aumenta. Ello se debe a que la tasa α_{Res} no varía para todos estos casos, ya que depende de $\mathcal{C}$ (5). Resumiendo, para conseguir una eficiencia alta, el servidor deberá utilizar una $m < \mathcal{C}$ que sea lo más pequeña posible, ya que a medida que m se aproxima a $\mathcal{C}$, la eficiencia disminuye hacia 0. El caso más favorable es para $m = 1$ (reserva de recursos para un cliente), aunque como veremos a continuación es necesario analizar la cantidad de señalización extra necesaria para determinar el valor óptimo a utilizar.

4.2 Evolución del número extra de mensajes RSVP en función de *m* y $\mathcal{C}$

En la Fig. 6, se puede observar la evolución de los mensajes de señalización extra debidos al inicio de sesión. Las cantidades indican el total de mensajes utilizados en una simulación donde los $\mathcal{C}$ clientes iniciaban 2 sesiones sucesivas. Similarmente, la Fig. 7 muestra la dinámica de los mensajes relacionados con el final de las sesiones. Estos dos casos presentan un comportamiento idéntico, debido a que se utiliza el mismo número de mensajes extra tanto al inicio como al final. Examinando las Figuras 6 y 7 en las tres zonas destacables se advierte lo siguiente: para $m < \mathcal{C}$, a medida que *m* crece, el número de mensajes extra aumenta, ya que es posible dar servicio a más clientes, y por tanto, la señalización necesaria es mayor, ya que depende de $\mathcal{C}$ mientras $m > \mathcal{C}$ (9) (10). En función de $\mathcal{C}$ la señalización disminuye, debido a que al iniciarse dos sesiones, la segunda se encuentra con el servidor dando servicio a otros clientes. Esto implica que para $\mathcal{C}$ suficientemente grande, el servidor se encuentre muy probablemente en estado $m < \mathcal{C}$ y por tanto, no sea necesaria señalización extra. A medida que *m* está más cercana a $\mathcal{C}$ esta probabilidad disminuye. Para $m = \mathcal{C}$, se observa un punto de inflexión a partir del cual la señalización extra se mantiene constante para $m > \mathcal{C}$. Cuando $m > \mathcal{C}$, la señalización extra se mantiene constante, disminuyendo a medida que $\mathcal{C}$ disminuye. Ello se debe a que en estos casos nunca se alcanza a *m*, por lo que la señalización sólo depende de $\mathcal{C}$. En este caso, y de forma similar al de la eficiencia, la zona óptima de trabajo es para $m < \mathcal{C}$, con *m* lo más pequeña posible.

La Fig. 8 muestra los mensajes de señalización extra debidos a la necesidad de actualizar la tasa que reservan algunos clientes, para evitar el rechazo de una nueva reserva. Examinando la Fig. 8 en las tres zonas anteriores se revela lo siguiente: para $m < \mathcal{C}$, a medida que *m* crece, el número de mensajes extra necesarios aumenta, ya que es posible dar servicio en modo ReR a más clientes y por tanto, la señalización necesaria será mayor, ya que depende de $\mathcal{C}_{ReR}$ (que será mayor con *m* (9)). En función de $\mathcal{C}$ la señalización disminuye, debido a que al iniciarse dos sesiones, la segunda se encuentra con el servidor prestando servicio a otros clientes. Esto implica que para $\mathcal{C}$ suficientemente grande, el servidor se encuentre muy probablemente en estado $m < \mathcal{C}$ y con todas las tasas actualizadas. A medida que *m* está más cercana a $\mathcal{C}$ esta probabilidad disminuye. Para $m = \mathcal{C}$, se observa un punto de inflexión a partir del cual la señalización extra se mantiene constante para $m > \mathcal{C}$. Cuando $m > \mathcal{C}$, la señalización extra se mantiene constante, disminuyendo a medida que $\mathcal{C}$ disminuye. Ello es debido a que en estos casos nunca se alcanza a *m*, por lo que la señalización sólo depende de la probabilidad de encontrar un determinado $\mathcal{C}_{ReR}$ que es mayor con $\mathcal{C}$. De nuevo, para estos mensajes de señalización se requiere que la zona óptima de trabajo sea $m < \mathcal{C}$, con *m* lo más pequeña posible. La evolución de los mensajes de señalización extra debidos a la necesidad de liberar alguna reserva por ser $m = \mathcal{C}_{ReR}$ para dar servicio a una nueva petición se muestra en la Fig. 9. Se observan nuevamente, tres zonas destacables: para $m < \mathcal{C}$, a medida que *m* crece, el número de mensajes extra necesarios disminuye, ya que la probabilidad de

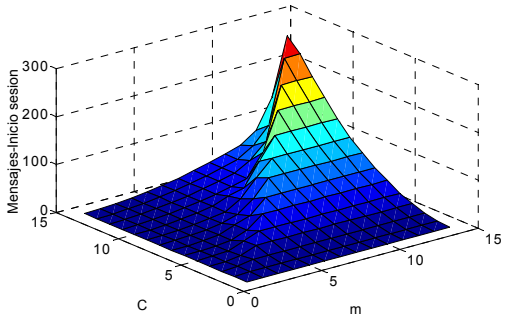


Figura 6: Mensajes de señalización extra (inicio sesión) en función de *m* y $\mathcal{C}$.

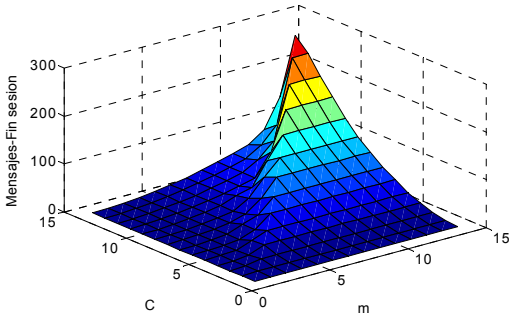


Figura 7: Mensajes de señalización extra (fin sesión) en función de *m* y $\mathcal{C}$.

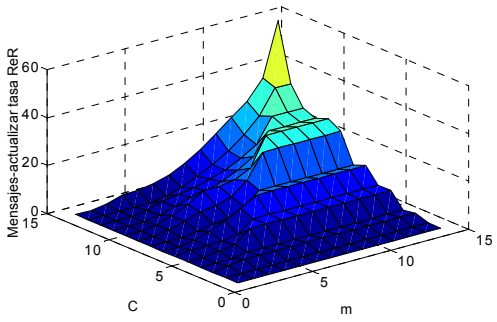


Figura 8: Mensajes de señalización extra (actualizar tasa de las reservas) en función de los parámetros *m* y $\mathcal{C}$.

que una nueva petición se encuentre con $m = \mathcal{C}_{ReR}$ se reduce rápidamente. Para $m = \mathcal{C}$, se observa un punto de inflexión a partir del cual la señalización extra es nula para $m > \mathcal{C}$. Cuando $m > \mathcal{C}$, la señalización extra es nula, ya que *m* siempre es mayor que $\mathcal{C}_{ReR}$ (10). Este caso, a diferencia de los demás, requiere que se trabaje en la zona $m > \mathcal{C}$, o próximo a $m = \mathcal{C}$. Finalmente, la Fig. 10 muestra la señalización extra total necesaria, suma de las anteriores. Como se puede deducir de la visualización de las gráficas anteriores, la zona óptima de trabajo está en $m < \mathcal{C}$, pero dado que la señalización debida a la necesidad de liberar reservas si $m = \mathcal{C}_{ReR}$ es mayor para *m* menor, se aprecia una zona óptima donde la señalización es mínima, que es en la que debería operar el servidor. Esto implica que el valor de *m* empleado dependerá del número de clientes a los que se dé servicio en cada momento, para reducir así la señalización trabajando en la zona óptima. Por otro lado, cabe recordar que *m* deberá ser lo más pequeña posible para maximizar la eficiencia media del sistema.

5 Conclusiones

En este trabajo, se ha analizado cómo debe realizarse el servicio simultáneo de varios flujos semi-elásticos. El principal problema que se plantea es la gestión del ancho de banda de acceso a la red que posee el servidor. Este ancho de banda se puede agotar fácilmente si no se gestionan debidamente las reservas de recursos. Se ha propuesto un método genérico de servicio de flujos semi-elásticos, basado en permitir que *m* clientes reserven recursos simultáneamente, repartiéndose el ancho de banda de acceso de igual forma. Para gestionar este ancho de banda es necesario utilizar señalización extra respecto al servicio a un solo cliente. Esta señalización extra se produce, si es necesaria, al inicio y final de una sesión, para actualizar reservas realizadas por clientes anteriores y para liberar reservas menos urgentes (las que llevan más tiempo) que la última petición de reserva recibida. Tal y como se ha visto, para realizar correctamente esta gestión se requiere que el protocolo de señalización de reserva de recursos permita al servidor, además de al cliente, también poder cambiar la reserva. Como casos concretos se ha analizado el servicio con reserva de recursos por cliente (con *m* igual al número máximo de clientes a los que se puede dar servicio) y el servicio con reserva de recursos para un cliente (con *m* igual a 1). El sistema se ha simulado para analizar como evoluciona la eficiencia media del sistema, así como la señalización extra necesaria. De ello se deduce que la eficiencia es mayor a medida que *m* es menor, dado que el ancho de banda que reserva cada cliente también es mayor. En cuanto a la señalización extra se observa una zona donde la señalización es mínima con *m* menor que el número de clientes $\mathcal{C}$, pero aumentando con $\mathcal{C}$. Ello permite concluir que el servidor deberá variar el valor de *m* en función del número de clientes para trabajar en la zona óptima, utilizando siempre el valor más pequeño posible para además maximizar la eficiencia media.

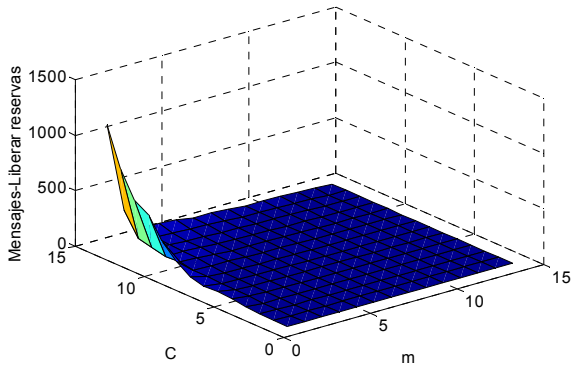


Figura 9: Mensajes de señalización extra (liberar reserva) en función de *m* y $\mathcal{C}$.

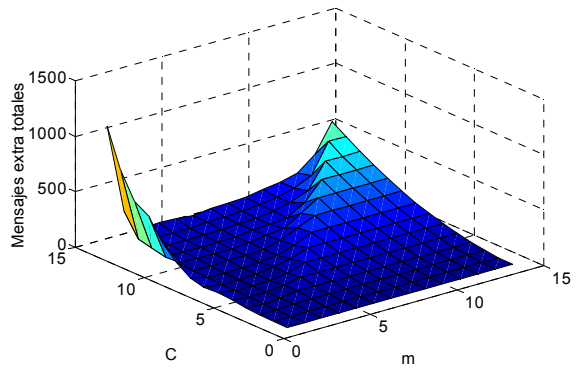


Figura 10: Mensajes de señalización extra en función de *m* y $\mathcal{C}$.

Agradecimientos

Este trabajo ha sido parcialmente financiado por los proyectos de investigación PRIME-IP (TIC2000-1734-C03-01), FAR-IP (TIC2000-1734-C03-03) y DISQET (CICYT - TIC2002-00818).

Referencias

[1] V. Fineberg, "A Practical Architecture for Implementing End-to-End QoS in an IP Network", IEEE Communications Magazine, pp. 122-130, Enero 2002.

[2] B. Stiller, P. Reichl, S. Leinen, "Pricing and Cost Recovery for Internet Services: Practical Review, Classification, and Application of Relevant Models", NETNOMICS: Economic Research and Electronic Networking, vol. 3, num. 2, pp. 149-171, Sept. 2001.

[3] M. Postigo-Boix, J. García-Haro, M. Aguilar-Igartua, "Cost Minimization Study of Semi-Elastic Flows Using Internet", in the Proceedings of the IEEE International Conference on Communications 2002 (ICC 2002), New York, USA, pp. 2237-41, Abril 2002.

[4] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) – version 1 functional specification," RFC 2205, IETF, Sept. 1997.

[5] M. Postigo-Boix, J. García-Haro, M. Aguilar-Igartua, "Cost Minimization Study in the Client-Server Transmission of Semi-Elastic Flows Using Internet", in the Proceedings of the 2001 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria, B.C., Canada, pp. 188-191, Agosto 2001.

[6] The Network Simulator – ns-2, http://www.isi.edu/nsnam/ns/

Sesión 1B

Aplicaciones y servicios telemáticos

Modelo de servicios de colaboración basados en SIP

Miguel Gó mez y Tomás P. de Miguel
Departamento de Ingeniería Telemática. Universidad Politécnica de Madrid
E.T.S.I. Telecomunicación, Ciudad Universitaria s/n. 28040 Madrid
Teléfono: 91 336 7366 Ext. 436 Fax: 91 336 7333
E-mail: gomez@dit.upm.es y tmiguel@dit.upm.es

Abstract. *Collaborative work services, which are mainly based on video-conferencing systems, have evolved intensively during last years pulled by recent innovations in networks and communication protocols. This paper tries to identify requirements for the creation of next generation collaborative services taking advantage of the new capabilities offered by the SIP signalling protocol. It also proposes a generic model and application architecture to support different conference topologies, control frameworks and interaction schemes. Finally, it describes some experiences that have been carried out to prove the easy customization, rapid prototyping and fast deployment capabilities offered by the model, both in IPv4 and IPv6 network environments.*

1 Introducción

La introducción de nuevos protocolos y redes de nueva generación que aúnan mayor ancho de banda, mayor fiabilidad y nuevas facilidades, está extendiendo el ámbito de aplicación de los servicios de colaboración multimedia. Frente a los rígidos sistemas antiguos, los nuevos servicios ofrecen más interactividad y la posibilidad de intercambiar medios más sofisticados.

El acceso del mercado doméstico a las nuevas tecnologías de banda ancha mediante cable o DSL abre la posibilidad de utilizar mensajería multimedia y aplicaciones de videoconferencia, hasta hace poco reservadas únicamente a empresas y entidades que contasen con redes privadas de alta capacidad.

A esto se une el desarrollo de la Nueva Generación de Internet basada en el estándar IPv6 que, además de resolver el actual problema de direccionamiento, va a facilitar la introducción de funcionalidades avanzadas que permitirán desarrollar servicios hasta ahora solo accesibles en entornos muy restringidos.

Finalmente, el panorama se completa con el despliegue de las redes de telefonía móvil de tercera generación, que empiezan a disponer de terminales con los que poder diseñar nuevas formas de colaboración mucho más flexibles que hasta ahora.

Este artículo presenta un modelo general de servicios de colaboración especialmente adecuado para facilitar la construcción rápida de servicios adaptados a las necesidades que demande el usuario en cada momento. Así mismo se exponen las experiencias piloto realizadas y las conclusiones y líneas de investigación futuras que estas experiencias han permitido extraer.

2 Nuevos servicios de colaboración

Los servicios de colaboración más difundidos hasta ahora han sido los basados en el protocolo H.323 de la Unión Internacional de las Telecomunicaciones (ITU) [1]. H.323 es un protocolo maduro, robusto y completo, sin embargo no puede decirse que haya logrado una aceptación masiva en el ámbito de las comunicaciones multimedia en tiempo real sobre IP. Existen varios factores que permiten explicar este fenómeno. El primero de ellos es sin lugar a dudas que H.323 fue una norma adelantada a su tiempo, ya que en el momento de su aparición sólo un conjunto reducido de empresas y entidades educativas contaban con las redes y dispositivos necesarios para implantar este tipo de entornos de colaboración.

Sin embargo esto no explica por qué su adopción no es mayor en la actualidad. Hay que tener en cuenta otros factores como, por ejemplo, que H.323 es un protocolo de escalabilidad limitada; debido a la ineficiencia de ciertos protocolos (por ejemplo T.124). Además, es un protocolo pesado y complejo. Está basado en ASN.1 que es muy pesado, difícil de implementar y con características poco utilizadas en la práctica. Al mismo tiempo es un protocolo que no ofrece muchas facilidades de adaptación, y los sistemas de colaboración modernos emplean otros conceptos de coordinación.

Por último, y quizás se trate de la razón más importante, puede decirse que H.323 es una norma más cercana al mundo de los operadores de telecomunicaciones que al de Internet. Pretende replicar sobre redes de conmutación de paquetes los servicios ya existentes en las redes PSTN y, en todo caso, proporcionar una serie de valores añadidos. Podemos concluir que este tipo de herramientas no aprovecha las capacidades de las redes actuales, y mucho menos las de las redes de nueva generación por venir.

Los nuevos servicios de colaboración demandan esquemas de interacción mucho más ligeros, flexibles y escalables. Además, la disponibilidad de redes de banda ancha ha cedido el interés por la eficiencia en línea de los tradicionales protocolos de señalización frente a una aproximación más simp lista, pero a la vez más efectiva. Esto hace que estas nuevas aplicaciones encuentren su paradigma en el nuevo protocolo SIP (Session Initiation Protocol) del IETF ([2] y [3]).

SIP es un protocolo destinado al establecimiento, modificación y liberación de sesiones multimedia, orientado sobre todo a la creación de servicios de voz sobre IP. Al tratarse de un estándar proveniente del mundo de Internet, se trata de una propuesta mucho más cercana y adecuada a las necesidades del sector. SIP reutiliza múltiples conceptos de probada solvencia como, por ejemplo, un modelo transaccional similar al de HTTP, intercambio de mensajes codificados en ASCII, arquitectura basada en un modelo peer to peer, etc. En resumen, se trata de un protocolo sencillo, ligero, escalable y altamente extensible, ya que se haya preparado para la incorporación de nuevos servicios y tecnologías conforme estos vayan apareciendo.

Las aplicaciones de colaboración de nueva generación también se verán ampliamente beneficiadas por el despliegue de la versión 6 del protocolo IP [4], ya que la Internet de Nueva Generación aporta una serie de mejoras altamente aprovechables para el desarrollo de nuevos servicios de colaboración.

Una de las principales mejoras es la ampliación a 128 bits del rango de direccionamiento. Aunque en principio puede parecer irrelevante, si deseamos que los nuevos terminales 3G así como multitud de utensilios de uso cotidiano cuenten con una dirección IP, la mejora obtenida se hace evidente. Además, la eliminación de la escasez de direcciones existente en la actualidad nos permitirá desprendernos de herramientas como los NAT, que al eliminar el ámbito global de las direcciones IP dificultan enormemente el establecimiento de sesiones de colaboración y arquitecturas peer to peer como las propuestas por el protocolo SIP.

IPv6 también mejora el soporte a los mecanismos de calidad de servicio con la inclusión de dos nuevos campos en la cabecera IP destinados a tal efecto: una etiqueta de flujo de 20 bits y un identificador de tipo de tráfico de 8 bits. Con la adopción de IPv6 se pretende mejorar la implantación y escalabilidad de mecanismos como DiffServ e IntServ, que resultan de gran utilidad para el envío de tráfico multimedia con requisitos de tiempo real a través de Internet.

Esta nueva versión de IP también incorpora un soporte multicast mejorado, ya que la existencia de diversos ámbitos de direccionamiento permiten controlar mucho mejor su alcance y dotarlo de una utilidad práctica de la que carece en la actualidad, ya que hoy en día prácticamente sólo es posible realizar difusión en entornos de área local.

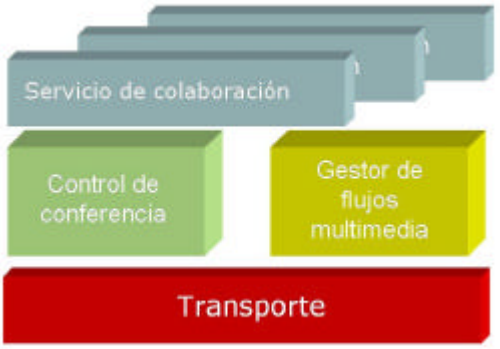


Figura 1: Arquitectura de un servicio de colaboración.

Por último, IPv6 también proporciona una serie de servicios de valor añadido de resultan de gran interés para el desarrollo de servicios de colaboración avanzados, y de los que las aplicaciones pueden beneficiarse de forma transparente sin más que adoptar este nuevo protocolo a nivel de red. Entre las más interesantes citamos los servicios de movilidad y seguridad a nivel IP.

Todavía es pronto para aventurar si se cumplirán las expectativas depositadas sobre estas nuevas tecnologías, ya que se encuentran en una fase muy incipiente de desarrollo, pero el amplio respaldo con el que cuentan en el entorno académico y empresarial permite una cierta dosis de optimismo. También cabe resaltar la favorable acogida por parte de los usuarios que han tenido las primeras herramientas que hacen uso de un reducido conjunto de las funcionalidades de movilidad, presencia y mensajería instantánea. El ejemplo más destacado de este grupo de herramientas diseñadas recientemente es Microsoft Windows Messenger.

3 Arquitectura

La arquitectura propuesta en la Fig. 1 identifica las funciones típicas de un servicio de colaboración moderno, donde el número de participantes y el grado de interactividad entre ellos puede llegar a ser muy elevado. Eso hace que cobren especial interés los servicios de control de escenario o la gestión de políticas y derechos de los usuarios que participan en una sesión.

Una sesión es una instancia de un servicio de colaboración y se define como el mecanismo que soporta la comunicación de personas a través de ordenadores. En un servicio de colaboración el conjunto de participantes intercambia flujos multimedia o comparte estados de una aplicación [5].

Aunque el comportamiento puede variar mucho, todos los servicios comparten en mayor o menor medida ciertas funcionalidades. Los elementos comunes a todo servicio de colaboración son: el gestor de servicio, el gestor de conferencia, el gestor de flujos multimedia y el módulo de transporte. A estos hay que unir la interfaz de usuario, que puede actuar a uno o varios niveles, dependiendo de la capacidad que se de al usuario de interactuar con los niveles inferiores.

El control principal recae sobre el gestor del servicio de colaboración, que es el encargado de seleccionar y particularizar el servicio que se va a prestar en cada sesión. Cada servicio demanda unos componentes diferentes (audio, video, etc.) que al configurarlos de diversas formas proporcionan diferentes escenarios de interacción. Por ejemplo, en una tele-clase sólo se difunde un audio, un vídeo y una pizarra (los del profesor) y el resto de sitios asisten de forma pasiva, mientras que en una tele-reunión todos los participantes intercambian su audio y su vídeo simultáneamente.

El gestor de servicio define qué escenarios de interacción estarán disponibles durante la sesión de colaboración, la configuración de los componentes de la aplicación para dichos escenarios y la forma de participación o rol con el que será capaz de intervenir cada uno de los participantes. En el caso más sencillo, la aplicación contará con un único escenario de interacción y el módulo de gestión del servicio de colaboración será una mera entidad lógica. En casos más complejos, se programarán varios escenarios durante el establecimiento de la sesión y la gestión del servicio consistirá en la activación de uno u otro a través del control de sesión.

3.1 Gestor de conferencia

El gestor de conferencia determina cuando y como se establece y gestiona cada sesión del servicio. La sesión de colaboración queda definida por los flujos multimedia que están siendo intercambiados, sus propiedades asociadas (codec, calidad, tamaño, etc.) y la jerarquía lógica establecida entre las estaciones participantes.

Por tanto, el gestor de conferencia es el encargado de crear, modificar o borrar conferencias. Consiste en un conjunto de protocolos que sirven para coordinar a todos los participantes de la sesión. Los mensajes de control de conferencia pueden ser de dos tipos: órdenes y notificaciones. Las órdenes son instrucciones o peticiones para modificar el estado de la sesión en curso, mientras que las notificaciones son precisamente los anuncios de dichos cambios de estado como, por ejemplo, cambios de participantes, cambios de políticas de operación o cambios en la configuración de flujos.

Como ya hemos comentado anteriormente, SIP es el protocolo de señalización más habitual en las aplicaciones de colaboración de nueva generación. Existen tres topologías básicas de conferencia multipunto en SIP [6].

La primera de ellas, denominada conferencia multienvío a gran escala, se basa en la utilización de una o varias direcciones multicast para la conferencia. Cada participante se une a bs grupos multicast y envía sus flujos de datos a dichos grupos. En este caso la señalización SIP se emplea únicamente para indicar a los participantes a qué grupos multicast deben unirse. De hecho, si los usuarios conocen las direcciones multicast que se están usando en la conferencia pueden unirse a la sesión sin necesidad de intercambiar ningún tipo de señalización SIP. Por ejemplo, si obtienen su configuración de un fichero a través de una página Web utilizando SDP [7] u otro mecanismo análogo. Esta es la topología de conferencia más apropiada cuando se desea incorporar un alto número de participantes, ya que tanto el control como el procesado de flujos se encuentran distribuidos entre todos los participantes, permitiendo una gran escalabilidad. Sin embargo, en IPv4 sólo es posible el multicast dentro de la propia subred y en IPv6 no todas las familias de routers incorporan todavía soporte multicast, por lo que este tipo de topologías de conferencia sólo resulta útil en entornos muy concretos.

Otra posible topología de conferencia pasaría por la utilización de un servidor de conferencia, que actúa como un agente de usuario SIP más. Cuando los usuarios desean unirse a la conferencia cursan una llamada SIP al servidor de conferencia, que establece relaciones SIP punto a punto con cada uno de ellos. El servidor acepta los flujos multimedia de cada uno de los participantes, los mezcla o conmuta si procede, y envía a cada usuario los flujos correspondientes. Puede verse que este modelo se haya limitado por el ancho de banda y la capacidad de procesado disponibles en el servidor de conferencia, por lo que generalmente sólo resulta conveniente para conferencias de tamaño medio.

Por último existe el modelo que podríamos llamar de control centralizado pero medios distribuidos. En este caso también existe un servidor central de conferencia, pero se encarga sólo de la señalización, mientras que los flujos multimedia se envían directamente entre los participantes. El envío de flujos puede realizarse mediante multicast o multi-unicast. En este modelo de conferencia el servidor hace uso de las capacidades de control de llamada por parte de terceros que proporciona SIP. Si no hacemos uso de direcciones multicast, puede verse que cada participante ha de enviar una copia de sus flujos multimedia a los restantes miembros, por lo que la escalabilidad de este modelo quedaría comprometida por el ancho de banda disponible en las estaciones participantes.

Al margen de estos modelos, SIP soporta otros tipos de conferencia como, por ejemplo, las conferencias ad-hoc o derivadas de la combinación de diversas llamadas SIP no-programadas. También es posible incluir múltiples servidores de conferencia para permitir la extensión de las topologías en forma de estrella expuestas a topologías en forma de árbol jerárquico, posibilitando así abordar escenarios más complejos en los que no todos los participantes tengan conectividad entre sí. Sin embargo todos estos tipos pueden verse como derivados o casos particulares de los expuestos anteriormente.

A la hora de diseñar una aplicación de colaboración podemos preseleccionar una topología de conferencia de entre las enumeradas o incorporar mecanismos que permitan su selección dinámica en función de los medios disponibles y el número y características de los participantes.

3.2 Gestor del servicio

El gestor del servicio es el encargado de seleccionar y configurar la funcionalidad que la aplicación ofrece para presentar una determinada experiencia de colaboración al usuario. Los múltiples flujos de información que la aplicación es capaz de prestar (audio, video, pizarra compartida, etc.) pueden configurarse de diversas formas para proporcionar diferentes escenarios de uso. El gestor del servicio tiene dos componentes: la interfaz de usuario y el control de sesión.

La interfaz de usuario permite la interacción entre el usuario y el servicio de colaboración ofertado por la aplicación. Ha de estar íntimamente relacionada con la lógica de presentación y el gestor de servicios, ya que la interfaz a mostrar dependerá fuertemente del escenario de control seleccionado y los medios que estén siendo intercambiados en cada momento.

A la hora de definir la interfaz de usuario de una aplicación de colaboración existen dos tendencias: la personalizable y la fija. Dado que las aplicaciones de colaboración suelen estar destinadas a mejorar la productividad y permitir el trabajo en equipo, resulta conveniente permitir a los usuarios personalizar el escritorio de trabajo y adecuarlo a sus preferencias. Por otro lado, el éxito de la colaboración se basa en que todos lo usuarios reciben la misma información y, por tanto, dicha personalización podría desvirtuar el entorno de colaboración. El equilibrio podría hallarse en proporcionar a los usuarios herramientas de personalización controladas tipo "*skins*", permitiendo respetar así una serie de normas que mantengan la uniformidad en el conjunto de aplicaciones cliente.

El control de sesión es un elemento esencial en todo servicio de colaboración, pues gestiona el acceso a los recursos compartidos a través de la política de control de escenario. Hay cuatro tipos básicos de políticas de control de escenario [8]:

1. *Sin control*. Cada participante puede utilizar cualquier recurso sin restricción.

2. *Control implícito*. Los permisos se asignan a un grupo de participantes antes de empezar la sesión.

3. *Control explícito*. Los permisos se obtienen durante la sesión. Por ejemplo, pulsando un botón para poder hablar.

4. *Control por moderador.* Un participante otorga o deniega el permiso en cada momento

En todo caso, cuando un usuario pide un cambio de escenario es necesario notificar al resto de los usuarios. Para ello será necesario establecer un mecanismo de intercambio de mensajes de control sobre la topología de conferencia subyacente. En el caso de las conferencias establecidas mediante señalización SIP como las que nos ocupan, existen diversas opciones para implementar un control de sesión distribuido ([9] y [10]).

En primer lugar es posible encapsular la señalización de control sobre los propios mensajes SIP. Esta opción es la que emplean en la actualidad algunas herramientas como, por ejemplo, Windows Messenger para el establecimiento de sesiones T.120. Otra opción sería emplear SDP sobre SIP para negociar el establecimiento de un flujo de control independiente.

Funcionalmente, las prestaciones son idénticas en ambas opciones cuando los usuarios hagan uso de la misma aplicación cliente, pero las diferencias quedan patentes a la hora de garantizar la compatibilidad con otros agentes de usuario SIP. Si encapsulamos la señalización sobre mensajes SIP genéricos (por ejemplo, una primitiva de mensajería instantánea), un agente de usuario SIP cualquiera será capaz de recibirlos y procesarlos, pero no entenderá su contenido y, o bien lo ignorará, o bien hará un uso indebido de dicho contenido. Además, el emisor no tendrá constancia de problema alguno y será incapaz de identificar si el mensaje ha sido correctamente procesado. Esto haría necesarios mecanismos más complejos, como procesar la información sobre el agente de usuario contenida en los campos SDP y enviar información particularizada a las diversas versiones, etc. Sin embargo, si el flujo de control de sesión se negocia como una fuente de datos más, aquellos agentes incapaces de gestionar dicho flujo lo rechazarán durante la negociación SDP, permitiendo así su identificación en el emisor y, si se desea, un tratamiento particularizado.

Por supuesto, ninguna de estas soluciones garantiza una interoperabilidad total con otras herramientas de colaboración, tan sólo permite garantizar un subconjunto mínimo de funcionalidades. La interoperabilidad completa pasaría por la definición de un protocolo estándar de control de sesión acoplable a SIP, al igual que se le acoplan otros

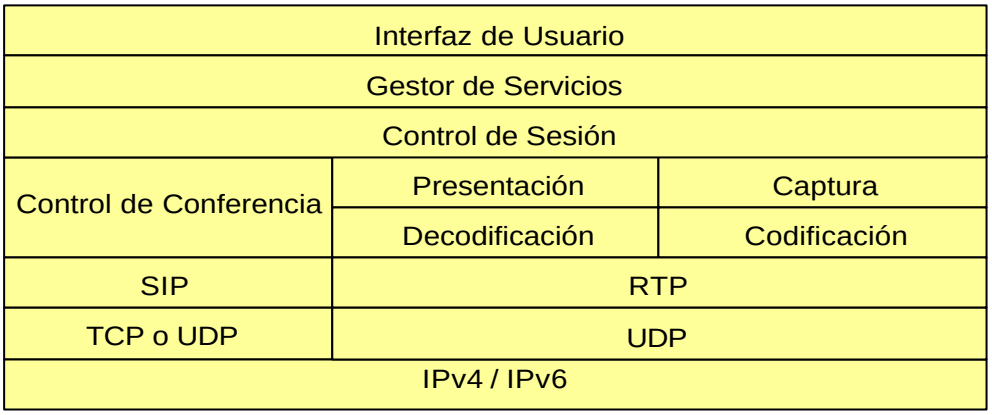


Figura 2: Arquitectura propuesta

protocolos, como SDP para la negociación de flujos multimedia.

3.3 Nivel de transporte

El nivel de transporte está formado por tres componentes: arquitectura de distribución, transporte de medios y nivel de red.

Como se ha mencionado anteriormente, el modelo contempla diferentes topologías de distribución. Desde la difusión utilizando protocolos multienvío a la red en estrella pasando por la utilización de un árbol de distribución jerárquico, que es la más probable pues se puede ajustar bien a las condiciones de la red física y puede conectar a un gran número de participantes en la sesión.

El transporte tiene como misión encapsular los datos de los distintos flujos de la aplicación y permitir su entrega con los requerimientos específicos de cada uno de ellos. Típicamente, en las aplicaciones de colaboración se ha empleado TCP para el transporte de la información de señalización y control y UDP para los flujos multimedia. Sin embargo, dado que la información de control representa un porcentaje muy pequeño respecto al tráfico total y además suele producirse en momentos muy concretos, esta práctica conlleva el mantener establecidas a lo largo de la sesión de colaboración una serie de conexiones TCP por las que se intercambia tráfico de forma muy esporádica. Esto hace que en la actualidad se tienda a emplear micro-conexiones TCP para la señalización o incluso UDP combinado con algún mecanismo de retransmisión.

Aunque es posible enviar directamente los flujos multimedia sobre UDP o encapsularlos empleando algún protocolo propietario, el estándar de Internet para el transporte de información de tiempo real es el protocolo RTP. Este protocolo no garantiza la entrega en tiempo real, pero proporciona los mecanismos necesarios para gestionar flujos multimedia como, por ejemplo, sincronización, reconstrucción, detección de pérdidas, seguridad e identificación de contenido. Si se combina con RTCP permite además obtener realimentación en el emisor sobre la calidad de recepción.

Finalmente el nivel de red es necesario para comunicar los distintos ejemplares de la aplicación. La adopción de un transporte de red u otro está acotada a este módulo; ya que el resto de las comunicaciones de la aplicación son locales a la máquina. Aunque IPv6 se perfila como el protocolo de la nueva generación, la plataforma actual es en su mayoría IPv4. En consecuencia lo ideal es diseñar la arquitectura para que sea posible funcionar en modo dual; permitiendo la conexión simultanea de terminales IPv4 e IPv6 en una misma sesión.

A la vista de lo expuesto anteriormente, la Fig. 2 muestra la propuesta de arquitectura detallada para la aplicación de colaboración de nueva generación.

4 Experiencias realizadas

Con objeto de realizar un análisis de viabilidad del modelo anteriormente expuesto, se han desarrollado una serie de prototipos que han permitido evaluar su rendimiento y prestaciones ante diversos casos de uso. Dado que estas implementaciones no pretenden ser aplicaciones comerciales a corto plazo, sino meras maquetas destinadas al prototipado y evaluación de resultados, los factores que han primado han sido la rapidez y facilidad de despliegue y la posibilidad de reutilización de bibliotecas ya existentes. Por ello, el lenguaje de programación elegido ha sido Java.

En una primera aproximación, no se ha pretendido implementar una aplicación de colaboración en su totalidad. Dado que el marco de control de sesión y gestión de servicios sobre SIP se encuentra aún en una fase muy temprana de desarrollo y existen diversas alternativas al respecto, se ha optado por implementar exclusivamente los niveles de control de conferencia, señalización y gestión de flujos multimedia. Esto posibilita que en un futuro estos prototipos sirvan como base para el estudio comparativo de diversas arquitecturas de control y gestión de servicios y usuarios.

4. 1 Componentes empleados

Como ya hemos comentado anteriormente, el objetivo de estos experimentos no ha sido el desarrollo de una implementación de la pila del protocolo SIP ni de un sistema de difusión RTP, sino demostrar que el modelo de colaboración propuesto proporciona un marco completo y flexible para la provisión de servicios de colaboración de nueva generación. Por ello, se ha tratado de reutilizar en la medida de lo posible aquellos componentes del modelo ya disponibles de forma comercial y/o experimental. Los principales módulos externos en los que se han apoyado los prototipos realizados son una biblioteca SIP, una biblioteca de captura y reproducción de medios, una biblioteca RTP y un proxy SIP con funcionalidades de servidor de registro y localización.

A la hora de seleccionar una biblioteca SIP en Java se presentan dos alternativas. Por un lado existen bibliotecas sencillas y comprensibles que proporcionan un subconjunto reducido de las funcionalidades del protocolo como, por ejemplo, la implementación Java del protocolo SIP proporcionada en la biblioteca jSIP [11]. La principal ventaja de este tipo de bibliotecas es su sencillez de manejo y la facilidad para introducir en ellas modificaciones y mejoras. Por otra parte, la aparición del API estándar JAIN SIP 1.0 [12] promovido por Sun Microsystems ha originado que un gran número de implementaciones SIP en Java opten por adoptar esta interfaz. Una de las principales ventajas de estas bibliotecas es su fiabilidad y garantía, ya que para poder obtener la certificación JAIN han de superar una serie de pruebas y controles. Además, por el hecho de implementar un API estándar, es posible sustituir la pila SIP subyacente por cualquier otra implementación conforme al API JAIN de forma completamente transparente para el resto del código. Sin embargo, no todas estas bibliotecas son de dominio público y, en caso de serlo, la elevada complejidad que presentan dificulta la modificación del código fuente para realizar cambios o mejoras.

A la hora de realizar los prototipos de aplicación de colaboración, se seleccionó en una primera fase de desarrollo la versión 0.8 de la biblioteca jSIP [11] por tratarse de una implementación reducida, comprensible y fácilmente modificable. Para la implementación de los prototipos IPv6 fue necesario adaptar esta biblioteca para posibilitar el manejo de de direcciones y conexiones IPv6, ya que no se encontraban soportadas. Conforme los prototipos se estabilizaron surgió la necesidad de hacer uso de funcionalidades SIP no incluidas en jSIP, por lo que se optó por utilizar la biblioteca SIP ofertada por el NIST (Nacional Institute of Standards and Technology) en el ámbito de su proyecto de voz sobre IP [13]. Las principales motivaciones para la elección de esta biblioteca de entre los diversos productos que cumplían la certificación JAIN SIP 1.0 fueron su gratuidad, la pertenencia al dominio público y las garantías de estabilidad y continuidad que ofrece por el hecho de formar parte de un proyecto activo del NIST.

Aunque en principio la utilización de un proxy SIP externo no era estrictamente necesaria en los prototipos, existen motivos que recomendaban su inclusión. En primer lugar, el proxy SIP actúa también como servidor de registro y localización, proporcionando servicio de movilidad SIP en el entorno de demostración y permitiendo emplear otros agentes de usuario independientes como, por ejemplo, MS Windows Messenger. También, al actuar como intermediario SIP, garantiza la coherencia sintáctica de los mensajes intercambiados por los agentes de usuario, que de otro modo podría quedar enmascarada al dialogar entre implementaciones semejantes. En los prototipos realizados se ha empleado la versión 0.5.0 del proxy partysip [14] para los experimentos en redes IPv4 y el SIP Express Router (SER) 0.8.10 de iptel.org [15] para las experiencias en entornos IPv6.

Una vez negociado mediante SIP el establecimiento de la sesión multimedia, es necesario contar con la lógica apropiada para el intercambio de información de tiempo real. Java Media Framework (JMF) 2.2.1 [16] de Sun Microsystems es un entorno completo e integrado en la plataforma Java que permite realizar de forma sencilla todas las fases que comprende este proceso: captura de flujos multimedia (audio y vídeo), codificación, intercambio mediante RTP y reproducción en destino. En el entorno IPv4 se ha utilizado el JMF 2.1.1b *performance pack* para Windows, mientras que para las experiencias IPv6 se ha seleccionado la implementación para Linux proporcionada por blackdown.org.

4.2 Prototipos IPv4

Como base para futuros entornos de colaboración, la primera experiencia consistió en la implementación de un cliente SIP IPv4 capaz de establecer comunicaciones de audio, vídeo y mensajería instantánea. Con vistas a su inclusión en posteriores entornos de conferencia más complejos, no se limitó la capacidad del cliente a la recepción de un único flujo de audio y vídeo, sino que se concibió como un cliente capaz de recibir y presentar un número teóricamente ilimitado de flujos multimedia. Este cliente ha sido probado en entornos de colaboración punto a punto como el reflejado en la Fig. 3, interaccionando tanto con clientes semejantes como con clientes SIP independientes como, por ejemplo, Microsoft Windows Messenger.

En una segunda fase se implementó un entorno de conferencia multipunto basado en la utilización de un servidor de conferencia. En nuestro caso el servidor de conferencia no realiza ningún tipo de conmutación

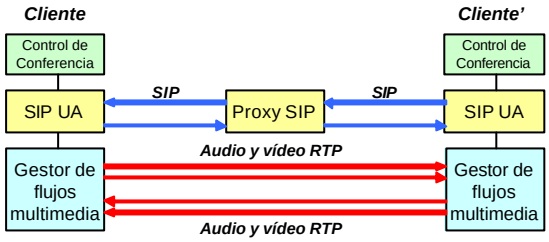


Figura 3: Entorno punto a punto.

o mezcla de medios, sino que se limita a replicar los flujos que aporta cada usuario y reenviarlos al resto de participantes. Como es lógico, para que esto sea posible los clientes han de ser capaces de reproducir localmente múltiples flujos de audio y vídeo. Como no resulta habitual que los clientes SIP comerciales presenten este comportamiento se introdujo lógica adicional en el servidor de conferencia que posibilitase emplear clientes SIP convencionales, enviando un único flujo de audio y vídeo a aquellos clientes que rechazasen sesiones con mayor número de elementos multimedia.

En la Fig. 4 puede verse el entorno de colaboración obtenido. Al no realizar el servidor ningún tipo de conmutación o mezcla de flujos RTP, con cada anexión o abandono de la conferencia es necesario además renegociar la sesión de intercambio de flujos con el resto de participantes. Este modelo cuenta con la ventaja de que reduce la complejidad y capacidad de procesado necesaria en el servidor de conferencia y además, al incrementar el intercambio de señalización SIP entre los participantes, proporciona un caso de estudio de mayor interés. Sin embargo, también es necesario tener en cuenta que el consumo de ancho de banda en el servidor de conferencia aumenta exponencialmente conforme lo hace el número de participantes, por lo que se convierte en el cuello de botella de este tipo de entornos de colaboración.

4.3 Prototipos IPv6

A la hora de realizar los experimentos sobre una red IPv6 fue necesario abandonar el entorno de desarrollo Windows, ya que el J2SDK actual no presenta soporte IPv6 en su versión para este sistema operativo. Por tanto, se ha migrado a Linux el entorno de experimentación y pruebas sobre IPv6, y deberá continuar así hasta que la pila IPv6 integrada en los sistemas Windows se consolide y Sun haga pública una versión para Windows del JDK con soporte IPv6.

Al ser código 100% Java, la migración no debería haber tenido impacto alguno, pero no obstante se han presentado dificultades en el ámbito de la captura de flujos multimedia. La versión para Linux de JMF, a pesar de estar respaldada por Sun, se trata de una versión bajo licencia de blackdown.org. Esta versión aún se encuentra en fase de desarrollo y presenta diversas carencias, en particular a lo que a captura de audio se refiere. Para solucionar estas carencias ha sido necesario hacer uso de los mecanismos de extensión que proporciona el entorno JMF para implementar un sistema propio de captura y reproducción de audio que acceda directamente al dispositivo /dev/dsp bajo Linux. Si bien esta solución ha resuelto satisfactoriamente los problemas de audio, cabe resaltar que resta portabilidad a la solución. La reciente aparición (5 de Mayo de 2003) de la versión 2.1.1e de Java Media Framework, que incorpora un *performance pack* para Linux, podría resolver estas deficiencias en el subsistema de audio y hacer innecesaria la implementación de un sistema propietario de reproducción y captura. Lamentablemente, a fecha de hoy no nos ha sido posible evaluar sus prestaciones.

Las experiencias realizadas sobre IPv6 han tratado de emular los entornos punto a punto y multipunto ya comentados en IPv4 (ver Fig. 3 y Fig. 4), obteniéndose resultados ampliamente satisfactorios. Cabe destacar que en este caso no ha sido posible probar la interoperabilidad con otros clientes SIP comerciales, dado que no existía en el momento de las pruebas ninguno disponible con soporte IPv6, por lo que sólo se han establecido sesiones entre ejemplares del cliente desarrollado. Hemos tenido noticias de que el grupo de investigación del Prof. Tim Chown en la Universidad de Southampton ha realizado una migración a IPv6 del cliente SIP contenido en el entorno VOCAL de Vovida.org [17]. En cuanto se encuentre disponible será posible emplear dicho cliente para realizar pruebas de interoperabilidad que garanticen la compatibilidad del entorno implantado.

Una de las principales ventajas de emplear IPv6 es que la aplicación se beneficia de forma transparente de los servicios avanzados que proporciona el nivel de red. Una de las experiencias realizadas en este ámbito ha consistido en emplear terminales dotados de servicio de movilidad a nivel IPv6 y realizar cambios de subred durante el transcurso de sesiones de colaboración establecidas en los entornos anteriormente enumerados (Fig. 3 y Fig. 4). Los resultados fueron ampliamente satisfactorios: tras un breve periodo de inactividad (<3s.) en el que se producía un corte de audio y la imagen de video quedaba congelada, la sesión continuaba de forma automática con normalidad en la nueva subred.

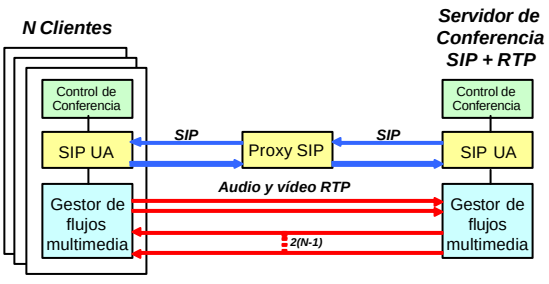


Figura 4: Entorno multipunto mediante Srv. de Conferencia.

5 Conclusiones

Basándonos en experiencias previas, presentamos un modelo de realización de servicios de colaboración adaptado a las nuevas redes y nuevas necesidades de los entornos distribuidos.

Este modelo permite crear servicios escalables, fáciles de entender, genéricos, fiables y seguros. A la vista de la información anteriormente expuesta y de las experiencias realizadas, puede concluirse que el protocolo SIP constituye un marco idóneo para la creación de este tipo de servicios en el ámbito del modelo propuesto. Por ello se perfila como el estándar de facto para el establecimiento de sesiones multimedia, por ser muy flexible extensible y fácil de adaptar a las arquitecturas de comunicación más utilizadas.

El modelo de aplicación de colaboración expuesto presenta una arquitectura enormemente flexible, ya que es capaz de trabajar tanto con las redes y protocolos existentes en la actualidad como con las redes, protocolos y servicios futuros. Además, las características de estos nuevos protocolos pueden incorporarse y complementarse para proporcionar nuevos servicios de colaboración, sin necesidad de revisar completamente toda la aplicación. Esto ha podido comprobarse en las experiencias realizadas sobre IPv6, ya que con unas leves modificaciones los prototipos han sido capaces de hacer uso de las nuevas redes y de los servicios avanzados que estas ofrecen.

También puede verse que se trata de un modelo ampliamente soportado, ya que cuenta con el respaldo tanto da la industria como de la comunidad académica y las primeras aplicaciones están contando con una gran aceptación entre los usuarios. Esto permite contar con una alta disponibilidad de software de base, por lo que es posible centrarse exclusivamente en el desarrollo de nuevos servicios a aplicaciones de alto nivel sin necesidad de preocuparse en implementar los niveles inferiores. Además, la alta disponibilidad de clientes SIP y su rápida difusión permite contar con un gran número de usuarios potenciales para los nuevos servicios desarrollados.

El modelo está alineado con los estándares que se están produciendo principalmente en IETF. Los protocolos que se están produciendo están todavía en las primeras fases de estandarización, pero las experiencias realizadas permiten asegurar que son fácilmente integrables en el entrono de generación de servicios de colaboración. Además, esto puede servir para evaluar las nuevas propuestas y proponer mejoras.

El entorno de los servicios de colaboración de nueva generación basado en el nuevo modelo se encuentra en una fase inicial de desarrollo, pero pensamos que en un breve plazo de tiempo podría completarse en su totalidad.

Referencias

[1] INTERNATIONAL TELECOMUNICATIONS UNION. *"Packet-based multimedia communication systems"*. ITU-T Recommendation H.323, Telecommunication Standardization Sector, ITU. Febrero de 1998.

[2] J. Rosenberg, H. Schulzrinne et al. *"SIP: Session Initiation Protocol"*. RFC 3261. IETF. Junio de 2002. http://www.ietf.org/rfc/rfc3261.txt

[3] William Stallings. *"Session Initiation Protocol"*. The Internet Protocol Journal, pp. 20-30, vol. 6, no. 1. Marzo de 2003.

[4] S. Deering, R. Hinden. *"Internet Protocol, Version 6 (IPv6) Specification"*. RFC 2460. IETF. Diciembre de 1998. http://www.ietf.org/rfc/rfc2460.txt

[5] V. Hilt, W. Geyer. *"A Model for Collaborative Services in Distributed Learning Environments"*. IDMS'97, Interactive Distributed Multimedia Systems and Services. Darmstadt, Alemania. Septiembre de 1997.

[6] J. Rosenberg, H. Schulzrinne. *"Models for Multi party Conferencing in SIP"*. IETF draft-rosenberg-sip-conferencing-models-02.txt. Julio de 2001.

[7] M. Handley, V. Jacobson. *"SDP: Session Description Protocol"*. IETF. RFC2327. Abril de 1998. http://www.ietf.org/rfc/rfc2327.txt

[8] F. Fluckiger. *"Understanding networked multimedia applications and technology"*. Prentice Hall. Nueva York, USA. 1995.

[9] C. Bormann, et al. *"Simple conference control protocol service specification"*. Internet draft, IETF. Marzo de 2001.

[10] Wu, Koskelainen, Schulzrinne. *"Use of Session Initiation Protocol (SIP) and Simple Object Access Protocol (SOAP) for Conference Floor Control"*. IETF draft. Enero de 2003.

[11] The jSIP Java SIP stack implementation: http://jsip.sourceforge.net/

[12] The JAIN APIs: http://java.sun.com/products/jain/

[13] The NIST Internet Telephony / VOIP project: http://snad.ncsl.nist.gov/proj/iptel/

[14] The partysip SIP proxy server: http://www.partysip.org/

[15] SIP Express Router: http://www.iptel.org/ser/

[16] Java Media Framework API: http://java.sun.com/products/java-media/jmf/

[17] The Vovida Open Communication Application Library (VOCAL): http://www.vovida.org

Verificación Biométrica mediante Firma On-Line Basada en Modelos Ocultos de Markov y Redes Neuronales

J.J. Igarza[1], R. Méndez[1], I. Hernáez[1], C. Vivaracho[2], I. Moro[2], D. Escudero[2]
[2]Departamento de Informática. Universidad de Valladolid
[1]Departamento de Electrónica y Telecomunicaciones. Universidad del País Vasco
Alameda Urquijo, s/n 48013 Bilbao
Teléfono: 946 01 41 24 Fax: 946 01 42 59
E-mail: jtpigugj@ehu.es, {raul,inma}@bips00.bi.ehu.es, {cevp,isaac,descudr}@infor.uva.es

Abstract. *Most people are used to signing documents and because of this, it is a trusted and natural method for user identity verification, reducing the cost of password maintenance and decreasing the risk of eBusiness fraud. With this method, identity is securely verified and an authentic electronic signature is created using biometric dynamic signature verification. Shape, speed, stroke order, off-tablet motion, pen pressure and timing information are captured and analyzed during the real-time act of signing the handwritten signature. The captured values are unique to an individual and virtually impossible to duplicate. This paper presents an investigation of various techniques for signature verification (HMM based and Neural Networks - NN). Different HMM and NN topologies are compared in order to obtain an optimized high performance signature verification system and signal normalization preprocessing makes the system robust with respect to writer variability.*

1 Introducción

El acceso seguro y natural a los Sistemas es un tema que está adquiriendo cada vez mayor importancia. Día a día crece la necesidad de verificar la identidad de las personas de una forma sencilla, rápida, cómoda y fácil de usar.

Tradicionalmente, en el proceso de identificación y control de acceso a sistemas o aplicaciones, se ha confiado en objetos, es decir, en sistemas basados en posesión, por ejemplo llaves o tarjetas inteligentes y en sistemas basados en conocimientos como PINs, o contraseñas. Sin embargo, los objetos se puede perder y los conocimientos se pueden olvidar, y ambos se pueden robar o copiar.

La *Biometría* se basa en la medición de diferentes características físicas personales e intransferibles, como por ejemplo la huella dactilar, el iris o la retina y en características del comportamiento individual, como la forma de hablar, escribir, firmar o teclear. Estas características personales, bien físicas o de comportamiento, permiten realizar la identificación de cada individuo de forma unívoca y ofrecer de esta forma la solución al problema generado por la seguridad convencional. Por ello, se considera que la solución biométrica es uno de los métodos más naturales y fiables de identificación y acceso seguro a sistemas y aplicaciones.

En general, la mayoría de los ciudadanos se muestran recelosos ante sistemas de identificación biométrica basados en características físicas como las huellas, el iris o la retina, ya que los encuentra asociados con cuestiones de índole penal. En cambio las características asociadas con su comportamiento como la voz, la escritura o la firma, aunque proporcionan menor fiabilidad que los anteriores, gozan de mayor aceptación social.

Cuatro grupos de Investigación de cuatro Universidades: Politécnica de Madrid, Politécnica de Cataluña, Valladolid y País Vasco participan conjuntamente en el Proyecto de Investigación [1] *"Aplicación de la Identificación de Personas mediante Multimodalidad Biométrica en Entornos de Seguridad y Acceso Natural a Servicios de Información"*, cuyo primer resultado común ha sido la creación de una Base de Datos Biométricos Multimodal [2] (huellas digitales, firmas y voz) que sirve de base para el resto de tareas de investigación de los grupos participantes.

El presente artículo es consecuencia de los trabajos de investigación posteriores sobre firma manuscrita a partir de dicha base datos.

2 Verificación de firmas

La firma manuscrita es uno de los modos de identificación de mayor uso y aceptación social; firmamos frecuentemente para manifestar los contenidos de cualquier tipo documento o para autenticar transacciones financieras. La verificación de las firmas se limita normalmente a una inspección visual, como si se tratara de la comparación de dos fotos, pero este método no es eficaz frente a falsificadores y en la mayoría de las ocasiones no se lleva a cabo proceso de verificación alguno. La automatización del proceso de verificación pretende mejorar la situación actual y eliminar el fraude.

La verificación automática de firmas se divide en dos áreas principales dependiendo del método de

adquisición de los datos: Por un lado, en la verificación de firmas *off-line*, la firma está disponible en un documento escrito que es escaneado para obtener la representación digital de la imagen. Por otro lado, en la verificación de firmas *on-line* se utiliza hardware como tabletas digitalizadoras, para registrar los movimientos del bolígrafo sobre el papel durante la escritura.

La verificación off-line se aplica a firmas de documentos pasados no adquiridas en ningún formato digital y en ella sólo interviene la forma de la firma. En cambio, en la verificación de firmas on-line, además de la representación espacial 2D se obtiene información de la dinámica de la firma como presión o inclinación del bolígrafo, con lo que resulta más difícil de falsificar. También se necesita la presencia del firmante en el momento de la captura digital.

3 Descripción del sistema

A continuación se hace una descripción de los módulos esenciales del sistema de verificación:

- módulo de adquisición de firmas on-line
- base de datos de firmas
- preprocesado de las firmas
- entrenamiento de los modelos
- selección de umbrales

3.1 Adquisición de firmas on-line

El sistema utiliza como dispositivo de captura una tableta digitalizadora de Wacom [3] modelo Intuos A-6 con interfaz USB. Dicha tableta proporciona 100 muestras por segundo de valores de presión y los cuatro grados de libertad: posición X, Y, acimut e inclinación del lápiz para cada muestra.

Los trazos sin presión, también llamados *pen-ups,* son igualmente muestreados, por lo que se conoce la trayectoria tanto con tinta como sin ella, lo cual supone disponer de una información extra que enrobustece al sistema.

La información de la firma, una vez digitalizada y convertida en una matriz, se almacena en un archivo, pudiendo pasar a engrosar la base de datos de firma o a ser verificada a continuación.

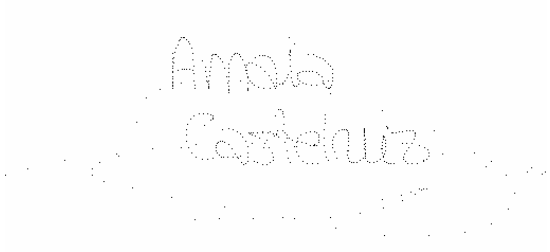


Fig 1 *Pen-downs* de una firma de la BD. Escala x2

La firma digitalizada consiste en una secuencia temporal de puntos muestreados a lo largo del acto de la firma a una frecuencia de muestreo fija determinada por el dispositivo de captura. Su longitud es directamente proporcional al tiempo de firma. En la figura 1 se observan los puntos muestreados con presión mayor que cero o pen-downs de una firma dinámica.

3.2 Base de datos de firmas on-line

El sistema posee una de base datos de firmantes [4], de cada uno de los cuales se dispone de 25 firmas propias. A su vez, cada firmante ha registrado 5 falsificaciones de cada uno de los 5 firmantes inmediatamente anteriores a él en la base de datos. Esto quiere decir que por cada firmante se disponen de 25 firmas falsificadas realizadas por 5 personas diferentes y de 25 firmas propias.

Cada subcorpus de la base de datos consta de un total de 75 individuos enlazados en una cola circular. Dado que la adquisición se realizó en cinco equipos distintos (Mataró, E.U.I.T. de Telecomunicación y E.T.S.I. Telecomunicación de la UPM, Valladolid y Bilbao) se dispone de un total de 375 individuos.

Los estudios que se presentan en este artículo han sido realizados sobre los subcorpora de Valladolid y Bilbao, esto es, sobre un total de 150 firmantes, con 3.750 archivos de firmas originales y otras tantas falsificaciones.

Las falsificaciones de la base de datos son de tipo entrenadas, en inglés *skilled forgery*, ya que el impostor repite varias veces la firma de la víctima antes de que la falsificación sea adquirida y almacenada definitivamente en la base de datos. Con el ánimo de mejorar aún más la calidad de las falsificaciones se estimuló a los participantes con un premio. Todo esto hace que la calidad de las falsificaciones sea alta.

3.3 Preprocesado de las firmas

Debido a que cada vez que firmamos lo hacemos de forma diferente, factores como la variabilidad en la velocidad de la firma, distinto tamaño, distinta ángulo de rotación y distinta zona de la tableta deben ser tenidos en cuenta en el preprocesado, de modo que, en la medida de lo posible, obtengamos una representación independiente de estos factores.

El módulo de preprocesado realiza una normalización en el tiempo de manera que todas las firmas resultantes tienen la misma longitud (número de muestras). Para ello se lleva a cabo una simple labor de interpolación o extrapolación dependiendo del número de muestras. Esta normalización temporal es opcional, ya que se puede mantener la longitud original de la firma a voluntad.

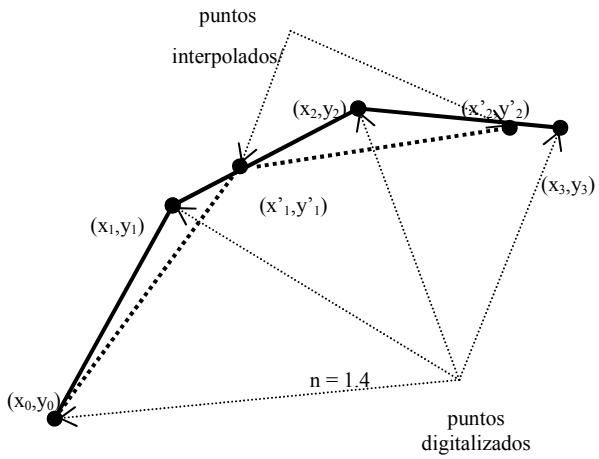


Fig 2 Algoritmo de normalización temporal

En el ejemplo de la figura 2, n = 1.4 representa la relación entre la longitud de la firma adquirida (420 muestras) y la de la firma normalizada (300 puntos). Los puntos adquiridos se representan como (x_j,y_j), con j = 0 hasta 420, y los puntos normalizados como (x'_i,y'_i), con i = 0 hasta 300.

Independientemente de que la relación sea mayor o menor que 1, las coordenadas tras la normalización serán:

$$(x'_i, y'_i) = (a * x_j + b * x_{j+1}, a * y_j + b * y_{j+1})$$

$$\text{siendo:} \begin{cases} j = \text{truncar}\,(i * n) \\ b = (i * n) - j \\ a = 1 - b \end{cases}$$

El mismo algoritmo se puede aplicar al resto de coordenadas como azimut, inclinación o incluso presión.

Una manera sencilla de eliminar la variabilidad en el tamaño (coordenadas X-Y) y rotaciones respecto a la tableta consiste en aplicar el método de Yang, Widjaja y Prasad [5]. Dichos autores proponen utilizar el valor absoluto del ángulo del segmento que une puntos consecutivos, a partir de las coordenadas X-Y de los mismos, mediante la siguiente fórmula:

$$\phi(k) = \arctan\left[\frac{\sum_{l=i+1}^{i+n} s_l^{(k)} \sin\theta'^{(k)}_l}{\sum_{l=i+1}^{i+n} s_l^{(k)} \cos\theta'^{(k)}_l}\right]$$

Siendo $\theta'^{(k)}_l = \theta_l^{(k)} - \theta_1$, donde θ_1 es el ángulo absoluto del primer segmento.

Esta fórmula realiza la normalización al mismo tiempo que resta el ángulo absoluto del primer segmento.

Para facilitar el cálculo del ángulo y mejorar la eficiencia computacional de dicho algoritmo hemos realizado unas pequeñas transformaciones en las fórmulas propuestas por Yang y otros, adaptándolas al algoritmo representado en la figura 2.

Como:

$$s_l^{(k)} \sin\left(\theta_l^{(k)} - \theta_1\right) = s_l^{(k)} \sin\theta_l^{(k)} \cos\theta_1 - s_l^{(k)} \cos\theta_l^{(k)} \sin\theta_1$$
$$s_l^{(k)} \cos\left(\theta_l^{(k)} - \theta_1\right) = s_l^{(k)} \cos\theta_l^{(k)} \cos\theta_1 + s_l^{(k)} \sin\theta_l^{(k)} \sin\theta_1$$

y además, $\Delta y_l^{(k)} = s_l^{(k)} \sin\theta_l^{(k)}$ y $\Delta x_l^{(k)} = s_l^{(k)} \cos\theta_l^{(k)}$

$$s_l^{(k)} \sin\theta'^{(k)}_l = \Delta y_l^{(k)} \cos\theta_1 - \Delta x_l^{(k)} \sin\theta_1$$
$$s_l^{(k)} \cos\theta'^{(k)}_l = \Delta x_l^{(k)} \cos\theta_1 + \Delta y_l^{(k)} \sin\theta_1$$

La fórmula de Yang, Widjaja y Prasad toma esta nueva forma:

$$\phi(k) = \arctan\left[\frac{\left(y_{i+n}^{(k)} - y_i^{(k)}\right)\cos\theta_1 - \left(x_{i+n}^{(k)} - x_i^{(k)}\right)\sin\theta_1}{\left(x_{i+n}^{(k)} - x_i^{(k)}\right)\cos\theta_1 + \left(y_{i+n}^{(k)} - y_i^{(k)}\right)\sin\theta_1}\right]$$

Esta fórmula, aunque aparentemente sea más compleja, resulta mucho más eficiente, puesto que $\cos\theta_1$ y $\sin\theta_1$ sólo se calculan una vez, ya que son valores constantes para todos los puntos de la firma. Además si este cálculo lo aplicamos a la longitud ya normalizada, el resultado final será:

$$\phi(i) = \arctan\left[\frac{\left(y'_{i+1} - y'_i\right)\cos\theta_1 - \left(x'_{i+1} - x'_i\right)\sin\theta_1}{\left(x'_{i+1} - x'_i\right)\cos\theta_1 + \left(y'_{i+1} - y'_i\right)\sin\theta_1}\right]$$

En su trabajo, Yang, Widjaja y Prasad discretizan a 16 ó 32 los valores obtenidos en esta fase para su procesado final. Según el modelo propuesto en este trabajo esta discretización no es absolutamente necesaria, razón por la cual se permite optar por un nivel determinado de discretización o por no utilizar ninguno.

3.4 Entrenamiento del sistema

El entrenamiento de cualquier sistema de verificación consiste en crear un modelo del objeto a verificar a partir de un conjunto de muestras del mismo. Los modelos así generados son almacenados en una base datos.

La eficacia del sistema depende fuertemente de lo representativa que sean la muestras de la firma del usuario durante el proceso de creación de la base de datos, para lo cual, es vital que el firmante se muestre cooperativo. Esta muestra debe representar la variación natural de la firma del usuario y éstos no deben ser aleccionados para que sus firmas sean totalmente iguales, tratando así de que las muestras sean tomadas en unas condiciones lo más cercanas posibles a una situación real.

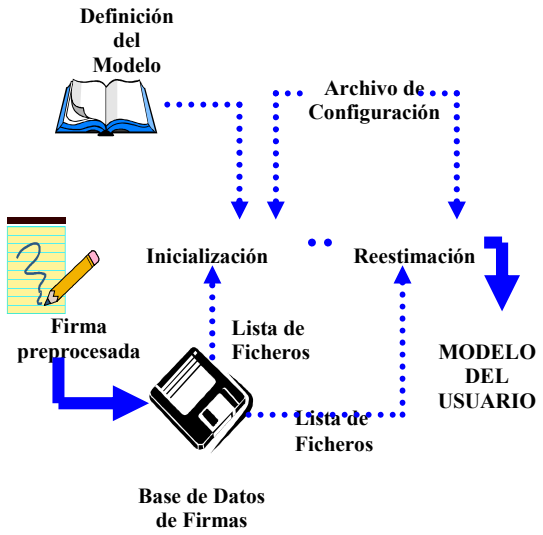


Fig 3 Obtención del modelo del usuario

Basándonos en el algoritmo descrito anteriormente se han creado dos tipos de modelos distintos: Por un lado, los miembros de Valladolid han creado modelos basados en Redes Neuronales o NN. Por otro lado, paralelamente, los miembros de Bilbao han optado por la vía de los Modelos Ocultos de Markov o HMM (Hidden Harkov Models).

Dichos modelos y sus resultados correspondientes son ampliados en el apartado 4.

3.5 Verificación y selección del umbral

La verificación consiste en calcular la similitud con el modelo entrenado y en función del valor de dicha similitud establecer el umbral que determine si un usuario es quien dice ser o se trata de un impostor.

A la hora de definir este umbral, se debe tener en cuenta el nivel de seguridad del que se desea dotar a la aplicación, esto es, si deseamos primar una FAR (*False Acceptance Rate*) o una FRR baja (*False Reject Rate*) dado que priorizar una implica deteriorar la otra y viceversa. En general, un sistema de seguridad debe garantizar una FAR próxima a cero, lo cual suele suponer pagar el tributo de una FRR alta, ya que, como se puede observar en la figura 5, ambas están ligadas.

Para comprobar la mayor o menor fiabilidad del algoritmo se estudiarán sus curvas DET (*Detection Error Tradeoff*) del algoritmo, estableciendo el punto de menor coste:

$$DCF = Cmiss * Pmiss * Ptrue + Cfa * Pfa * Pfalse$$

donde Cmiss es la constante que define cuánto ponderamos un falso rechazo, Cfa define cuánto ponderamos una falsa aceptación, Ptrue la probabilidad a priori y Pfalse, su complementario (1-Ptrue). Esta función se irá evaluando para cada uno de los puntos de la curva DET, hallándose de esta manera el punto donde el valor de la función es mínimo. Este punto define el umbral para el que el rendimiento del sistema es óptimo.

Otra referencia válida es la EER o tasa de error igualitario que corresponde al punto en que FAR y FRR coinciden.

El cálculo de las curvas ha sido realizado conforme al software de libre distribución del NIST [6] (Instituto de estándares norteamericano), al cual se han realizado algunas adaptaciones. Estos programas varían dinámicamente el umbral de aceptación y calculan la FAR y la FRR para distintas situaciones.

Cuanto más cooperativo se muestre el usuario ante el sistema conseguirá una FRR menor y también su FAR. Como resultado, su curva DET estará más cerca de los ejes de coordenadas y la EER será también menor.

4. Modelos desarrollados

4.1 Firma on-line mediante HMM

Los modelos ocultos de Markov o Hidden Markov Models – HMM [7] han demostrado su eficacia principalmente en el terreno del reconocimiento de voz, campo en el que son una técnica frecuentemente utilizada.

Los HMM modelizan un doble proceso estocástico gobernado por una cadena de Markov con un número finito de estados y un grupo de funciones asociadas a cada estado. En instantes discretos de tiempo, el proceso está en uno de los estados y genera un determinado símbolo de acuerdo con el estado en el que se encuentra. Finalmente, se obtiene un conjunto de símbolos que definen las características básicas de la firma; sin embargo, el estado que genera cada uno de los símbolos es desconocido: está "oculto".

En nuestro modelo inicial las firmas se describen por la función que viene dada por el ángulo normalizado direccional en función de la distancia a lo largo de la trayectoria de la firma.

Una parte importante en el estudio ha consistido en la definición del número de estados, el número de símbolos por estado, la matriz de probabilidades de transición entre estados, así como la probabilidad inicial de distribución de estados.

Para ello, el modelo de cada firmante se ha inicializado con sus firmas personales 00, 01, 02, 03, 04, que son las introducidas en la primera tanda, y ha sido reestimado con las cinco mismas y las firmas personales 05, 10, 15, 20, que son las primeras de las cuatro tandas posteriores. Con ello se pretende que el modelo contemple la variabilidad de la firma en el tiempo, teniendo en cuenta el proceso de captura de la base de datos.

A fin de determinar cuál de las arquitecturas de HMM presenta mejor comportamiento, se han realizado pruebas con firmas normalizadas a 300 muestras y cuantizadas a 32 símbolos. Se ha comprobado que los modelos L-R, *left-right*, de 6 estados (el inicial, cuatro intermedios y el final) se comportan mejor que otros modelos L-R con menos estados y con más estados. Los modelos con peores resultados han sido los ergódicos o generalizados, aquellos que permiten transiciones de cualquier estado a cualquier estado.

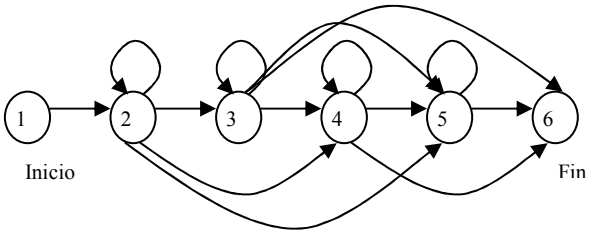


Fig 4 Arquitectura HMM L-R utilizada

Una vez determinada la arquitectura de HMM a emplear se han hecho pruebas para determinar cuál era el comportamiento frente a distintas longitudes de normalización. Se han probado normalizaciones a 100, 200, 300, 400, 500, 600 muestras y también firmas sin normalización (manteniendo su tamaño original) y se ha observado un mejor comportamiento con valores en el rango de 300 a 500. El valor 300 guarda una relación clara con la longitud media de las firmas, que suele corresponderse con una duración de escritura de aproximadamente 3 segundos.

Una vez fijado el modelo de 6 estados y la normalización a 300 muestras, el último factor a probar es el grado de cuantización. Se han hecho pruebas con 16 valores posibles, con 32, 64 y sin cuantización, llegándose a la conclusión de que este factor apenas influye en la calidad del modelo.

4.2 Firma on-line mediante NN

En el caso de las Redes Neuronales -NN- se ha optado por la arquitectura del Perceptrón Multicapa (MLP) con algoritmo de aprendizaje de retropropagación del error.

La red se usa como Red Autoasociativa, esto es, se intenta que reproduzca la entrada a la salida. En el entrenamiento la entrada y la salida deseada coinciden. Yegnanarayana y otros [8] han demostrado que una red de este tipo con una capa oculta no lineal y con un número de neuronas inferior al de las capas de entrada/salida es capaz de modelar la distribución de probabilidad de los vectores de entrenamiento.

El algoritmo de preprocesado es común al utilizado con los HMM y al igual que con los HMM se obtienen mejores resultados normalizando las firmas a un tamaño fijo, observando que la longitud que mejores resultados genera es 100.

En cuanto al vector de características usado, esto es, la entrada a la red, se compone de un número N consecutivo de ángulos. Los 100 ángulos de la firma normalizada se dividen en grupos de N, y esos son los vectores de características usados.

Se ha probado con distintos valores de N, observando que cuanto mayor es N, mejores son los resultados, pero menos vectores tenemos para entrenar la red, por lo que hay que llegar a un valor de compromiso. El valor de N utilizado es 20, lo cual supone que se extraen 5 vectores de características por firma. De ello se deduce que el tamaño de las capas de entrada y salida de la red es de 20 neuronas.

Es importante tener en cuenta que para que el algoritmo de entrenamiento funcione correctamente es conveniente normalizar los valores del vector de entrada, lo que significa que todos estén incluidos en un determinado intervalo. Todos los valores del vector de características deben estar comprendidos entre 0 y 1, para lo cual se divide cada componente por el valor máximo que pueden tener 2*PI.

La arquitectura de la red usada consta de 3 capas: una de entrada de 20 neuronas, una oculta con 9 neuronas, y una capa de salida de 20 neuronas. Las neuronas en las dos últimas capas tienen función de activación sigmoide [9].

Se han probado distintos valores de neuronas en la capa oculta, observando que el valor óptimo se obtiene con 9 neuronas. Con menos y más de 9 los resultados aunque no por mucho, son peores.

En la fase de prueba, una firma se compone de N vectores de características, y para cada uno se obtiene una salida de la red. Lo primero es estimar la probabilidad de pertenencia de cada vector al modelo representado por la red. Esto se realiza calculando el error cuadrático entre la salida de la red y la salida que se esperaba, que recordemos, es la entrada.

Tenemos entonces N estimaciones de pertenencia para cada firma, una por vector, para obtener la de la firma completa se calcula la media de los errores para cada uno de esos N vectores.

4.3 Resultados obtenidos

Se ha realizado un estudio del rendimiento de los dos métodos aplicado al campo de la verificación de la identidad, esto es, en la comparación entre la firma del individuo que trata de ganar el acceso al sistema y el modelo de la identidad que el usuario ha proporcionado previamente.

Estas pruebas han sido realizadas sobre los subcorpora de Valladolid y Bilbao, con un total de 150 modelos de cada tipo de prueba.

	FAR	FRR	EER	$Cdet_1$
HMM	0,295%	48,21%	17,565%	0,051
NN	0,32%	51,28%	19,47%	0,054

Tabla 1 Resultados iniciales

En la tabla 1 se muestran los resultados obtenidos con HMM y NN para los 150 firmantes, según el preprocesado similar al propuesto por Yang y otros, que sólo incluye ángulos. Como se aprecia, si bien la FAR es próxima a 0, la tasa de falso rechazo ronda el 50%, lo cual supone que un usuario debería firmar de promedio 2 veces para que se le admita en un sistema.

Para definir el punto óptimo de funcionamiento, hallamos el punto de menor coste según las funciones del NIST, con los siguientes valores Cfa = 1 Cmiss = 10 y Ptarget = 0.01. Esto supone priorizar una FAR muy baja con una ponderación de 10 a 1, aún a costa de una FRR elevada.

En la figura 5 se representa la curva DET (Detection Error Tradeoff) del usuario número 8 de Bilbao con las FAR y FRR como ejes de coordenadas X e Y respectivamente.

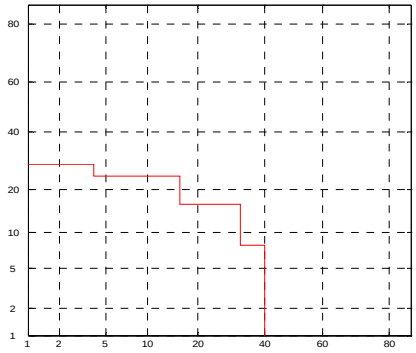


Fig 5 Curva DET del usuario 8

Si bien las FAR eran aceptables las FRR eran exageradas, esto tal vez se debiera a la calidad de las imitaciones entrenadas de la base de datos. Se pensó pues, en hacer el mismo experimento, pero con imitaciones aleatorias: a cada individuo se le intentarían colar las firmas originales del resto de los componentes de la base de datos. Los resultados con los 75 firmantes de Bilbao fueron los siguientes:

HMM 75 usuarios	ERR Random forgery	ERR Skilled Forgery	Diferencia
Media	14,345%	18,703%	4,358%

Tabla 2 Falsificación aleatoria/entrenada inicial

El rendimiento mejora algo, pero no todo lo que cabría esperar. Esto se debe a que al utilizar las falsificaciones aleatorias, mejora la tasa de falsa aceptación pero no la de falso rechazo y por ello, aunque mejoremos sustancialmente la FAR como la FRR no varía, la mejora relativa de la EER no será significativa.

En siguientes fases de pruebas se pensó en eliminar la resta del ángulo inicial a todos los ángulos propuesta por Yang y otros e implementada en el preprocesado, ya que debido a la forma en la que fue adquirida la base de datos (los usuarios firmaban dentro de una rejilla orientada) cabía pensar que dicha corrección no era necesaria.

Se obtuvieron resultados en los cuales los ERR se reducían a la mitad, de lo cual se deduce que esa rotación introduce una información ruidosa que no beneficia a la verificación, por lo que ha sido eliminada de la implementación definitiva del algoritmo. En lugar de eliminar este ángulo, una alternativa a evaluar consistiría en remplazar el valor del ángulo del primer segmento por el ángulo del eje principal de inercia de la firma, ya que éste presenta un comportamiento más estable.

Finalmente, para poder determinar cómo influyen las otras tres coordenadas proporcionadas por la tableta, se entrenaron los sistemas incluyendo presión, acimut e inclinación, se crearon otros 150 nuevos modelos y que fueron probados de nuevo.

	FAR	FRR	EER	$Cdet_1$
HMM	0,00%	31,52%	9,253%	0,032
NN	0,16%	36,24%	9,9%	0,085

Tabla 3 Resultados con el nuevo preprocesado incluyendo ángulos, presión, acimut e inclinación.

Se observa que la introducción de estos parámetros redunda en una notable mejoría de los resultados. Queda claro pues, que conforme se van incluyendo más parámetros en el entrenamiento del sistema mayor es la capacidad de verificación del mismo.

5. Conclusiones y trabajo futuro

Los resultados obtenidos con idénticos parámetros por los Modelos Ocultos de Markov y las Redes Neuronales guardan bastante similitud, si bien en la presente implementación son ligeramente mejores los obtenidos con HMM. En la actualidad se está trabajando con redes recurrentes y los primeros datos obtenidos crean nuevas expectativas.

La introducción de información adicional, con datos como velocidad, aceleración, centro de masa, ejes de inercia, longitud de segmentos lineales y circulares [10], radios de curvatura, etc. supondría una mejora

de los resultados, hasta el punto de conseguir un producto utilizable en soluciones comerciales. Futuros proyectos que en la actualidad se desarrollan en el departamento, abordarán este tema.

La *fusión multimodal* [11] de distintos métodos biométricos (huellas, voz, firmas...) permite mejorar aun más la verificación del individuo. En el mismo sentido, cabría hablar de *fusión intramodal* aplicando la combinación de varios métodos de verificación basados todos en el mismo rasgo biométrico. En el presente trabajo se presentan los resultados obtenidos por dos métodos independientes, HMMs y Redes Neuronales, sobre firma on-line. Los resultados de la fusión intramodal de ambos métodos supondrá una mejora sobre los resultados de cada método en particular. Paralelamente otros equipos del proyecto trabajan en otros rasgos biométricos como voz y huellas, con lo cual pronto estaremos en disposición de fusionar los diferentes sistemas en un sistema multimodal.

Como se ha mencionado al comienzo, por regla general, la información dinámica de la firma del usuario suele ser muy característica. Esto redunda en una mayor precisión de los métodos de verificación que utilizan la técnica on-line. Sin embargo, en usuarios con una firma poco estable también puede provocar un aumento de los falsos rechazos. En la mayoría de los casos una captura on-line de los datos implica una mayor FRR (si bien una FAR mucho menor) y una captura off-line una FAR mayor, lo cual se puede considerar más grave dependiendo de la seguridad exigible a la aplicación. La utilización combinada de ambos métodos puede generar resultados con un alto grado de fiabilidad.

Para finalizar, cabe destacar que este proyecto en Biometría supone el comienzo de una nueva línea de trabajo en las actividades del grupo de Seguridad en Sistemas Distribuidos del Área de Ingeniería Telemática de la Escuela de Ingenieros.

Agradecimientos

Este trabajo ha sido parcialmente financiado por la Comisión Interministerial de Ciencia y Tecnología, CICYT, bajo el proyecto TIC2000-1669-C04-03.

Referencias

[1] Sitio WEB del Proyecto TIC2000-1669-C04-03 [Online] http://www.infor.uva.es/biometria

[2] J. Ortega, D. Simón, M. Faúndez, V. Espinosa, I. Hernáez, J.J. Igarza, C. Vivaracho, Q.I. Moro. "MCYT: A Multimodal Biometric Database" COST275-The Advent of Biometrics on the Internet. (2002). pp. 123-126

[3] Wacom Technology Co. [Online] Available http://www.wacom.com.

[4] C.E. Vivaracho, Q.I. Moro, D. Escudero, I. Hernáez, J.J. Igarza. "Creación de una Base de Datos para Reconocimiento de Personas Mediante Multimodalidad Biométrica". II Taller Iberoamericano de Informática Industrial. (2002) pp. 105-111

[5] L.Yang, B.K.Widjaja, R. Prasad. "Application of Hidden Markov Models for Signature Verification", (1995) Pattern Recognition, 28(2) pp. 161-170

[6] National Institute of Standards and Technology. [Online] Available http://www.nist.gov

[7] L.R. Rabiner, B.H. Juang "An Introduction to Hidden Markov Models", IEEE ASSP Magazine, January 1986

[8] B. Yegnanarayana y S. P. Kishore, "AANN: An Alternative to GMM for Pattern Recognition". Neural Network, vol. 15, no. 3, pp. 459-469, Abril 2002

[9] B. Yegnanarayana, S. P. Kishore y A. V. Anjani, "Neural Networks Models for Capturing Probability Distribution of Training Data", Proc. 4th International Conference of Congnitive and Neural Systems, Mayo 2000.

[10] Iñaki Goiricelaya, Juan José Igarza, Juan José Uncilla, Federico Pérez, Jesús Romo, Koldo Espinosa "Modelización de Contornos Mediante la Búsqueda de Segmentos Lineales y Circulares en Imágenes de Nivel de Gris" (1997) URSI Vol. I pp. 203-206

[11] J Kittler, K Messer and J Czyz. "Fusion of Intramodal and Multimodal Experts in Personal Identity Authentication Systems" COST275-The Advent of Biometrics on the Internet. (2002). pp. 17-24

Sistema Avanzado de Gestión de Billetes para Transporte Público con Tarjetas Inteligentes

Luis R. López, Jesús Martínez, Pedro Merino
Dpto. de Lenguajes y Ciencias de la Computación. Universidad de Málaga.
Campus de Teatinos s/n. 29001. Málaga. España.
E-mail: lramonl@terra.es, jmcruz@lcc.uma.es, pedro@lcc.uma.es

Abstract. *Smartcards-based applications have opened a new range of possibilities in e-commerce. This paper presents a new solution for the integration of smartcards in a public transport ticketing service. We propose an architecture based on "digital tickets", which are easily managed by the electronic equipment involved. These tickets, stored on smartcards, offer some new exciting possibilities as transferability, tampering detection or remote selling, which can add some value to customers and service providers. Everything is done under a high security environment. To achieve all these requirements, authentication and public key ciphering are employed.*

1 Introducción

Las empresas de transporte de viajeros han experimentado una gran transformación en los últimos 20 años, no sólo por la mejora tecnológica de los vehículos, sino también por la incorporación de redes de comunicaciones en su infraestructura interna. Actualmente, estas compañías suelen realizan la venta de billetes mediante ventanillas, desde las que se accede de forma directa a una base de datos centralizada. Esta infraestructura implica la necesidad de la presencia física del cliente en la ventanilla para adquirir el billete. El problema se agrava con el hecho de que en los minutos anteriores a la salida de un vehículo suelen producirse largas colas de clientes, reduciendo aún más las posibilidades de poder adquirir una plaza libre antes de la hora de salida

La tendencia es dotar a la infraestructura con la posibilidad de comprar billetes sin que sea imprescindible la presencia del cliente. En este escenario se identifican fundamentalmente tres problemas:

(1) Evitar la entrega material del billete, que no es posible al encontrarse el emisor y el cliente en localizaciones distintas.

(2) Garantizar al emisor del billete, que ingresará el coste del mismo. Es necesario por tanto un mecanismo de autenticación del cliente, es decir, un mecanismo para comprobar que el cliente sea quien dice ser, además de habilitar una forma de pago que no sea "en metálico".

(3) Finalmente, y una vez en el vehículo, hay que comprobar que el billete es auténtico, teniendo en cuenta que éste no ha sido entregado al cliente físicamente por la empresa de transportes. Dicha comprobación debe ser rápida y eficaz para no interrumpir el flujo habitual desubida de pasajeros en el vehículo mientras se evita el fraude.

La entrega remota del billete podría resolverse usando servicios de fax, pero no suelen ser habituales en las casas de los particulares, aparte de que implicaría incluir medidas adicionales para evitar modificaciones no autorizadas del billete enviado a distancia. La autenticación sin ambigüedades presenta más dificultades aún, pues no se puede comprobar sin presencia física el documento de identidad del cliente.

De la misma forma, el pago con tarjeta de crédito, único método ampliamente extendido de pago a distancia para los usuarios, implica riesgos para ambas partes: el cliente puede pretender utilizar una tarjeta de crédito de la que no tiene autorización, lo que haría necesario por parte de la empresa prestadora del servicio el tener que validar en línea los datos de la tarjeta, o bien la empresa podría intentar cargar al cliente una cantidad de dinero distinta a la acordada, lo que se solucionaría con la emisión de una factura en el momento de la compra. En cualquier caso, el coste adicional de esta forma de pago podría ser excesivo para este tipo de servicio.

Por último, la comprobación a bordo del vehículo tendría que considerar la posibilidad de que el cliente hubiera realizado copias ilegales del billete, modificadas o no, para requerir la prestación de servicios para los que no ha pagado; más aún, esa comprobación ha de ser extremadamente rápida.

El billete electrónico almacenado en una tarjeta inteligente es una solución óptima para resovler todos los problemas comentados anteriormente de una forma eficaz y fácilmente implementable con la

tecnología actual[1]. Sin embargo, la mayoría de las aplicaciones actuales para transporte público sólo emplean tarjetas para albergar simples contadores de viajes, y no explotan otras posibilidades.

En este artículo se presenta una solución factible desde el punto de vista técnico para una manipulación más flexible de los billetes electrónicos en transporte público. Para ello, se ha diseñado e implementado una arquitectura software de servicio que contempla el uso de tarjetas inteligentes con capacidad de proceso para el almacenamiento y manipulación de los billetes electrónicos. Las principales contribuciones del trabajo, desde el punto de vista de una aplicación de comercio electrónico, se refieren a la infraestructura de servicio, el formato del billete, el protocolo para su manipulación y el uso combinado de tarjetas con y sin contacto.

La organización del artículo es la siguiente. En la Sección 2 se hace una introducción a las características de los tickets electrónicos y las tarjetas inteligentes. La Sección 3 describe los servicios contemplados y el papel de cada uno de los agentes en la aplicación. En la Sección 4 se describe el uso de la tarjeta inteligente. La Sección 5 presenta los detalles sobre la seguridad del sistema. La Sección 6 presenta los aspectos más relevantes de la implementación actual. En la Sección 7 se enumeran algunas conclusiones y futuras líneas de trabajo.

2 Preliminares

2.1 El billete electrónico

En todo billete, independientemente de su naturaleza, se distinguen tres partes: el emisor, la promesa y el propietario. Un billete no es más que un certificado que garantiza al propietario el derecho a demandar al emisor las prestaciones o servicios (promesa) escritos en él.

El "billete electrónico", "billete digital" o "digital ticket" realiza la misma función que el billete tradicional de papel, aunque añade a éste numerosas posibilidades como privacidad de su contenido, facilidades para la emisión, almacenamiento y entrega remota o la firma por el emisor. Además, los billetes electrónicos poseen una gran facilidad para la parametrización de sus propiedades, de tal forma que se simplifica enormemente la gestión de los billetes en una infraestructura de transporte público. Esto da pie a nuevos servicios, tales como la delegación de los derechos del billete entre dos clientes o la inclusión de un registro de trayectos ya realizados. Más aún, al estar codificados en forma digital pueden ser transferidos de forma electrónica, no siendo necesaria la presencia física del proveedor del servicio para la emisión del billete.

Al no tratarse de billetes de naturaleza física, como los tradicionales, es necesario encontrar un contenedor adecuado para su almacenamiento por parte del cliente, y es aquí donde tienen un campo natural de aplicación las tarjetas inteligentes.

2.2 Las tarjetas inteligentes

El uso de tarjetas inteligentes[2,3,4] como depósito ideal de los billetes electrónicos está motivado por las características especiales de estos últimos: los billetes digitales están compuestos de datos sensibles, pueden contener gran cantidad de información y deben estar protegidos. Las tarjetas inteligentes ofrecen la flexibilidad necesaria para almacenar este tipo de información; además, son altamente fiables, tanto en términos de protección de miradas no autorizadas como de modificaciones indeseables. Esto es esencial en aplicaciones de comercio electrónico. Existen otras alternativas que, si bien han sido consideradas, no cumplen con todos los requisitos necesarios para el almacenamiento de los billetes electrónicos, como son las *tarjetas de banda magnética* o las *tarjetas con código de barras* .

Las tarjetas inteligentes, también llamadas *SmartCards*, poen un "chip" con gran capacidad de memoria (típicamente varios kilobytes), y son capaces incluso de ejecutar programas de forma similar a un ordenador, para lo que se emplean lenguajes de programación como Basic o Java. Actualmente, los esfuerzos de las compañías punteras se concentran en dotar a estas tarjetas de más capacidad de almacenamiento y de procesamiento[2]. Gracias a ello, se dipone de sistemas criptográficos implementados en las tarjetas, tanto en la vertiente simétrica como en la asimétrica, siendo a veces incluido un coprocesador criptográfico para acelerar los cálculos[5]. La información sensible puede estar además protegida por un código conocido por el usuario, generalmente denominado PIN (Personal Identification Number).

En función de la "inteligencia" del circuito, podemos hacer una primera clasificación entre tarjetas de memoria y tarjetas microprocesadas. Las primeras son las usadas en aplicaciones tipo tarjeta de cabina telefónica. El segundo tipo de tarjetas, las dotadas de procesador, son mucho más potentes, pues permiten realizar operaciones complejas, no son simples almacenes de información. Cuando en las siguientes secciones aparezca el término tarjeta inteligente, siempre se referirá a este último tipo de tarjetas.

Hay variedades muy interesantes en la forma que tiene el circuito de comunicarse con el exterior: mientras que algunas tarjetas necesitan ser insertadas físicamente en un lector, otras se comunican a distancia vía radiofrecuencia. Las tarjetas que no necesitan insertarse físicamente en

ningún lector para interactuar son las denominadas tarjetas "sin contacto" (contactless cards), y reciben la alimentación directamente del campo electromagnético generado por el lector, lo que las hace especialmente aptas para servicios donde prime la comodidad del cliente. Su mayor inconveniente es el alto precio de los lectores. Como derivación de las tarjetas sin contacto nacieron las tarjetas combo, que son aquellas que presentan dos interfaces: el clásico con contacto y otro sin contacto.

3 El sistema de gestión de billetes

En esta Sección se describe la infraestructura para la gestión avanzada de billetes cuyo esquema se muestra en la Figura 1. Primero se enumeran los servicios que proporcina y a continuación se detalla el papel de cada agente de la aplicación.

3.1 Servicios de la Infraestructura

Los servicios que ofrece esta aplicación pueden resumirse en emisión y comprobación de billetes, y consultas de servicios realizados. La gestión de pagos se contempla fuera del sistema, al realizarse a través de entidades bancarias.

Emisión de billetes

Como puede verse en la Figura 1, existen tres puntos donde se pueden emitir billetes: en una ventanilla, en el sistema de venta automático (SVA) o en un ordenador remoto conectado a Internet que disponga de un lector de tarjetas.

El caso más favorable de cara a la seguridad es la ventanilla, pues se supone que los terminales de venta están conectados a la base de datos central mediante red privada o, en su defecto, a través de una red privada virtual (VPN) vía Internet.

Con el sistema de venta automático ocurre algo similar: la comunicación con la central se realiza mediante redes privadas, o, si no es el caso, cifrando la comunicación. Sin embargo, hay que procurar que no exista información sensible dentro del terminal para evitar que ésta pueda ser extraída por la fuerza (por ejemplo, mediante robos).

Por último, la emisión remota de billetes requiere más medidas de seguridad. El programa debería actuar como simple interfaz entre la aplicación servidora y la tarjeta para evitar modificaciones del comportamiento por parte de un usuario malicioso. Esto no impide que la aplicación pueda almacenar información pública en el disco duro del usuario para reducir el tiempo de las transacciones.

En este diagrama se supone que la red privada de la empresa es segura. Si algún segmento de la comunicación ha de realizarse a través de una red pública, sería más que recomendable utilizar conexiones VPN para asegurar la privacidad de las transferencias.

También ha de destacarse que en el caso de que el cliente opte por comprar el billete de forma remota, en ningún momento se le deberá permitir el acceso directo a la red de la empresa para evitar posible problemas de seguridad. Como se puede apreciar, estas normas cautelares se aplicarán con rigor en el resto del diseño.

El formato del billete emitido por estos medios se muestra en la Figura 2. La mayor parte de sus elementos son auto-explicativos, otros se discuten más adelante.

Comprobación del billete en el vehículo

La comprobación del billete se realiza fuera de línea, esto es, sin una conexión en tiempo real con la base de datos de la empresa. Por ese motivo se incrementan las medidas para evitar falsificaciones.

Cuando el cliente presenta su tarjeta a bordo del vehículo, un programa comprobará si la suscripción al proveedor es correcta y si tarjeta es auténtica. Sólo en el caso de que el cliente sea reconocido, se analizan los billetes. Cada billete aceptado será almacenado en la base de datos del vehículo para su descarga sobre la base de datos central cada cierto período de tiempo (cada 48 horas). Cuando se produzca la descarga de información en la base de datos central podrá comprobarse si todas las suscripciones aceptadas en el vehículo eran auténticas y si los billetes fueron efectivamente emitidos. En caso de la comprobación arrojara un resultado negativo habría que activar las alarmas de seguridad.

Consulta de servicios

La consulta de todos servicios realizados puede efectuarse únicamente utilizando dos agentes: el sistema de venta automático y un ordenador remoto conectado a Internet. En este último caso, es posible incluso consultar información acerca de los últimos servicios sin contar con conexión a Internet, pues la tarjeta del cliente guarda un registro de éstos.

3.2 Agentes de la infraestructura

El proveedor o emisor de tarjetas

Es el encargado de personalizar las tarjetas para su uso y de emitir los billetes para éstas. La personalización implica crear un sistema de ficheros en la tarjeta con una estructura determinada y generar las claves iniciales, además de registrar la nueva tarjeta en la base de datos correspondiente. Por otra parte, la emisión de billetes necesita la generación del par de claves para cada instancia de servicio que pueda realizarse.

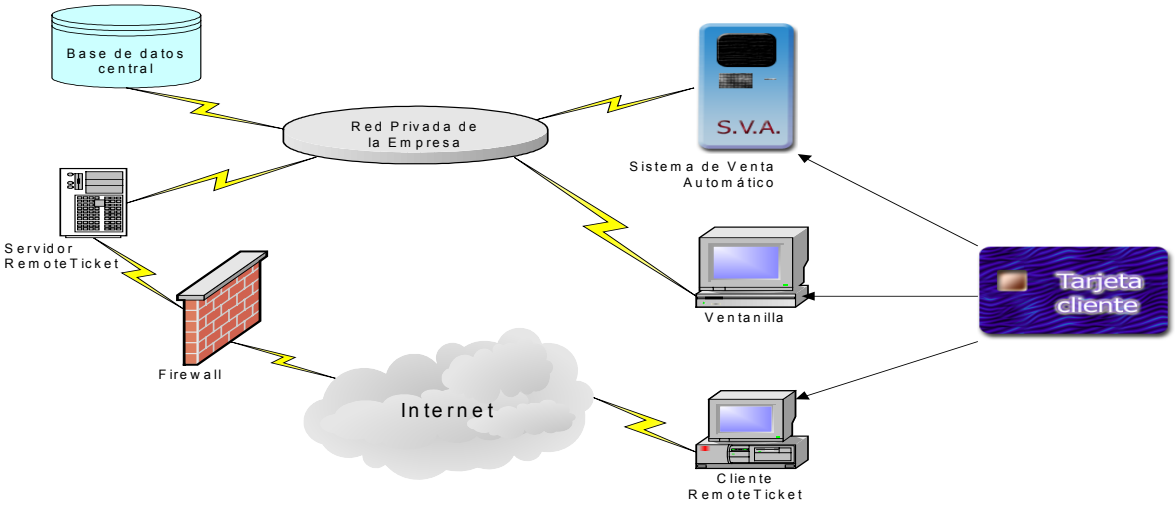


Figura 1: Infraestructura de emisión de billetes

Este agente de la infraestructura, localizado en las oficinas de la empresa de transportes, trabaja en un entorno seguro, entendiéndose como tal aquel en el que no hay acceso público. Aún así, debe controlarse la seguridad de las claves, en especial las de proveedor y de instancia de servicio, pues su conocimiento por agentes maliciosos podría provocar la falsificación de identidades y/o billetes.

El sistema de venta automático (SVA)

Es un terminal con pantalla táctil y lector de tarjetas. Una vez instalados en lugares públicos y conectados a la base de datos central permiten realizar operaciones con la tarjeta del cliente tales como cambio de PIN, consulta, compra o anulación de billetes, etc.

La presencia de estos terminales tiene dos claros objetivos: extender el periodo de atención al cliente al 100% del tiempo de apertura de la estación y descargar la cola de la ventanilla.

La ventanilla

Responde al rol clásico de la ventanilla tradicional: desde ella se pueden comprar billetes o anularlos. Incluso es posible que en algunos casos se le permita al cliente obtener por escrito un registro de los últimos servicios recibidos. Todo ello está asistido por un operador humano.

El vehículo

Lugar donde se presta el servicio. Es el encargado de verificar que el cliente presenta un billete electrónico válido contenido en una tarjeta válida. Dicha verificación se realiza con la ayuda de otra tarjeta inteligente, transportada por el conductor, y que contiene la información necesaria para validar los billetes.

Acceso remoto desde Internet

Tiene como objetivo el facilitar al usuario los servicios del Sistema de Venta Automático en cualquier ordenador dotado de un lector de tarjetas inteligentes y una conexión a Internet. En este agente la seguridad debe ser crítica para evitar que se produzca fraude o intercepción de información sensible por parte de terceros.

Por todo ello, la comunicación con el servidor de la empresa deberá estar cifrada y, si es posible, autenticada. También es conveniente que los ordenadores de la empresa ejecutando el programa servidor estén protegidos de ataques provenientes del exterior mediante firewalls y similares.

La base de datos

Aquí se almacenan todas las claves, información de las tarjetas y los servicios realizados para su posterior cobro. La base de datos puede tener gran volumen de transacciones, así que hay que buscar soluciones sólidas y suficientemente flexibles como para aguantar la carga de trabajo. Se supone que el acceso a la base de datos sólo se puede realizar a través de la red privada de la empresa, con lo que muchas restricciones de seguridad podrán ser relajadas, pero no han de desaparecer. Es un requisito que la base de datos soporte múltiples usuarios con distintos permisos.

4 La tarjeta del cliente

Idealmente, la tarjeta cliente debería proporcionar la máxima comodidad al cliente sin comprometer la seguridad. La facilidad de uso y la fluidez en el abordaje del vehículo pueden solucionarse utilizando tarjetas sin contacto. Sin embargo, los lectores de este tipo de tarjetas son bastantes más caros que sus equivalentes de contacto. Si bien esto no debería suponer ningún problema de cara a la empresa, al ser el número de vehículos es

Versión	Proveedor	CSN	ID de usuario	Número de billete	Fecha y hora emisión	Fecha y hora servicio	Línea	Est. origen	Est. destino	Plaza	Precio	Tarifa	FIRMA
8bits (0x01)	16bits	64bits	24bits	64bits	32bits	32bits	16bits	32bits	32bits	16bits	32bits	16bits	1024bits

Figura 2: Formato del Billete

relativamente pequeño, puede ser un factor crítico si se quiere ofrecer un servicio de adquisición remota de billetes: por ello, la solución ideal implica tarjetas cliente con doble interfaz de acceso (tarjetas combo).

Para que la solución sea completamente eficaz, ambas interfaces deben ser capaces de compartir información. De esta forma, los billetes comprados pueden ser almacenados mediante un lector de contacto y comprobados en el vehículo mediante la interfaz de acceso sin contacto.

Como efecto lateral, destacar que, por normal general, las interfaces de acceso de contacto suelen presentar mayores opciones de seguridad, con lo cual se cumplen simultáneamente las necesidades anteriormente mencionadas de máxima comodidad con máxima seguridad.

Por tanto, en el diseño del sistema se considera una tarjeta con dos interfaces y una zona de memoria compartida, que están disponibles en el mercado desde hace ya algunos años.

4.1 La tarjeta GemCombi/MPCOS

En el desarrollo de la aplicación se ha escogido el modelo GemCombi/MPCOS del fabricante francés Gemplus, pues reúne todos los requisitos anteriores. Esta tarjeta pertenece a la familia "combo", y por tanto presenta doble interfaz de acceso. La interfaz de contacto es compatible con cualquier lector ISO 7816, mientras que la interfaz sin contacto es compatible MIFARE (ISO 14443A). Sin embargo, no se trata simplemente de la unión de dos tarjetas inteligentes en un mismo soporte físico, como es el caso de la tarjeta GemTwin del mismo fabricante: en la GemCombi ambas interfaces no son independientes, ya que pueden compartir información; esto les aporta gran flexibilidad. La parte con contacto es compatible prácticamente al 100% con otra tarjeta del mismo fabricante: la MPCOS/EMV; la parte sin contacto, es una versión avanzada de la GemEasy 8000, todavía en producción. La tarjeta GemCombi/MPCOS presenta un mapa de memoria dividido en dos partes: el COMA y el SMA. El COMA, o "Área de Memoria de Sólo Contacto" (*Contact-Only-Memory-Area*), tiene una capacidad aproximada de 1 kilobyte y un sistema de ficheros tipo ISO 7816. Por otra parte, el SMA o "Área de Memoria Compartida" (*Shared-Memory-Area*) soporta 4 kilobytes distribuidos en forma de sectores y bloques, siguiendo el estándar MIFARE. La memoria del SMA se reparte en 32 sectores de 4 bloques y 8 sectores de 16 bloques. Cada bloque almacena 16 bytes de información. Mientras que el COMA sólo puede ser accedido a través de la parte con contacto de la tarjeta, el SMA presenta zonas que pueden ser configuradas para permitir cualquier combinación de acceso.

4.2 Mapa de memoria

Desde la interfaz de contacto se pueden acceder a varios ficheros que contienen información acerca del PIN/PUK de la tarjeta, así como los distintos ficheros SEBT que sirven de puente entre la memoria de contacto y la sin contacto. Por otra parte, la interfaz sin contacto tiene su información dividida en sectores.

Ya que la sección de memoria que comparten ambas interfaces se corresponde de forma directa con 8 sectores del SMA (los numerados de 32 al 39), la información a almacenar en la tarjeta se estructura en forma de sectores (ver Figura 3). Los dos primeros sectores incluyen los datos identificativos del cliente e información sobre la suscripción. Esta última información va acompañada de una firma criptográfica que le da

Sector	Block	Description		Contactless access	Contact access
Sector 32	0	Suscription Info 0		*Read*: Public (A0A1A2A3A4A5) *Write*: Forbidden	*Read*: PIN *Write*: PIN + Admin key
	1				
	2				
	3				
	4				
	5				
	6				
	7				
	8				
	9	User Info			
	10				
	11				
	12	Ticket Info 0	Ticket Info 1		
	13	Ticket Info 2	Ticket Info 3		
	14	Ticket Info 4	Ticket Info 5		
	15	SECURITY(ACCESS CONDITIONS)			
Sector 33	0	Suscription Info 1		*Read*: Public (A0A1A2A3A4A5) *Write*: Forbidden	*Read*: PIN *Write*: PIN + Admin key
	1				
	2				
	3				
	4				
	5				
	6				
	7				
	8				
	9	(RFU)			
	10				
	11				
	12				
	13				
	14				
	15	SECURITY(ACCESS CONDITIONS)			
Sectors 34/35/36/37/38/39	0	Ticket 0/1/2/3/4/5		*Read*: Restricted (Issuer Key) *Write*: Forbidden	*Read*: PIN *Write*: PIN + Admin key
	1				
	2				
	3				
	4				
	5				
	6				
	7				
	8				
	9				
	10				
	11	Service Log 0/2/4/6/8/10		*Read*: Forbidden *Write*: Restricted (Issuer Key)	
	12				
	13	Service Log 1/3/5/7/9/11			
	14				
	15	SECURITY(ACCESS CONDITIONS)			

Figura 3: Mapa del SMA en la tarjeta cliente

validez para evitar el fraude. Adicionalmente, se incluye un mapa de billetes que servirá como tabla de búsqueda (lookup table) a la hora de comprobar los billetes desde el vehículo.

Los seis sectores restantes almacenan los billetes electrónicos (descritos en la Figura 2), a razón de uno por sector. Su contenido va firmado con una clave que es diferente para cada realización del viaje, reduciendo el daño potencial provocado por el hecho de que comprometan estas claves.

5 Consideraciones sobre seguridad

Ya se ha comentado con anterioridad que uno de los objetivos claves de la infraestructura es evitar el fraude. Este requisito de seguridad implica varios aspectos a tener en cuenta. Algunos de los más importantes se detallarán en las siguientes subsecciones.

5.1 Esquema de cifrado

En esta infraestructura se usan pares de claves asimétricaspara firmar y autenticar, ya que permiten a cualquier agente exterior comprobar la integridad de los distintos elementos sin proporcionarle la información necesaria para falsificar información [6,7]. Existen dos tipos de claves privadas críticas: la de proveedor y la de instancia de servicio.

La clave de proveedor es la que permite a los vehículos autenticar las tarjetas y comprobar de forma "off-line" la suscripción del cliente al servicio, por lo que si llegara a hacerse pública sería posible falsificar identidades, aunque no billetes. Además, hay que tener en cuenta la imposibilidad de modificación de la información de subscripciones sin tener la contraseña adecuada. Más aún: una tarjeta con la identidad falsificada no podría realizar ninguna operación a través de ventanilla, SVA o Internet debido a los mecanismos de autenticación avanzados que se realizan por la interfaz de contacto (éstos incluyen la utilización de una clave secreta simétrica sólo conocida por la base de datos central y la tarjeta original). Eso excluye la posibilidad de que una tarjeta falsificada pueda realizar compras.

Aún así, debe plantearse un procedimiento a ejecutar en el caso de que la clave privada de proveedor sea comprometida y sea necesaria su sustitución por otra nueva. Con este objetivo se ha decidido incluir en la tarjeta espacio para otro proveedor adicional, de forma que se puedan incluir dos certificados de suscripción: si la clave original se compromete, los nuevos billetes (que irán firmados con la correspondiente nueva clave) podrán utilizar otro código de proveedor. El hacerlo así permite que los billetes antiguos no se vean afectados, pues la suscripción antigua permanecerá en la tarjeta al menos durante el margen de tiempo en el que los billetes tengan validez. La adición o reemplazo de proveedores se realizará de forma automática en cualquiera de los tres puntos habituales de gestión: la ventanilla, el SVA o el programa de acceso a Internet.

Si además de la clave privada de proveedor se compromete la de instancia de servicio, un cliente con una tarjeta auténtica (y con los medios adecuados para modificar la información de la tarjeta) podría falsificar un billete para la instancia de servicio en cuestión. Como es obvio, el vehículo no detectará nada extraño en el billete, pues la firma será completamente correcta, y permitirá al usuario recibir el servicio.

En este caso, el falsificador se arriesga a que el terminal del vehículo detecte una colisión al tener dos clientes asignadas la misma plaza, es decir, el fraude se detectaría antes de realizar el servicio. Si no se produjera la colisión, sería igualmente detectado al realizar el volcado periódico del registro del vehículo en el ordenador central.

Considerando la posibilidad de que este diseño pueda ser implementado en tarjetas con capacidades criptográficas asimétricas, se recomienda utilizar tamaños estándar de clave (p.ej. 512, 1024, 2048). Una longitud de 512 bits puede considerarse débil hoy en día, por lo que es aconsejable descartarla. La seguridad ofrecida por una clave de 1024 bits es más que suficiente para la seguridad requerida, al menos durante los próximos años. Es por ello el valor recomendado. Una clave de 2048 bits provocaría problemas con aquellas tarjetas que tienen poca capacidad de almacenamiento, pues una única firma ocuparía 256 bytes; por tanto sólo es recomendada en aquellos casos concretos donde la tarjeta no presente problemas con un gran tamaño de datos.

Dado que la clave privada de instancia de servicio sólo tiene validez hasta la ejecución del servicio, que existe una posibilidad muy pequeña de que realizar un ataque con éxito a las dos claves de 1024 bits y que conlleva un gran riesgo para el estafador que intenta la falsificación, no se considera conveniente establecer un sistema de revocación de claves que complicaría en exceso la infraestructura.

Por último, hay que considerar la posibilidad de que se descubriera la clave privada que genera la firma de los billetes. Para dificultar que esto suceda, se creará un par de claves para cada instancia de un servicio. Así se acotará el tiempo que tiene un asaltante para descubrirla y poder emitir billetes de forma fraudulenta.

5.2 Duplicado de tarjetas

En el caso de que se lograra clonar una tarjeta cliente, el sistema sería incapaz, al menos inicialmente, de distinguir entre la auténtica y la copia. Es por ello necesario el establecimiento de mecanismos que dificulten la realización de un duplicado y haga más difícil cometer fraude.

Con este objetivo en mente, se ha optado por utilizar el número de serie de la tarjeta (CSN) que, según el fabricante, es único entre todas las tarjetas. Además, si es posible, la aplicación debería comprobar el código de emisor de la tarjeta (Issuer Number) para comprobar si coincide con el de la propia empresa.

Por tanto, para realizar un clonado habrían de copiarse ambos números, que sólo están disponibles con el acceso físico a la tarjeta original. A pesar de que la copia se lograra (caso poco probable, pues habría que modificar la ROM de la tarjeta), este tipo de fraude podría ser detectado por el cliente original al comprobar su registro de servicios en línea.

5.3 Cambio de la promesa del billete

Los cambios en la promesa de un billete pueden ser muy apetecibles para un usuario malintencionado, ya que le permitiría modificar a su voluntad los parámetros de un billete auténtico. La promesa del billete no está cifrada con el objetivo de permitir al usuario comprobar su contenido, así que la protección de la información ha de realizarse utilizando algún tipo de firma criptográfica: un cambio en cualquier campo del billete provocará un error en la comprobación que impedirá aceptar el billete como válido.

Además, el mecanismo de la firma, si está convenientemente implementado mediante criptografía asimétrica, podría permitir a cualquier usuario o aplicación comprobar la validez del billete accediendo a información públicamente disponible, sin tener posibilidad alguna de alteración.

5.4 Emisión fraudulenta de billetes

Otra posibilidad a contemplar sería el intento por parte de un usuario de añadir un billete de forma fraudulenta a una tarjeta auténtica, bien creando el billete desde cero o bien copiándolo desde un billete válido de otra tarjeta. Aquí se presentan dos dificultades: las barreras que se encontrará para modificar la información almacenada en la tarjeta y qué efectos tendrían dichas modificaciones en caso de llevarse a la práctica.

El primer problema debería poder solucionarse estableciendo unas condiciones de acceso muy restrictivas a los ficheros que contienen la información de los billetes. Estos permisos deberían establecerse de forma que sólo el emisor de la tarjeta posea las claves necesarias para actualizar los billetes. Se recomienda anular el acceso de escritura en la interfaz sin contacto de la tarjeta.

Para el segundo problema pueden considerarse muchas situaciones. Por ejemplo, la copia exacta de la información se evita asociando el billete a la tarjeta a través del número de serie. Lógicamente, hay que impedir que esa asociación se pueda modificar. Es por ello que se ha optado por incluir el número de serie en el cuerpo del billete: el CSN quedará protegido con la firma de validez.

Otro problema puede ser el intento de reutilización de un billete válido ya utilizado en algún otro servicio. La solución es de corte similar a la anterior: especificar en el billete la fecha y la hora de realización del servicio. Si se intentara modificar, la firma impediría la validación del billete.

5.5 La venta en el vehículo

La capacidad de vender un billete a bordo del vehículo implica ciertas consideraciones a lo que seguridad se refiere: Recordemos que para la creación de un billete válido, éste debe ir acompañado de una firma de verificación correcta obtenida con la clave privada de la instancia del servicio. Dicho con otras palabras, es necesario transportar la clave privada del servicio al vehículo. El lugar ideal para transportar esta clave es la tarjeta del conductor.

Además existen otras complicaciones: la actualización de los datos de los billetes no puede realizarse desde un lector sin contacto, por lo que habrá que recurrir a otro lector o bien reutilizar el del conductor, con los consiguientes problemas de incomodidad que esto generaría para el cliente.

Otro enfoque completamente distinto para resolver este problema consistiría en la solicitud del billete en línea desde el vehículo, por ejemplo mediante mensajes cortos (SMS). De esta forma el vehículo enviaría a un servidor de la central información acerca del usuario y del servicio, devolviendo como respuesta el billete ya firmado. Aun así, continúa el problema de que es necesario contar con otro lector de contacto para introducir el billete en la tarjeta:

A tenor de lo expuesto, no se ha considerado conveniente implantar actualmente esta posibilidad en la infraestructura, aunque en caso de ser introducida finalmente, debería optarse claramente por la segunda opción.

6 Estado de la implementación

La implementación actual consiste en un prototipo funcional que implementa todos los agentes de esta arquitectura. Para esta implementación se ha empleado el lenguaje C++, la librería PC/SC[9,10], OpenSSL[11] para las comunicaciones, la base de datos Firebird (versión *Open Source* de Interbase 6.0) y la tarjeta GemCombi/MPCOS con acceso dúal. La evaluación de este prototipo ha sido positiva tanto en términos de funcionalidad como de tiempos de respuesta.

Hay que destacar que, para realizar la implementación, se ha construido una biblioteca de clases para el desarrollo de software de acceso a la tarjeta desde el lenguaje C++, que puede considerarse alternativa al marco OCF[8] desarrollado para Java. Este nuevo marco permite la adaptación del software a diferentes entornos. Entre otras funciones, permite acceder a lectores sin contacto utilizando el protocolo TLP por línea serie (PC/SC no soporta ese acceso). Con lo que se aporta por tanto independendencia respecto al fabricante y respecto al tipo de acceso.

7 Conclusiones

En este artículo se ha presentado el diseño de una aplicación de comercio electrónico para el transporte público de viajeros. Las principales contribuciones del trabajo son el diseño de la infraestructura de servicio, la definición del billete y el mecanismo para la gestión de billetes electrónicos en tarjetas con doble acceso (contacto y sin contacto). Ambos diseños hacen posible la venta segura de billetes y el chequeo de los mismo en el vehículo sin estar en línea con la base de datos.

Entre los trabajos futuros, el primero es realizar una implementación completa del sistema, más alla del prototipo actual. En este sentido sería deseable disponer de otroas alternativas a la tarjetas empleadas, de manera que se puede contrastar realmente el grado de independencia de la aplicación respecto a las mismas.

Otra ampliación interesante supondría la incorporación de mecanismos de delegación de billetes entre usuarios.

Agradecimientos

Este trabajo se ha realizado en el marco de los proyectos e-ticket (Referencia 1FD97-1269-C02-02) y TIC-2002-04309-C02-02.

Referencias

[1] Fujimura, K.; Nakajima Y., *General-Purpose Digital Ticket Framework.* Proceedings of the 3rd USENIX Workshop on Electronic Commerce. Pp 177-186.1998. Available in: http://www.usenix.org/publications/library/proceedings/ec98/fujimura.html

[2] International Organization for Standardization (ISO), *ISO 7816 Integrated Circuit Cards with Electrical Contacts. Part 1: Physical characteristics. Part 2: Dimensions and Location of Contacts. Part 3: Electronic Signals and Transmission Protocols*. http://www.iso.ch

[3] Sun Microsystems, *Java Card 2.1.1 API Specification*, avaliable in http://java.sun.com/products/javacard

[4] Hansman, U., et al., *Smart Card Application Development Using Java*., Springer-Verlag, 2000.

[5] Castellá-Roca, J.; Domingo-Ferrer, J.; Herrera-Joancomartí, J., Planes J.A. *Comparison of Java Cards for Micropayment Implementation*. Proceedings of CARDIS'2000, pp 19-38. Kluwer Academic Publishers. 2000.

[6] Diffie, W.; Hellman, M. *New Directions in* Cryptography, IEEE Transactions on Information Theory. IT-22, n. 6. pp. 644-654. 1976.

[7] Rivest, R.L.; Shamir, A.; Adleman, L. M., *A method for obtaining digital signatures and public-key cryptosystems.* Journal of the ACM, 21(2):120-126, February 1978.

[8] The OpenCard Framework (www.opencard.org)

[9] PC/SC Workgroup: (http://www.pcscworkgroup.com/)

[10] MUSCLE Project (www.linuxnet.com)

[11] OpenSSL: www.openssl.org

Asistente para la Automatización de Operaciones de Comercio Electrónico B2C en Internet

Paula Montoto, Juan Raposo, Manuel Álvarez, Ángel Viña
Departamento Tecnologías de la Información y las Comunicaciones
Facultad de Informática. Universidad de A Coruña. 15071 A Coruña
Teléfono: 981 167 000 Fax: 981 167 160
E-mail : {pmontoto, jrs, mad,avc}@udc.es

Justo Hidalgo, Alberto Pan
Denodo Technologies. Calle Real 22, 3º. 15003 A Coruña
Teléfono: 981 100 200 Fax: 981 100 205
E-mail : {jhidalgo,apan}@denodo.com

Abstract. *Placing an order in an online shop often involves filling detailed request forms, introducing payment data, performing password-based authentication, etc. All these tasks can make the buying process tedious and cumbersome for customers. In this paper, we present an Internet navigation assistant which is able to perform these tasks automatically to a great extent, thus improving B2C usability. The system architecture relies on a personal agent which is installed in the customer browser as a toolbar, and on a set of secure web servers which securely store and manage the user-profile data and the descriptions of the supported electronic shops. The shops descriptions are automatically learned by the system, thus avoiding the need for its generation and maintenance. The agent also includes other services such as performing secure payments using a virtual-card generator, using an Europay's SPA/UCAF wallet, etc.*

1 Introducción

La gran cantidad de información que un usuario maneja diariamente en sus transacciones en Internet, a menudo convierte la experiencia de compra B2C en un proceso farragoso y propenso a errores. Entre la información que el usuario debe gestionar manualmente, por sí mismo, se encuentra:

- Información personal de registro en sitios web: nombre, apellidos, NIF, dirección, código postal, nacionalidad, e-mail, etc.
- Los identificadores de usuario y *passwords* para el acceso a sitios que requieren autenticación.
- Información referida a la realización de pedidos, como el número de tarjeta de crédito, dirección de facturación, dirección de entrega, etc.

Una aplicación que permita al usuario despreocuparse del manejo de toda esta información, incrementaría la amigabilidad del proceso de compra.

En este artículo se presenta un agente personal de ayuda al proceso de compras B2C. El sistema se instala en el navegador del cliente (Microsoft Internet Explorer [1] -en adelante MSIE-), adoptando la forma de una barra de herramientas, y se encarga de gestionar los datos del usuario relevantes para el proceso de compra, automatizando tareas como el rellenado de formularios de pedido en las tiendas o el pago electrónico mediante tarjeta de crédito. El sistema aporta importantes ventajas respecto a asistentes existentes, como Gator[4] o RoboForm [5].

Para el rellenado automático de formularios, el sistema utiliza un mecanismo colaborativo, que aprende automáticamente cómo rellenar los formularios de compra de las diversas tiendas, basándose en la experiencia previa de la globalidad de los usuarios del sistema. De esta forma, cuando varios usuarios realizan compras en una misma tienda, el sistema es capaz de aprender las características del proceso de compra en la misma y, a partir de ese momento, dicho proceso podrá ser ya automatizado para todas las compras de cualquier usuario del sistema en la mencionada tienda.

Además, el agente integra mecanismos para la realización de pagos electrónicos seguros utilizando tarjetas de crédito, mediante el uso de técnicas tales como números de tarjeta virtuales o el protocolo SPA/UCAF (Secure Payment Application / Universal Cardholder Authentication) [2][3].

Este artículo está estructurado como sigue. En la sección 2 se plantea la arquitectura global del sistema. En la sección 3 se describe el rellenador de formularios. En la sección 4 se describen los principales servicios finales implementados, así como el proceso para la incorporación de nuevas funcionalidades en el agente. Finalmente, la sección 5 plantea las conclusiones obtenidas del desarrollo realizado y esboza las líneas de trabajo futuro.

2 Arquitectura del Sistema

Para gestionar la información de los usuarios, el sistema utiliza una arquitectura distribuida basada en el uso de un servidor seguro. Esto permite, soportar la movilidad de los usuarios y minimizar la exposición de los datos sensibles del usuario a lo estrictamente imprescindible, evitando así los riesgos de seguridad asociados a elementos tales como almacenamiento de ficheros locales, cachés de navegadores, etc.

2.1 Componentes del sistema

El sistema de ayuda a la navegación aquí descrito consta de los siguientes componentes (como se puede ver en la Fig. 1):

- El servidor de autenticación. Gestiona la autenticación de los usuarios en el sistema.

- El servidor de perfiles de la barra, que es quien almacena versiones de los ficheros de configuración de la barra para diferentes distribuciones software y envía actualizaciones al agente cliente cuando es necesario.

- El servidor de perfiles de usuario. Almacena la información personal de los usuarios y la configuración de la barra de herramientas cuando el usuario se autentica en el sistema. Proporciona servicios web para editar esta información.

- El agente, instalado en el cliente como una barra de herramientas, que contiene los servicios especificados en el perfil de la barra o del usuario, dependiendo de si está autenticado o no.

Además, para el servicio de rellenado automático se necesita otro componente:

- El servidor de información de rellenado. Almacena como han rellenado los usuarios los formularios. Proporciona servicios web para obtener, añadir y eliminar dicha información, además de un promocionador de formularios.

Los datos personales del usuario y los datos de rellenado se almacenan en sendas Bases de Datos, y se accede a ellos y se manipulan a través de servicios web, manteniéndolos así totalmente independientes y desacoplados entre sí y de la implementación del agente de la parte cliente. El servidor de datos personales ofrece servicios web para añadir, eliminar o modificar los datos. El agente utilizará esos servicios, cargando la URL correspondiente que contendrá un formulario que debe rellenar el usuario. La comunicación entre los servicios web y la barra se realizará a través de campos ocultos en *frames* no visibles de las páginas de respuesta.

En lo referente al agente, se necesita que tenga acceso a los objetos cargados en el navegador del usuario, monitorizando ciertos eventos que se producen sobre ellos y accediendo o modificando algunas de sus propiedades. Por tanto, el agente va a tener que estar acoplado al navegador, lo que sugiere que será necesaria una implementación diferente en función del navegador considerado. En esta primera implementación del sistema se ha considerado el navegador MSIE, debido a su amplia difusión entre los usuarios potenciales del sistema. El tipo de *plug-in* elegido ha sido una barra de herramientas por su adaptación a los requisitos técnicos y de utilización.

2.2 Interacción entre los componentes del sistema

El funcionamiento del sistema y la relación entre sus componentes son los que siguen:

a) Cuando el usuario abre una instancia del navegador MSIE y se inicia la barra de herramientas, se lee el perfil de la misma -que tiene almacenado en local- y muestra una barra de espera mientras se conecta con el servidor de perfiles de la barra indicándole la distribución software que está instalada, y su versión del perfil. El servidor de perfiles, si detecta que hay una versión software posterior, le envía un perfil nuevo para actualizar el software. En caso contrario comprueba la versión del perfil, de manera que si tiene una versión más reciente se la envía.

b) La barra de herramientas, una vez que sabe qué perfil debe utilizar, lo carga y muestra en la barra los servicios que se indican en ese perfil, entre los que se encontrarán los servicios que permiten autenticarse en el sistema. La autenticación se realizará a través de un formulario web - para ello habrá un servicio para abrir la URL correspondiente y servicios de escucha que analizan información contenida en los frames cargados en el navegador -.

c) Tras producirse la autenticación a través del servidor de autenticación, el navegador posee una *cookie* que lo habilita para acceder a los servidores de perfiles de usuario y de formularios. Además el servicio web del servidor de autenticación envía a la barra un *frame* oculto con información que le indica que tiene que refrescar los datos del usuario. El agente correspondiente lo detecta y pide al servidor de perfiles de usuario los datos del usuario, que contendrán sus datos personales más la configuración personalizada de la barra. Cuando se recibe dicha información, se muestran en la barra los servicios que indique el perfil.

d) Si el perfil de usuario incluye el servicio de rellenado de formularios

   - La barra puede pedirle al servidor de formularios la información de rellenado de

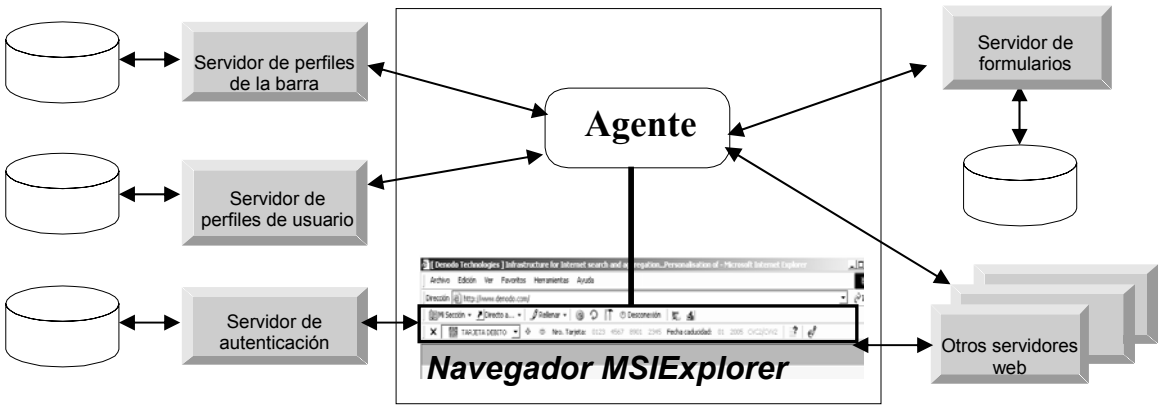


Figura 1: Arquitectura de Sistema

ese usuario justo después de autenticarse o podría esperarse a pedírsela bajo demanda según la fuese necesitando. Cuando el usuario rellena un nuevo formulario, o modifica la forma de rellenar uno del que ya hubiese información, el agente envía al servidor de formularios los nuevos datos.

- Si el servidor de formularios no está accesible, la barra sólo podrá rellenar los formularios que el usuario ha rellenado de forma manual durante la sesión actual.

e) Para salir del sistema, se hace también a través de un servicio web que elimina la *cookie* del navegador. La barra destruye la información del usuario de la sesión, y recupera su aspecto inicial, previo a la autenticación.

2.3 Arquitectura del agente

Se pretende que el sistema construido no se quede simplemente en un rellenador automático de formularios de compra, sino que esté integrado dentro de un esquema más amplio, y mucho más rico funcionalmente tanto en el ámbito de compras en Internet como en otros ámbitos.

Como se ha comentado en la introducción es necesaria la existencia de medios transaccionales de pago en los que exista seguridad tanto para el cliente como para el comerciante. En este aspecto, se ha decidido que nuestro agente pueda integrarse con una aplicación de tarjetas virtuales, y pueda actuar como *wallet* dentro del protocolo SPA/UCAF. Es interesante también una integración del rellenador automático con estos dos servicios.

Estos son solamente dos servicios que podría ofrecer el agente de la parte cliente pero existe una gran gama de posibilidades. Una alternativa es la construcción de versiones personalizadas del agente para cada perfil de usuario; esta posibilidad tiene como principal dificultad la relacionada con las reinstalaciones en la parte cliente, lo cuál obliga a crear un sistema genérico donde se puedan añadir servicios de una manera cómoda y eficiente tanto para el usuario como para el desarrollador del servicio. De esta manera, incluso podría darse la opción al usuario de personalizar la lista de servicios presentes en su agente, de manera dinámica.

La idea consiste en tener una serie de servicios que pueden ser desarrollados de forma independiente, de manera que en una distribución software de la barra de herramientas se incluyen los servicios que se desee que puedan formar parte de la barra. Para cada usuario de una distribución software de la barra se puede configurar el conjunto de servicios presentes para ese usuario. Este esquema permite generar fácilmente versiones software de la barra, personalizadas para distribuidores de servicios, que a su vez puede ser personalizada para cada usuario final. También permite controlar y actualizar fácilmente versiones software de la barra de herramientas añadiendo o eliminando servicios. El rellenador automático, *wallet* SPA/UCAF, y la generación de tarjetas virtuales serían algunos de los servicios que forman parte de la barra de herramientas (ver sección 4.1 para más detalle).

Para construir esta barra de herramientas genérica el software puede configurarse vía ficheros XML (eXtensible Markup Language), para fijar el contenido de la barra para una distribución concreta. Inicialmente esta configuración se almacenará en local en el proceso de instalación y nos referiremos a ella como perfil de la barra. Contendrá la lista de tipos de servicios incluidos en esa distribución software, la lista de servicios que podrán ser incluidos en la barra -instancias de cada tipo de servicio parametrizado-, y el aspecto de la barra, es decir la lista de servicios incluidos y en que orden, cuando el usuario aún no se ha autenticado en el sistema - cuando abre el navegador -.

El tipo de un servicio indica la representación gráfica del mismo - si la tiene - y responde ante un conjunto concreto de eventos sobre la barra y/o el navegador. En el siguiente apartado se profundizará más sobre este tema.

Figura 2: Agente visualizándose como una barra de herramientas

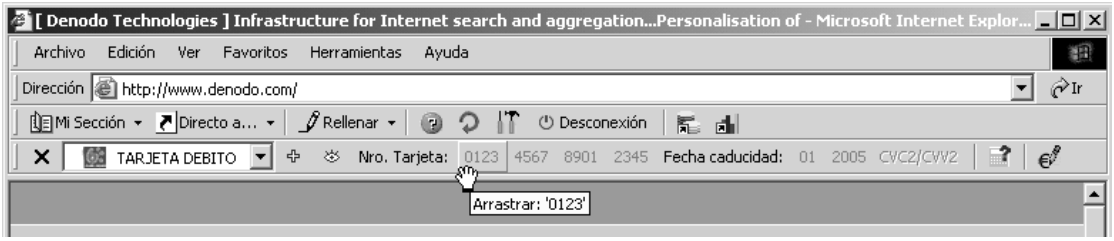


Figura 3: Agente con las dos barras de herramientas

Los aspectos que pueden personalizarse para cada servicio, vienen determinados por su tipo, y harán referencia a su aspecto gráfico y a sus parámetros de funcionamiento - que pueden incluir relaciones con otros servicios -.

Una vez que el usuario se autentica en el sistema, el contenido de la barra de herramientas cambia, puesto que ya se han cargado sus datos y se le pueden ofrecer los servicios que dependen de ellos, tales como el rellenado automático. Por tanto el servidor que almacena los datos personales del usuario también va a almacenar la configuración de la barra para ese usuario cuando se autentica en el sistema.

Anteriormente se ha comentado que el perfil de la barra se almacena en local en la instalación, pero es necesario un mecanismo que nos permita actualizarlo para poder cambiar la configuración de un servicio o la lista de servicios presentes en la barra, sin tener que hacer una reinstalación. Para ello se utiliza un mecanismo de control de versiones, consistente en que cada vez que se inicia la barra, le envía al servidor de perfiles un identificador de la distribución software, y la versión de la configuración local que posee para actualizarse si es necesario. Por ejemplo, en caso de que el servidor detecte que hay una versión software posterior, éste podrá enviar a la barra un perfil con un servicio para actualizar la versión software.

En el proceso de instalación de la barra puede seleccionarse el idioma de la misma, y una vez instalada el usuario puede decidir cambiar el idioma. Por ese motivo, los textos e incluso las imágenes de la barra serán internacionalizables.

Para potenciar la usabilidad de la barra y ofrecer un mayor dinamismo se incluyen dos tipos de servicios especiales que no son servicios finales, sino contenedores de otros servicios: los servicios compuestos y los servicios de segunda barra (que se describen en la sección 4). El agente siempre va a tener una barra de herramientas visualizándose y a veces tendrá abierta una segunda barra, cuyo contenido variará en función del servicio de segunda barra que la haya abierto (Fig. 2 y Fig. 3).

3 Rellenador de formularios

3.1 Planteamiento inicial del rellenador de formularios

Una primera aproximación para agilizar los procesos de compra es la de que el usuario tenga disponible gráficamente su información de tarjetas y direcciones y que pueda emplearla sin tener que teclearla cada vez- por ejemplo arrastrándola sobre el campo del formulario que desee- . Para esto es necesario que el usuario registre previamente su información, pudiendo modificarla y dar de alta o baja nuevos datos de tarjetas o direcciones en cualquier momento. Por la seguridad y comodidad del usuario, no es recomendable que esa información se almacene en su máquina local, sino que ésta se almacene en un servidor seguro, al que acceda el sistema de rellenado previa autenticación, de manera que la información confidencial del usuario se destruya una vez finalizada la sesión correspondiente.

Si un usuario rellena un formulario, el sistema podría recordar como ha rellenado ese formulario y ofrecerse posteriormente a rellenarlo de la misma manera. El siguiente paso sería que el sistema fuese capaz de rellenar automáticamente los formularios que el usuario ya ha rellenado con anterioridad, pero permitiéndole elegir el dato con el que el usuario desea rellenarlo esta vez - por ejemplo el usuario puede poseer varias tarjetas y querer realizar la compra con otra diferente a la que usó la primera vez. Nótese que para ello el sistema debe aprender asociaciones lógicas, como por ejemplo en qué

campo de un formulario el usuario introduce su número de tarjeta de crédito. Para poder aprender estas asociaciones, el sistema debe monitorizar las acciones del usuario, sobre los formularios que va rellenando. El funcionamiento sería el siguiente:

- En todo momento se está monitorizando cómo el usuario rellena los campos de los formularios. Es decir, se "escuchan" las operaciones de *drag&drop* que hace de sus datos sobre los campos de los formularios.

- Cuando se envía un formulario, se almacenan las asociaciones que se refieren a ese formulario, junto con un identificador del formulario, que identificará la página en la que está y qué formulario es dentro de esa página.

La información de cómo el usuario ha rellenado los formularios es interesante que se almacene en un servidor central, en lugar de la máquina local, porque ello nos permitirá utilizar esa información para "aprender" formas comunes de rellenar formularios y poder ofrecérselas a usuarios aunque nunca hayan rellenado el o los formularios en cuestión.

Por tanto habrá un servidor, al que llamaremos "servidor de formularios" que almacenará la información de asociaciones entre datos de usuarios y campos de formularios. Cada usuario tendrá su propia información, generada con sus compras, a la que llamaremos "asociaciones temporales" - habrá una por cada formulario que el usuario haya rellenado y enviado - . En el servidor se ejecuta periódicamente un proceso que va analizando las asociaciones temporales de todos los usuarios que hacen referencia a un mismo formulario, para extraer asociaciones que podrían ser consideradas válidas para los usuarios del sistema que nunca han rellenado ese formulario. Por ejemplo, un criterio sería considerar una asociación validada, cuando un porcentaje de todos los usuarios ha rellenado un formulario de la misma manera.

El servidor de formularios enviará a cada usuario la información de sus asociaciones temporales generadas en sesiones anteriores, y la información de asociaciones validadas, aprendidas por el servidor. Existen dos posibilidades configurables para recibir esa información (a) Recibirla en el momento de autenticarse en el sistema, además de recibir los datos personales del usuario, y (b) recibirla bajo demanda, es decir, cuando el usuario decida utilizar el rellenador automático, éste solicitaría al servidor de formularios, las asociaciones que tuviese para la página actual. La opción elegida dependerá del volumen de información tratado, lo cual dependerá del número de usuarios y del uso que hagan del sistema.

3.2 Generalización del rellenador de formularios

El rellenador de formularios planteado en el apartado anterior actúa sobre información de tarjetas y direcciones de envío, pero podría utilizarse para rellenar formularios de otros dominios sin variar su filosofía básica de funcionamiento. Por ejemplo, un dominio muy útil en el que podría ser utilizado es el de los formularios de identificación.

La representación utilizada es la siguiente: un usuario tiene asociados una serie de tipos de datos, cada uno de los cuales estará identificado por un nombre - tarjetas, direcciones, identificadores, etc. - y tendrá asociado un conjunto de datos de ese tipo. Cada dato tendrá un identificador y una lista de atributos, identificados por su nombre y tendrán asociados un valor. Para representar los datos en este formato, el servidor de datos de usuario se los comunicará al rellenador de formularios en XML.

Considerando los datos de esta forma genérica, cada asociación debe almacenar el nombre del tipo de dato y el nombre del atributo que se ha utilizado para rellenar el campo correspondiente. Por ejemplo, tendríamos una asociación para el formulario F y el usuario U que indicase que el campo "X" se ha rellenado con el atributo "número de tarjeta" del tipo de dato "tarjetas". Con este nuevo enfoque, el rellenador de formularios puede parametrizarse con el nombre de los dominios a utilizar en el rellenado automático.

En las asociaciones consideradas hasta el momento se asume que el usuario siempre rellena un campo con el valor completo de un atributo de un tipo de dato. Siguiendo con el mismo ejemplo de tarjetas, hay formularios en los que el número de la tarjeta debe introducirse en cuatro campos de cuatro en cuatro cifras. Para soportar este y otros casos parecidos, se pueden definir patrones para seleccionar parte de un valor de un atributo.

3.3 Proceso de aprendizaje

La filosofía que sigue el proceso de validación de formularios es la siguiente: cada cierto tiempo configurable se lanzará el proceso, en el que un módulo Clasificador clasifica los formularios de los que hay información en categorías de formularios. Para aquellos de una misma categoría se analizan las asociaciones temporales de los usuarios que lo han rellenado, y se clasifican en categorías de asociaciones a través del mismo Clasificador. Acto seguido decide, a través de un Promocionador, qué categorías cumplen los requisitos necesarios para poder ser consideradas como asociaciones validadas, y por último, selecciona a los representantes de esas categorías para dar lugar a asociaciones validadas.

El proceso de aprendizaje de asociaciones validadas se ha implementado conforme a unas interfaces, de manera que se pueda redefinir fácilmente.

Mediante propiedades de configuración se puede seleccionar el Clasificador y el Promocionador a utilizar. Un Clasificador básico sería el de igualdad, es decir, dos formularios pertenecen a la misma categoría si su identificador es igual y dos asociaciones temporales pertenecen a la misma categoría si tienen exactamente las mismas asociaciones simples entre datos de usuario y campos del formulario. Un ejemplo de Promocionador es uno que decida que una categoría de asociaciones se puede validar cuando hay un número mínimo de formularios que pertenecen a la categoría de formularios y el porcentaje de mapeos pertenecientes a una categoría de asociaciones con respecto al total de asociaciones de la categoría de formularios supera un umbral.

Este Promocionador combinado con el Clasificador anterior constituyen una estrategia simple de aprendizaje: cuando un cierto número mínimo de usuarios han rellenado un mismo formulario y un porcentaje determinado lo han hecho de la misma forma, se añade esa forma de rellenarlo como una asociación validada. Pueden elegirse estrategias más complicadas, por ejemplo haciendo comparaciones de clasificación que no sean de igualdad u otorgando diferentes puntuaciones a la forma de rellenar los formularios de distintos usuarios.

4 Tipos de Servicios

Para poder incluir un servicio dentro de la barra, se debe especificar: un nombre para el servicio, que lo identifica unívocamente, el tipo de servicio al que pertenece y un conjunto de parámetros específicos.

Un tipo de servicio identifica un comportamiento específico, es decir, reacciona ante uno o varios eventos sobre la barra y/o el propio navegador, realizando una determinada acción.

El sistema diferencia varios tipos de servicios. En primer lugar, clasifica los tipos de servicios en estáticos y no estáticos. Los tipos de servicios de naturaleza estática, son aquellos que no tienen definida ninguna representación gráfica dentro de la barra. Por el contrario, los tipos de servicios no estáticos, sí tienen representación gráfica, y por tanto, además de manejar determinados eventos, se encargan de su visualización. Por otro lado, dentro de la barra de herramientas, un servicio no estático, puede incluirse como estático si el usuario no lo invoca de forma directa, sino que desde otros servicios.

Una segunda clasificación, ya comentada con anterioridad, diferencia los tipos de servicios en simples y en servicios contenedores de otros servicios. Un servicio simple es aquel servicio final que tiene un comportamiento definido que reacciona ante determinados eventos. En cambio, el segundo grupo, lo componen aquellos que se configuran en función de otros servicios, que forman parte de él.

4.1 Algunos servicios importantes

A continuación se explican algunos de los tipos de servicios que proporciona la barra de herramientas.

Servicio URL

Un tipo de servicio de naturaleza simple y no estático, que se encarga de visualizar una dirección web en el propio navegador o en una nueva ventana. Este tipo de servicio admite un amplio conjunto de parámetros de configuración que especifican aspectos como la URL a la que debe navegar, el método de acceso a utilizar en la navegación, los parámetros constantes, o referentes a elementos del perfil del usuario que se desean enviar en la petición, si se desea realizar la navegación en una nueva ventana, y en caso positivo, el tamaño de la misma, si se desea mostrar las barras de herramientas, entre otros.

Servicio de Rellenado

Es un servicio simple y no estático cuya función es la de realizar el rellenado de los formularios de la página que se visualiza en el navegador de manera automática, si tiene la información necesaria para hacerlo. El servicio para realizar dicha operación, debe disponer de dos tipos de información: la información de perfil del usuario, que se obtiene en el proceso de autenticación, y la meta-información de mapeos de los datos del perfil y los elementos del formulario web, que conoce el servidor de formularios.

El servicio de rellenado comprueba si existe información de mapeo para alguno de los formularios de la página actual, y en caso positivo establece como valores de los elementos del formulario, los elementos del perfil que se indican en la meta-información de mapeo.

Además, el servicio de rellenado, antes de proceder a rellenar con un elemento perteneciente al tipo de dato relativo a las tarjetas, si la tarjeta que le corresponde utilizar en el rellenado admite la generación de tarjetas virtuales, simula un servicio de tipo URL estático, que permitirá al usuario crear una tarjeta virtual a través de un formulario web. Si el usuario decide crear una tarjeta virtual, el servicio de rellenado utiliza la nueva tarjeta en el proceso de rellenado, y en caso contrario, rellena con la tarjeta que originó el proceso.

Servicio de Tarjetas Virtuales

Mediante un servicio de tarjeta virtual, el cliente puede solicitar una tarjeta virtual, distinta de la suya, particular y diferente para cada pago. Este servicio le

ofrece los datos de una tarjeta virtual que puede trasladar al formulario en sustitución de la suya, con un nuevo número, una fecha de caducidad próxima a la fecha del pago y un importe determinado. Será luego el gestor del servicio de tarjeta virtual quien cargue el importe de la compra a su propia tarjeta garantizándole seguridad y confidencialidad.

El servicio de Tarjetas Virtuales es un servicio simple y estático que integra la barra con la aplicación externa encargada de generar las tarjetas virtuales. Tiene como misión escuchar los documentos de las páginas que se cargan en el navegador, con el fin de detectar la llegada de un *frame* especial que notifica a la barra la generación de una nueva tarjeta virtual o la cancelación de la creación de una tarjeta virtual. Cuando el servicio detecta la existencia de alguno de los *frames*, notifica al servicio de rellenado que inició el proceso de generación de la tarjeta virtual, para que rellene con la nueva tarjeta o que realice el rellenado con la tarjeta seleccionada o por defecto, según le corresponda.

Servicio *drag&drop*

El servicio de tipo *drag&drop* es un servicio simple y no estático que permite realizar al usuario el rellenado manual de los formularios web de una manera fácil e intuitiva y permite a la barra identificar de forma directa las relaciones de mapeo entre la información del perfil del usuario y los elementos de los formularios web. Un servicio de este tipo, es el encargado de visualizar dentro de la barra el valor de un atributo de un tipo de dato del perfil del usuario a través de uno o varios botones con texto, para que el usuario pueda “arrastrar” el valor del atributo o el valor de un fragmento del valor del atributo, a elementos editables de un formulario.

Además, un servicio de *drag&drop* podrá configurarse para que fragmente el valor del atributo que representa según un criterio - por espacios o por rango -, y añada tantos botones como fragmentos se identifiquen. Cada uno de esos botones, tendrá como texto el valor del fragmento que representa, y cuya función final es la de permitir arrastrar dicho valor. Esto se utiliza, por ejemplo, para que el usuario pueda introducir en un formulario web el número de tarjeta en grupos de 4 dígitos, a través de operaciones de *drag&drop*.

Servicio compuesto

Un servicio de tipo compuesto es un servicio estático contenedor de otros servicios. Se representa como un botón "especial" que al pulsarlo despliega un menú de selección, dónde cada opción del menú desplegable representa a un servicio concreto formando una estructura de n niveles. No todos los tipos de servicios pueden ser incluidos dentro de un servicio compuesto.

Servicio SPA/UCAF

Otro tipo de servicio que incluye la barra de herramientas es el denominado SPA/UCAF. Este tipo de servicio implementa un *wallet* SPA/UCAF, es decir, el elemento que comunica el cliente que realiza la compra con un servidor SPA, permitiendo a la barra actuar como un monedero que soporta la normativa SPA/UCAF. El tipo de servicio SPA/UCAF es un servicio simple y estático encargado de comprobar -por cada página que recibe el navegado- si se trata de una página UCAF –una página con un formato específico- relativa a una compra en un comercio que utiliza ese sistema de pago. Si el servicio, identifica una página UCAF, realiza modificaciones sobre la misma, como el establecimiento de valores específicos sobre elementos del formulario de la página e incluso - si es necesario - realiza el *submit* sobre el formulario.

Posteriormente, comprueba que el usuario se ha autenticado - en caso negativo, muestra una página de autenticación a través de un servicio URL - y muestra al usuario una ventana modal con cierta información relevante de la compra que le permite seleccionar entre sus tarjetas, la que desea utilizar en la compra. Acto seguido, el servicio solicita el AAV a una aplicación externa, rellena los campos UCAF con la respuesta - tarjeta, fecha caducidad, etc. - y realiza el *submit* del formulario de compra.

Servicio lista de selección

Un tipo de servicio de lista de selección permite la selección de uno de los elementos que pertenecen a un tipo de dato concreto.

Un servicio de este tipo, se visualiza en la barra de herramientas como una lista de selección, que muestra en cada elemento de la lista un texto que identifica el elemento junto con una imagen.

Dicho servicio tiene asociado un tipo de dato del perfil, que es configurable, y que indica qué elementos debe mostrar en su interior. Cada elemento del tipo de dato especificado, se corresponde con un elemento de la lista de selección.

Servicio de segunda barra

Un servicio de segunda barra es un servicio simple no estático, contenedor de otros servicios. Se muestra en la barra de herramientas como un botón que al pulsar, muestra una segunda barra con una lista de servicios representados gráficamente en su interior.

4.1 Implementación de un nuevo tipo de servicio

El sistema posibilita la adición de servicios de valor añadido. Para añadir un nuevo tipo de servicio en la barra de herramientas, es necesario conocer su comportamiento, es decir, saber ante qué eventos desea responder con una determinada acción. Cuando se define un nuevo tipo de servicio, el tipo de servicio

se suscribe como observador de uno o varios grupos de eventos.

Los tipos de eventos más importantes que se identifican en el sistema son:

- Los eventos que indican que un servicio se ha activado, bien de forma directa por parte del usuario o bien desde otros servicios.
- El evento que se produce cuando se selecciona un botón que muestra un menú desplegable.
- El evento que se produce cuando el usuario inicia una operación de *drag&drop* sobre algún elemento de la barra de herramientas.
- Los eventos que representan acciones de navegación que se producen en el navegador: inicio de navegación, documento o *frame* cargado, página completa cargada, envío de un formulario, etc.
- El evento que indica que el control de la ventana de la barra de herramientas, en la que se encuentra el servicio, cambia de tamaño.
- El evento que solicita al servicio un texto de ayuda que se representa a través de un *tooltip*.
- El evento que se produce cuando el usuario mueve el ratón sobre alguno de los controles que forman parte de su representación gráfica.

3 Conclusiones y Trabajos Futuros

En este artículo se ha presentado un agente personal de compras que convierte el proceso de compra B2C en un proceso más amigable, más seguro y menos propenso a errores. El sistema se ha desarrollado para los usuarios de banca electrónica de diversos bancos regionales españoles.

En el trabajo previo, quizá los sistemas más similares al aquí presentado son rellenadores automáticos de formularios como Gator [4] o RoboForm [5], etc. El presente trabajo proporciona un marco más general que el de dichos sistemas previos, proporcionando, entre otras, las siguientes ventajas:

- Almacenamiento centralizado y seguro de la información sensible de los usuarios.
- Por lo anterior, se posibilita la movilidad de los usuarios, es decir, que un usuario no dependa de una máquina concreta para poder utilizarlo.
- El sistema aprende, de la experiencia del total de usuarios, de forma que un usuario que nunca haya realizado una compra en una tienda podrá beneficiarse del sistema igualmente, si otros usuarios han realizado compras en dicha tienda.
- Está integrado en una plataforma general que permite incorporar servicios de valor añadido relacionados con el pago electrónico, como los ya integrados de SPA/UCAF y tarjetas virtuales.
- Debido a su generalidad y extensibilidad, es posible añadir nuevos servicios de manera cómoda y eficiente, posibilitando variar su composición de forma dinámica, sin necesidad de tener que realizar actualizaciones del sistema en las localizaciones de todos los agentes.

En cuanto a líneas de trabajo futuro:

- La versión actual del sistema sólo soporta el navegador MSIE como contenedor del agente configurable. Aunque se trata del navegador más ampliamente utilizado, se considerará la migración del sistema a otros contenedores, como Mozilla/Netscape.
- Se podría aprovechar la capacidad de monitorización de las navegaciones de los usuarios del sistema, para obtener información relevante acerca de sus comportamientos, que podrían posibilitar la adición de nuevos servicios personalizables al agente.

Referencias

[1] Microsoft Corporation, Microsoft Internet Explorer 5.5 o superior, http://www.microsoft.com, 2001.

[2] SPA and UCAF Detailed Specification, Global e-Business & Emerging Technologies, Mastercard International, 2001.

[3] Mastercard SPA/UCAF, http://www.mastercardintl.com/newtechnology/ecommercesecurity/spa/.

[4] Gator eWallet, The Smart Online Companion, http://www.gator.com.

[5] RoboForm: Free Password Manager, Form Filler, Password Generator, AutoLogin. http://www.roboform.com.

Protocolo de Intercambio con Atomicidad para el Pago de Cantidades Elevadas Mediante Moneda Electrónica

Magdalena Payeras Capella, Josep Lluís Ferrer Gomila, Llorenç Huguet Rotger
Departament de Ciències Matemàtiques i Informàtica. Universitat de les Illes Balears.
Carretera de Valldemossa, Km. 7.5, 07122 Palma de Mallorca
Teléfono: 971171390
E-mail: {mpayeras, dijjfg, dmilhr0}@uib.es

Abstract. *Payments involving high amounts of money must be extremely secure. Security aspects have to be implemented in the payment system (in this case electronic coins) and in the fair exchange between the coin and the product. A secure payment scheme for high amounts has to provide prevention of forgery, overspending, double spending and robbery. This paper describes a fair exchange protocol that can be used together with a payment system for high amounts. This way, the fair exchange of elements represents the addition of atomicity to the payment system.*

1 Introducción

Entre los protocolos de comercio electrónico presentados hasta la fecha, aparecen, por una parte, propuestas para la realización de pagos mediante moneda electrónica, y por otra, protocolos que permiten que en la realización de una compra se produzca un intercambio equitativo entre la moneda utilizada en el pago y el producto comprado o un recibo de la compra [3,4,7,11]. Entre las primeras, las propuestas para el pago de cantidades elevadas permiten prevenir el fraude (falsificación, robo, reutilización) con el objetivo de asegurar la validez de los pagos [5].

La compra de productos con precios elevados requiere, a la vez, la prevención del fraude en el uso de monedas y el intercambio equitativo entre producto y moneda. Este hecho conduce a la definición de un protocolo específico para este tipo de transacciones. El protocolo de intercambio para pagos para cantidades elevadas se basa en la utilización de un protocolo de pago que proporcione la seguridad adecuada para los pagos de grandes cantidades, como el protocolo descrito en [5]. El protocolo de intercambio mantendrá las características de seguridad y anonimato de la moneda utilizada, a las que añadirá las características deseables del intercambio. Estas características son entrega certificada bilateral y participación de una tercera parte de confianza sólo para la resolución de conflictos.

2 Intercambio entre moneda electrónica y producto o recibo de la compra

El pago a cambio de un recibo (PpR) o el pago a cambio de un producto (PpP) se dan siempre y cuando se utilice un medio de pago electrónico en la compra de un bien o servicio adquirido en un establecimiento electrónico. En un pago con tarjeta de crédito, la orden de compra que incluye el número de la tarjeta (y la firma) se intercambia por el recibo del pago o por el producto. En los pagos con moneda electrónica, la moneda forma parte del intercambio y se convierte en el elemento a intercambiar por parte del comprador.

Nos centraremos en el caso de los pagos con moneda electrónica, puesto que los intercambios de orden de compra firmada a cambio de un recibo del pago se pueden considerar una aplicación de los protocolos de firma de contratos. Por otra parte, el intercambio que involucra moneda electrónica no se puede considerar resuelto mediante los protocolos de firma de contratos, puesto que existen situaciones específicas donde la interrupción del intercambio provocaría la pérdida de la moneda para las dos partes o la pérdida del anonimato de alguna de ellas.

Posibles fallos de la red por una parte, y el comportamiento fraudulento de cualquiera de las partes involucradas en el intercambio por la otra, pueden ocasionar problemas. El comportamiento fraudulento puede conducir a situaciones donde una de las dos partes proporcione el bien o realice el pago y no reciba el pago o el bien de la otra parte. La atomicidad permite vincular una serie de operaciones y obligar a que estas se ejecuten en la totalidad o no se ejecuten en absoluto.

Los fallos de la red pueden dar lugar a situaciones donde las monedas pasen a estar en un estado ambiguo, donde se podría llegar, incluso, a perder su valor. Por ejemplo, en un sistema de moneda electrónica *off-line* en el que un error provoca que un pagador no pueda saber si el receptor ha recibido o no la moneda, no puede arriesgarse a utilizarla de nuevo puesto que si lo hiciera y el pago se había finalizado, el usuario no tan sólo sería identificado sino que también sería acusado de reutilización.

En un intercambio atómico se da la propiedad de equitatividad. Además, es deseable que las partes

puedan demostrar cuál es el objeto que la otra parte ha recibido, y por lo tanto, en caso de disputa posterior, se puedan aportar pruebas del intercambio. En función de las prestaciones de los protocolos de PpP o de PpR, se hablará de intercambio atómico y/o compra certificada para alguna/todas las partes involucradas en el intercambio.

Existen, en función de sus características, diferentes tipos de protocolos. En [11] se define *atomicidad del dinero* como la característica que impide que en la transferencia de fondos exista creación o destrucción de dinero. Por lo tanto en este caso no existe intercambio equitativo sino que tan sólo se protege la transacción de la moneda de manera que esta se haga de forma atómica.

También en [11] se define *atomicidad del bien.* Esta segunda definición comprende los protocolos que no sólo presentan atomicidad del dinero sino que también permiten el intercambio equitativo entre el bien y la moneda.

La *entrega certificada* [11] permite la atomicidad de moneda y bien, y proporciona, a las dos partes involucradas en la compra, pruebas de lo que han enviado y de lo que la otra parte ha recibido. En la *entrega certificada unilateral* [3], el cliente puede probar qué bienes recibió, en caso de disputas, por ejemplo cuando los bienes recibidos no se ajustan a su descripción, pero el vendedor no puede probar que el cliente recibió el bien. El vendedor tiene garantizado el cobro si y sólo si el cliente obtiene el bien digital correctamente. Por otra parte, el vendedor no puede probar que el comprador recibió correctamente los bienes. En la *entrega certificada bilateral* [3], a diferencia del caso unilateral, todas las partes tienen pruebas de recepción.

La *entrega certificada y atómica* [8] proporciona atomicidad del bien y la moneda. Las partes se han puesto de acuerdo en la negociación inicial y el intercambio da pruebas de que los bienes y la moneda se han recibido. Tiene, a la vez, atomicidad del bien y entrega certificada.

Finalmente, la *compra atómica distribuida* [8] proporciona atomicidad del dinero y del bien cuando en la compra intervienen más de un vendedor. En muchas aplicaciones de comercio electrónico, la compra de un cliente no se limita a un vendedor independiente. Si, por ejemplo, un usuario compra un software a un vendedor y para ejecutarlo necesita un sistema operativo concreto que sólo está disponible en otro vendedor, ninguno de los dos bienes son útiles por separado, el usuario necesita la garantía de que la compra de los bienes se puede hacer de forma atómica, para recibir los dos bienes o no pagar ninguno.

2.1 Objetivos

En la aplicación de pago por recibo, como en la de pago por producto, el objetivo es conseguir que la compra sea certificada y el pago atómico. De esta manera un vendedor podrá demostrar que un cliente ha recibido el bien, a la vez que el cliente podrá probar que el vendedor ha recibido el pago así como demostrar cuál es el objeto que él ha recibido en el intercambio.

El anonimato de la compra es una característica deseable. Interesará que la compra del producto (PpP) sea anónima, al menos para el usuario que realiza el pago. En los intercambios PpR, la recepción del producto suele ser una etapa en la que el usuario receptor no es anónimo, pero incluso en este caso puede interesar que el usuario pagador permanezca anónimo. Por esta razón, es necesario definir protocolos de PpP y PpR de manera que se mantenga el anonimato que proporciona el sistema de pago. Así, un pago con comprador anónimo dará lugar a un protocolo de intercambio con las mismas características.

Otra de las características deseables en los protocolos de PpP o PpR es la capacidad para ejecutarse sin la intervención de una tercera parte de confianza (TTP). La TTP se utilizará para resolver los conflictos cuando el protocolo no se complete, sea cual sea la razón que ha provocado esta situación. En los intercambios PpP o PpR, se requerirá no sólo que el intercambio no requiera la participación de la TTP en cada ejecución, sino que también será fundamental que el pago (que formará parte de uno de los pasos del intercambio) no requiera la validación de la moneda por parte del banco.

2.2 Soluciones Existentes

Las soluciones existentes hasta la fecha para PpP y PpR se pueden agrupar en función de la intervención de terceras partes. En la solución adoptada en [10], existe un coordinador que conoce la identidad de todas las partes, de ahí que el sistema no permita pagos anónimos. Esta solución es útil en caso de fallos del sistema, pero no es útil en el caso de intento de fraude.

Algunas soluciones, como [4] no precisan de una tercera parte. En este caso se opta por dividir la moneda en dos partes que se enviarán antes y tras la recepción del bien, respectivamente. El vendedor no está protegido, puesto que no puede recurrir a una tercera parte en caso de no recibir la segunda parte de la moneda. Si no se completa la operación simplemente no cobra. Las monedas pueden tener un estado ambiguo si el comprador no se arriesga a reutilizarlas para no ser identificado. Como conclusión no proporciona atomicidad, y sólo proporciona cierta protección al pagador.

Otros protocolos realizan los intercambios con la mediación de una tercera parte, como [3, 6, 9,11]. En [6] existe una tercera parte activa, concretamente un *blackboard* donde todos los usuarios pueden leer y escribir. En [3] el banco, que actúa como TTP, interviene en el pago. El esquema proporciona entrega certificada unilateral. [11] presenta un pago *on-line* donde el banco también actúa como TTP y garantiza el intercambio equitativo durante el pago. Esquemas parecidos son [9] y [8] donde se utiliza un coordinador de pago *on-line*.

Existen algunas soluciones donde la TTP es pasiva. Estas soluciones son optimistas puesto que la tercera parte sólo interviene en caso de excepción. Las soluciones existentes con TTP pasiva no consiguen protocolos ideales para el PpP o el PpR. Por ejemplo, en [13] si no se acaba el intercambio de forma satisfactoria, el cliente no podrá conseguir el bien, sólo podrá recuperar el dinero, es decir, existe la posibilidad de cancelar el intercambio, pero no se puede forzar la finalización. Además, si el vendedor no deposita la moneda recibida, entonces no lo pueden relacionar con la recepción del pago.

Otra solución es [12]. En este protocolo no se hace el análisis del sistema de pago que se utilizaría en el intercambio. La compra no es certificada, es decir, el vendedor no puede demostrar que el cliente haya recibido el bien.

3 Descripción del protocolo de pago de cantidades elevadas

El protocolo descrito en [5] es un sistema de pago electrónico que permite la prevención de la reutilización con independencia del uso de dispositivos resistentes a manipulación a la vez que realiza la captura del pago de forma off-line. El esquema está protegido contra el fraude, evitando la falsificación, la sobreutilitzación y los robos, además de la reutilización. Estas características hacen que el sistema, desde el punto de vista de la seguridad, sea adecuado para la transferencia de cantidades elevadas.

Otro aspecto a considerar, común en los pagos de cualquier magnitud, es la posibilidad de pago y cobro anónimos. Si esta característica se añade a las anteriores (prevención de la reutilización y otras medidas de seguridad), hace que este esquema sea conveniente para los pagos anónimos de cantidades elevadas. Todas las transacciones (excepto la retirada y el depósito en las que se identifica una cuenta) pueden llevarse a término de forma anónima. Los clientes pueden realizar pagos anónimos y además estos no pueden ser rastreados por una confabulación entre el comerciante que va a recibir el pago y el banco.

Analizando el anonimato del receptor, se puede decir que las monedas recibidas en un pago pueden ser utilizadas de manera anónima. Los comerciantes pueden elegir entre gastar de manera anónima las monedas recibidas o depositarlas. En el caso de utilización anónima de las monedas, los pagos realizados tampoco podrán ser rastreados. Los receptores pueden utilizar las monedas con el mismo anonimato que tendrían si utilizasen monedas retiradas del banco.

La posibilidad de llevar a cabo actividades ilícitas es una de las consecuencias del anonimato. Para resolver el problema se utiliza la revocación de anonimato. Aunque en el esquema presentado en [5] las actividades ilícitas pueden ser prevenidas si se sospechan, los usuarios implicados en actividades ilícitas denunciadas a posteriori pueden ser identificados gracias a un protocolo de revocación del anonimato.

El esquema de pago involucra tres tipos de usuarios: el cliente o pagador (*C*), el Vendedor o receptor (*M*) y el banco (*B*). El conjunto de subprotocolos incluye la *retirada*, un *protocolo de fines múltiples* (*MPP*, utilizado para las operaciones de *pre-pago* y a*uto transferencia*), la *captura del pago* y el *depósito*. *B* está implicado en la retirada, el pre-pago, la auto transferencia y el depósito, pero no es necesario durante el pago.

La notación utilizada en la descripción de los protocolos se lista en la siguiente tabla.

Tabla 1. Notación

X, Y	**Concatenación de dos mensajes X e Y**
$i \rightarrow j$: X	**El usuario i envía un mensaje X al usuario j**
i: *operación*	**El usuario i realiza *operación***
$x \stackrel{?}{=} y$	**Comprobación de una igualdad**
$E_K(x)$	**Cifrado simétrico del elemento X mediante la clave K**
$D_K(x)$	**Descifrado simétrico del elemento X mediante la clave K**
H(x)	**Función de hash aplicada al elemento X**
$S_u(x)$	**Firma del elemento X realizada por el usuario u**
$K_{Pu}(x)$	**Cifrado del elemento x con la clave pública del usuario u**
$K_{Su}(x)$	**Cifrado del elemento x con la clave privada del usuario u**
T	**Tercera parte de confianza**
W_0	**Identificador de la moneda**

W_1	**Prueba secreta relacionada con la moneda**
M_x	**Moneda identificada con el número X**
Q_x	**Valor de la moneda X**
MPP	**Subprotocolo multipropósito**
W_{0m}	**Identificador generado por el receptor del pago**
W_{1m}	**Prueba secreta generada por el receptor del pago**
Ack	**Reconocimiento generado por el banco enviado cuando se acepta una moneda para ser cambiada o depositada**
Nack	**Mensaje que envía el banco cuando detecta un intento de reutilización**

3.1 Subprotocolo de retirada

Este subprotocolo permite la creación de monedas. El usuario que actúa como cliente (*C*) solicita una moneda nueva al banco. Como consecuencia *B* autentifica a *C*, crea una moneda nueva y descuenta el importe de la nueva moneda de la cuenta de *C*.

Tabla 2. Subprotocolo de retirada.

1. C:	**W_1, Q_1, $W_0 = h(W_1)$**
2. C → B:	**Solicitud = $=\{Q_1, W_0, Id_c, S_c(Q_1, W_0)\}$**
3. B:	**Verificación de la firma: $K_{pc}(S_c(Q_1, W_0)) \stackrel{?}{=} h(Q_1, W_0)$**
4. B → C:	**$M_1 = \{Q_1, W_0, K_{sb}(Q_1, W_0)\}$**

W_0 y la cantidad Q_1, junto con la firma bancaria sobre ellos formaran la moneda. W_0 *(identificador de moneda)* será utilizado como un número de serie para reconocer la moneda y prevenir reutilizaciones. Recordando que el identificador es el resumen de la prueba secreta W_1, *C* puede demostrar la posesión de la moneda con el conocimiento *de* W_1 y M_1 (sólo *C* conoce las dos partes).

Cuando cualquier usuario quiere utilizar una moneda, ha de demostrar el conocimiento de su prueba secreta (W_1), pero no está obligado a revelar su identidad. Cuando *B* guarda toda la información disponible sobre la moneda, dispone de suficiente información para reconocer la moneda en el depósito, pero gracias al uso del *MPP*, el banco (o la confabulación de *B* y *M* en el depósito de la moneda) no puede revelar en que lugar *C* ha utilizado la moneda (como es verá posteriormente), entonces los pagos del usuario son anónimos y no rastreables.

Resumiendo, este protocolo de retirada proporciona monedas no anónimas (el anonimato se consigue en etapas posteriores), con seguridad contra la falsificación de monedas, autenticación del propietario de la cuenta y con la existencia de una prueba secreta de validez (conocida sólo por *C*) que será requerida para el uso de la moneda.

3.2 Subprotocolo de fines múltiples aplicado a pre-pago

El primer paso para la realización de un pago de *C* a *M* es la ejecución del subprotocolo de fines múltiples.

Tabla 3: Subprotocolo multipropósito aplicado a pre-pago.

1a. C→M:	**Solicitud**
1b. M:	**W_{1m}, $W_{0m} = h(W_{1m})$**
1c. M→C:	**W_{0m}, $S_m(W_{0m})$, Q_2**
1d. C:	**Verificación de la firma**
2. C→B:	**$E_{Kpb}(K)$, $E_K(M_1, W_{0m}, W_{0x})$**
3. B:	**Comprobación de reutilización**
4. B→C:	**Ack = $S_b(W_0, W_{0m}, W_{0x})$ o Nack = $S_b(W_1)$**
5. C→B:	**$E_K(Q_2, W_1)$**
6. B:	**$W_0 \stackrel{?}{=} h(W_1)$, $Q_3=Q-Q_2$, $M_2 = \{Q_2, W_{0m}, K_{sb}(Q_2,W_{0m})\}$, $M_3 = \{Q_3,W_{0x}, K_{sb}(Q_3,W_{0x})\}$**
7. B→C:	**$E_K(M_2, M_3)$**

Por cada petición de compra, *M* genera una nueva prueba secreta W_{1m} y calcula W_{0m}, el identificador para un pago futuro. Este paso del *MPP* define su funcionalidad (en este caso pre-pago).

Para la preparación de la moneda, *C* envía al banco la moneda retirada cifrada con una clave simétrica de sesión. El banco calcula W_0 y explora la lista de monedas gastadas. Si W_0 aparece en la lista, el banco descubre que alguien intenta utilizar una moneda por segunda vez (reutilizar). Estos intentos de reutilización se previenen abortando la operación.

C muestra la prueba secreta (W_1) de la moneda M_1 y solicita la creación de una moneda nueva con el identificador dado por *M* (W_{0m}). *C* puede elegir entre usar la fracción restante para la creación de otra moneda para él mismo o para la preparación de otro pago. *B* no puede averiguar la identidad del creador o creadores de los identificadores nuevos que serán los receptores finales de las monedas. No obstante, inicialmente una confabulación entre *C* y *B* seria capaz de rastrear *M* ya que *C* puede mostrar a *B* el identificador, mientras que la identidad de *M* puede ser extraída del depósito. Para solucionar este problema, la aplicación de auto-transferencia del *MPP* se puede utilizar para que *M* elimine la posibilidad de rastreo.

3.3 Subprotocolo de pago

Durante el subprotocolo de pago, un mensaje único entre *C y M* se utiliza para transferir la moneda. La captura del pago se realiza off-line, sin la mediación del banco. En este momento, *C* conoce la moneda M_2 y su identificador W_{0m}, pero *C* no conoce la prueba secreta de validez W_{1m}. Sólo *M* conoce W_{1m} y por esta razón *M* puede estar seguro de que la moneda no se ha utilizado previamente en otro pago antes de la ejecución de este. No hay ninguna necesidad de ponerse en contacto con el banco durante el pago para evitar la reutilización: esta no es posible.

Tabla 4. Subprotocolo de pago

1. C → M:	**M_2**
2. M:	**Q_2, $W_{0m} \stackrel{?}{=} K_{pb}(M_2)$,** **Comprobación de reutilización en la lista de M**

3.4 Subprotocolo de fines múltiples aplicado a auto-transferencia

Una vez *M* ha recibido la moneda, puede elegir entre:

a) depositarla identificando su cuenta,

b) Cambiar la moneda gastada por una nueva para él mismo, con una auto-transferencia sin identificación, o

c) Usar la moneda para preparar un pago nuevo a otro comerciante de manera anónima utilizando el *MPP* para una operación de pre-pago.

M no puede distinguir una moneda recibida de una obtenida con el protocolo de retirada, pero hay una diferencia: *C* conoce la identidad de *M* y el identificador de la moneda. Por esta razón, los pagos podrían ser rastreados por la confabulación de *C* y *B* si las monedas fuesen usadas directamente. La operación de auto-transferencia es otra función del *MPP* que podrá ser utilizada para eliminar la posibilidad de rastreo.

Tabla 5: Subprotocolo multipropósito aplicado a auto transferencia.

1. M:	**W_{1m}', W_{1m}'', Q_3, W_{0m}'=h(W_{1m}'), W_{0m}''= h(W_{1m}'')**
2. M → B:	**E_{Kpb}(K), E_K(M_2, W_{0m}', W_{0m}'')**
3. B:	**Comprobación de la reutilización**
4. B → M:	**Ack o Nack**
5. M → B:	**E_K(Q_3, W_{1m})**
6. B:	**$W_{0m} \stackrel{?}{=} h(W_{1m})$,** **$M_3$={$Q_3$, W_{0m}', K_{sb}(Q_3, W_{0m}')},** **M_4= {Q_4, W_{0m}'', K_{sb}(Q_4, W_{0m}'')}**
7. B → M:	**E_K(M_3, M_4)**

En este paso *B* no conoce quien es el usuario que autotransfiere la moneda. Además, *B* no puede distinguir si el usuario se autotransfiere la cantidad total de la moneda, o si prepara un pago con una fracción de la moneda y se autotransfiere la parte restante, o si prepara dos pagos.

3.5 Subprotocolo de depósito

El protocolo de depósito es similar al protocolo de auto-transferencia con la diferencia de la identificación de la cuenta de *M* y sin la exigencia de un identificador nuevo.

Tabla 6. Subprotocolo de depósito.

1. M → B:	**M_2**
2. B:	**Comprobación de reutilización**
3. B → M:	**Ack o Nack**
4. M → B:	**E_{Kpb}(W_{1m}, M_{id})**
5. B:	**$W_{0m} \stackrel{?}{=} h(W_{1m})$, Crédito**

3.6 Conclusión

El protocolo de pago de cantidades elevadas presentado en [5] satisface los requisitos de seguridad necesarios para el pago de cantidades elevadas además de ofrecer anonimato.

Sin embargo, la realización de un pago representa un intercambio, que podría interrumpirse por diferentes causas, provocando una situación no equitativa entre las partes. La incorporación de atomicidad al protocolo de pago puede realizarse utilizando un protocolo de intercambio equitativo. Este protocolo se describe en la sección siguiente.

4 Descripción del Protocolo de Intercambio

El protocolo está formado por un subprotocolo de intercambio que se ejecutará en cada operación de compra que requiera el pago de cantidades elevadas. Este subprotocolo no requiere la intervención de la tercera parte y permite la finalización de la operación de compra.

Cuando el subprotocolo anterior no se finaliza, las partes pueden solicitar la finalización del intercambio poniéndose en contacto con la tercera parte y ejecutando el subprotocolo de finalización. A diferencia de los protocolos de intercambio presentados anteriormente, este protocolo no requiere un subprotocolo de cancelación debido a las características del sistema de pago utilizado. Gracias a que el protocolo descrito en [5] no permite reutilizar monedas, no será necesaria una etapa reto-respuesta (que se utiliza para identificar a los reutilizadores), y por lo tanto el intercambio se simplifica respecto a protocolos de intercambio adecuados para este tipo de sistemas de pago y como consecuencia se podrá reducir el número de pasos del intercambio.

4.1 Subprotocolo de intercambio

El subprotocolo de intercambio consta de un paso inicial que expresa la intención de compra e indica el producto seleccionado.

Tabla 7. Intercambio para pagos de cantidades elevadas: subprotocolo de intercambio.

0. C→M:	**Solicitud**	**Paso 1a del subprotocolo de pre-pago.**
1. M→C:	**W_{0m}, $S_m(W_{0m})$, Q_2, Id_M c=E_k(producto/recibo), K_t=K_{Pt}(k), H_M=K_{Sm}{H[H(c), k_T, W_{0m}], Id}**	**Paso 1c del subprotocolo de pre-pago modificado**
2. C→M:	**M_2**	**Paso 1 del sub. de pago.**
3. M→C:	**K**	**Paso nuevo. Equivalente a un tercer paso del sub. de pago**

A continuación se realiza el intercambio mediante tres pasos o transferencias. Estos pasos son los siguientes:

- Paso 1: Envío del recibo o producto cifrado del vendedor al comprador.
- Paso 2: Envío de la moneda del comprador al vendedor.
- Paso 3: Envío de la clave del vendedor al comprador.

La moneda utilizada en el pago es la moneda descrita en el apartado 2 sin ninguna modificación, es decir, *moneda = {Q_1, W_0, $K_{sb}(Q_1, W_0)$, W_1}*, dónde W_0 es el identificador de la moneda y W_1 es la prueba secreta conocida sólo por el receptor. Los pasos del subprotocolo de intercambio se corresponden con las transferencias de información entre *C* y *M* en los subprotocolos de pre-pago y pago del protocolo descrito en [5]. Las correspondencias se muestran en la tercera columna de la tabla 7.

Con la modificación del paso 1c del subprotocolo de pre-pago y la incorporación de un tercer paso en el subprotocolo de pago, el protocolo completo de intercambio y pago es el siguiente:

Tabla 8. Protocolo completo de intercambio para el pago de cantidades elevadas.

1a. C→M:	**Solicitud**
1b. M:	**W_{1m}, W_{0m} = h(W_{1m})**
1c. M→C:	**W_{0m}, $S_m(W_{0m})$, Q_2, c=E_k(producto/recibo), K_t=K_{Pt}(k), H_M=K_{Sm}{H[H(c), k_T, W_{0m}], Id}**
1d. C:	**Verificación de la firma**
2. C→B:	**$K_{pb}(K_1)$, $E_{K1}(M_1, W_{0m}, W_{0x})$**
3. B:	**Comprobación de reutilización**
4. B→C:	**Ack = $S_b(W_0, W_{0m}, W_{0x})$ o Nack = $S_b(W_1)$**
5. C→B:	**$E_K(Q_2, W_1)$**
6. B:	**$W_0 \stackrel{?}{=} h(W_1)$, M_2={Q_2,W_{0m}, $K_{sb}(Q_2,W_{0m})$}, M_3 ={Q_3,W_{0x}, $K_{sb}(Q_3,W_{0x})$}**
7. B→C:	**$E_K(M_2, M_3)$**
8. C→M:	**M_2**
9. M:	**Q_2,$W_{0m} \stackrel{?}{=} K_{pb}(M_2)$, Comprobación de reutilización en la lista de M**
10. M→C:	**K**

4.2 Subprotocolo de finalización

Si el subprotocolo de intercambio no se finaliza, entonces es posible que una de las partes se vea perjudicada respeto a la otra. La interrupción del intercambio puede ser debida a un comportamiento incorrecto de una de las partes, que podría intentar impedir que la otra parte consiguiera retornar el intercambio a una situación equitativa.

En este protocolo la interrupción puede darse tras la recepción del mensaje del paso 1 o del mensaje del paso 2. En el primer caso, ninguna de las partes tiene el elemento deseado, y no se han intercambiado elementos que puedan comprometer a ninguna de las partes, por lo tanto no es necesario utilizar un subprotocolo de cancelación. El subprotocolo de finalización permitirá obtener los elementos deseados poniéndose en contacto con la tercera parte. Este subprotocolo puede ser ejecutado por las dos partes.

En el subprotocolo de finalización se utilizan dos variables booleanas, *finalizado* y *probado*, que indican sí *C* ha proporcionado la moneda y como consecuencia ha recibido la clave, y que *M* ha proporcionado la prueba secreta, respectivamente. El valor por defecto de las dos variables es falso.

Tabla 9. Intercambio para pagos de cantidades elevadas: subprotocolo de finalización de M.

1. M → T:	**Solicitud, W_{0m},** **E_k(producto/recibo), k_T, h_M**
2. T:	**IF (finalizado=cierto)** **T→M: M_2** **ELSE** **T→M: Solicitud de prueba** **M→T: W_{1m}** **T: probado = cierto**

M ejecutará el protocolo si no recibe el mensaje del paso 2. Esto puede ser debido a que *C* no haya enviado el mensaje o bien a qué este se haya perdido, y aunque fue enviado no ha llegado a su destino. En cualquiera caso *C* tampoco recibirá el mensaje del paso 3, por lo tanto en esta situación las dos partes pueden ejecutar el subprotocolo de finalización.

Tabla 10. Intercambio para pagos de cantidades elevadas: subprotocolo de finalización de C.

1. C → T:	**$S_m(W_{0m})$, Q_2, M_2,** **E_k(producto / recibo), k_T, h_M**
2. T→ C:	**K**
3. T:	**finalizado=cierto** **IF (probado = cierto)** **T→B: M_2, W_{1m},** **Solicitud depósito (Id_M)**

Si *C* no recibe el mensaje del paso 3 cuando *M* ya ha recibido el mensaje del paso 2, entonces *M* ya dispone de todos los elementos y *C* puede ejecutar el subprotocolo de finalización para obtener la clave.

En el subprotocolo de finalización de *C*, si la variable probado vale cierto (*T* dispone de la clave secreta de la moneda), *T* puede solicitar al banco que la moneda sea depositada en la cuenta de *M*, finalizando el intercambio.

5 Evaluación del Protocolo

La realización de la compra utilizando el protocolo de intercambio para pagos de cantidades elevadas se puede concluir utilizando únicamente el subprotocolo de intercambio o utilizando también el subprotocolo de finalización. Las situaciones en las que se puede encontrar el intercambio son las siguientes:

- C no envía el mensaje del paso 2

  Si una vez enviada la solicitud el cliente decide no realizar la compra, no solicitará la moneda al banco para realizar el pago. Como consecuencia tampoco ejecutará el subprotocolo de finalización.

  Sí *M* solicita finalización, se comprobará que *C* no ha solicitado finalización (finalizado = falso) y no se le proporcionará la moneda.

- C envía el mensaje del paso 2 y *M* no lo recibe.

  En esta situación *C* espera recibir la clave (tiene la moneda correspondiente) y *M* espera recibir la moneda. Por lo tanto las dos partes pueden solicitar finalización. Estas solicitudes se pueden realizar en diferente secuencia.

  - C finaliza, *M* finaliza

    C envía la moneda a la tercera parte y recibe la clave, entonces la variable finalizado pasa a valer cierto. *M* recibe la moneda.

  - M finaliza, *C* finaliza

    M solicita finalización, y la variable finalizado vale falso, por lo que no se le proporcionará la moneda. En cambio, *M* recibirá una solicitud de depósito de la prueba secreta de la moneda, que será útil si a continuación *C* solicita finalización.

    Cuando *C* solicita finalización encontrará la variable *probado = cierto* y recibirá la clave al tiempo que se deposita en la cuenta de *M* la moneda revelada en la solicitud de finalización de *C* .

- C envía el mensaje del paso 2, *M* lo recibe y miente. (No envía el mensaje del paso 3).

M no puede impedir que C, una vez haya creado la moneda relacionada con la compra, obtenga la clave siempre y cuando proporcione esta a la tercera parte. Puesto que M, en este caso, dispone del pago, esta situación es equitativa.

- M envía el mensaje del paso 3 y *C* no lo recibe.

Este caso es equivalente al anterior, puesto que *C* puede obtener la clave sin ningún impedimento si proporciona la moneda.

6 Conclusiones

Los pagos que representen la transferencia de cantidades elevadas requieren grandes medidas de seguridad. La compra de bienes o servicios de coste elevado supone un intercambio entre un producto o recibo y el correspondiente pago. No tan sólo es necesario que, mediante un sistema de pago seguro se garantice que la moneda utilizada sea válida, no utilizada ni robada, sino que también se tiene que garantizar que la moneda llegará al vendedor si este proporciona el producto. Del mismo modo, desde el punto de vista del usuario pagador, es muy importante que la compra represente un intercambio equitativo, puesto que el comprador no se sentirá seguro, y por lo tanto no utilizará una moneda de gran valor si no se puede asegurar de que recibirá el producto comprado. Por estos motivos se puede asegurar que la necesidad de atomicidad es aún mayor cuando en la compra intervienen cantidades de dinero elevadas.

Los aspectos de seguridad relacionados con la moneda se pueden satisfacer mediante la utilización de un sistema de pago adecuado para cantidades elevadas, como el protocolo FAEC-HQ.

La compra se podrá realizar utilizando el subprotocolo de intercambio, que consiste en la secuencia de pasos del pago con algunos elementos adicionales. Este intercambio representa una solicitud de compra más tres transferencias entre vendedor y comprador.

Un subprotocolo de finalización, ejecutado entre una de las partes y la tercera parte de confianza se utilizará cuando el intercambio conduzca a una situación no equitativa. Debido a que el número de pasos del intercambio es reducido, las posibilidades de interrupción en diferentes puntos son también reducidas. En todas estas situaciones, el uso del subprotocolo de finalización permite llegar a una situación equitativa.

Referencias

[1] Adi, K., Debbadi, M. and Mejri, M.: "A new logic for electronic commerce protocolos.", AMAST'00, LNCS 1816, páginas 499-513, Springer Verlag, 2000.

[2] Asokan, N., Herreweghen, E. Van and Steiner, M.: "Towards a framework for handling disputes in payment systems", 3rd Usenix workshop on electronic commerce, páginas 187-202, 1998.

[3] Camp, J.,Harkavy, M., Tygar, J.D. and Yee, B.: "Anonymous atomic transactions", 2nd USENIX workshop on electronic commerce, páginas 123-133, 1996.

[4] Jakobsson, M.: "Ripping coins for a fair exchange", Eurocrypt'95, LNCS 921, páginas 220-230, Springer Verlag, 1995.

[5] Payeras, M., Ferrer, J.L. and Huguet, L.: "A fully anonymous electronic payment scheme for B2B", To appear in ICWE'03 Proceedings.

[6] Pagnia, H. and Jansen, R.: "Towards multiple payment schemes for digital money", Financial Cryptography' 97, LNCS 1318, páginas 203-216, Springer Verlag, 1997.

[7] Schuldt, H., Popovivi, A. and Schek, H.: "Give me all I pay for – Execution guarantees in electronic commerce payment processes", Informatik'99 –Workshop "Unternehmensweite und unternehmensübergreifende Workflows: Konzepte, Systeme, Anwendungen", 1999.

[8] Schuldt, H., Popovivi, A. and Schek, H.: "Execution guarantees in electronic commerce payments.", 8th international workshop on foundations of models and languages for data and objects (TDD'99), LNCS 1773, Springer Verlag, 1999.

[9] Su, J. and Tygar, J.D.: "Building blocs for atomicity in electronic commerce", 6th USENIX security symposium, 1996.

[10] Tang, L.: "Verifiable transaction atomicity for electronic payment protocols", IEEE ICDCS'96, páginas 261-269, 1996.

[11] Tygar, J.D.: "Atomicity in electronic commerce", 15th annual ACM symposium on principles of distributed computing", páginas 8-26, 1996.

[12] Vogt, H., Pagnia, H. and Gärtner, F.C.: "Modular fair exchange protocols for electronic commerce" 15th Annual Computer Security Applications Conference, ACSAC'99, páginas 3-11, 1999.

[13] Xu, S., Yung, M., Zhang, G. and Zhu, H. "Money conservation via atomicity in fair off-line e-cash", International security workshop ISW'99, LNCS 1729, páginas 14-31, Springer Verlag, 1999.

Sistema integral de pago telemático para entornos de comercio electrónico

Juan José Unzilla, Jon Matías, Eduardo Jacob, Mariví Higuero, Cristina Perfecto, Puri Saiz
Departamento de Electrónica y Telecomunicaciones. Area de Ingeniería Telemática.
Escuela Superior de Ingenieros. Universidad del País Vasco – Euskal Herriko Unibertsitatea
Alda. Urquijo s/n - 48013 - Bilbao
Teléfono: 94 601 4035 - Fax: 94 601 4259
E-mail: {jtpungaj, jtamafrj, jtpjatae, jtphiapm, jtppeamc, jtpsaagp}@bi.ehu.es

Abstract. *Nowadays, Internet is accepted by the international business community as a viable medium for doing business, despite the absence of security mechanisms. However the new cryptographic techniques are bringing security and anonymity to the electronic transactions. These cryptographic techniques allow to develop fully-anonymous and secure electronic payment systems. This paper presents a payment system that offers a general solution to the most usual scenarios in electronic commerce. In this approach, credit card, micropayment and electronic coins are available in an integrated system based on web application platforms, Java, XML and free-license programs.*

1 Introducción

Este artículo recoge el trabajo realizado en el grupo de Ingeniería Telemática (GIT) relacionado con los sistemas de pago telemático, partiendo de los desarrollos realizados en el campo del dinero no trazable [1].

El sistema desarrollado pretende dar una solución integral al pago electrónico, de forma que sea seguro y escalable. Es evidente que la seguridad y la garantía de que las operaciones de pago son las que las partes acuerdan representan un elemento determinante, tanto en las contrataciones con presencia física como en la que se realizan en los entornos de comercio electrónico. Otro aspecto importante es la falta de conocimientos informáticos y de seguridad del usuario potencial, lo que exige una elevada automatización de las operaciones de pago a la vez que la claridad suficiente para conseguir la confianza de los usuarios. Para que el uso de sistemas de pago telemático se generalice, es preciso desarrollar sistemas flexibles y sencillos de usar que garanticen la confidencialidad de los datos enviados por los usuarios a un coste bajo (al menos inferior al de los sistemas tradicionales).

En la actualidad, la forma de pago más utilizada en el comercio minorista (B2C) se basa en el intercambio de información crítica de usuario (como el número de su tarjeta de crédito) a través de un canal de comunicación seguro (empleando SSL) [2]. El principal problema de este sistema es que se está facilitando el número de tarjeta de crédito a un tercero, lo que requiere bien un conocimiento previo que ofrezca confianza o un sistema de pago con garantías suficientes para las partes.

Existen un gran número de protocolos y sistemas de pago [3] [4], la mayoría de ellos diseñados con el objetivo de responder a un determinado escenario. En el sistema que se presenta se han integrado tres modalidades de pago complementarias que permiten responder a las necesidades de pago de la mayoría de los usuarios, mediante el uso exclusivo de software libre.

1.1 Objetivos y motivación del trabajo

El objetivo principal del sistema es crear una herramienta de pago que sea genérica y que permita a un vendedor ofrecer diferentes escenarios de pago según cual sea el caso. Además, el sistema deberá de ser seguro y permitir opcionalmente el pago anónimo. Seguro en el sentido de que proporcione un medio confidencial y que asegure la integridad de las transacciones. En el caso de que sea anónimo deberá permitir que nadie conozca la identidad del comprador, qué ha comprado, dónde o cuándo.

El escenario de aplicación del sistema surge dentro del proyecto NOTACON [5], y se orienta fundamentalmente al comercio de contenidos o de material digital. Así se diseña una herramienta de pago electrónico que debe satisfacer los siguientes requerimientos:

- Debe garantizar la seguridad:
  - Debe resultar imposible que nadie excepto el comprador cargue pagos a su cuenta.
  - Tanto el comprador como el vendedor, tienen que contar con medios necesarios para probar ante terceros que el pago se realizó con éxito.
  - El sistema como un todo debe ser resistente al fraude.
- Proporcionar privacidad a las transacciones:
  - Debe impedirse que observadores externos puedan obtener algún tipo de

información útil a través de las actividades de los usuarios.

- Debe primar la sencillez en el diseño, para poder obtener un sistema altamente escalable y con elevada eficiencia computacional.
- Se primará la sencillez de uso, de instalación y de configuración.
- Se deberá permitir el pago anónimo, al igual que se deberá de poder tener una factura donde se pueda demostrar la autoría de un pago. Se deja en manos del usuario la posibilidad de mostrar o no su identidad.

2 Visión general del sistema

El sistema ha sido desarrollado en lenguaje Java, buscando una herramienta de pago que sea portable y multiplataforma, y para su realización se ha partido de trabajos previos del grupo [1] como ya se ha indicado. La utilización de las implementaciones actuales requiere el registro previo en cada uno de los sistemas y, antes de realizar el pago, seleccionar la forma en que se desea pagar. En el caso de pagos anónimos, ésta es una opción ofrecida por poquísimos vendedores. Con la solución adoptada, el escenario de pago se decide al realizar el desembolso y se asegura que el vendedor acepta todas las formas de pago al ser una herramienta integral.

Otro valor añadido del sistema es que se ha diseñado de forma modular, lo que permite su ampliación o adaptación a nuevos requerimientos. En el diseño actual se ha optado por analizar las características de cada uno de los sistemas de pago más empleados en cada escenario, y adaptarlas a las necesidades del proyecto.

La arquitectura del sistema esta formada por tres entidades:

- Cliente: sujeto dispuesto a pagar para obtener un objeto, contenido digital o servicio.
- Vendedor: entidad que posee aquello que desea el cliente y que cobrará por transferírselo.
- Banco: encargado de controlar la validez de las transferencias. Tanto el cliente como el vendedor confiarán en él.

En el cliente se trabaja mediante applets firmados, con el objetivo de facilitar las tareas de actualización y minimizar las labores de configuración en el mismo, ofreciendo a la vez garantías de seguridad. Para ello deberá de disponer de un browser con una máquina virtual Java.

En el caso del vendedor se hace uso de servlets para dotarlo de inteligencia y controlar la secuencia de instrucciones. Se emplean páginas jsp para facilitar la presentación al cliente y para el acceso a base de datos se utilizará JDBC. Como Sistema Gestor de Bases de Datos se ha utilizado MySQL [6]. Para poder responder a las peticiones de páginas que le haga el cliente, el vendedor debe disponer un servidor Web y un motor de servlets y páginas jsp (en este caso Apache [7] con TomCat [8]).

De forma similar al vendedor, el subsistema que se implementa en el banco está basado en servlets. Este servidor Web debe atender las peticiones del cliente para conseguir monedas, para cambiar configuraciones o para consultar datos. Al igual que en subsistema del vendedor para el acceso a la base de datos se utiliza JDBC y como Sistema Gestor de Bases de Datos se usa también MySQL.

En la Fig. 1 se muestra la arquitectura general del sistema, donde pueden verse los tres subsistemas y los componentes software principales de cada uno de ellos.

3 Pago mediante tarjeta

La solución consta de tres entidades: un cliente, un vendedor y un banco. Cada uno de ellos debe cumplir los siguiente requisitos:

- Cliente: debe tener una cuenta en el banco y adquirir una tarjeta de crédito o de débito, y para poder realizar los pagos necesitará fijar una clave de seguridad para accesos a través de Internet. Deberá acceder al certificado del banco para poder obtener su clave pública ($K_{PUBBANCO}$).
- Vendedor: debe poseer una cuenta en el banco a la que se cargarán los pagos que realicen los clientes. Debe poseer identidad digital avalada por un certificado firmado por una Autoridad de Certificación *de la confianza del cliente*.
- Banco: de la misma forma que el vendedor, debe poseer una identidad digital en forma de certificado firmado por una Autoridad de Certificación *de la confianza del cliente.*

El sistema requiere que todas las partes involucradas tengan capacidad de procesamiento, lo que se consigue en base a applets en la parte del cliente y en base a servlets tanto en el vendedor como en el banco.

En el caso de pago con tarjeta se lleva un control de las cuentas de clientes y vendedores por medio de sendas bases de datos en el banco. En estas bases se guardará una clave asociada a cada cuenta, cuyo conocimiento sólo conocerá su propietario, que avalará el uso de la misma para realizar un pago, presuponiendo de esta forma que estará autorizada por el titular. De cara a posibles reclamaciones posteriores se llevará el control de todas las transacciones realizadas en otra base de datos situada en el banco.

En la Fig. 2 se muestra la arquitectura para el soporte de pago con tarjeta. El proceso se describe, de forma resumida, en la Fig. 3.

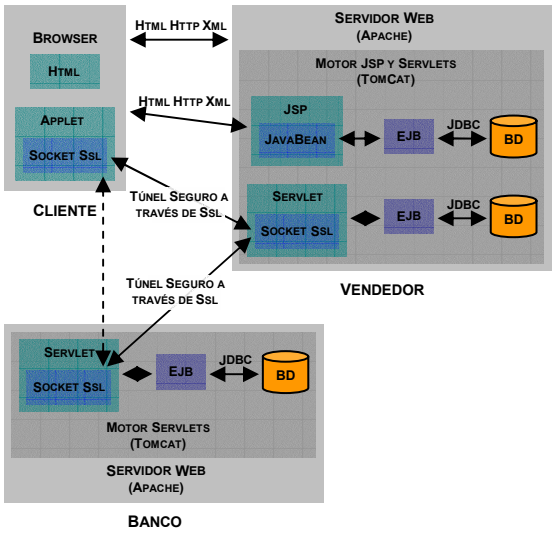


Fig. 1: Arquitectura general del sistema

4 Pago mediante dinero electrónico

Para el pago mediante dinero electrónico el sistema implementa dos tareas principales, la consecución de las monedas y la realización de un pago con posibilidad de que haya devolución. Al igual que en el pago a través de tarjeta, la solución constará de tres entidades: un cliente, un vendedor y un banco. Los requisitos que cada entidad tendrá que cumplir son los mismos que en el caso anterior. Todas las entidades precisan de capacidad de procesamiento, utilizándose applets firmados en el cliente y servlets tanto en el vendedor como en el banco.

Para conseguir la no trazabilidad de las monedas, se hace uso de la técnica de firma ciega [9]. Esta técnica consiste en obtener una firma de una entidad sin que ésta conozca el contenido de lo firmado. Para conseguir esto, el poseedor de la información que pretende firme la entidad, aleatoriza esta información antes de enviársela y desaleatoriza la firma que la entidad le envía.

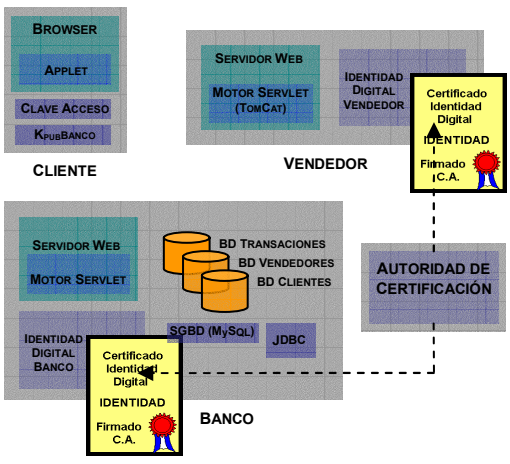


Fig. 2: Arquitectura para el pago con tarjeta.

En este caso el banco debe tener una base de datos con todas las monedas que ha generado para poner en circulación. Para llevar el control de si estas monedas están en posesión del banco o de si por el contrario las tiene un usuario, habrá que crear una nueva base de datos. En esta base de datos de control, también se vigilará si la moneda es válida o esta pendiente de actualización. Como en cualquier banco también habrá que llevar el control de las cuentas de clientes y vendedores por medio de sendas bases de datos situadas en el banco. También habrá que tener una base de datos para anotar los números de secuencia de los tickets se hayan amortizado por monedas.

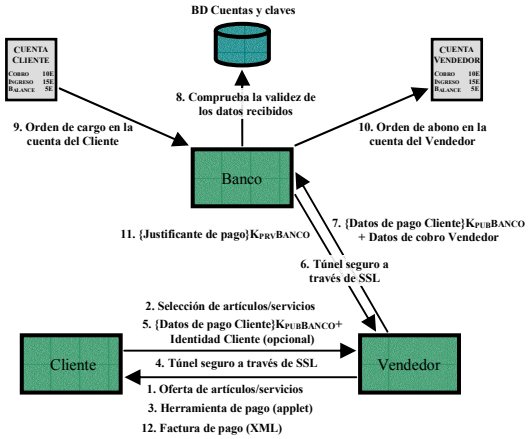


Fig. 3: Proceso de pago mediante tarjeta.

En la Fig. 4 se muestran gráficamente los elementos que componen cada una de las entidades del sistema para el pago mediante moneda electrónica.

Las monedas son documentos XML que tienen los siguientes campos:

- Número de Identificación o de Serie: número único que identifica unívocamente a una moneda. Es asignado por el banco.
- Valor: es el valor nominal, entre un conjunto finito de ellos, asociado a cada moneda por el banco.
- Número de Secuencia: garantiza que la misma moneda no pueda ser gastada más de una vez, ya que no pueden existir dos monedas válidas iguales. Al cambiar de manos la moneda se incrementará en una unidad este número, haciendo que sólo el receptor del pago conozca la nueva moneda válida. El banco deberá de actualizar su base de datos de monedas para registrar esta nueva moneda.
- Emisor: identidad del banco expendedor de la moneda.
- Fecha: fecha de creación de la moneda.
- Firma: el banco concatena todos los datos anteriores y los firma con la clave privada asociada al valor nominal de la moneda en cuestión (existe una K_{PRV} por cada valor nominal posible). De esta forma cualquiera que tenga el certificado asociado a cada valor nominal podrá

comprobar el valor de la moneda y que realmente fue firmado por el banco, pero nunca podrá asegurar la vigencia de esta moneda. El único que podrá asegurar su validez será el banco.

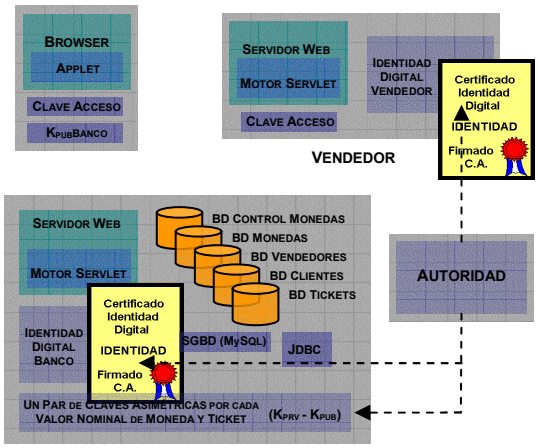


Fig. 4: Arquitectura para el pago con moneda electrónica.

Cuando alguien realiza un pago y envía unas monedas, el receptor de éstas debe comprobar si son válidas, para lo cual las envía al banco. Este confirma la vigencia de las monedas y en caso de que sean válidas devuelve al receptor del pago las nuevas monedas que sólo este último conocerá. La nueva moneda se obtendrá de incrementar en una unidad el número de secuencia, de poner la fecha correspondiente al día en que se genere esa nueva moneda y de obtener la nueva firma con la clave correspondiente. Una vez hecho esto se actualizará la base de datos de monedas con estos nuevos valores.

Mientras se produce esta actualización, la base de datos que lleva el control de las monedas mantendrá como no válida la moneda correspondiente a ese número de serie. Con ello se consigue que el individuo que realiza el pago no conozca las nuevas monedas válidas asociadas a los números de secuencia que utilizó. Será el receptor del pago el único que conozca estas nuevas monedas asociadas a esos números de serie. De esta forma se resuelve el problema del double spending (gastar dos veces la misma moneda) y se permite que el receptor de la moneda no tenga que amortizar inmediatamente el valor de la moneda.

4.1 Procedimiento para la consecución de monedas de forma anónima

La solución adoptada para la generación de monedas se basa en eCash [10]. Mediante el uso de técnicas de firma ciega el banco firma unos tickets de valores predefinidos que previamente han sido enmascarados por el cliente y deduce el importe correspondiente de la cuenta o lo carga en la tarjeta de crédito. Los tickets son documentos XML con los siguientes campos:

- Número de Serie: número aleatorio único propuesto por el usuario suficientemente largo para que la probabilidad de repetición sea prácticamente cero. Cuando un ticket sea amortizado por monedas, el banco apuntará este número de serie evitando que pueda ser amortizado de nuevo. Este número se aleatoriza antes de enviarlo al banco para que este lo firme de forma ciega.
- Valor: valor nominal, entre un conjunto finito de ellos, asociado al ticket que se descontará al usuario que lo crea.
- Firma: el banco firma (de forma ciega) el número de secuencia con una clave privada correspondiente al valor del ticket.

Para evitar que se usen tickets ya gastados, el banco mantendrá una base de datos con el número de serie de los tickets ya amortizados por monedas. Se limita el tamaño de la base de datos de tickets cambiando las claves de cifrado de los tickets cada 1000 tickets expendidos. Una vez que reciban todos los números de secuencia de los 1000 tickets expedidos con las claves correspondientes, se podrán eliminar o guardar para realizar estadísticas de uso en un sistema de almacenamiento alternativo. Pueden coexistir más de una tabla al mismo tiempo mientras que se recuperan todos los números de serie de cada una.

El proceso de obtención de monedas requiere previamente la obtención de ticket. La Fig. 5 muestra el proceso a seguir para ello. Los tickets válidos se pueden cambiar con posterioridad por monedas que no se podrán asociar a ninguna identidad. El procedimiento a seguir para el canje de tickets por monedas se muestra en la Fig. 6.

4.2 Procedimiento para la consecución de monedas de forma no anónima

Si no se exige anonimato, el procedimiento es más sencillo que el anterior. En cualquier caso, una vez que entre en juego el dinero electrónico y cambie de manos, será imposible seguir su rastro salvo que el receptor de estas monedas se identifique. Las monedas involucradas en una posible devolución por parte del vendedor al cliente siguen guardando el anonimato debido a que no se conoce la identidad del cliente que las recibe.

La consecución de monedas de forma no anónima puede ser utilizada tanto por vendedores como por clientes. El procedimiento se muestra gráficamente en la Fig. 7.

4.3 Procedimiento de pago con posibilidad de devolución

Una vez que tanto vendedores como clientes poseen monedas digitales, se está en disposición de realizar pagos y de poder recibir una devolución de monedas en caso de que el importe de pago no sea exacto.

Si por algún tipo de requisito legal el receptor de un pago tiene que identificarse, cuando se desee que el banco devuelva monedas nuevas, se tendrá que aportar una identidad a la hora de comprobar la validez de las mismas. En este caso, el último paso del procedimiento exige identificar al cliente.

En la Fig. 8 se muestra el proceso de pago con devolución. Si el vendedor quiere amortizar directamente las monedas que recibe del cliente en forma de pago, tendrá que enviar al banco el número de cuenta junto con el documento XML con las monedas. El banco se encargará de comprobar la validez y vigencia de las monedas y si todo es correcto, incrementará la cuenta del vendedor con el valor correspondiente al total de las monedas.

Tras esto, el banco señalará como no válidos (pendientes de actualización) los números de serie de las monedas amortizadas, esto se llevará a cabo en la base de datos que se encarga del control de monedas. En la misma base de datos y para los mismos números de serie se cambiará la posesión de las monedas poniéndolas como propiedad del banco. De forma similar a la descrita, cualquier usuario que tenga monedas electrónicas podrá amortizar su valor en cualquier momento.

5 Sistema de micropago

Finalmente se expondrá la solución adoptada para el sistema de micropago. Las entidades que entran en juego son tres, al igual que en los otros dos sistemas analizados: un cliente, un vendedor y un banco.

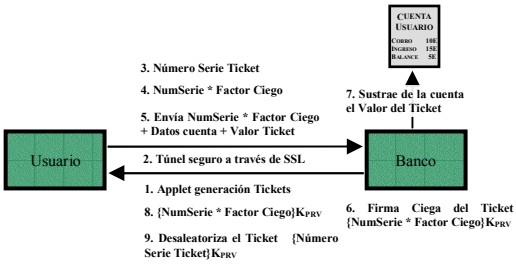


Fig. 5: Proceso de obtención de tickets.

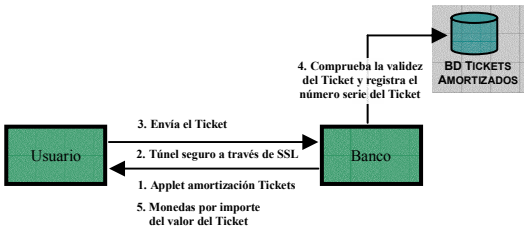


Fig. 6: Proceso de canje de tickets por monedas.

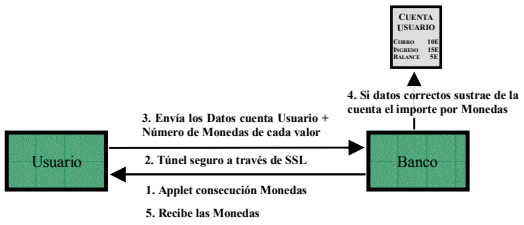


Fig. 7: Proceso de obtención de monedas de forma no anónima.

- Cliente: tendrá una cuenta en el banco o poseerá monedas electrónicas para poder conseguir la calderilla que da acceso al sistema de micropago.
- Vendedor: deberá de tener obligatoriamente una cuenta en el banco a la que cargará el importe que los clientes le paguen por la calderilla. Será el propio vendedor el que controle la validez de la misma y a la hora de utilizar micropago como tal, no será necesario contactar con el banco.
- Banco: únicamente será necesario para garantizar al vendedor el cobro de los importes correspondientes a la calderilla que éste venda.

Todas las entidades deben poseer capacidad de procesamiento. Se hace uso de applets firmados en el cliente y tanto el vendedor como el banco se basan en servlets.

El banco lleva el control de las cuentas de clientes y vendedores por medio de sendas bases de datos situadas en el banco. En la modalidad de micropago será el vendedor el encargado de comprobar la validez de la calderilla, por lo tanto debe haber en el vendedor una base de datos con toda la calderilla que hay en juego. Éste también tendrá un registro con todas las claves simétricas utilizadas en la generación de calderilla. Estas claves se renovarán mensualmente.

En la Fig. 9 se muestra de forma gráfica los elementos necesarios que debe tener cada entidad del sistema para que pueda funcionar el micropago. La calderilla es un documento XML con los siguientes campos:

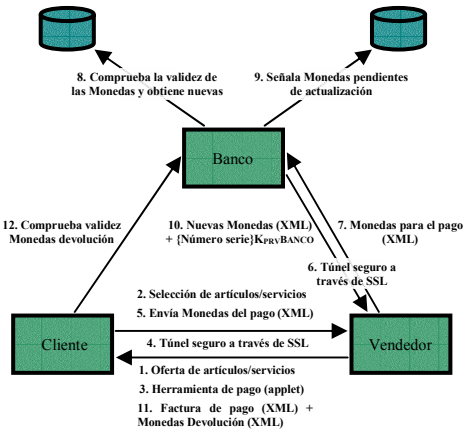


Fig. 8: Proceso de pago con devolución.

- Vendedor: entidad que genera la calderilla y única capaz de validarla.
- Número de Identificación o de Serie: número único que identifica unívocamente a cada unidad. Es el vendedor el encargado de generar de forma centralizada estas referencias y de validar la calderilla asociada a cada número de serie.
- Valor: es el valor que tiene la calderilla y que, por lo tanto, se pueda gastar en el vendedor. Podrá tendrá cualquier valor posible que esté avalado por la firma del vendedor, y podrá tener precisión de décimas de céntimo (0,001€).
- Fecha: la fecha de creación de la calderilla con precisión mensual. Es decir, tendrá el mes y año correspondiente al momento en que el cliente paga al vendedor y este le envía la calderilla por valor correspondiente al importe. Cabe reseñar que mientras una misma unidad esté en posesión de un cliente y no se agote, tendrá siempre la misma fecha (la creación es única aunque se actualice con posterioridad).
- Firma: se concatenan todos los campos anteriores y se cifra con la clave simétrica correspondiente al mes de creación (el mes y año que figura en el campo fecha) de la calderilla. La asociación entre el mes-año y esta clave simétrica se guarda en la base de datos de claves simétricas.

La calderilla representa una cantidad prepagada y a modo de comparación se puede asociar a las tarjetas prepago telefónicas o bonobus. La falsificación es un proceso computacionalmente mucho más costoso que el posible valor defraudado.

El vendedor expenderá calderilla con un número de serie único para todas las unidades válidas en cada momento. Esto quiere decir que simultáneamente sólo existe una unidad válida para cada número de secuencia, lo cual no quiere decir que cuando un usuario gaste una unidad ese número de secuencia se deseche. Lo que ocurrirá es que se creará calderilla nueva con ese identificador, no habrá problema de fraude debido a que la clave de cifrado variará de una unidad a otra al generarse en fechas diferentes. Incluso aunque esta clave fuese la misma, el usuario deberá de aportar los datos que están vigentes en la base de datos de calderilla.

El funcionamiento de la calderilla será el siguiente. Un cliente conseguirá una unidad por un método definido anteriormente, bien sea a través de tarjeta de crédito o a través de dinero electrónico. De la primera forma el sistema podría no ser totalmente anónimo si todas las partes se ponen de acuerdo. Sin embargo, si se compra calderilla a través de monedas anónimas el sistema será completamente anónimo. Por lo tanto el anonimato del sistema depende de la forma de pago en la consecución de la calderilla.

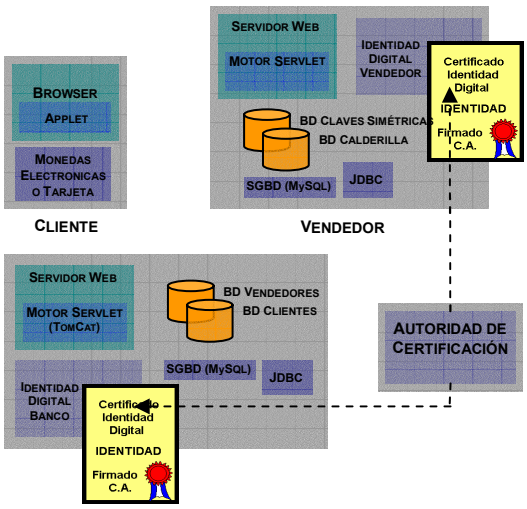


Fig. 9: Arquitectura general de sistema de micropago.

En este punto el usuario tiene calderilla en su poder por valor del pago que haya realizado. Ahora con la calderilla podrá realizar pagos (solamente en el vendedor en el que ha conseguido las unidades) de cualquier valor con precisión de décimas de céntimo (0,001€). Para realizar este pago enviará la unidad al vendedor, este comprobará la vigencia y validez del mismo. Si se da el visto bueno, el vendedor restará al valor de la unidad el importe correspondiente al pago realizado y lo cambiará en la base de datos de la calderilla.

Analizando la fecha de creación de la unidad, cogerá de la base de datos de claves simétricas la clave correspondiente al mes-año de creación de la calderilla y cifrará los datos con el nuevo valor para la unidad. Esta nueva firma se introducirá en la base de datos de la calderilla actualizándola de esta forma para posteriores pagos. Una vez que se realizan todos estos pasos se envía la nueva unidad al cliente.

En el caso que se llegue a gastar por completo la unidad, el valor que figurará en la base de datos de la calderilla será cero. Un cero en este campo indica que la unidad no pertenece a nadie y se puede entregar a otro usuario. Por razones de seguridad no se entregará dos veces dentro del mismo mes el mismo número de serie, ya que la clave de cifrado para ambos sería la misma y habría una posibilidad de fraude. Conseguir esto es bien fácil, ya que en la base de datos de la calderilla quedará la fecha de generación de la última unidad con ese número de serie. Si la fecha coincide con el mes-año en curso no se empleará ese número de serie para generar una nueva calderilla.

El campo fecha, para una unidad con el mismo número de serie, sólo actualizará su valor cuando se gaste y se genere uno nuevo para dárselo a otro cliente.

Como ya se ha comentado, la asociación entre el mes-año y la clave simétrica correspondiente se

guarda en la base de datos de claves simétricas. Esta base de datos no crecerá indefinidamente puesto que cuando en la base de datos de calderilla no quede ninguna entrada para un determinado mes-año, la asociación de este mes-año con la clave simétrica que tuvo en su momento ya no será necesaria, y se podrá borrar la entrada de este mes-año en la base de datos de claves simétricas.

La utilización de claves simétricas de validez mensual tiene un enfoque diferente del de las claves empleadas en las monedas electrónicas. En éstas se tenía una clave para cada valor nominal, mientras que en el caso de la calderilla, el valor puede ser cualquiera y las claves se controlarán por la fecha de creación. Este control se llevará mediante la asociación previamente comentada.

Suponiendo que el usuario ya ha conseguido calderilla para pagar en un determinado vendedor mediante el pago con monedas electrónicas (el más recomendado) o mediante el pago a través de tarjeta de crédito, se analizará en detalle los pasos que se darán en el pago a través de micropago. Cabe destacar que a partir de este momento la entidad del banco ya no será necesaria, puesto que el vendedor se vale perfectamente para validar y actualizar la calderilla. En la Fig. 10 se muestra gráficamente el proceso de pago a través de micropago.

Como se ha podido ver, en este sistema sólo son necesarias las tres entidades a la hora de obtener la calderilla. La obtención de calderilla en realidad no es nada más que un pago a través de monedas electrónicas o a través de tarjeta. En el caso más habitual, para describir el funcionamiento de la solución adoptada para micropago sólo son necesarias dos entidades.

6 Conclusiones y trabajo futuro

El sistema presentado trata de responder a la creciente necesidad de disponer de sistemas de pago telemático eficientes que ofrezcan la versatilidad suficiente para ser empleados en diferentes escenarios. Así, ofrece de forma telemática las alternativas más utilizadas en las operaciones de pago habituales (la tarjeta de crédito y pago en metálico), junto con un sistema de pago específico, el micropago.

En el caso de pagos mediante tarjeta de crédito, el sistema ofrece garantías y seguridad a cada una de las partes, ya que pueden justificar la realización de la transacción. El cliente a través de la factura, ya que está firmada por el vendedor y posee el justificante de pago del banco, el vendedor a través del justificante de pago del banco y de los datos que el cliente le envió y que cifró con la clave del banco y el banco porque con todos los datos es él que realiza la transacción.

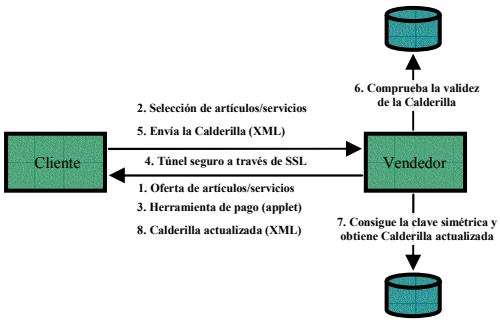


Fig. 10: Proceso de pago mediante micropago.

Además el sistema permite el anonimato del cliente frente al vendedor. Aunque esto sea así, el sistema no será anónimo puesto que el banco conocerá tanto la identidad de uno como la del otro a la hora de realizar la transferencia.

En cuanto al pago basado en moneda electrónica cabe destacar que el sistema ofrece anonimato en su funcionamiento. No es necesario amortizar las monedas cada vez que traspasan de un usuario a otro y en este intercambio se preserva el anonimato de los usuarios. Con la solución adoptada se resuelve el problema del double spending (que un usuario intente gastar dos veces la misma moneda) sin tener que mantener en una base de datos los números de secuencia de las monedas gastadas. El problema que puede producirse es que estas bases de datos crezcan de forma incontrolada. En la solución adoptada, la base de datos de las monedas se dimensiona teniendo en cuenta la cantidad de dinero que el banco quiera tener en forma digital.

También se permite que el receptor de la moneda, no amortice (pase el valor de la moneda a su cuenta en el banco) inmediatamente el valor de la moneda y lo mantenga en forma de dinero digital de una forma segura. La principal ventaja de la solución adoptada frente a un sistema como eCash [10] es que se mantienen los mismos los números de serie, mientras que en eCash una vez que una moneda es amortizada se tiene que eliminar de la circulación su número de serie.

Finalmente, en el modo de pago basado en calderilla, la creación de ésta se lleva a cabo de forma centralizada y controlada por el vendedor correspondiente. A pesar de la creación centralizada, el sistema en si es anónimo puesto que no se requiere la identidad del cliente en ningún momento. Es a la hora de conseguir la calderilla cuando se puede perder el anonimato, asociando el número de serie de la unidad a la identidad autora del pago. Pero si este pago se realiza de forma anónima, el sistema de micropago también lo será.

Otra característica que agiliza el sistema, frente a sistemas como el de pago con monedas electrónicas, es el hecho de que la calderilla que hace falta para hacer un micropago es sólo una, mientras pueden ser muchas las monedas involucradas en la realización de un pago. No sólo se gana velocidad al ser menos la información que se intercambia entre las partes, sino que también se ahorra tiempo a la hora de comprobar la validez de lo intercambiado. Calderilla sólo habrá que validar una, mientras que las monedas a validar pueden ser varias. Además, las monedas hacen uso de criptografía asimétrica bastante más lenta que la criptografía simétrica utilizada en la calderilla.

El sistema presentado ofrece una clara mejora del rendimiento en lo que se refiere al pago mediante moneda electrónica frente al sistema descrito en [1], en el que sólo se ofrecía esta modalidad. Las operaciones de creación de monedas, pago y devolución se realizan en tiempos inferiores, con una reducción media de un 15%. La falta de sistemas de prueba para software de pago electrónico, así como la inexistencia de métricas específicas para ello dificultan la comparación real y objetiva entre las diferentes propuestas existentes. Este es uno de los campos de investigación abiertos en el se pretende profundizar en el futuro.

Además de la ya citada, las líneas de trabajo futuro que se pretende abordar son la integración del sistema con PKIs adaptadas a entidades finales no personales, así como la creación de un sistema de gestión para la venta de contenidos digitales. Para ello se dispone de un prototipo de servicios de contenidos educativos basado en estándar IMS y un sistema de sindicación de contenidos.

Agradecimientos

Este trabajo ha sido financiado en parte por la aportación de la empresa Iberdrola SA a través del Aula Iberdrola de la Escuela Superior de Ingenieros de Bilbao (2001) y por el Gobierno Vasco y la empresa Sarenet SA a través del proyecto NOTACON (OD01UN14), de la convocatoria de proyectos de Investigación Oferta-Demanda del Plan de Ciencia Tecnología e Innovación (PCTI), 2001-2002.

Referencias

[1] J. Unzilla, A. Muñoz, J. Eguiluz, C. Perfecto. “Sistema de comercio de contenidos con anonimato mediante dinero no trazable”. Actas de las III Jornadas de Ingeniería Telemática (JITEL'01). Barcelona, 19-21 Septiembre 2001. ISBN: 84-7653-783-2.

[2] Estudios del uso de sistemas de pago. http://www.aece.es/.

[3] Network payment mechanisms and Digital cash. http://ntrg.cs.tcd.ie/mepeirce/project.html.

[4] D. O’Mahony, M. Peirce, H. Tewari. *Electronic Payment Systems for e-Commerce*. 2nd Ed. Artech House. ISBN: 1-58053-268-3 (2001).

[5] Proyecto NOTACON: Definición de nuevos servicios telemáticos a empresas: proveedores de servicios de aplicación, servicios de notaría digital y gestión de mercados de contenidos digitales. Plan de Ciencia Tecnología e Innovación del Gobierno Vasco. 2001-2002.

[6] mysql. http://www.mysql.com.

[7] Apache. http//www.apache.org.

[8] Tomcat. http://jakarta.apache.org/tomcat.

[9] D. Chaum. “Achieving Electronic Privacy”. Scientific American, pp. 96-101. (1.992).

[10] eCash. http://www.ecash.net.

Sesión 2A

Redes y servicios multicast

Evaluación de Rendimiento de un Algoritmo para Recuperación de Errores en Comunicaciones Multipunto

Marta Barría [1] Reinaldo Vallejos [2] Mónica Aguilar [3]
[1]*Departamento de Computación, Universidad de Valparaíso, Valparaíso, Chile*
marta.barria@uv.cl
[2]*Departamento de Electrónica, UTFSM, Valparaíso, Chile*
reinaldo@elo.utfsm.cl
[3]*Departamento de Telemática, UPC, Barcelona, España*
maguilar@mat.upc.es

Abstract. *In* [*3,4*] *the authors of this publication presented an algorithm for error recovery (packets lost) in a multipoint transmission. The aim of the present paper is to analyze the performance of this algorithm. The performance evaluation is valid for any tree topology that connects the participants of the multipoint structure. Performance measures are average value of latency, implosion, number of NACKs sent to recover a given packet and the number of retransmissions that were needed for errorless reception at all the end points.*

1. Introducción

Las comunicaciones multipunto ofrecen una manera eficiente de difundir información desde un (o varios) transmisor (es) a un grupo de receptores.

A grandes rasgos, las aplicaciones multipunto pueden ser divididas en aplicaciones que necesitan que la transmisión de la información sea fiable y aplicaciones que no necesitan de esta fiabilidad. Aplicaciones tales como videoconferencia, audio en Internet, kioscos electrónicos, etc., no requieren que la transmisión sea fiable, ya que pueden aceptar un cierto nivel de pérdida de información sin que el usuario detecte un deterioro significativo en el servicio que recibe. Por otra parte, aplicaciones tales como distribución de *software*, pizarras compartidas, juegos interactivos, transmisión de cuentas bancarias, replicación de bases de datos, etc., sí necesitan que la transmisión de la información sea fiable, lo cual significa que todos los paquetes transmitidos *deben* ser recibidos sin errores por *todos* los receptores de la información [1]. Sin embargo, a causa de la naturaleza de "mejor esfuerzo" de Internet, ésta red no permite entregar un servicio que garantice la fiabilidad de la comunicación, por lo que es crucial desarrollar protocolos que permitan recuperar los paquetes con errores [7]. La solución más simple para generar mecanismos de comunicación multipunto confiable, sería adaptar los protocolos que se usan para este propósito en las comunicaciones punto a punto. Lamentablemente esta estrategia no resulta conveniente, ya que como resultado de esta adaptación se obtienen protocolos muy ineficientes [2], motivo por el cual surge la necesidad de diseñar protocolos en base a otras estrategias.

Los mecanismos de recuperación de errores están constituidos básicamente por una fase de detección del error y otra de corrección del mismo. La fase de detección normalmente se lleva a cabo en los receptores, ya que este esquema es más eficiente que el basado en la fuente [2]. Usualmente los receptores descubren la pérdida de un paquete cuando detectan un salto en la secuencia de la numeración de los paquetes que le están llegando desde una determinada fuente. Para recuperar un paquete, los receptores envían al transmisor (o fuente) un paquete especial llamado NACK, el cual indica a la fuente que debe retransmitir el paquete perdido. Aunque el mecanismo de recuperación de errores recién descrito no parece presentar dificultades, en la práctica si las hay, ya que si cada uno de los receptores afectados por la pérdida de un paquete enviara un NACK a la fuente, en ella se produciría la llegada sincronizada de (posiblemente muchos) NACKs. Este fenómeno se conoce con el nombre de "implosión de NACKs". La implosión de NACKs es indeseable porque produce congestión en la fuente y alrededor de ella, lo que obviamente causa degradación en las medidas de rendimiento globales de la red, tales como el retardo y el *throughput* [2].

Otra medida que permite evaluar el rendimiento de un algoritmo de recuperación de errores en una comunicación multipunto, es la latencia. La latencia, corresponde al intervalo de tiempo que transcurre desde el instante en que un determinado paquete se pierde hasta que todos los receptores lo reciben correctamente. Similarmente al caso de la implosión, la latencia también es una medida importante, ya que afecta el rendimiento global de la red. Para resolver en forma óptima el problema de la implosión de NACKs sería necesario que el mecanismo de recuperación de la información consiga, en un tiempo mínimo, que solo uno de los receptores afectados envíe el NACK. Esta no es una tarea fácil cuando se quiere resolverla cumpliendo simultáneamente algunas otras restricciones, como por ejemplo que los nodos de la red no conozcan la topología de ella (lo que simplifica el mecanismo de recuperación de errores). Además, en cualquier mecanismo diseñado

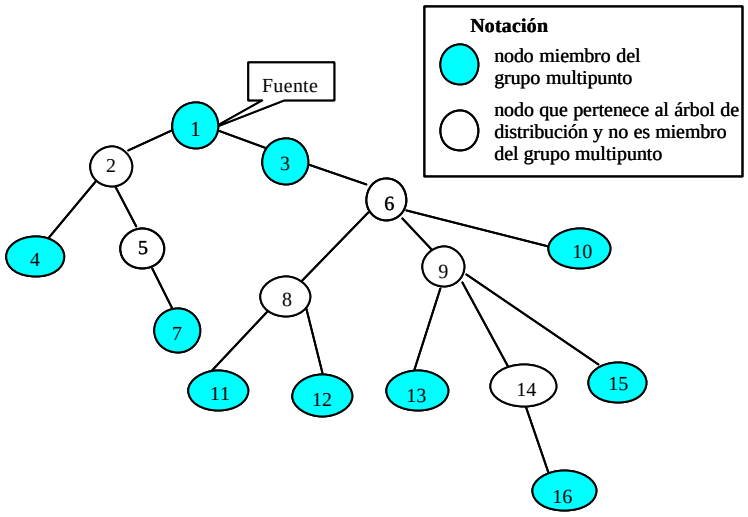


Figura 1. Árbol de distribución D.

para disminuir el problema de la implosión, el subgrupo de receptores afectados por la pérdida de información debe ponerse de acuerdo (de alguna manera) en quién (o quiénes) enviarán NACKS, lo que con gran probabilidad aumentará el retardo en la recuperación de la información (latencia). En otras palabras: la disminución de la implosión tiende a causar un aumento en la latencia, lo que es indeseable. La situación ideal consiste en que el tamaño de la implosión sea igual a 1 y la latencia sea mínima. Por este motivo, los mecanismos para recuperación de errores que han sido propuestos en la literatura buscan disminuir simultáneamente la implosión y la latencia, pues esto implica mejorar las medidas de rendimiento globales de la red. Sin embargo, como estos objetivos son conflictivos entre sí, los algoritmos intentan obtener un compromiso entre el tamaño de la implosión y el tiempo de latencia.

En [3, 4] los autores de este trabajo propusieron un nuevo algoritmo para la recuperación de errores (o pérdida de paquetes) que ocurren en una transmisión multipunto. Este algoritmo tiene algunos atributos que lo hacen competitivo con respecto a los protocolos existentes. En particular, la recuperación del paquete con error se realiza en forma local, es decir, en la zona donde éste se produce; además posee un bajo tiempo de latencia y el tamaño de la implosión es cercano al ideal. Para justificar esta afirmación, en este trabajo se hace un análisis matemático del rendimiento del protocolo propuesto.

Específicamente, en este trabajo se evalúan los valores medios de la latencia, la implosión, el número de NACKs enviados para la recuperación de un paquete y el número de retransmisiones efectuadas hasta que todos los miembros del grupo reciben en forma correcta un determinado paquete. Además, el análisis de rendimiento efectuado en este trabajo es válido para cualquier topología del árbol de recuperación, a diferencia del análisis efectuado en [3] donde se supuso que el árbol de recuperación estaba restringido a una topología tipo estrella.

El resto de este escrito está estructurado de la siguiente forma. En primer lugar, con el objeto de que el artículo sea autocontenido, en la sección 2 se describe el algoritmo; luego, en la sección 3 se evalúan las diferentes medidas de rendimiento. En la sección 4 se muestran algunos resultados numéricos, y finalmente, en la sección 5 se entregan las conclusiones del trabajo realizado.

2. Descripción del Algoritmo

Para representar la red se utiliza un grafo $G=(V, E)$, donde $V=\{i, 1£ i £ V\}$ corresponde al conjunto de nodos de la red, V es el número total de nodos de la red; y $E=\{e_{i,j}; £ i,j£ V, i¹j\}$ es el conjunto de arcos representan los enlaces de la red.

En una comunicación multipunto el intercambio de información se realiza entre un usuario que transmite información y múltiples usuarios que la reciben. Los usuarios de la comunicación multipunto se denominan *miembros del grupo*. El *miembro* que origina la información se denomina *fuente, f,* y los demás miembros del grupo se denominan *receptores*. Sea R el conjunto de receptores, con $R=\{r, 1£ r £ R\}$, donde R es el número de todos los receptores del grupo multipunto. El rol de fuente puede ser asumido por diferentes miembros de la comunicación multipunto en diferentes intervalos de tiempo de la misma. Cada miembro del grupo se conecta a la red a través de algún nodo de ella. Con el objeto de simplificar el escrito, en adelante se usará la expresión *"miembro del grupo"* para referenciar tanto al usuario que participa de la comunicación multipunto como al nodo a través del cual dicho usuario se conecta a la red. Consecuentemente con la definición anterior, se define $M = R \cup \{f\}$ como el conjunto de nodos que son miembros del grupo multipunto, por lo que $M \subseteq V$. Un determinado nodo i pertenece a M cuando existe al menos un participante de la comunicación multipunto que está conectado a la red a través del nodo i. Además se asume que existe un árbol de distribución de la

información, que es denotado por *D*, el cual interconecta a todos los nodos de *M* [1, 2]. Este árbol es generado por algún protocolo de encaminamiento multipunto [5, 6]. Lógicamente, *D* contiene a todos los miembros del grupo multipunto y, adicionalmente, puede contener a algunos nodos de la red que no son miembros de *M*, pero que son necesarios para su interconexión. Los nodos de *D* que no son miembros del grupo, necesariamente son nodos internos del árbol de distribución, tal como se muestra en la Fig. 1. Los paquetes que se originan en la fuente *f* son transmitidos a cada uno de los miembros de *M* a través de *D*. Cuando uno de estos paquetes llega a un nodo del árbol, este nodo envía una copia del paquete a cada uno de sus hijos dentro del árbol. De esta forma los paquetes transmitidos por *f* llegan a todos los miembros del grupo. Además, cada miembro de *M* mantiene una copia de los paquetes que ha recibido correctamente, que en caso de ser requerida, puede ser retransmitida a sus descendientes. Por su parte, aquellos nodos de *D* que no son miembros del grupo multipunto, sólo actúan como transmisores intermediarios de cualquier paquete que reciban.

Los receptores son los responsables de mantener la fiabilidad de la comunicación. Esto significa que los receptores tienen la misión de detectar la pérdida de paquetes y solicitar la retransmisión de los paquetes perdidos. La solicitud de la retransmisión la realizan mediante el envío de un NACK hacia su padre en *M*. La pérdida de paquetes normalmente se detecta por los receptores al notar un salto en la secuencia de numeración de los paquetes que llegan a él desde la fuente. Para modelar la pérdida de paquetes, cada canal de la red tiene asociada una probabilidad de pérdida de paquetes de datos y una probabilidad de pérdida de NACKs. Estas probabilidades son diferentes, debido fundamentalmente a que el tamaño de los paquetes de NACKs es menor que el de los paquetes de datos. Tanto las probabilidades de pérdida de paquetes de datos como de NACKs son homogéneas en el tiempo, lo que significa que dos paquetes del mismo tipo que son transmitidos por un mismo canal en instantes diferentes tienen asociada la misma probabilidad de ser afectados por un error en el canal. Por otro lado, diferentes canales de la red pueden tener asociadas diferentes probabilidades de error.

Cuando por algún motivo se produce la pérdida de un paquete, se activa el mecanismo de recuperación de errores. Antes de proceder a definir formalmente este mecanismo, a continuación se definen algunos conceptos auxiliares; luego, en base a estos conceptos se entrega una descripción intuitiva del algoritmo de recuperación de errores, y posteriormente se define formalmente el algoritmo.

Árbol de recuperación. Sea T_i el sub-árbol de *D* cuya raíz es el nodo $i \in \mathsf{M}$; todos sus nodos interiores no son miembros del grupo multipunto; y el conjunto de sus hojas, $\mathsf{H}_i = \{h \mid h \in T_i\}$ corresponde a todos los miembros del grupo multipunto que son descendientes del nodo *i* en el árbol *D*, que cumplen la condición de que en el camino entre el nodo *i* y alguna hoja *h* en *D* no existe ningún miembro del grupo multipunto. Por ejemplo, en el árbol de distribución de la Fig. 1, hay dos árboles de recuperación: el árbol de recuperación T_3, que tiene como raíz al nodo 3, el conjunto H_3 de sus hojas son los miembros 10, 11, 12, 13, 15,16 (H_3 ={10, 11, 12, 13, 15,16}), y sus nodos interiores (no miembros) son los nodos 6, 8, 9, y 1. El otro árbol de recuperación de *D* es el árbol T_1, cuya raíz es el nodo 1, $\mathrm{H}_1 = \{4,7,3\}$ y los nodos interiores (no pertenecientes al grupo multipunto) son los nodos 2 y 5. Una consecuencia de la definición de árbol de recuperación es que cualquier miembro del grupo multipunto que es un nodo interior de *D*, es simultáneamente raíz de un árbol de recuperación, y hoja de otro árbol de recuperación; por ejemplo, en la Fig. 1, el nodo 3 es hoja de T_1 y raíz de T_3.

Una observación que se usará más adelante, en la definición del algoritmo de fiabilidad, es notar que la pérdida de un paquete ocurre en un único enlace y dicho enlace pertenece a un único árbol de recuperación. Por ejemplo, si en el árbol de la Fig. 1 se pierde un paquete en el canal $e_{3,6}$ o en el canal $e_{6,9}$, el árbol de recuperación asociado a ambas pérdidas es el mismo, el árbol T_3; en cambio, si el paquete se pierde en el canal $e_{5,7}$, el árbol de recuperación asociado a la pérdida es el árbol T_1.

Considere que la pérdida de un paquete ocurre en un determinado enlace $e_{j,k} \in \mathsf{D}$, donde el nodo *j* es el padre del nodo *k* en *D*. Bajo estas condiciones se define el árbol de error, F_k , al sub-árbol de *D*, cuya raíz es el nodo *i* (donde *i* es el ancestro perteneciente a *M* más cercano al nodo *k)* contiene el camino entre el nodo *i* y el nodo *k*, y contiene al sub-árbol de T_i cuya raíz es el nodo *k*. Cabe destacar que los primeros miembros del grupo multipunto que detectan la pérdida de un paquete son las hojas de F_k. Volviendo al ejemplo de la Fig. 1, si el error ocurre en el enlace $e_{3,6}$, los primeros miembros que se dan cuenta del error son todas las hojas de T_3, en este caso $F_6 = T_3$. Por otro lado, si el error ocurre en enlace $e_{6,9}$, de las hojas de T_3 solamente los miembros 13, 15 y 16 son afectados por el error, en este caso el árbol de error está compuesto por los nodos 3, 6, 9, 13, 14, 15 y 16 y todos los arcos que interconectan a estos nodos. Nótese que en este caso $F_9 \neq T_3$.

Cada hoja *h* que pertenece a un árbol de error F_k tiene asociado un único árbol de espera Q_h . La raíz de Q_h es el nodo (miembro) *h*; y además Q_h contiene a todos los descendientes de $h \in \mathsf{D}$. Por ejemplo, suponga que en el árbol de la Fig. 1, el

nodo 9 también es miembro del grupo; bajo esta condición se tendrían asociados los siguientes árboles de espera: Q_9, Q_{10}, Q_{11}, y Q_{12}.

Las definiciones *de árbol de recuperación, árbol de error y árbol de espera* son útiles para explicar de manera intuitiva el funcionamiento del algoritmo; sin embargo, para la ejecución del algoritmo ningún nodo del árbol de distribución necesita conocer la existencia de estos árboles. A continuación se entrega una explicación intuitiva del algoritmo usando las definiciones anteriores.

Cuando un determinado miembro detecta la pérdida de un paquete, envía un mensaje de inhibición por su árbol de espera. Los miembros del grupo que reciben el mensaje de inhibición quedan imposibilitados de solicitar la repetición del mensaje perdido. El objetivo de esta inhibición es doble: por un lado limitar la implosión (limitando el número de miembros candidatos a enviar el NACK) y, por otro lado, lograr una baja latencia. Esto es debido a que los únicos miembros afectados por el error que no reciben ningún mensaje de inhibición son las hojas del árbol de error, y son éstos los miembros del grupo más cercanos al lugar de ocurrencia del error. Por este motivo éstos son los nodos encargados de solicitar la retransmisión del paquete perdido. Después de enviar el mensaje de inhibición, cada una de las hojas del árbol de recuperación que fue afectada por el error ejecuta un experimento aleatorio, con distribución Bernoulli. Si el resultado del experimento es exitoso, la hoja envía un NACK hacia arriba del árbol de distribución. El parámetro del experimento Bernoulli se escoge de forma que con alta probabilidad se envíe un único NACK. Los objetivos de la ejecución del experimento aleatorio son lograr simultáneamente baja latencia y baja implosión. Se logra baja latencia debido a que cada nodo decide si enviar o no un NACK de manera casi inmediata (el único retardo corresponde al tiempo que demora una CPU en efectuar el experimento aleatorio), la baja implosión se logra en la medida que efectivamente se logre que se envié solamente un único NACK.

Cuando el NACK es propagado hacia arriba del árbol de distribución, el primer miembro que lo recibe es precisamente la raíz del árbol de recuperación asociado al error. Debido a que esta raíz tiene una copia del paquete perdido, elimina el mensaje de la red y retransmite (hacia abajo) el paquete solicitado. Nótese que al eliminar el NACK, la raíz del árbol de recuperación asociado al error es el único miembro, de los que poseen una copia del paquete perdido, que se entera de la pérdida. Al proceder de esta forma el algoritmo consigue que: se retransmita una única copia del paquete perdido (lo que evita implosión de retransmisiones), y que esa copia sea retransmitida por el nodo más cercano al lugar del error que posee una copia del paquete perdido (lo que ayuda a obtener una baja latencia).

Después de que la raíz del árbol de recuperación retransmite el paquete perdido, este paquete se propaga normalmente hacia abajo del árbol de distribución. Cuando el paquete llega a miembros que lo habían perdido, éstos miembros se recuperan del error; y cuando el paquete llega a un miembro que no había perdido el paquete, este miembro descarta el paquete de la red, con lo cual evita que se propague inútilmente hacia abajo del árbol de distribución.

2.1 Algoritmo de fiabilidad Multipunto

A continuación se entrega una definición precisa del algoritmo de fiabilidad utilizando la notación introducida anteriormente.

1. Si un miembro del grupo multipunto detecta la pérdida de un paquete, transmite un mensaje a todos sus nodos descendientes, informándoles de este evento. Este mensaje se propaga normalmente hacia abajo a través de *D*. Los miembros del grupo que reciben dicho paquete, se auto inhiben de enviar un NACK respecto de ese paquete (delegando de esa forma la responsabilidad de recuperar el paquete perdido a otros miembros del grupo, motivo por el cual este mensaje se denomina *mensaje de inhibición*). Después de transmitir el mensaje de inhibición, el nodo continúa transmitiendo normalmente a sus descendientes los paquetes correctos que le llegan desde la fuente.

2. Después de transmitir el mensaje de inhibición, el *miembro* que detectó el error (suponga que es el nodo *h*), decide si envía o no un NACK hacia arriba en el árbol de distribución. Para esto, efectúa un experimento aleatorio con distribución Bernoulli de parámetro $p_1(h)$ (más adelante se explica cómo se escoge el parámetro $p_1(h)$). Si el resultado del experimento aleatorio es un éxito, entonces el miembro envía un NACK hacia su padre en *D*.

3. Después, independientemente de haber enviado o no el NACK, el nodo inicia un *Timeout*, $TO(T_i)$, con el objeto de esperar la retransmisión del paquete perdido.

4. Si el $TO(T_i)$ expira antes de que el paquete solicitado sea recibido, el nodo *h* repite los pasos 2 y 3 de este algoritmo, pero esta vez el parámetro del experimento Bernoulli vale $p_n(h)$, donde $(n-1)$ corresponde al número de veces que el nodo *h* ha efectuado el paso 2 del algoritmo en su intento de recuperar el paquete perdido.

5. Cuando un nodo del árbol de distribución que no es un miembro del grupo recibe un NACK desde uno de sus hijos en el árbol de distribución, lo retransmite a su padre en el árbol de distribución.

6. Cuando un miembro del grupo recibe un NACK con respecto a un paquete determinado, éste nodo retransmite el paquete solicitado a cada uno de sus hijos en el árbol de distribución y elimina el mensaje NACK de la red.

7. Cuando el nodo *h* recibe la retransmisión del paquete solicitado, lo retransmite a cada uno de sus hijos en el árbol de espera Q_h. Dicho paquete se propaga normalmente hacia abajo en el árbol de distribución (con lo cual cada uno de los miembros del grupo que son descendientes del nodo *h* recuperan el paquete perdido, sin haber participado activamente en su recuperación).

Finalmente, cuando un nodo que no fue afectado por el error recibe el paquete retransmitido, descarta el paquete de la red.

3. Análisis de Rendimiento

3.1 Evaluación de la Latencia

Sea E[L] el valor medio de la latencia. Debido a que la pérdida de un paquete puede ocurrir en cualquier enlace del árbol de distribución D, para evaluar E[L] se condiciona en el enlace en el cual ocurre el error, con lo cual se obtiene:

$$E[L]=\sum_{\forall e_{j,k}\in D} E\left[L \mid error\ en\ e_{j,k}\right]\Pr\left(error\ en\ e_{j,k}\right) \quad (1)$$

Suponiendo que la red está compuesta por un mismo tipo de enlace, lo que implica que éstos tienen la misma probabilidad de ser afectados por los errores, se tiene que:

$$\Pr\left(error\ \ en\ e_{j,k}\right)=\frac{1}{C(\mathrm{D})}; \quad (2)$$

donde *C(D)* es el número de enlaces que posee el árbol de distribución *D*.

Para calcular $E\left[L \mid error\ en\ e_{j,k}\right]$ se condiciona en la iteración en la cual se envía por primera vez al menos un NACK y se aplica el teorema de probabilidades totales, con lo cual se obtiene:

$$E\left[L \mid error\ en\ e_{j,k}\right]=\left(\sum_{n=1}^{N(F_k)} E\left[L \mid e_{j,k}, n\right]P\left(n \mid e_{j,k}\right)\right) \quad (3)$$

donde *n* identifica el número de iteraciones efectuadas por el algoritmo hasta que se envía el primer NACK; $N(F_k)$ corresponde al máximo valor de *n*, dado que el error ocurrió en el enlace $e_{j,k}$, es decir $N(F_k)$ corresponde al número de la iteración del algoritmo en la cual al menos una de las hojas afectadas por el error (las hojas del árbol F_k) envía con probabilidad igual a uno un NACK; $E[L \mid e_{k,j}, n]$ es el valor medio de la latencia dado que el error ocurrió en el enlace $e_{j,k}$ y que se envió por primera vez al menos un NACK en la *n*-ésima iteración del algoritmo; y $P(n|e_{j,k})$ es la probabilidad de que se envíe al menos un NACK por primera vez en la *n*-ésima iteración del algoritmo, dado que el error ocurrió en el enlace $e_{j,k.}$

Para evaluar $E\left[L|n,e_{j,k}\right]$, se supone que no existe pérdida de NACKs ni de paquetes retransmitidos. Esta suposición es razonable ya que la ocurrencia de un error es un fenómeno de baja probabilidad de ocurrencia . Debido a esto, esta suposición no altera de manera significativa el resultado, sin embargo simplifica el análisis (nótese que el error (o perdida de un paquete) que causa la activación del algoritmo de fiabilidad también es un fenómeno que ocurre con baja frecuencia relativa, sin embargo en esta sección se está evaluando la latencia del algoritmo justamente en esas ocasiones). La suposición anterior implica que:

$$E\left[L|n,e_{j,k}\right]=n\cdot TO\left(T_{r(F_k)}\right) \quad (4)$$

donde $r(F_k)$ es una función que entrega el nodo raíz del árbol de error F_k .
Efectuando los cálculos intermedios necesarios (para más detalles consultar [8]), se concluye que la latencia media está dada por la ecuación (5).

$$E[L]=\sum_{\forall e_{j,k}\in D}\frac{TO\left(T_{r(F_k)}\right)}{C(D)}\left(\left(1-\prod_{\forall h\in F_k}\left(1-p_1(h)\right)\right)+\sum_{n=2}^{\min_{h\in F_k}\left(\left\lceil 1+\lg_2\left(\frac{1}{p_1(h)}\right)\right\rceil\right)} n\left(1-\prod_{\forall h\in F_k}\left(1-p_1(h)g^{n-1}\right)\right)\prod_{m=1}^{n-1}\prod_{\forall h\in F_k}\left(1-p_1(h)g^{m-1}\right)\right) \quad (5)$$

$$E[I]=\sum_{\forall e_{j,k}\in D}\frac{1}{C(\mathrm{D})}\left(\sum_{i=1}^{H(F_k)} i\cdot\sum_{\forall \bar{k}_{H(F_k),i}\in K_{h(F_k),i}}\left(\prod_{\forall h\in \bar{k}_{H(F_k),i}} p_1(h)\prod_{\forall h\notin \bar{k}_{H(F_k),i}}\left(1-p_1(h)\right)\right)\right)\left(1-\prod_{\forall h\in F_k}\left(1-p_1(h)\right)\right)$$

$$+\sum_{\forall e_{j,k}\in D}\frac{1}{C(\mathrm{D})}\left(\sum_{n=2}^{N(F_k)}\left(\sum_{i=1}^{H(F_k)} i\cdot\sum_{\forall \bar{k}_{H(F_k),i}\in K_{h(F_k),i}}\left(\prod_{\forall h\in \bar{k}_{H(F_k),i}} p_n(h)\prod_{\forall h\notin \bar{k}_{H(F_k),i}}\left(1-p_n(h)\right)\right)\right)\left(1-\prod_{\forall h\in F_k}\left(1-p_n(h)\right)\right)\prod_{m=1}^{n-1}\prod_{\forall h\in F_k}\left(1-p_m(h)\right)\right) \quad (6)$$

3.2 Evaluación de la Implosión

Sea E[I] el valor medio de la implosión. Análogamente al procedimiento usado para evaluar la latencia media, para calcular E[I] se condiciona en el enlace en el cual puede ocurrir un error, en el número de hojas que pueden enviar simultáneamente un NACK y en la iteración en que puede acontecer este hecho. Realizando los cálculos intermedios, la implosión media está dada por la ecuación (6), donde $K_{l,u}$ corresponde al conjunto de *l*-tuplas distintas que pueden formarse a partir de *u* elementos diferentes. En el caso bajo análisis, *l* correspondería al número total de hojas de F_k y *u* correspondería a las hojas que envían un NACK. Sea $\vec{k}_{l,u}$ una de las *l*-tuplas del conjunto $K_{l,u}$, y *h* es una de los componentes de $\vec{k}_{l,u}$. Como ejemplo de la notación recién definida considere el siguiente caso: $K_{8,3}$ representa al conjunto de todas las 8-tuplas diferentes que se pueden formar a partir 3 elementos diferentes; $\vec{k}_{8,3}$ es una de las 8-tuplas que forman parte de $K_{8,3}$ y *h* uno de los componentes de $\vec{k}_{8,3}$; y sea $H(F_k)$ el número de hojas del árbol F_k.

4. Resultados numéricos

Considere una topología del árbol de recuperación es de tipo estrella, compuesta por *R* miembros del grupo multipunto, y un nodo que no es miembro. Los canales del árbol de recuperación tienen longitud igual a 1, y la misma probabilidad *x* de que se pierda un paquete transmitido por ellos. Usando estos valores, se tiene que la probabilidad de que una determinada hoja *h* envíe un NACK la primera vez que se detecta la pérdida de un determinado paquete (es decir, en la primera iteración del algoritmo), está dada por:

$$p_1(h) = \min\left[1, \frac{a}{R-1}\right], \qquad 1 \le h \le R-1$$

donde α es un parámetro del algoritmo, utilizado para obtener equilibrio entre la latencia y la implosión. La probabilidad de que una hoja envíe un NACK en iteraciones posteriores ($n>1$) se calcula suponiendo que *g= R-1* (el número de hojas del árbol de recuperación) En consecuencia, se tiene que: $p_n(h) = 1;\ 1 \le h < R$.

A continuación se muestran algunas de las medidas de rendimiento obtenidas para este tipo de topología del árbol de recuperación. Para este caso, el valor de la latencia media está dado por: $E[L] = \frac{1}{R}\left\{1 + (1-p)^{(R-1)} + (R-1)(2-p)\right\}$; y la implosión media está dada por: $E[I] = \frac{R-1}{R}\left\{p + (1-p)^{R-1} + 1\right\}$

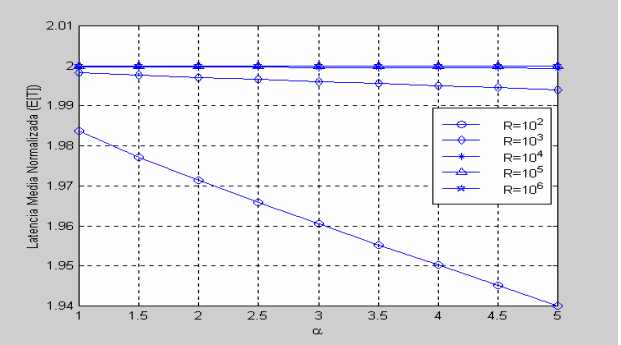


Figura 2. Latencia media en función de α.

En la Fig. 2 se ilustra la latencia media en función de diferentes valores de α, para tamaños de grupo que varían entre 10^2 y 10^6 miembros. En el gráfico puede verse que para tamaños de grupo mayores o iguales que 10^3 miembros la latencia media se mantiene alrededor de 2, independientemente del valor de α. Esto posiblemente se debe a que, con probabilidad cercana a 1, el error afecta sólo a un miembro (fenómeno que para este caso específico ocurre con probabilidad (R-1)/R), el cual envía con probabilidad 1 un NACK en la segunda iteración del algoritmo.

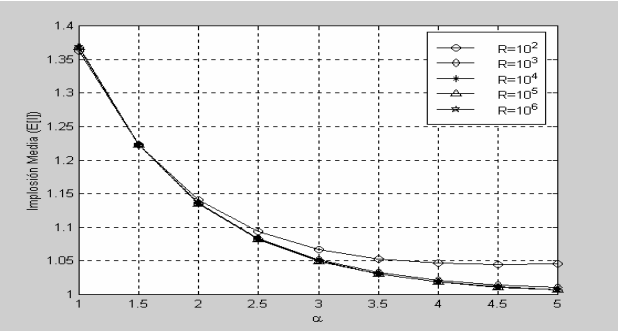


Figura 3. Implosión media en función de α.

En la Fig. 3 se ilustra la implosión media en función de α, para un árbol de recuperación con más de 100 hojas. En este caso se nota que a medida que el tamaño del grupo de recuperación aumenta la implosión media tiende a 1 (el valor ideal). Esto se debe a que en estos casos E[I] corresponde aproximadamente a (1+*e-a*). Otra forma, más práctica, de explicar esta tendencia de E[I] es notar que en estos casos con alta probabilidad el error afecta sólo a una hoja.

4.1 Otras medidas de Rendimiento

Para el árbol de recuperación tipo estrella se calculan en forma adicional dos nuevas medidas. Estas son: el número medio de retransmisiones de un paquete y el número medio de NACKs enviados hasta que los *r* receptores reciben correctamente un paquete.

Para este propósito el comportamiento del algoritmo propuesto se puede representar por medio de una cadena de Markov (CM), en la que los estados identifican al número de receptores que todavía no han recibido correctamente un determinado paquete transmitido por la fuente. Esto significa que si la CM se encuentra en el estado i, entonces i receptores no han recibido correctamente el paquete. Lógicamente que el instante en que el transmisor envía el paquete por primera vez, ningún receptor lo ha recibido. Este es el motivo por el cual el estado inicial de la cadena de Markov es el estado R (donde R es el número de hojas del árbol de recuperación).

Sea $p_{i,j}$ la probabilidad de que CM transite del estado i al estado j en una transmisión del paquete. Una transmisión del paquete corresponde a una iteración de los pasos 2 y 3 del algoritmo. (En adelante se usará el vocablo "paso" para referirnos a una de estas iteraciones). Evidentemente se cumple que $p_{i,j} = 0$, $i>j$. Por otro lado, para que la CM transite del estado i al j en un paso, lo que debe ocurrir es que: $(i-j)$ receptores de los i que no tenían el paquete lo reciban correctamente. Esto ocurre con probabilidad $\binom{i}{i-j}(1-f(c))^{i-j}$; además, los otros j receptores no lo deben recibir, lo que ocurre con probabilidad $f(c)^j$. Entonces la transición entre el estado i y el estado j en un paso está dada por:

$$p_{i,j} = \begin{cases} \binom{i}{i-j}(1-f(c))^{(i-j)} f(c)^j; & 0 \le j \le i \le r \\ 0 & i < j \end{cases} \quad (7)$$

donde $f(c)$ es la probabilidad de que un paquete se pierda en el camino c que va entre el transmisor y el receptor [8].

Evaluación del número medio de retransmisiones de un paquete.

Sea T_r la variable aleatoria que representa el número de transmisiones necesarias para que un paquete sea recibido exitosamente por todos los r receptores pertenecientes al árbol de recuperación mostrado en la Fig. 2. Este número de retransmisiones equivale al número de pasos que la CM demora en transitar por primera vez desde el estado r al estado 0. Sea $E[T_r]$ el valor medio de T_r. El número medio de retransmisiones está dado por:

$$E[T_r] = \frac{1}{(1-f(c)^r)}\left(1+\sum_{j=1}^{r-1}\binom{r}{i} f(c)^j (1-f(c))^{r-j} E[T_j]\right) \quad (8)$$

donde p_{rj} está dada por la ecuación (7). Para más detalles ver [8].

Evaluación del número medio de NACKs enviados hasta que los *r* receptores reciben correctamente un paquete.

Se considera un caso especial, en el cual la probabilidad de enviar un NACK es igual para todas las hojas del árbol de recuperación y no varía en el tiempo, es decir $p_n(h)=p$. Además se define q como la probabilidad de pérdida de un NACK. Entonces, puede escribirse la probabilidad de que un NACK sea enviado por el receptor y recibido efectivamente por el transmisor como $\tilde{p} = p\ (1-q)$. Como en el caso anterior, sea $f(c)$ la probabilidad de perder un paquete. Sea $E[\tilde{N}_r]$ el número medio de NACKs que efectivamente llegan al transmisor hasta que los r receptores hayan recibido el paquete perdido. Sea $E[N_r] = \frac{E[\tilde{N}_r]}{q}$ el número medio de NACKs enviados hasta que los r receptores se hayan recuperado del error. Entonces:

$$E[\tilde{N}_r] = r\tilde{p} + \sum_{j=1}^{r-1}\binom{r}{j} f(c)^j (1-f(c))^{r-j} E[\tilde{N}_j] + f(c)^r E[\tilde{N}_j] \quad (9)$$

Luego, usando funciones generadora exponencial y manipulando algebraicamente la expresión resultante, se llega a que la expresión para el número medio de NACKs enviados está dada por: En la Fig. 5 se muestra número medio de NACKs enviados hasta recibir un paquete en forma correcta versus el número de hojas del árbol de recuperación, para diferentes probabilidades de pérdida de paquete en un canal, denotada por x.

$$E[N_r] = \frac{r\tilde{p}}{q(1-f(c))} \quad (10)$$

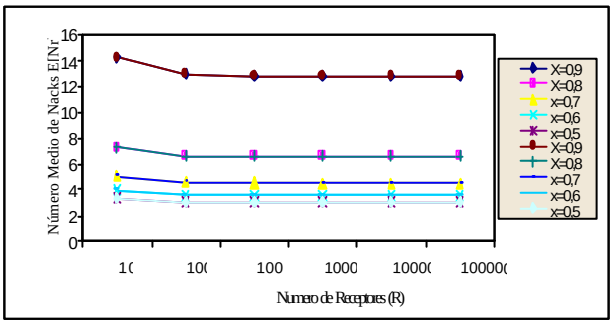


Figura 4. Número medio de Nacks versus número de hojas del árbol de recuperación.

Puede verse en el gráfico que el número medio de NACKs enviados hasta recuperar un paquete es casi independiente del número de receptores, a partir de 100 receptores y para cualquier valor de x. Esto demuestra la alta escalabilidad del algoritmo propuesto. También se puede observar que a medida que la probabilidad de pérdida de paquetes aumenta, a cantidad de NACKs enviados también se incrementa.

5. Conclusiones

En este trabajo se ha realizado el análisis de rendimiento del algoritmo para la recuperación de errores en una transmisión multipunto que fue propuesto en [3]. En este algoritmo, la recuperación de errores se realiza solamente por algunos de los miembros del grupo vecinos al error. Estos nodos delimitan lo que hemos denominado *árbol de recuperación.* El algoritmo propuesto atribuye mayor probabilidad de solicitar la retransmisión de un paquete perdido a las hojas del árbol de recuperación más lejanas a la raíz de este árbol, pues son ellas quienes tienen mayor probabilidad de ser afectadas por la pérdida de un paquete.

El algoritmo consigue una baja latencia y una baja implosión en la recuperación de errores, para cualquier tamaño del grupo multipunto. Esto se debe a que la recuperación de errores se realiza en forma local y por otra parte, a la forma con que se determina la probabilidad de que las hojas del árbol de recuperación afectadas por el error soliciten la retransmisión del paquete perdido. En base a los ejemplos analizados puede decirse que el algoritmo presenta una alta escalabilidad, ya que tanto la latencia como la implosión media presentan valores cercanos al ideal, para cualquier tamaño del grupo multipunto.

Entre las medidas de valor medio se calcularon la latencia, la implosión, el número de NACKS y el número de retransmisiones respecto de un paquete perdido. Las medidas fueron evaluadas en base a modelos probabilísticos y de cadenas de Markov. La solución del modelo es válida para cualquier topología del árbol de distribución y para cualquier número de miembros que tenga un grupo multipunto.

Agradecimientos

Este trabajo fue financiado en parte por los proyectos FONDECYT Nº 1000055 /2000, por el proyecto DIPUV 19/ 2001 de la Universidad de Valparaíso, por el proyecto 230223 de la Universidad Técnica Federico Santa María, y por el proyecto DISQET (CICYT TIC2002-00818).

Referencias

[1] C. Diot, W.Dabbous and J.Crowcroft, “Multipoint Communication: A Survey Protocols, Functions and Mechanism", *IEEE Journal on Selected Areas in Communications* Vol. 15, No. 3, April, 1997.

[2] D. Towsley, J.Kurose and S. Pingali, " A Comparison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols", *IEEE Journal on Selected Areas in Communications*, April 1997 .

[3] M. Barría, R. Vallejos, L.F. Soares, “A Failure Tree Reliable Multicast Algorithm Based on Recovery Tree (RMART)” , in Proceedings 9th IFIP Conference on Performance Modelling and Evaluation of ATM & IP Networks 2001, Julio 2001, Budapest.

[4] M. Barría, R. Vallejos, L.F. Soares , Algoritmo para la recuperación de errores de una transmisión multipunto, CLEI 2000, México.

[5] T. Ballardie, *“Core based tree (CBT) multicast: Architectural overview and specification*” Internet Draft RFC, July 1994.

[6] T. Bilhartz, J.B. Cain, D. Fieg and S.G. Batsell, “Performance and Resource Cost Comparisons for the CBT and PIM Multicast Routing Protocols *”, IEEE JSAC*, 15(3), April 1997.

[7] C. Kenneth Miller, "Multicast Networking and Applications", Addisson - Wesley, 1999.

[8] M. Barría, R. Vallejos, M. Aguilar. “Evaluación de Rendimiento de un Algoritmo para Recuperación de Errores en Comunicaciones Multipunto”, Informe Técnico Depto. Electrónica Universidad Técnica Federico Santa María, 2003.

Protocolo de transporte punto a multipunto fiable con control de congestión (PMFCC)

Javier Muñoz Kirschberg, Marta Solera Delgado.
Departamento de Ingeniería de comunicaciones.
E. T. S Ingeniería de Telecomunicaciones. Universidad de Málaga.
Emails: javiermunozk@eresmas.com, msolera@ic.uma.es (1)

Abstract. *This paper presents an overview of the current research in the topic of single-rate transport multicast protocols and it introduces a new protocol, called RCCMP (Reliable, Congestion Controlled Multicast Protocol). This protocol has been designed to be simple, scalable (NAK suppression), reliable and TCP-friendly. The congestion control is a central part of the protocol, where the feedback of the worse receiver is used to control a transmission window in a TCP-like fashion. The scalability issue is addressed with an exponential timers scheme, that is also used to estimate the number of receivers involved in the communication. The protocol does not need support from network elements neither maintains state information dependent of the number of receivers.*

1. Introducción

La comunicación multipunto en Internet ha sido un tema de investigación constante durante los últimos años. El organismo regulador de estándares en Internet, IETF (Internet Engineering Task Force), ha desarrollado una incesante labor para apoyar este esfuerzo, ya que la transmisión punto a multipunto resulta beneficiosa tanto para los proveedores de servicios, que ven aumentada la eficiencia de uso de sus elementos intermedios de red y reducido el tráfico que circula por sus redes, como para los usuarios, que pueden disfrutar de nuevas aplicaciones colaborativas (juegos multimedia mejorados, aplicaciones de trabajo en grupo, etc.) y de difusión (retransmisión en directo de conciertos, eventos deportivos, etc.) con un uso eficiente de la red.

En este artículo se van a analizar algunas propuestas de protocolos de transporte multipunto y se va a presentar el protocolo PMFCC que cumple con los requisitos de sencillez, escalabilidad y fiabilidad necesarios para las comunicaciones multipunto.

1.1 Protocolos de transporte multipunto

Los protocolos de transporte fiable, en conexiones multipunto o en punto a punto, aseguran la recepción ordenada y sin errores y controlan el flujo de transmisión.

Las necesidades de las aplicaciones multipunto, mayores que las de las aplicaciones punto a punto, junto a dificultades propias de este tipo de transmisión, han determinado un amplio número de protocolos de transporte, cada uno de los cuales está diseñado para satisfacer algunas de estas necesidades, entre las que cabe resaltar las siguientes:

- Fiabilidad: Algunas aplicaciones necesitan asegurar que la información transmitida llega sin errores a todos los receptores.
- Recepción ordenada: Relacionada con la anterior, consiste en asegurar que la recepción de la información se produce ordenadamente.
- Control de flujo y de congestión: Es, quizás, la necesidad más importante, ya que sin el control de congestión presente en TCP (Transport Control Protocol), Internet, probablemente, no existiría. Con el control de congestión se asegura que la velocidad de transmisión se adapta a las cambiantes circunstancias de la red.
- Robustez: El protocolo de transporte debe no fallar, o, al menos, limitar las circunstancias en las que se producen fallos y especificarlas claramente.
- Soporte para llegada tardía: Algunas aplicaciones requieren que los receptores que se unen a la comunicación una vez que ésta ha comenzado sean capaces de recuperar los paquetes que se han transmitido antes de que dichos receptores se unieran.
- Gestión de grupo: Esta necesidad consiste en controles de admisión para determinar

(1) Este trabajo ha sido parcialmente financiado por CICYT en el contrato TIC2000-1042.

que receptores pueden unirse a la comunicación.

- Seguridad.

Así, por ejemplo, la transmisión de ficheros mediante mecanismos multipunto posiblemente necesite fiabilidad, recepción ordenada, control de congestión, gestión de grupo, soporte para llegada tardía, etc. sin imponer demasiados requisitos en cuanto a retardo, mientras que la transmisión de información multimedia en tiempo real, como un concierto, no requiere la transmisión fiable (por cuestiones de retardo no tiene sentido recuperar los paquetes ya transmitidos) pero si robustez, control de congestión, gestión de grupo, etc.

Nuestro campo de interés se centra en los protocolos que aseguran, simultáneamente, fiabilidad, robustez, recepción ordenada y control de flujo y de congestión.

1.2 Características deseables en un protocolo de transporte multipunto

Las características deseables en un protocolo de transporte multicast, dependen de las necesidades de la aplicación, aunque existen algunas comunes a todas ellas:

- Escalabilidad: Los protocolos multicast se utilizan en comunicaciones en las que el número de receptores puede ser muy elevado. Por lo tanto, deben incluir mecanismos de control del tráfico generado, para evitar que este tráfico crezca proporcionalmente al número de receptores.

- Comportamiento justo con respecto a TCP: Los protocolos multipunto que han sido diseñados para ser usados en Internet deben tener en cuenta que la mayoría del tráfico que circula por la red es tráfico TCP. Con objeto de que dicho tráfico ya existente no se vea negativamente influido por los nuevos protocolos multicast, éstos deben asegurar que se comportan con justicia con respecto a TCP, es decir, que su presencia no dificulta la transmisión de tráfico TCP más de lo que lo haría la presencia de un nuevo agente TCP.

1.3 Mecanismos de fiabilidad

Para asegurar la fiabilidad, los protocolos multicast no suelen usar un mecanismo de confirmación positiva o explícita (ACK) por parte de cada uno de los receptores, debido a los problemas de escalabilidad que esta solución presenta para un número elevado de receptores; por lo tanto, los protocolos multicast dependen de un mecanismo de confirmación negativa o implícita (NAK), mediante el cual un receptor se comunicará con el agente emisor de la información siempre que detecte la pérdida de ésta. Evidentemente, dicha solución, en caso de que se produzcan pérdidas en muchos receptores, puede dar lugar a la transmisión de un alto número de NAKs al emisor, dando lugar a lo que se conoce como *implosión de NAKs*.

Evitar dicha implosión, es, por tanto, fundamental para el éxito de un protocolo multicast de transporte, y diversos mecanismos han sido sugeridos para este fin. Desgraciadamente, muchos de ellos requieren, al menos en parte, colaboración de los elementos intermedios de red, como es el caso del popular protocolo PGM (Pragmatic General Multicast)[1] o una cierta jerarquización de los receptores, con la complicación inherente que eso conlleva, como RMTP (Reliable multicast Transport protocol)[2]. El protocolo presentado en este artículo utiliza un esquema basado en temporizadores, descrito en [3], [8] y [9], que no necesita la colaboración de los elementos de red ni la jerarquización de los receptores.

1.4 Control de congestión

El control de congestión es uno de los problemas más complicados de resolver en cualquier protocolo de transporte multicast; tanto es así, que protocolos como PGM [1] obvian dicho asunto en el proceso de estandarización. Sin embargo, cabe destacar dos grandes tendencias en el panorama actual:

- Control de congestión unitasa: Este esquema se basa en transmitir toda la información a una única tasa, que irá variando según las circunstancias de la comunicación. Su principal ventaja es la sencillez, tanto conceptual como de implementación.

- Control de congestión multitasa: Estos esquemas transmiten información a cada receptor a la tasa más adecuada para ellos. Sin embargo, su dependencia de los protocolos de enrutamiento (nivel de red), el hecho de que no se puedan aplicar a cualquier tipo de información y su dificultad para responder ante la congestión, representan problemas aún por resolver.

En este artículo nos centraremos en los protocolos con control de congestión unitasa, entre los cuales cabe destacar los que se basan en una ventana de transmisión como PGMCC[4] (PGM Control Congestion) o los que se basan en una tasa de transmisión como TFMCC [5] (TCP friendly Multicast Congestion Control) y ORMCC [6] (Output Rate Multicast Congestion Control). En la siguiente sección, presentaremos brevemente dos de estos esquemas, PGMCC y ORMCC, y

señalaremos algunos de los problemas aún no resueltos.

2. Esquemas de control de congestión relacionados: PGMCC y ORMCC

Los esquemas de control de congestión unitasa se basan, generalmente, en la selección de un representante (generalmente, el peor receptor), que regula la tasa o la ventana de transmisión del agente emisor de la información. Sin embargo, los distintos mecanismos difieren tanto en el proceso de selección del representante (métricas usadas) como en la supresión de la información de realimentación. A continuación, analizamos dos esquemas de control de congestión.

2.1 PGMCC

PGMCC usa la información enviada por el representante, llamado *acker* (confirmador) en ese esquema, en forma de confirmaciones positivas (ACKs) para regular una ventana de transmisión de una manera similar, aunque ligeramente distinta, a la de TCP.

Para escoger al peor representante, se utilizan estimaciones de la tasa de pérdidas y del RTT (Round Time Trip), éste último medido en paquetes y no en instantes temporales, que se computan a través de la llamada ecuación de equilibrio de TCP [7]:

$$Tasa \propto \frac{1}{RTT * \sqrt{p}} \qquad (1)$$

donde RTT es el tiempo que tarda un paquete en ir a un receptor y volver al emisor y p es la tasa de pérdidas.

Para comunicar las condiciones actuales, cada receptor adjunta, cuando transmite una confirmación negativa (NAK), sus medidas de tasa de pérdidas y de RTT. Sin embargo, PGM depende de los elementos intermedios de red para realizar una supresión efectiva de los NAKs, lo que impide que se pueda asegurar que PGMCC escogerá al peor receptor como representante. Además, PGMCC no incluye ningún mecanismo de supresión de información de realimentación.

2.2 ORMCC

ORMCC, al contrario que PGMCC, es un protocolo que regula la velocidad de la fuente por tasa y no mediante un mecanismo de ventana. La tasa de transmisión aumenta continuamente al avanzar el tiempo, disminuyendo sólo en caso de que el agente emisor reciba una indicación de congestión por parte del representante.

Para escoger al representante, ORMCC usa una métrica propia, la llamada tasa de rendimiento en caso de congestión. La tasa del representante es enviada en cada paquete de datos transmitido, de tal manera que sólo aquellos receptores que pasen a encontrarse en peor situación que el representante puedan enviar una indicación de congestión. Como no existen las confirmaciones positivas (ACKs) de PGMCC, el agente emisor supone, que, transcurrido un cierto tiempo sin recibir una indicación de congestión del representante, es necesario realizar un cambio de dicho representante.

Dos son los principales problemas de ORMCC:

- No hay definida ninguna fórmula análoga a la que presenta PGMCC para ponderar las relación RTT-pérdidas, por lo que la elección del representante entre dos receptores de características similares puede resultar ambigua.

- Aún más importante puede ser el hecho de que ORMCC solo actualiza la tasa de rendimiento del representante cuando éste envía una indicación de congestión; por lo tanto, si el peor receptor mejora su situación repentinamente, no enviará las indicaciones de congestión, dejando a todos los receptores con estadísticas obsoletas.

3. Protocolo PMFCC

En este apartado presentamos una propuesta para un nuevo protocolo, simple, fiable, con comportamiento equitativo con respecto a TCP y escalable, llamado PMFCC (Protocolo de transporte punto a multipunto fiable con control de congestión).

3.1 Descripción de PMFCC

PMFCC incluye un mecanismo de congestión unitasa, lo que hace que necesite escoger a un representante entre todos los receptores (el peor). El mecanismo que permite escoger dicho representante es muy parecido al de PGMCC, y por tanto, se basa en computar la ecuación de equilibrio de TCP y escoger al receptor con peores estadísticas. Las métricas usadas por PMFCC también coinciden con PGMCC y son las siguientes:

- Tasa de pérdidas: Es una medida del número de paquetes perdidos por cada receptor. Al contrario que en PGMCC, no se usan filtros de primer orden para estimar esta tasa, sino una simple media del tipo

$$Tasa\ de\ pérdidas = \frac{paquetes\ perdidos}{paquetes\ recibidos} \quad (2)$$

- Retardo de ida y vuelta (RTT): El retardo se mide en paquetes y no en segundos, para, al igual que PGMCC, evitar que diferencias en los relojes de los receptores conlleven un sesgo a favor de un grupo de ellos.

Este representante se comunica con el emisor mediante el envío de confirmaciones positivas (ACK), que las usa para controlar una ventana de transmisión idéntica a la de TCP Reno. Estas confirmaciones son remitidas no sólo al emisor, sino también a todos los receptores, lo que diferencia a PMFCC de la mayoría de protocolos de transporte multipunto con control de congestión unitasa.

Todos los receptores, excepto el representante, estudian los ACKs que aquel envía y detectan que paquetes han sido confirmados y cuales no. Si observan que un paquete no recibido por ellos ha sido confirmado, tienen la posibilidad de solicitar una retransmisión mediante una confirmación negativa o NAK.

3.2 Escalabilidad y fiabilidad en PMFCC: Gestión de NAKs

Generalmente, la escalabilidad de un protocolo depende de cómo gestione el tráfico generado (particularmente los NAKs) y de si los agentes que intervienen en la comunicación deben mantener variables de estado en forma proporcional al número de receptores.

En el caso de PMFCC la escalabilidad se logra mediante a) un número constante de variables de estado y b) el control de los NAKs.

Para lograr este último control, se utiliza el esquema de temporizadores descrito en [3], [8] y [9]. Cuando un receptor advierte que el representante ha confirmado un paquete no recibido por él, compara su tasa de pérdidas con la del representante, que se adjunta en cada ACK, y sólo si es significativamente inferior, el receptor podrá enviar inmediatamente al emisor una confirmación negativa o NAK. Podemos observar, por tanto, que

- Para todos aquellos paquetes que no hayan sido recibidos por un grupo de receptores que incluya al representante, no se generan NAKs, puesto que el representante, al no confirmar la recepción de esos paquetes, provoca su retransmisión.

- Los NAKs generados sólo se enviarán inmediatamente a la fuente en el caso de que el receptor que los genere tenga una tasa de pérdidas significativamente superior a la del representante. Además, a la hora de comparar las tasas de pérdidas que acompañan a un ACK, sólo se tienen en cuenta los paquetes con número de secuencia igual o inferior al número de secuencia del paquete confirmado. Esto, que llamamos **estadísticas retardadas**, evita un problema presente en esquemas como ORMCC y PGMCC, donde los receptores siempre comparar sus estadísticas actualizadas hasta el último paquete recibido con las que tenía el representante en el momento en que generó las estadísticas.

Aquellos receptores que hayan generado NAKs, pero que no tengan una tasa de pérdidas superior a la del representante, retardan su NAK una cantidad de tiempo determinada por un temporizador exponencial. Si durante el tiempo de espera para enviar dicho NAK se recibe el paquete que provocó la generación del NAK, entonces se cancela el envío. En [3], [8] y [9] puede observarse que se logra una supresión muy efectiva de los NAKs con este esquema de temporizadores (por ejemplo, recepción de solo 4 NAKs para 1.000.000 de receptores que hayan perdido el paquete).

Sin embargo, para que este esquema funcione es necesario disponer de una estimación, no necesariamente muy exacta, del número de receptores. Por ello, cada cierto tiempo la fuente pide a todos los receptores que generen un informe de su tasa de pérdidas y del RTT y la retarden antes de enviarla en una cantidad de tiempo también determinada por estos temporizadores exponenciales. En cuanto se recibe el primer informe, el emisor ordena cancelar el envío del resto de ellos, y, usando el número de respuestas que hayan llegado y el algoritmo descrito en [10], el emisor es capaz de estimar, con un margen de error adecuado para las necesidades de los temporizadores, el número de receptores.

El número de NAKs que se reciban por paquete perdido es regulable: cuanto mayor sea dicho número, mayor será el ancho de banda ocupado pero menor el retardo. En general, el mecanismo de supresión de NAKs se comporta de una manera muy ajustada a las predicciones teóricas que pueden realizarse teniendo en cuenta el número de receptores y la probabilidad de error por paquete recibido. En la fig. 1 se pueden observar los resultados de una simulación matemática, realizada en Java, de las cantidad de paquetes NAKs recibidos en el emisor según el número de receptores que generan NAKs (con retardo emisor-receptor uniforme entre 0 y 500 milisegundos y 4 respuestas deseadas; los resultados son la media de 30 simulaciones); dichos resultados muestran que el número de respuestas recibidas se puede controlar, incluso para un número de receptores elevado.

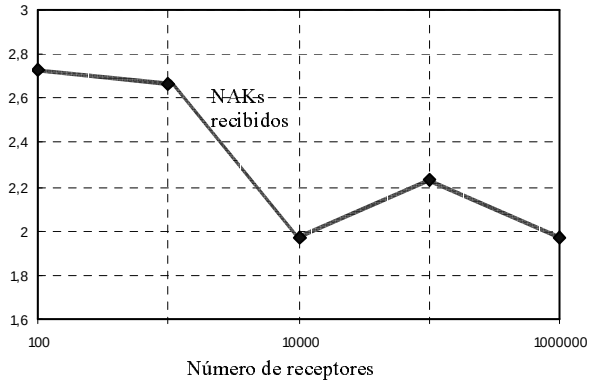


Figura 1: Número de respuestas obtenidas según el número de receptores que reciben el paquete erróneo.

3.3 Selección o cambio del representante

Al comenzar la comunicación, los informes mencionados en el apartado anterior se usan para escoger un representante. El emisor estima constantemente el RTT, en este caso en segundos y no en paquetes, entre el emisor y el representante, para poder aplicar un comportamiento idéntico al de TCP Reno.

Cuando un receptor envía una confirmación negativa (NAK) al emisor, la acompaña de sus estadísticas, para que éste pueda compararlas con las del actual representante, y, si es necesario, realizar un cambio de representante.

Todo representante tiene asegurado un periodo mínimo de permanencia, lo que, unido a que el mecanismo de elección de representante está ligeramente sesgado a favor del receptor que ostenta el cargo de representante, evita un número excesivo de oscilaciones en el protocolo.

En la fig 2 es posible observar, como, al comenzar la transmisión, alguno de los receptores más cercanos al emisor (nodo 0), es seleccionado como representante, ya que sus informes llegan al nodo transmisor antes que los del resto de receptores.

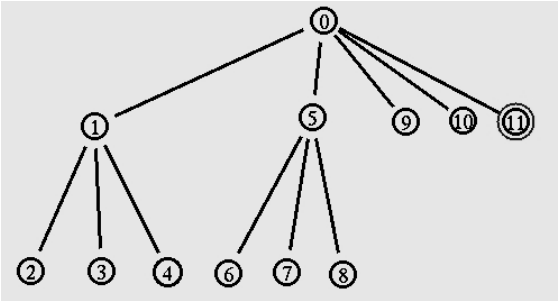


Figura 2: Primer representante (doble círculo).

Para compensar este hecho, la selección de representante incluye un *periodo de gracia*, similar al de ORMCC [6]: después de escoger representante, el emisor espera dos veces el máximo RTT observado hasta ese momento, y todo informe que llegue con características peores o iguales a la del actual representante provoca un cambio de éste. En la fig. 3 se puede observar como el receptor número 2 es seleccionado como representante, ya que su RTT es mayor que el del receptor número 11.

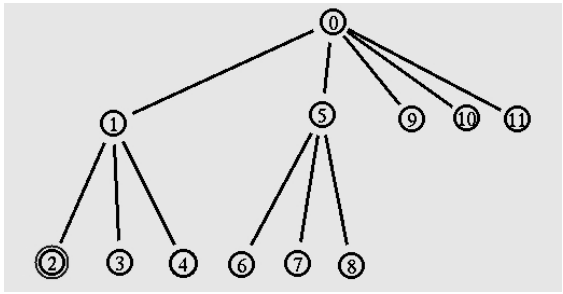


Figura 3: El nuevo representante es el receptor número 2 (doble círculo).

Todos los enlaces anteriores son idénticos, por lo que escoger al receptor 2 frente al 3,4,6,7 u 8 o al 11 frente al 9 o al 10 es producto del orden de llegada de sus informes.

Durante el resto de la comunicación, sólo se producirá un cambio de representante si alguno de los receptores empeora sus métricas por debajo de las estadísticas del representante número 2.

3.4 Comportamiento equitativo con respecto a TCP

Como se ha indicado con anterioridad, una de las características deseables de cualquier nuevo protocolo es que no se comporte de manera injusta con el tráfico ya existente en Internet (que principalmente es tráfico TCP). Lo que se entiende por comportamiento justo con respecto a TCP está sujeto a discusión, considerándose como requisito mínimo el que los nuevos protocolos no provoquen la caída de la velocidad de transmisión de TCP a un valor cercano a 0 y como requisito máximo el acercarse al comportamiento de TCP, tal como se define en [7].

Para medir lo equitativo del comportamiento de un protocolo con respecto a TCP, se han definido numerosas métricas, de las cuales destacamos el índice de justicia de [13], definido como:

$$\text{Índice de justicia} = \frac{\left(\sum_{i=0}^{n} X_i(t)\right)^2}{n * \left(\sum_{i=0}^{n} X_i(t)^2\right)} \qquad (3)$$

donde n es el número de agentes emisores (TCP y PMFCC), y $X_i(t)$ es el número medio de bits transmitidos en el intervalo de tiempo que

comienza en t y tiene duración fija τ (normalmente, τ equivale a dos veces el RTT).

En PMFCC se busca aproximarse lo máximo posible al comportamiento de TCP, para lograr que este nuevo protocolo pueda ser implementado en Internet sin provocar ningún problema. Para ello, como se especificó anteriormente, se sigue un control de flujo mediante ventana, con un algoritmo muy similar al de TCP Reno. En las fig. 5 y 6 se puede comprobar el comportamiento de 4 instancias PMFCC frente a 4 instancias TCP, con la topología mostrada en la fig. 4, mediante los resultados aportados por el simulador ns [12].

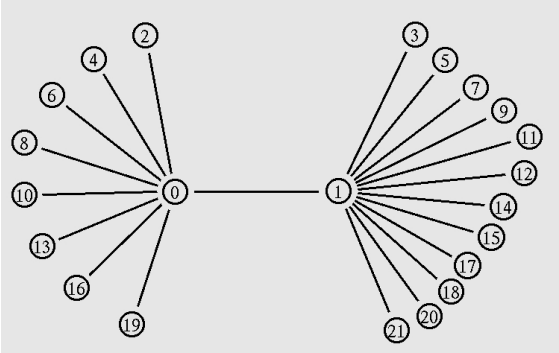


Figura 4: Topología empleada para realizar las gráficas de equidad. Los nodos transmisores (a 600kbps) se encuentran a la izquierda y los receptores a la derecha. El enlace principal, que es el que presenta la congestión, es de 500 kbps, con una cola de 30 paquetes y 50 ms de retardo.

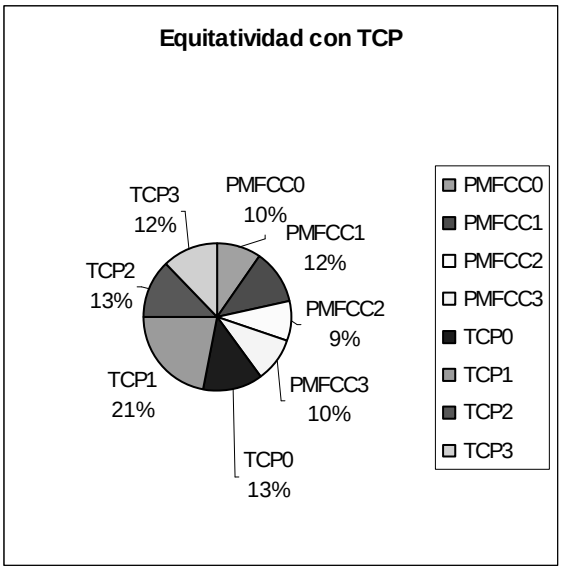


Figura 5: Comparación de 4 instancias TCP Tahoe frente a 4 instancias PMFCC. Con TCP Tahoe se usa una estrategia de ACKs no retardados.

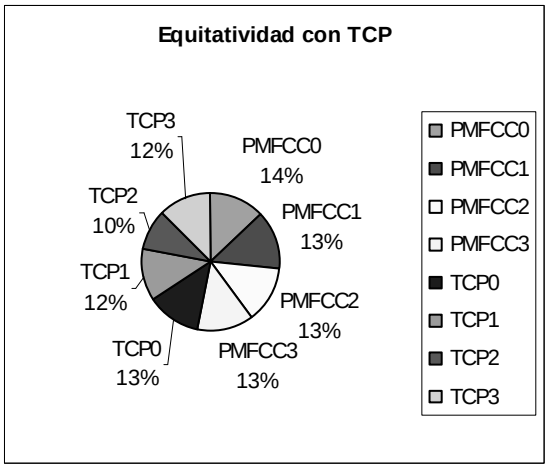


Figura 6: Comparación de 4 instancias TCP Reno frente a 4 instancias PMFCC. Con TCP Reno se usa una estrategia de ACKs retardados.

Puede observarse cómo las gráficas anteriores muestran un comportamiento bastante equitativo entre PMFCC y varias implementaciones de TCP, obteniéndose un índice de justicia del 91,37% cuando se compara con PMFCC con TCP Tahoe y un índice de justicia del 99,26% cuando se compara PMFCC con TCP Reno.

Con respecto a los ACKs, caben dos estrategias: el envío inmediato (estrategia inicial de TCP) o el envío retardado. Esta última presenta una serie de ventajas en cuanto a ancho de banda usado y es recomendada por la IETF [11] en todas las implementaciones de TCP. PMFCC, siguiendo las especificaciones de la IETF para TCP, usa también ACKs retardados.

4. Conclusiones

En este artículo se ha presentado una propuesta para un nuevo protocolo, PFMCC, basado en los últimos estudios sobre esquemas control de congestión unitasa disponibles. Este protocolo no requiere la colaboración de los elementos intermedios de la red, y alcanza sus objetivos de escalabilidad y fiabilidad, usando un esquema de temporizadores y escogiendo al peor receptor como representante, para establecer un lazo de control de flujo y de congestión entre el peor representante y el emisor muy similar al de TCP, lo que permite un comportamiento que cabe calificar de justo con respecto a este último protocolo.

Referencias

[1] Tony Speakman y otros, "PGM Reliable Transport Protocol Specification", *Internet Draft*, http://www.rfc.net

[2] J.C Lin, S. Paul, "RMTP: A reliable multicast transport protocol", *IEEE Infocom 1996*, Marzo 1996

[3] Jorg Nonnenmacher, Ernst W. Biersack, "Scalable Feedback for Large Groups", *IEE/ACM Transactions on Networking*, pp. 375-386, vol. 7(3), Jun. 1999.

[4] L. Rizzo, "PGMCC: A TCP-friendly single-rate multicast congestion control scheme", *ACM SIGCOMM 2000*, Agosto 2000.

[5] Jorg Widmer, Mark Handley, "Extending Equation-based Congestion Control to Multicast Applications", *ACM SIGCOMM 2001*, Agosto 2001.

[6] Jiang Li, Shivkumar Kalyaanaraman, "ORMCC: A Simple and Effective Single-Rate Multicast Congestion Control Scheme", pendiente de publicación.

[7] J. Padhy, V. Firoiu, D. Towsley, J. Kurose, "Modeling TCP Throughput: A Simple Model and its Empirrical Validation", *ACM SIGCOMM 1998*, Agosto 1998.

[8] Jorg Nonnenmacher, Ernst W. Biersack, "Optimal multicast feedback", *IEEE Infocom 1998*, Marzo 1998.

[9] Jorg Nonnenmacher, "Reliable Multicast Transport to Large Groups", *Tesis doctoral en el Insitituto Eurécom*, 1998.

[10] T. Friedman, D. Towsley, "Multicast session membership size estimation", *IEEE Infocom 1999,* Marzo 1999.

[11] M. Allman y otros, "TCP Congestion control", *Internet Draft*, http://www.rfc.net

[12] Ns development team, "Ns manual", http://www.isi.edu/nsnam/ns/ns-documentation.html

[13] Gianluca Iannaccone, Luigi Rizzo, "Fairness of a single-rate multicast congestion control scheme", *Sprintlabs Technical Report*, disponible en http://www.sprintlabs.com/People/gianluca/papers/fairness.pdf

Alternativas para el Control de Congestión en Redes Multicast

Miguel Rodríguez Pérez Manuel Fernández Veiga Cándido López García Sergio Herrería Alonso

Dept. de Ingeniería Telemática. Universidad de Vigo
E.T.S.E. Telecomunicación. Campus Universitario Lagoas-Marcosende s/n.
36 200 Vigo, España
Teléfono: +34 986 81 39 01. E-mail: Miguel.Rodriguez@det.uvigo.es

Abstract *The diffusion of multimedia content across the Internet is still a hot topic of active research. The main difficulty settles in having an efficient transmission protocol that preserves the stability of the current Internet at the same time. In this paper we present our work designing and developing two different protocols aimed at resolving these problems: a multiple layer based one, called LDP and a single-rate one, VLMCC. Both protocols are tuned for the transmission of real-time multimedia content to an unlimited number of receivers. Our simulations show that the protocols are fair against competing TCP traffic present in the network, and have a slowly varying throughput making both protocols suitable for multimedia transmission.*

1 Introducción

La difusión de contenidos multimedia a un gran número de receptores en Internet es todavía un problema al que no se le ha encontrado una solución plenamente satisfactoria. La principal dificultad reside en la elaboración de un protocolo de transporte eficiente y que sea capaz a un mismo tiempo de preservar la estabilidad actual de Internet. En los últimos años se han desarrollado multitud de estudios con este objetivo, obteniéndose como resultado diferentes propuestas que intentan solucionar los distintos problemas existentes. Entre los características que cualquier solución debe poseer podemos citar las siguientes:

Variación suave de tasa Se ha demostrado que los cambios abruptos en la tasa de una conexión TCP como consecuencia de la simple pérdida de un paquete puede degradar severamente la calidad percibida por el usuario en una transmisión multimedia [1, 2]. Por tanto, el protocolo debe ser diseñado de forma que tenga una variación instantánea de su tasa de transmisión lo más suave posible. Las variaciones lentas de la tasa tienen un efecto secundario positivo en los receptores, ya que éstos pueden ajustar más la memoria que dedican a almacenar las tramas antes de mostrarlas, reduciendo de esta manera tanto el consumo de memoria como el retardo en la transmisión.

Escalabilidad Es conveniente que el protocolo pueda ser empleado en una red multicast, para que pueda dar servicio a aplicaciones destinadas a una gran población de estaciones simultáneamente. Este apartado presenta algunos problemas de diseño comunes a todos los protocolos de congestión multicast. Probablemente, el más crítico sean las *implosiones de retroalimentación* que se producen cuando un porcentaje alto de los receptores se ven afectados por un mismo suceso en la red, y se lo comunican al emisor [3]. El protocolo debe prever esto de alguna manera y ser capaz de obtener la suficiente información de todos los receptores sin saturar al servidor.

Equidad Inter-flujos El protocolo debe repartir el ancho de banda disponible de una manera ecuánime entre todos los flujos presentes en la red. De una manera informal, si dos flujos observan las mismas condiciones en la red, entonces deben obtener el mismo ancho de banda. En este artículo nos concentramos en hacer el protocolo compatible con TCP, por ser ésta la clase de tráfico predominante en Internet [4]. Nótese que existen aplicaciones y protocolos que no realizan este reparto ecuánime (por ejemplo: UDP, RTP) y contribuyen a congestionar la red.

Entre las propuestas realizadas a la hora de resolver el problema del control de congestión en redes multicast, se pueden observar dos grandes tendencias: las propuestas *multicapa* y las *monocapa*.
La técnicas *monocapa* envían todos los datos a través de un único grupo multicast. El problema que esto conlleva es que la tasa de transmisión ha de ser válida para todos los miembros de la transmisión, lo que obliga a que ésta no sea superior a la máxima tasa admisible por el receptor con peores condiciones de red en cada instante. Las técnicas *multicapa* no presentan este inconveniente, pues los datos a transmitir son divididos en diferentes capas transmitidas por diferentes grupos multicast de manera independiente. Los receptores, según el ancho de banda del que dispongan, se suscriben al mayor número de grupos posibles, ya que a un mayor número de capas recibidas le corresponde una mayor calidad de reproducción. Un ejemplo de estas

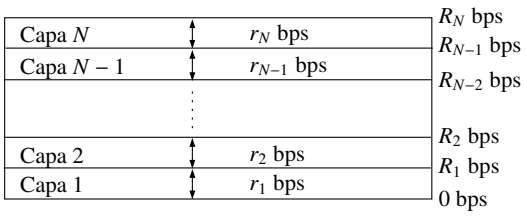


Figura 1: El sistema de capas empleado en la transmisión.

técnicas puede verse en [5].
Por tanto, la principal ventaja de los protocolos *multicapa* es que potencialmente se adaptan a un conjunto de receptores muy diferentes y a distintas rutas de la red de una forma óptima. Por otro lado, las propuestas *monocapa* son mucho más fáciles de analizar y comprobar.
En este artículo se presentan dos protocolos diferentes, uno *multicapa*, el Layered Datagram Protocol (LDP), y uno *monocapa*, el Vegas-like Multicast Congestion Control (VLMCC), que poseen las características anteriores.
El resto del artículo está organizado como sigue. En la Sección 2 se presenta el protocolo LDP y en la Sección 3, el VLMCC. Para finalizar, se muestran nuestras conclusiones en la Sección 4.

2 LDP

LDP es un protocolo de control congestión multicast basado en técnicas *multicapa* y que realiza dos modificaciones fundamentales al esquema propuesto en [6, 5]. En primer lugar, LDP hace uso de algoritmos binomiales [7, 2] a la hora de adaptar su tasa, de manera que minimiza la amplitud de las oscilaciones en la tasa instantánea. En segundo lugar, LDP es un protocolo en bucle abierto en el que los receptores no necesitan intercambiar información de control con el servidor u otros receptores, lo que mejora la escalabilidad. El funcionamiento en bucle abierto también evita los problemas de implosión de retroalimentación [1] que otros protocolos deben evitar a la hora de que el servidor recoja información de estado de todos los clientes.

2.1 Descripción del Protocolo LDP

En este apartado se explica el funcionamiento de LDP. Al tratarse de un protocolo *multicapa*, la información a transmitir se divide en N capas acumulativas diferentes, de tal manera que cuantas más capas reciba una estación mejor será la calidad de la recepción.
LDP no garantiza la recepción de todos los paquetes transmitidos, ya que ello no es necesario para la transmisión de contenidos multimedia, lo cual libera al protocolo de complejidades tales como temporizadores de retransmisión, agregación de asentimientos, etc. Esto simplifica enormemente el diseño del protocolo, al mismo tiempo que permite centrarse en los problemas inherentes al control de congestión.

El concepto del apilamiento de capas se ajusta perfectamente a la transmisión de contenidos multimedia ya que los datos pueden ser codificados de tal forma que las capas superiores simplemente mejoren la resolución de la imagen (o frecuencias superiores de audio, estéreo, ...) De esta manera, los receptores que no puedan acceder a las capas más altas siempre podrán recibir una versión de los contenidos adecuada a sus capacidades.
La Figura 1 muestra una representación de este tipo de transmisiones. Cada capa i tiene asociada una tasa de transmisión r_i. LDP funciona de modo que si un receptor está subscrito a la capa j, entonces también debe estarlo a los grupos multicast que transmitan las capas $i \leq j$. Por tanto, todo receptor recibe $R_j = \sum_{i=1}^{j} r_i$bps.
Como cada capa se transmite por un grupo multicast diferente, todo lo que un receptor debe hacer es estimar una tasa adecuada de recepción (r_{ap}) y unirse a todas las capas $k \leq j$ que cumplan

$$\sum_{i=1}^{j} r_i \leq r_{ap} \leq \sum_{i=1}^{j+1} r_i. \quad (1)$$

2.1.1 El Cálculo de r_{ap}

Para calcular la tasa máxima admisible en cada receptor, LDP emplea algoritmos binomiales [7]. Es bien conocido que estos algoritmos son compatibles con TCP cuando se utilizan para controlar el tamaño de la ventana de congestión. LDP utiliza estos algoritmos para el cálculo de la tasa máxima admisible, de manera que el reparto de ancho de banda con TCP sea ecuánime a largo plazo.
Los receptores LDP usan las pérdidas de paquetes como señal a la hora de decrementar r_{ap}. Esta disminución está limitada a una vez por cada RTT. Si tras un RTT un receptor no detecta ninguna pérdida, entonces r_{ap} se ve incrementado. La ecuación (2) se utiliza para incrementar r_{ap} y (3) se utiliza cuando se detectan pérdidas

$$r_{ap+1} \leftarrow r_{ap} + \alpha / r_{ap}^{k};\ \alpha > 0 \quad (2)$$

$$r_{ap+1} \leftarrow r_{ap} - \beta r_{ap}^{l};\ 0 < \beta < 1. \quad (3)$$

Es necesario que se cumpla $l \leq 1, k + l = 1$, para que el funcionamiento del protocolo pueda ser compatible con el de TCP. Dicha ecuación satisface los requisitos expresados en [8] para que un algoritmo de control de congestión sea compatible con TCP.

2.1.2 Estimación de RTT

En el apartado anterior se mencionó el hecho de que tanto los incrementos en r_{ap} como los decrementos en respuesta a cambios en las condiciones de la red se realizaban a lo sumo una vez por RTT. Fue un verdadero reto diseñar un método para obtener una buena estimación de RTT y, al mismo tiempo, preservar la naturaleza en bucle abierto del protocolo. Este mecanismo se describe a continuación.
Cuando un receptor se une a una transmisión, se subscribe a la capa 1. Sólo una vez que el primer paquete

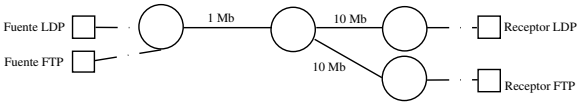


Figura 2: Red simulada.

es recibido, un paquete especial ECHO se transmite directamente al servidor marcado con la hora local del receptor. Este paquete es respondido por el servidor y, una vez que la respuesta haya llegado el nodo tiene una primera estimación de *rtt*. A partir de este momento el receptor ajusta *rtt* según

$$rtt \leftarrow rtt + \Delta_{rtt} \tag{4}$$

cada vez que se reciben dos paquetes consecutivos. Para calcular Δ_{rtt} los receptores hacen uso de un dato enviado por el servidor en todos los paquetes: el tiempo transcurrido desde la emisión del último paquete ($off set$). Por tanto

$$\Delta_{rtt} = ahora - (ultimaLlegada + off set). \tag{5}$$

Para finalizar, a la estimación de RTT se le aplica un filtro paso-bajo que elimina restos de ruido de alta frecuencia presente en las medidas

$$\widehat{rtt} \leftarrow (1-g)\widehat{rtt} + g \cdot rtt. \tag{6}$$

Aquí $\widehat{rtt}$ representa la estimación de RTT y *rtt* denota la última medida de RTT. LDP utiliza $g = 0{,}1$ como ganancia del filtro.

2.2 Resultados de Simulación

Este apartado se muestran los resultados obtenidos al simular el funcionamiento del protocolo LDP con el *ns-2* [9]. Debido a la dificultad de una demostración formal acerca de las propiedades de compatibilidad entre LDP y TCP, las simulaciones realizadas intentan comprobar de manera empírica que esta compatibilidad realmente existe.
Salvo que se indique lo contrario, se ha utilizado la red representada en la Figura 2 para las simulaciones. Esta topología permite experimentar con las interacciones entre un flujo LDP cuando se enfrena a distintas sesiones TCP en un cuello de botella.

2.2.1 Comportamiento Básico de LDP

La Figura 3 muestra la tasa promediada obtenida por un flujo TCP y una sesión LDP configurada como sigue: cuatro capas diferentes de 100, 200, 300 y 400 Kbps respectivamente. Como se esperaba, una vez que el flujo TCP comienza en el segundo 20, el flujo LDP converge rápidamente a 300 Kbps. Esto ocurre porque el receptor sólo puede suscribirse a las dos primeras capas sin obtener más ancho de banda que el que obtuviese TCP. De todas formas, el reparto de ancho de banda puede variarse modificando la relación α/β.
El segundo experimento consiste en comprobar la capacidad de adaptación de LDP a los cambios en el RTT.

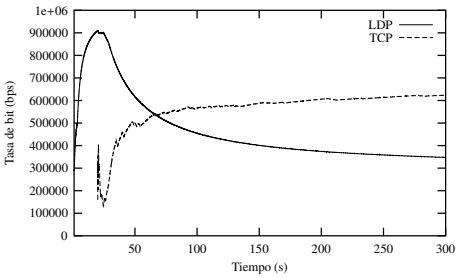


Figura 3: LDP y TCP compartiendo el ancho de banda disponible en un cuello de botella. LDP configurado con $\alpha = 0{,}8$ y $\beta = 1$.

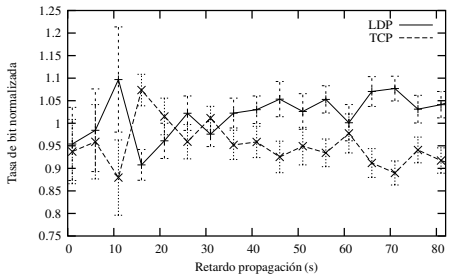


Figura 4: Tasa obtenida por TCP y LDP al variar el retardo de propagación.

Para ello, en el experimento se varía el tiempo de propagación en el enlace cuello de botella, comenzando por 1 ms y finalizando en 100 ms. En la Figura 4 se muestran los resultados normalizados, donde 1 se corresponde con una tasa de 500 Kbps. Para cada valor diferente del retardo de propagación, la simulación fue repetida 20 veces variando ligeramente el instante de comienzo del flujo TCP entre cero y un segundo. La gráfica muestra los intervalos de confianza para una calidad del 95 %. Como puede observarse, los resultados de ambos flujos no divergen al variar el RTT. Esto es consecuencia del algoritmo de estimación del RTT utilizado por LDP.
La siguiente simulación muestra el comportamiento de LDP en un escenario multicast complejo. La Figura 5 representa la topología de red utilizada. Hay un único emisor y cinco receptores diferentes, dos de los cuales están tras un cuello de botella de 1 Mbps, y los otros tres comparten un enlace de 0,5 Mbps.
Los enlaces en la red tienen un tiempo de propagación de 10 ms y todos los routers utilizan Random Early Drop (RED) como mecanismo de descarte de paquetes. Los receptores utilizan un algoritmo SQRT [7] (es decir: $k = l = 0{,}5$) con $\alpha = 0{,}08$ y $\beta = 0{,}06$, y que el emisor transmite diez capas de 100 Kbps cada una.

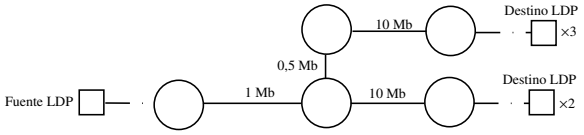


Figura 5: Representación de la red utilizada para la simulación de la Figura 6.

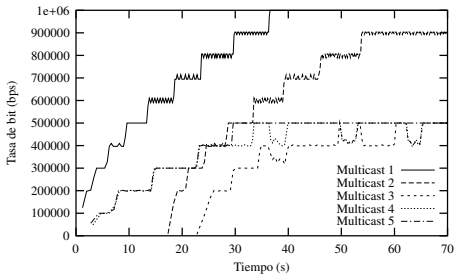


Figura 6: Simulación del protocolo LDP en un escenario multicast complejo.

Por tanto, la capacidad del principal cuello de botella es la misma que la tasa máxima de transmisión.
Tal y como muestra la Figura 6, cada receptor se une a la transmisión en un instante de tiempo diferente, de forma que es posible comprobar como todos convergen a la tasa máxima disponible para cada uno. El hecho de que sólo el primer receptor de cada ruta de la red sea capaz de utilizar exactamente la tasa máxima disponible se debe a que ésta coincide con la capacidad máxima de la red, de forma que la simple pérdida de un paquete evita que todos los receptores puedan obtener la tasa máxima. Este detalle no deja de ser anecdótico, ya que se trata de un caso extremo en el que la capacidad de la red coincide con R_i para algún i, y el resultado es una pequeña pérdida de eficiencia.

3 VLMCC

El protocolo VLMCC propuesto en este apartado intenta resolver el problema del control de congestión en redes multicast desde una perspectiva *monocapa*.
Al centrarse en las propuestas *monocapa*, aparecen dos distintas estrategias a la hora de preservar la estabilidad de la red, que, por otra parte, son compartidas con los protocolos unicast de difusión. Por un lado, los protocolos *regidos por ecuación* intentan ofrecer a la red la misma tasa que ofrecería un flujo TCP en las mismas condiciones, usando para ello una fórmula aproximada como ecuación de control. Normalmente emplean alguna variante de

$$T = \frac{s}{RTT\sqrt{\frac{2p}{3}} + 3t_{RTO}\sqrt{\frac{3p}{8}}p\,(1+32p^2)} \qquad (7)$$

siendo T la tasa, s el tamaño de los paquetes, p la probabilidad de pérdidas y t_{RTO} el tiempo de retransmisión. La principal dificultad surge del hecho de tener que disponer de medidas fiables de todos los parámetros presentes en la fórmula. Es por ello que estos protocolos tienen problemas a la hora de adaptarse a cambios rápidos en las condiciones de la red [10], pues necesitan bastante tiempo para obtener medidas fiables de todos los parámetros. Ejemplos de estos protocolos podemos encontrarlos en [11] para redes unicast o en [12, 1] para redes multicast.
Por otro lado los *emuladores-TCP* procuran emular directamente el comportamiento de dicho protocolo. Su principal problema en redes multicast es la obtención de suficiente información acerca de todos los receptores sin saturar al servidor, ya que el funcionamiento del algoritmo de control de congestión de TCP se basa en el intercambio continuo de información entre extremos para poder detectar la congestión.

En esta Sección presentamos un nuevo *emulador-TCP* basado en las ideas de TCP-Vegas y cuya principal característica reside en que utiliza información de retorno de un solo receptor, con lo que resuelve el principal problema de este tipo de protocolos. Además, el resto de las estaciones son capaces de adquirir suficiente información para calcular la tasa justa de ancho de banda que pueden permitirse y solicitar al emisor ser el nuevo responsable de enviar la información de retorno si la tasa actual de emisión supera a su tasa justa, solventándose de esta manera los problemas de *implosión de asentimientos* que pudiesen surgir [1, 3].

3.1 Decisiones de Diseño

Un problema a resolver por los protocolos que utilizan técnicas *monocapa* es cómo averiguar el estado de cada receptor presente en la sesión sin que el emisor se vea saturado por dicha información a medida que la cantidad de receptores aumente. Algunos autores proponen el uso de complicados algoritmos tanto a nivel de los receptores como de la red para agregar la información antes de que llegue al emisor [13]. Otra posibilidad es usar un receptor como representante de los demás [14]. La principal ventaja de este método radica en el hecho de que no introduce ningún retraso adicional a la información, lo que ayuda a disminuir la latencia. Además, estos esquemas no suelen requerir ninguna modificación en los routers de la red, facilitando su implantación. Ésta será la técnica que empleemos en VLMCC.

Otro problema al que nos enfrentamos en nuestro diseño fue el idear algún mecanismo que hiciese a nuestro protocolo compatible con TCP. Para este fin, se decidió emular el comportamiento de TCP entre el servidor y un representante del conjunto de receptores. De esta manera, podemos estar seguros de que si dicho representante se elige de un modo adecuado (por ejemplo, el receptor con peores condiciones de red) el protocolo repartirá el ancho de banda de la red de forma justa con TCP en todos los enlaces. La variante de TCP escogida para nuestra emulación ha sido TCP-Vegas [15], por dos motivos principales: 1) Suele exhibir pequeñas variaciones en su tasa de emisión; 2) TCP-Vegas posee características que facilitan el diseño de los receptores que han de analizar las condiciones de la red. Además, si la tasa de emisión fluctuase de manera amplia (como en el resto de las variantes de TCP), sería más complicado averiguar cuál es la tasa de emisión a largo plazo por los receptores, lo que les llevaría a tener que retrasar sus respuestas a condiciones de congestión hasta que se hubiese producido una medición suficientemente fiable.

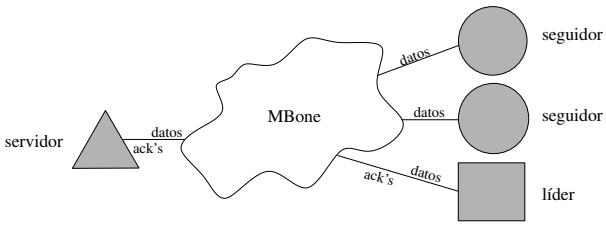


Figura 7: Elementos principales presentes en una transmisión VLMCC.

3.2 Descripción del Protocolo

3.2.1 Funcionamiento General

Comenzaremos proporcionando una descripción general del protocolo, mientras, al mismo tiempo, presentamos a sus principales actores. Hemos representado un escenario simple en la Figura 7. El nodo con forma triangular en la figura representa al nodo servidor. Su principal tarea consiste en enviar paquetes de datos a un grupo multicast y procesar las asentimientos provenientes del líder de la sesión. Monitorizando dichos asentimientos, el servidor debe poder adaptar la tasa de emisión a las condiciones de la red en cada momento, para evitar que se produzca congestión. Explicaremos cómo lo logra en la Sección 3.2.2. El nodo rectangular es el líder de la sesión. Para cada paquete emitido por el servidor, existe un, y sólo un, líder designado responsable de producir el asentimiento correspondiente y remitirlo de vuelta al emisor. Finalmente, los otros dos nodos son ejemplos de estaciones seguidoras que no pueden enviar información de retroalimentación al líder, pero que deben estimar la tasa máxima que pueden aceptar sin causar congestión. Siempre que esta tasa máxima admisible se sitúe por debajo de la tasa actual a la que está emitiendo el servidor, el nodo seguidor deberá solicitar al servidor pasar a ser el nuevo líder de la sesión, proporcionando de esta manera un mecanismo para que el servidor pueda adaptar su tasa al máximo denominador común de entre todos los receptores.

3.2.2 El Tándem Servidor-Líder

El tándem servidor-líder es responsable de llevar a cabo una correcta emulación del comportamiento de una conexión TCP-Vegas que estuviese en funcionamiento entre ellos dos. En lo que concierne a este apartado, asumiremos que el líder ya ha sido designado de alguna manera (véase el apartado 3.2.3).

Por tanto, el servidor actúa como una versión simplificada de un servidor TCP-Vegas, que no proporciona ninguna garantía acerca de la entrega de los paquetes, no teniendo que preocuparse por las retransmisiones. Por contraste, la tarea del líder es mucho más simple, puesto que solamente debe responder con un asentimiento a cada paquete que reciba.

Para emular correctamente el comportamiento de TCP-Vegas, el servidor debe mantener una ventana de congestión (W) y estimaciones tanto del Round Trip Time (RTT) como de su varianza ($\widehat{rtt}$ y $\hat{\sigma}^2_{rtt}$ respectivamente). Tanto $\widehat{rtt}$ como $\hat{\sigma}^2_{rtt}$ son estimados utilizando medias móviles (8) y (9), donde g y g_{σ^2} son factores de ganancia y rtt es el valor de RTT observado en el último paquete asentido

$$\widehat{rtt} \leftarrow (1-g)\,\widehat{rtt} + g \cdot rtt \tag{8}$$

$$\hat{\sigma}^2_{rtt} \leftarrow \hat{\sigma}^2_{rtt} + g_{\sigma^2}\left(\left|\widehat{rtt} - rtt\right| - \hat{\sigma}^2_{rtt}\right). \tag{9}$$

El servidor marca cada paquete enviado con su hora local, para así poder medir de forma precisa el RTT. Este sello temporal se incluye de vuelta en los asentimiento, pero corregido con la cantidad de tiempo que el receptor necesitó para procesar el paquete, de manera que sólo el tiempo de transmisión y los retrasos en las colas de los routers contribuyan a la medida de rtt. Cada $\widehat{rtt}$ el servidor ajusta el tamaño de su ventana de congestión, de acuerdo con las variaciones observadas en el RTT. Para esto, compara la tasa efectiva de envío obtenida ($paquetesEnCamino/\widehat{rtt}$), medida en paquetes por RTT, con el máximo esperado (W/rtt_base), donde rtt_base es el mínimo de las mediciones efectuadas de rtt. Si la diferencia está por debajo de un cierto umbral α, la tasa de emisión puede ser aumentada sin causar problemas a la red, y, por tanto, W se incrementa en un paquete.[1] Por contra, cuando la diferencia es considerada demasiado alta, es decir, cuando es superior a un cierto nivel β, la sesión está comenzando a causar congestión en la red, y, para corregirlo, W se decrementa en un paquete. En cualquier otro caso, el valor de W se mantiene constante.

Existe otra condición bajo la cual el valor de W puede ser modificado. Cada vez que se detecta la pérdida de un paquete (un paquete cuyo asentimiento no llega en los $\widehat{rtt} + 4\,\hat{\sigma}^2_{rtt}$ segundos tras su emisión), el valor de W se reduce a la mitad. Cuando esto ocurre, todas las pérdidas detectadas en el mismo RTT son ignoradas, puesto que la disminución del valor de W no ha tenido tiempo de surtir efecto.

Teniendo presente la exposición anterior, un servidor VLMCC puede enviar nuevos paquetes a la red cada vez que se cumplen todas las siguientes condiciones:

1. La aplicación tiene nuevos datos que enviar.
2. La cantidad de paquetes actualmente en la red es menor que el tamaño de la ventana de congestión.
3. Un tiempo adecuado ha transcurrido desde que se ha enviado la anterior ráfaga de paquetes. Esta condición es una consecuencia del hecho de que tanto TCP-Vegas como VLMCC intentan evitar enviar ráfagas largas de paquetes a la red. En nuestro protocolo la longitud de la ráfaga ha sido limitada a dos paquetes, tras los cuales el servidor ha de esperar el tiempo

$$espera = \widehat{rtt} \cdot \frac{MaxSegment}{W}, \tag{10}$$

[1] La diferencia es primero normalizada dividiendo $\widehat{rtt}$ entre rtt_base, de forma que tanto α como β puedan ser expresados en bytes o paquetes y no en bytes por segundo o paquetes por segundo. Esto permite que los diferentes parámetros de configuración no dependan de las características físicas de la red, como la longitud de los enlaces o sus capacidades.

donde *MaxSegment* es el tamaño máximo de un paquete, antes de enviar algún nuevo paquete. En cualquier caso, siempre es posible enviar una ventana completa de paquetes en la mitad de un RTT.

3.2.3 Los Seguidores

Los seguidores son nodos que están experimentando mejores condiciones que el líder. Por tanto, su trabajo consiste en monitorizar las condiciones actuales de la red y, en caso de que dichas condiciones empeoren, avisar al servidor solicitando pasar a ser el nuevo líder. Llamamos a esto *retar al servidor*.

La parte complicada es, por supuesto, monitorizar correctamente las condiciones de la red; sobre todo porque los seguidores no pueden enviar ninguna información al servidor de forma regular, ya que eso impediría que el protocolo pudiese escalar satisfactoriamente. El método que utilizan los seguidores para estimar las condiciones de la red consiste en calcular el tamaño de la ventana de congestión que tendría una conexión TCP-Vegas en sus mismas circunstancias.

Para ello, siguen el algoritmo descrito en la Sección 3.2.2, pero con una dificultad añadida: no pueden conocer directamente el valor de *rtt*, así que necesitan pedirle una pequeña ayuda al servidor y no calcular *rtt* directamente, sino su variación (Δ_{rtt}) entre dos paquetes consecutivos, para lo que hacen uso del algoritmo ya descrito en la Sección 2.1.2.

Con toda esta información, el seguidor puede calcular un valor para W que será utilizado indirectamente para decidir cuándo se debe retar al servidor. La última palabra, en todo caso, será para éste último, ya que puede ignorar el reto si un nuevo líder ha sido elegido en los últimos RTTs, dejando así tiempo a que la nueva situación se estabilice, o si considera que el nuevo líder tendría unas condiciones de red que degradasen demasiado la calidad de la transmisión, por lo que debe abandonarla.

En cada paquete enviado, existe también información acerca de la tasa de envío actual del servidor ($W_{servidor}/\widehat{rtt}_{servidor}$). Los seguidores utilizan está información para calcular una media móvil de la diferencia entre su tasa esperada ($W_{seguidor}/\widehat{rtt}_{seguidor}$) y la tasa anunciada por el servidor. Cuando la diferencia crece y sobrepasa el valor equivalente de añadir N_{extra} paquetes a $W_{seguidor}$, se genera un nuevo reto, es decir, siempre que

$$\frac{W_{servidor}}{\widehat{rtt}_{servidor}} > \frac{W_{seguidor} + N_{extra}}{\widehat{rtt}_{seguidor}} \qquad (11)$$

El valor N_{extra} es un compromiso entre una transmisión perfectamente ecuánime en el reparto del ancho de banda en todos los enlaces, pero con una frecuencia alta de cambio de líder, y una que puede tomar algo más de ancho de banda del que estrictamente le corresponde, aunque siendo mucho más estable. En todo caso, la cantidad extra de ancho de banda consumido está acotada por N_{extra}, con lo que VLMCC puede ajustarse para ser tan ecuánime como se estime oportuno.

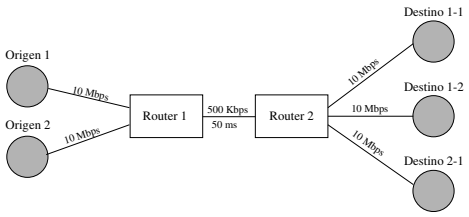


Figura 8: Topología de simulación (los tiempos de propagación para los enlaces donde no se ha especificado ninguno es 0. Se podrían haber utilizado tiempos de propagación mayores, pero sólo estamos interesados en el modo en que se compartían los recursos en el cuello de botella). Los enlaces utilizan colas FIFO.

3.2.4 Establecimiento de la Conexión

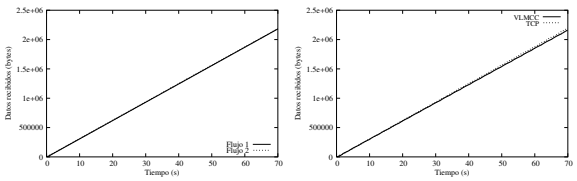
Hasta el momento hemos dejado de lado el problema de cómo se comporta el servidor al comenzar a transmitir, cuando no existe ningún líder designado. El servidor utiliza para este fin los paquetes especiales que envían los nodos para obtener su estimación inicial del *rtt*. Simplemente tiene que escoger como líder al primer nodo del que recibe uno de estos paquetes. Otro problema es el mecanismo necesario para que los nodos puedan abandonar la sesión. Hemos decidido ignorarlo por el momento, ya que nuestro principal interés reside en el funcionamiento del algoritmo de control de congestión.

3.3 Resultados Experimentales

En este apartado presentamos los resultados de las simulaciones del protocolo VLMCC realizadas con el simulador *ns-2* [9]. Comprobaremos las propiedades básicas del protocolo con la topología representada en la Figura 8, para luego comprobar que el protocolo comparte de forma equitativa el ancho de banda con otros protocolos en un escenario más complejo.

En nuestra primera simulación comparamos la cantidad de tráfico satisfactoriamente entregado por dos sesiones VLMCC que comparten la red de la Figura 8. El nodo etiquetado como *Origen 1* actúa como servidor de la primera conexión VLMCC, y *Origen 2* es el origen del segundo flujo VLMCC. *Destino 1-1* y *Destino 1-2* son dos nodos que reciben datos de la primera sesión VLMCC, y, finalmente, *Destino 2-1* es el receptor de los paquetes de la segunda sesión.

Los resultados pueden observarse en la Figura 9(a), y son completamente satisfactorios. Ambas sesiones obtienen exactamente la misma cantidad de ancho de banda sin importar la cantidad de receptores subscritos a cada una de las sesiones, como puede apreciarse observando la pendiente de las curvas. La Figura 9(b) muestra los resultados obtenidos al reemplazar al segundo servidor VLMCC por una fuente TCP, de forma que exista un flujo TCP entre *Origen 2* y *Destino 2-1*. Aunque los resultados son muy similares a los mostrados en la Figura 9(a), ahora ambos protocolos no obtienen exactamente la misma cantidad de ancho de banda, si bien la diferencia es ínfima.



(a) (b)

Figura 9: Datos recibidos por dos conexiones VLMCC (a) y una conexión VLMCC y un flujo TCP (b) cuando comparten un único cuello de botella.

Tabla 1: Pérdidas acumuladas (bytes) por las conexiones de la Figura 9(b).

Flujo	**1 s**	**24 s**	**47 s**	**70 s**
TCP	6 042	13 520	21 840	29 120
VLMCC	1 056	1 056	1 056	1 056

La Tabla 1 muestra las pérdidas acumuladas por ambas conexiones. Vemos que para el flujo TCP se producen pérdidas durante toda la simulación y a un ritmo prácticamente constante (si descontamos los instantes iniciales de la conexión, donde se producen más pérdidas al finalizar el *slow-start*), mientras que en el caso de VLMCC tan sólo se produce la pérdida de un paquete al comienzo de la simulación (aunque no se ve en la tabla, sucede en el segundo 0,609). Esto sucede porque mientras que TCP-Reno necesita de las pérdidas para ser capaz de ajustar su tasa, TCP-Vegas, y por ende VLMCC, pueden detectar indicios de congestión antes de que ocurra, y evitar así pérdidas innecesarias de paquetes.

La Figura 10 muestra una topología usada para comprobar las propiedades de estabilidad de VLMCC. Un flujo VLMCC emite durante toda la simulación a cuatro receptores diferentes. Tras 10 segundos de simulación se establece una conexión TCP entre la fuente TCP y *Destino TCP 1*. En el segundo 25 dicha conexión es cerrada y dos segundos después se establece una nueva conexión TCP, esta vez con *Destino TCP 2* como destino. En el segundo 40 la conexión con *Destino TCP-1* es restablecida, y ambas se cierran en el segundo 50.

En la Figura 11 se muestran los resultados de la simulación. La pendiente de ambas curvas es prácticamente idéntica, lo que indica que cada conexión obtiene la misma cantidad de ancho de banda disponible en el enlace cuello de botella. Las líneas verticales representan retos de los seguidores al líder de la sesión VLMCC. Vemos que algunos retos ocurren al principio de la simulación, cuando aún no se ha elegido un líder adecuado, y también cada vez que varían las condiciones de la red, debido a la presencia de nuevos flujos TCP en diferentes partes de la red. Unos instantes después de cada cambio, VLMCC alcanza un estado de régimen permanente en el que ya no suceden más retos.

La última simulación presentada en este artículo pre-

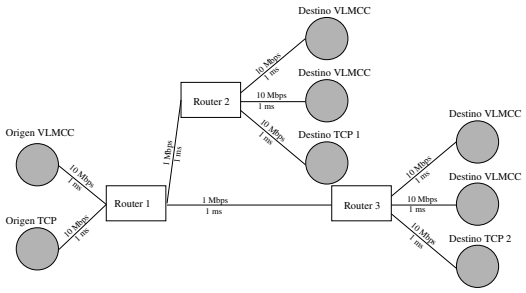


Figura 10: Topología usada para comprobar las propiedades de estabilidad de VLMCC. Todos los routers utilizan simples colas FIFO.

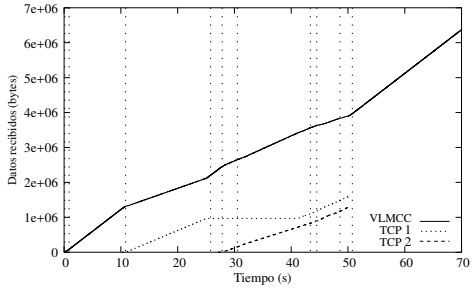


Figura 11: Datos recibidos para una sesión VLMCC y dos conexiones TCP en la red de la Figura 10.

tende mostrar el comportamiento del protocolo en un escenario realista (similar al usado en [7]), como es el representado en la Figura 12. En ella se establece una sesión VLMCC entre una fuente y tres distintos receptores, a la vez que una conexión TCP, tráfico UDP y algún tráfico HTTP consumen recursos de la red. La Figura 13 muestra que, incluso en una situación tan complicada como la descrita, VLMCC es capaz de comportarse de una manera ecuánime con el flujo TCP.

4 Conclusiones

Hemos presentado dos protocolos de control de congestión orientados a la creación de aplicaciones de difusión de contenidos multimedia de tiempo real en redes multicast. Aunque ambos protocolos son similares en espíritu e intentan que los receptores funcionen de la forma más independiente posible, presentan diferencias fundamentales. Así, LDP utiliza una estrategia *multicapa* a la hora de transmitir la información, lo que le permite adaptarse a un mayor espectro de re-

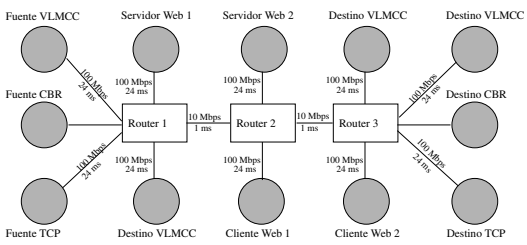


Figura 12: Topología de simulación. Todos los routers utilizan colas FIFO.

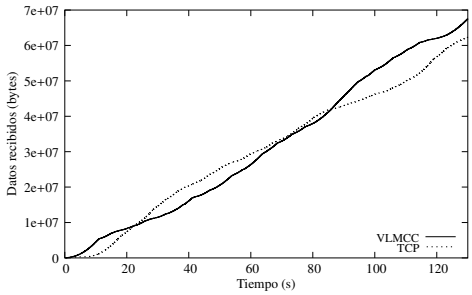


Figura 13: Datos recibidos por las conexiones VLMCC y TCP en la red de la Figura 12.

ceptores y configuraciones de red simultáneamente, a cambio de una mayor complejidad a nivel de red y de tener una resolución limitada a la hora de compartir el ancho de banda con flujos TCP, estando ésta limitada por el número de capas existentes y sus tasas nominales. Por otro lado VLMCC emplea una estrategia *monocapa* en la que emula directamente el funcionamiento de TCP, lo que nos permite tener una confianza mayor en su compatibilidad con TCP. Por otro lado, al ser una técnica *monocapa*, puede ser empleado en redes sin características multicast y, aunque en ese caso se pierde eficiencia, el algoritmo de control de congestión sigue siendo válido. Por lo demás, hemos mostrado en las simulaciones de ambos protocolos que los dos se adaptan muy bien a la transmisión multimedia, ya que minimizan las oscilaciones en la tasa de transmisión, permitiendo menor latencia en las comunicaciones bidireccionales y un menor uso de memoria en los búferes de recepción.

Agradecimientos

Este trabajo ha sido subvencionado por el proyecto: «TIC2 000-1 126 del Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica».

Referencias

[1] B. Widmer y M. Handley, "Extending equation-based congestion control to multicast applications," en *Proceedings of the 2001 conference on applications, technologies, architectures, and protocols for computer communications*. ACM Press, 2001, pág. 275–285.

[2] N. Feamster, D. Bansal, y H. Balakrishnan, "On the interactions between layered quality adaptation and congestion control for streaming video," en *11th International Packet Video Workshop*, Abr. 2001.

[3] Y. R. Yang y S. S. Lam, "Internet multicast congestion control: A survey," en *Proceedings of ICT 2000*. Acapulco, Mexico: ICT 2000, May 2000.

[4] S. Floyd y K. Fall, "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Transactions on Networking (TON)*, vol. 7, no. 4, pág. 458–472, 1999.

[5] S. McCanne, V. Jacobson, y M. Vetterli, "Receiver-driven layered multicast," en *ACM SIGCOMM*, vol. 26,4. New York: ACM Press, Ago. 1996, pág. 117–130.

[6] N. Sacham, "Multipoint communication by hierarchically encoded data," en *Proceedings of the eleventh annual joint conference of the IEEE computer and communications societies on One world through communications (Vol. 3)*. IEEE Computer Society Press, 1992, pág. 2107–2114.

[7] D. Bansal y H. Balakrishnan, "TCP-friendly congestion control for real-time streaming applications," *MIT Technical Report, MIT-LCS-TR-806*, 2000.

[8] N. R. Sastry, "Application specific unicast congestion control," Master's thesis, Department of Computer Sciences, University of Texas at Austin, Dic. 2001, also available as Technical Report TR-01-51, Department of Computer Sciences, The University of Texas at Austin.

[9] NS, "ns Network Simulator," Feb. 2003. [Online]. Disponible: http://www.isi.edu/nsman/ns/

[10] D. Bansal, H. Balakrishnan, S. Floyd, y S. Shenker, "Dynamic behavior of slowly-responsive congestion control algorithms," en *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM Press, 2001, pág. 263–274.

[11] S. Floyd, M. Handley, J. Padhye, y J. Widmer, "Equation-based congestion control for unicast applications," en *Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*. ACM Press, Ago. 2000, pág. 43–56.

[12] M. Luby, V. K. Goyal, S. Skaria, y G. B. Horn, "Wave and equation based rate control using multicast round trip time," en *ACM SIGCOMM*, Jan 2002, pág. 191–204.

[13] I. Rhee, N. Balaguru, y G. N. Rouskas, "MTCP: Scalable TCP-like congestion control for reliable multicast," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 38, no. 5, pág. 553–575, 2002.

[14] D. DeLucia y K. Obraczka, "Multicast feedback suppression using representatives," en *INFOCOM (2)*, 1997, pág. 463–470.

[15] L. S. Brakmo, S. W. O'Malley, y L. L. Peterson, "TCP Vegas: New techniques for congestion detection and avoidance," *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 4, pág. 24–35, 1994.

Encaminamiento Multicast Eficiente en Extensiones Ad Hoc a Redes IP Fijas: El Protocolo MMARP

Pedro M. Ruiz, Antonio Gómez-Skarmeta, Pedro Martínez Asensio
Departamento de Ingeniería de la Información y las Comunicaciones. Universidad de Murcia
Campus de Espinardo. Facultad de Informática
30071 Espinardo (Murcia)
Teléfono: 968 36 46 76 Fax: 968 36 41 51
E-mail: {pedrom, skarmeta, pma}@dif.um.es

Abstract. *Most of the typical IP multicast protocols which are used in fixed IP networks, like IGMP, assume that the terminals are in the same link. However, the multi-hop nature of ad hoc network extensions prevents standard-IP nodes from taking part in IP multicast communications through the ad hoc network. We propose a multicast architecture in combination with a new ad hoc multicast routing protocol called MMARP. MMARP nodes are challenged with special IGMP-handling capabilities allowing our solution to combine the efficiency of multicast ad hoc routing protocols with the support of standard-IP nodes without compromising the performance of the protocol. The use of the IGMP protocol as the interface between standard IP nodes, the fixed network and the ad hoc network extension allows a ready deployment of this approach in existing IP multicast networks.*

1 Introducción

IP Multicast ofrece comunicaciones multipunto eficientes entre un grupo de nodos y ha emergido como una de las áreas de investigación más trabajada dentro de las redes de comunicaciones. El problema de la distribución de datagramas a un grupo dinámico de receptores ha sido investigado desde los años 80 y la mayoría del equipamiento de red existente hoy día soporta los protocolos básicos de IP multicast. La principal ventaja de IP multicast es la enorme reducción del consumo de ancho de banda que consigue. Esto es especialmente interesante para las redes móviles e inalámbricas 'beyond 3G' en las que el número de terminales se espera que sea elevado y las aplicaciones sean principalmente multimedia y tenderán a consumir bastante ancho de banda. En este marco, IP multicast puede suponer un valor añadido importante para un operador dada la reducción de costos y posibilidad de nuevos servicios que requieran este tipo de transporte.

El proyecto IST MIND (Mobile IP-based Network Developments)[1] ha investigado la idea de emplear redes de acceso ad hoc inalámbricas basadas en el protocolo IP. La arquitectura general de red se basa en un núcleo IP que interconecta las diferentes redes de acceso. De entre las diferentes redes de acceso posibles, las 'Mobile Ad hoc Networks' (MANETs) han despertado gran interés como camino hacia redes 'beyond 3G' ya que permiten ampliar el área de cobertura sin necesidad de añadir nueva infraestructura. En esta extensión ad hoc, un terminal de usuario puede emplear los de otros usuarios como routers para obtener caminos multisalto hacia el núcleo de red. Un ejemplo de esta idea de redes 'beyond 3G' se muestra en la Fig.1.

La provisión de comunicaciones multicast en este tipo de extensiones ad hoc conectadas a redes IP fijas resulta bastante más complejo que en las redes IP tradicionales. A la complejidad de los protocolos de encaminamiento multicast para MANETs habría que añadir en este caso, la interacción con los protocolos

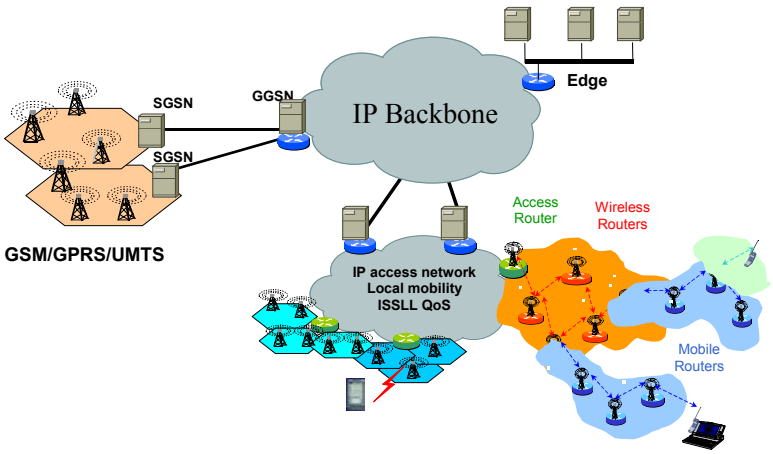


Figura. 1. Esquema de una futura red 'beyond 3G'

empleados en la red fija y el soporte de terminales IP estándar, que no están normalmente soportados por los protocolos de encaminamiento ad hoc.

Se han propuesto en la literatura muchos protocolos [1] de encaminamiento multicast para redes IP fijas. Sin embargo, estos protocolos no son capaces de ofrecer un buen rendimiento en redes ad hoc debido a que las estructuras de distribución que emplean son muy frágiles y su actualización muy lenta, como para soportar los frecuentes cambios en la topología que suceden en las redes ad hoc. Para resolver estas limitaciones han aparecido protocolos de encaminamiento multicast específicos para redes ad hoc [2]. Estos protocolos incorporan funcionalidades específicas para resolver los problemas que plantean las redes ad hoc, pero hasta el momento éstos protocolos sólo han sido pensados para funcionar en redes ad hoc puras sin interconexión a redes IP fijas y, por lo tanto, no son capaces de interoperar con redes IP ni de soportar nodos IP multicast estándar.

Hoy día ya se pueden encontrar algunos trabajos como [5], que consideran la interconexión de redes ad hoc a redes IP. Por el momento las soluciones presentadas en este sentido sólo se centran en la provisión de comunicaciones unicast y no ofrecen una interacción con la red fija para enviar tráfico multicast. Aún así, tampoco permiten que la extensión ad hoc se convierta en una red de acceso para terminales que no se encuentren en la zona de cobertura de los routers de acceso.

En este artículo presentamos el Multicast MAnet Routing Protocol (MMARP), un nuevo protocolo de encaminamiento multicast para redes ad hoc que es capaz de ofrecer encaminamiento eficiente en la red ad hoc a la vez que ofrecer, sin una penalización significativa en el rendimiento, una interacción adecuada con la red fija y con nodos IP estándar que pudiesen emplear la extensión ad hoc como una red de tránsito hacia la red fija. Para ello los nodos MMARP se extienden con la posibilidad de interceptar y procesar los mensajes IGMP[6], a la vez que posibilitan que los nodos IP multicast estándar que se encuentren en la red ad hoc participen en sesiones multicast exactamente como lo harían si estuviesen conectados a la red fija.

El resto del artículo se organiza como sigue: la sección 2 analiza los problemas y requerimientos así como la arquitectura multicast propuesta para extensiones ad hoc a redes IP fijas. La descripción del protocolo MMARP se encuentra en la sección 3. La sección 4 presenta unos resultados empíricos del empleo de MMARP en una maqueta real. Finalmente, la sección 5 recoge algunas conclusiones y describe líneas de actuación futuras.

2 Extensiones Ad hoc a Redes Fijas

Las extensiones ad hoc a redes IP fijas se han propuesto en el marco del proyecto IST-MIND para facilitar las comunicaciones entre nodos en la red de IP fija y nodos IP que pudiesen estar fuera del radio de cobertura de las estaciones base. La creación de este tipo de extensiones ad hoc formadas espontáneamente es muy económica para el operador debido al ahorro en infraestructura. Por supuesto, el ancho de banda en este tipo de redes es más limitado y variable que en un enlace inalámbrico tradicional, pero el área de cobertura es bastante más elevada. En estos escenarios es precisamente donde IP multicast puede ayudar en gran medida a reducir el consumo de ancho de banda sobre todo para las aplicaciones multimedia de tiempo real.

2.1 Requisitos

Los nodos ad hoc normalmente disponen de un pila de protocolos diferente a un router o terminal IP. De hecho, algunos protocolos de encaminamiento ad hoc no usan direcciones IP. Para marcar la diferencia entre este tipo de nodos, emplearemos el término ‘nodo IP estándar’ para denotar a un terminal o router IP empleando protocolos estándar, mientras que denominaremos ‘nodos ad hoc’ a los terminales ad hoc, cuya pila de protocolos es específica para redes ad hoc.

La interacción de las extensiones ad hoc con la red de acceso y con los terminales IP estándar plantea una serie de requisitos que no satisfacen actualmente ni los protocolos de encaminamiento multicast empleados en las redes fijas, ni los protocolos de encaminamiento multicast que se han propuesto hasta la fecha para redes ad hoc.

Nuestro objetivo para este tipo de extensiones ad hoc es ser capaces de llegar a un compromiso en el que al menos se satisfagan los siguientes requisitos:

- Interoperabilidad con el modelo IP multicast empleado en las redes fijas.
- Eficiencia, escalabilidad y baja sobrecarga de control.
- Tolerancia a fallos y robustez. Por ejemplo, que se pueda disponer de diferentes puntos de acceso a la red fija dentro de una misma extensión ad hoc.
- Compatibilidad con los protocolos de encaminamiento multicast entre dominios.

2.2 Problemas a superar

Para los terminales el proceso de participación en sesiones multicast es muy sencillo. Para enviar datagramas multicast es suficiente con enviar datagramas usando como dirección IP destino una dirección de grupo (Ej. en el caso de IPv6 en el rango 224.000 a 239.255.255.255). Para recibir tráfico multicast los terminales han de emplear el Internet Group Management Protocol (IGMP[6]) para solicitar a su router multicast, situando en su misma subred IP, su interés en unirse a ese grupo. Estos mecanismos no son soportados por los protocolos de

encaminamiento multicast propuestos hasta el momento. Como veremos, aparecen una serie de problemas que dificultan el empleo de este modelo.

1)Problemas con el Time To Live

El protocolo IGMP se basa en el empleo de datagramas IP multicast que se transmiten con un Time to Live (TTL) igual a uno porque los routers multicast se supone que están a un único salto de los terminales. Dada la naturaleza 'multihop' de las redes ad hoc, éstos datagramas IGMP no podrían transitar la extensión ad hoc para llegar al router de acceso situado en la red fija..

2)Naturaleza 'multihop' de las redes ad hoc

Los paquetes enviados por los emisores que se encuentren a más de un salto de la red fija no serán recibidos automáticamente por el router de acceso. Sin embargo, los nodos intermedios tendrían que garantizar esto, ya que es un requerimiento para los protocolos de encaminamiento de la red fija. Por ejemplo, en el caso de Protocol Independent Multicast Sparse Mode (PIM-SM [7]), ésta es la única forma de que el router de acceso registre a esas fuentes en el Rendezvous Point (RP).

Permitir que los terminales IP estándar puedan usar la red ad hoc como una red de acceso hacia la red fija, requiere que los nodos ad hoc sean capaces de procesar los mensajes IGMP, ya que ésta es la forma en la que los nodos IP estándar se unen a los grupos multicast. Sin embargo, los protocolos de encaminamiento para redes ad hoc propuestos hasta la fecha asumen que todos los nodos son ad hoc y no contemplan la existencia de terminales IP tradicionales.

3)Direccionamiento plano de las redes ad hoc

Otro problema adicional es la diferencia en el modelo de direccionamiento entre las redes ad hoc y las redes fijas. Mientras que en Internet es necesario emplear un direccionamiento jerárquico en el que las direcciones IP tienen un significado topológico, en las redes ad hoc se usa un direccionamiento totalmente plano. El principal problema es que los routers multicast aplican a cada paquete entrante el llamado 'RPF-Check'. Este proceso descarta todo paquete que llegue por un interfaz diferente al que el router usaría para llegar a la fuente de dicho paquete.

2.3 Arquitectura Propuesta

Durante el proyecto MIND evaluamos diferentes alternativas para conseguir un soporte eficiente de comunicaciones multicast entre nodos de la extensión ad hoc y los nodos en la red fija. Básicamente se pueden agrupar en dos enfoques: emplear túneles entre los nodos IP estándar y el router de acceso o emplear un protocolo específico de encaminamiento multicast en la extensión ad hoc. Este segundo enfoque es el seleccionado ya que, como demostramos en [7], resulta más eficiente en términos de escalabilidad, simplicidad y rendimiento.

Como se aprecia en la Fig. 2, la clave para obtener estas ventajas es la idea de confinar en los nodos de la extensión ad hoc toda la nueva funcionalidad necesaria para interaccionar con los nodos IP estándar y la red fija, empleando protocolos estándar para dichas interacciones. De este modo se consigue una independencia total del protocolo de encaminamiento que se emplee en la red fija y permite la participación de los nodos IP estándar sin necesidad de incorporar ningún cambio o modificación a su comportamiento normal.

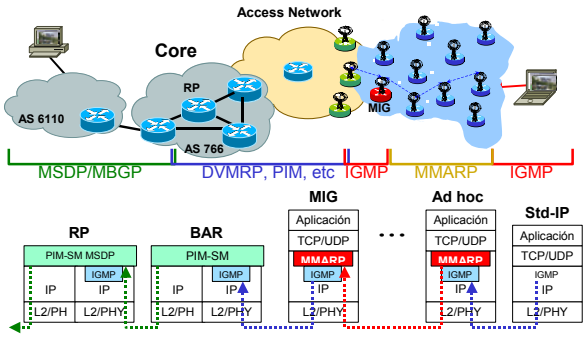


Figura. 2. Arquitectura multicast para extensiones ad hoc

Los protocolos usados en las redes IP fijas como ARP, IGMP, etc, resultan costosos de emplear en una red ad hoc porque suponen una sobrecarga adicional. Para poder soportarlos de un modo eficiente, el propio protocolo MMARP incluirá los mecanismos necesarios para reducir la sobrecarga al mínimo. La descripción detallada de estos mecanismos se encuentra en la siguiente sección.

La Fig. 3. muestra las interacciones necesarias para recibir un flujo multicast según el enfoque propuesto. Como se aprecia, el nodo IP estándar aparece marcado como 'MN', los nodos ad hoc como 'MMR' mientras que el 'BAR' sería el router de acceso a la red fija, y el router identificado como'RP' representa al RP para este grupo multicast, supuesto que estamos empleando PIM-SM. La clave de este planteamiento, es que el protocolo MMARP es capaz de interceptar los mensajes IGMP de todos los nodos de la extensión ad hoc, y hacer llegar al router de acceso los IGMP Reports correspondientes, ofreciendo caminos eficientes dentro de la red ad hoc y sin penalizar en términos de sobrecarga de control.

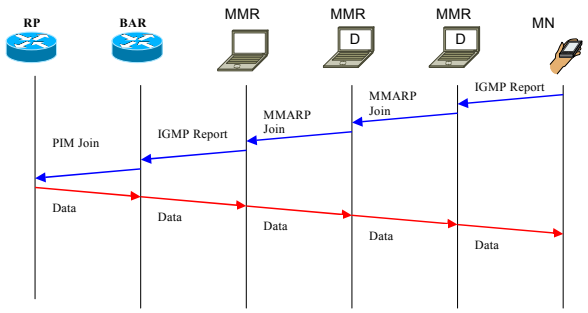


Figura. 3. Interacción para recibir un flujo multicast

3 El Protocolo MMARP

El protocolo MMARP está especialmente diseñado para redes ad hoc móviles (MANETs). Es totalmente compatible con el modelo estándar de IP Multicast [8] y permite que los nodos IP estándar participen en sesiones multicast sin necesidad de ningún cambio a su operación habitual. MMARP es el encargado de procesar y generar los mensajes IGMP necesarios para interoperar tanto con los nodos IP estándar como con los routers de acceso.

La interacción con los routers de acceso se realiza mediante los denominados Multicast Internet Gateways (MIGs). Denominaremos 'MIG' a cada uno de los nodos MMARP que se encuentran situados a un único salto de la red IP fija. Es decir, a un único salto de los routers de acceso. Cualquier nodo MMARP puede convertirse en MIG en cualquier momento. La única diferencia entre un MIG y un nodo MMARP normal es que los MIGs son los responsables de notificar a los routers de acceso los grupos multicast para los que hay receptores interesados dentro de la red ad hoc.

Tal cual funciona IP multicast, al router de acceso no le preocupa qué nodos concretos están interesados en unirse un grupo multicast determinado. Lo que el router necesita saber es si hay algún nodo interesado en unirse o no. Es precisamente esta anonimicidad del servicio IP multicast la que explota MMARP con el concepto de MIG. El MIG a todos los efectos es como un host para el router de acceso, pero en este caso representará los intereses de todos los nodos (tanto ad hoc como IP estándar) que hay en la extensión ad hoc. Se puede decir que el MIG actúa como un host virtual formado por múltiples terminales que emplearán MMARP como protocolo para distribuir eficientemente el tráfico multicast dentro de la extensión ad hoc. De este modo, el protocolo MMARP consigue funcionar con cualquier protocolo de encaminamiento en la red fija, a la vez que lo independiza del protocolo de encaminamiento entre dominios que se emplee.

MMARP usa un modelo híbrido para construir una malla de distribución. Las rutas entre nodos ad hoc se establecen reactívamente (es decir, bajo demanda). Por el contrario, las rutas hacia fuentes o receptores en la red fija se establecen proactívamente mediante un anuncio periódico por parte de los MIGs hacia la red ad hoc.

3.1 Funcionamiento General

MMARP usa una estructura de distribución con topología de malla similar a la empleada por ODMRP. Este tipo de estructura de distribución ofrece una buena protección frente a la rotura de los enlaces causada por la movilidad de los nodos (ver Fig. 4). Tanto la parte reactiva como la parte proactiva del protocolo participan en la creación de dicha malla de distribución.

La parte reactiva consta de una fase de solicitud y otra de respuesta. Cuando un nodo ad hoc tiene nuevos datos que enviar, periódicamente envía mediante un broadcast con TTL de un único salto un mensaje MMARP_SOURCE que se propaga por toda la red ad hoc, para actualizar el estado de los nodos en el camino hacia los receptores así como las rutas multicast. Estos mensajes incluyen en su cabecera un identificador único que permite a los nodos intermedios detectar duplicados y evitar que un mismo nodo distribuya varias veces un mensaje MMARP_SOURCE que pudiera llegarle por diferentes caminos. Cuando un mensaje MMARP_SOURCE llega por primera vez a un nodo, éste almacena la dirección

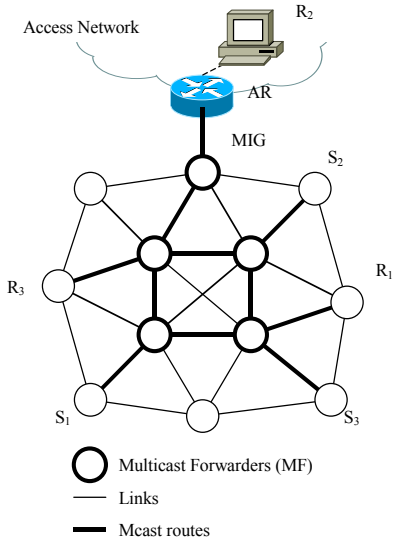


Figura 4. Malla de distribución entre fuentes (S) y destinos (R).

IP del vecino que se lo envió y lo vuelve a reenviar a sus vecinos con un broadcast limitado a un salto. Cuando el mensaje llegue a un receptor o a un nodo ad hoc que se encuentre a un único salto de un nodo IP estándar que se ha unido a ese grupo multicast, el nodo envía de nuevo con un broadcast limitado a un salto un mensaje MMARP_JOIN que incluye la dirección IP del vecino del cual se recibió el MMARP_SOURCE, que será el nodo elegido para formar parte del camino hacia la fuente. Al detectar su dirección IP en un mensaje MMARP_JOIN, un nodo se da cuenta de que está en el camino entre una fuente y un destino. En ese momento activa su Multicast Forwarder Flag (MF_FLAG) para el grupo

multicast correspondiente y envía un mensaje MMARP_JOIN incluyendo la dirección IP del siguiente salto elegido hacia la fuente. Con la repetición sucesiva de este proceso, se consigue crear el camino más corto entre la fuente y el receptor. Los nodos con su MF_FLAG activo para un grupo multicast simplemente reenviarán los datagramas multicast que les lleguen dirigidos a dicho grupo. Al haber múltiples fuentes y receptores, este proceso genera una malla de distribución multicast como la mostrada en la Fig. 4.

La parte proactiva del protocolo consiste en el anuncio periódico del MIG como router por defecto hacia la red fija. Como el TTL de los mensajes IGMP está prefijado en un único salto, la recepción de un mensaje IGMP Query puede emplearse como mecanismo para que los MIGs se den cuenta de que lo son y activen su MIG_FLAG. Los MIGs periódicamente enviarán por broadcast mensajes MMARP_DFL_ROUTE que serán restransmitidos de modo similar a los MMARP_SOURCE y servirán para crear caminos multicast entre los receptores en la extensión ad hoc y los emisores que estén en la red fija. Cuando el mensaje MMARP_DFL_ROUTE llega a un receptor o a un vecino de un receptor IP estándar, dicho nodo comenzará el proceso de unión hacia la fuente enviando un mensaje MMARP_JOIN incluyendo en la cabecera el siguiente salto en el camino inverso al de la propagación del MMARP_DFL_ROUTE. Al igual que en la parte proactiva, el mensaje MMARP_JOIN se va propagando hasta llegar al MIG. Cuando el MIG recibe el mensaje MMARP_JOIN construye un IGMP Report que es enviado al router de acceso para informarle del interés de al menos un nodo por unirse a ese grupo multicast en la red fija.

En el caso particular de que un nodo IP estándar pase a enviar datagramas multicast, el proceso de creación de la malla de distribución es similar sólo que en este caso, como las fuentes IP estándar no han de notificar nada, serán los sus vecinos MMARP más cercanoslos que, al recibir los datagramas multicast, construirán el mensaje MMARP_SOURCE para informar de la existencia de esta nueva fuente en la extensión ad hoc.

El protocolo MMARP incorpora mecanismos para la reparación local de enlaces que permiten solventar las roturas de enlaces que pudieran darse durante la fase de creación de la malla de distribución. Cuando tras cuatro intentos un nodo comprueba que le es imposible entregar un mensaje MMARP_JOIN a su vecino en el camino más corto hacia la fuente, éste envía con un broadcast limitado a un salto un mensaje MMARP_NACK indicando a todos sus vecinos su incapacidad para construir el camino. Sus vecinos, al recibir este mensaje activarán su MF_FLAG y usarán sus propias rutas para proseguir la construcción del camino hacia la fuente. Si alguno de estos nodos no conociese una ruta hasta la fuente, repetiría el proceso enviando de nuevo un MMARP_NACK a sus vecinos. Este mecanismo garantiza que se encontrará un camino hacia la fuente. Si bien este nuevo camino posterior a la rotura del anterior no es óptimo, si que ofrece una recuperación rápida hasta la siguiente actualización de la topología.

Una vez creada la malla de distribución, la distribución del tráfico multicast es muy simple: los paquetes de datos dirigidos a un grupo multicast serán sólo propagados por aquellos nodos MMARP que tengan su MF_FLAG activo para ese grupo. Cuando tales paquetes de datos llegan a un nodo cuyo MF_FLAG para ese grupo aún no ha expirado, éste chequea que no se ha enviado ya antes (para evitar duplicados) y lo reenvía. En cualquier otro caso, el paquete es descartado.

3.2 Soporte eficiente de nodos IP estándar

Los protocolos que usan los nodos IP estándar para realizar sus funciones básicas tales como ARP o IGMP fueron pensados para funcionar en redes de medio compartido BMA (Broadcast Medium Access). Sin embargo, el funcionamiento de las redes ad hoc a nivel de enlace es algo diferente. Mientras que en las redes BMA se garantiza que todos los nodos que son capaces de recibir una trama de otros nodos son capaces también de comunicarse entre sí, esto no se puede garantizar en las redes ad hoc, en las que dos nodos pueden recibir datos de un tercero sin estar éstos últimos alcanzables entre sí.

Para los protocolos de encaminamiento ad hoc usuales, al no soportar terminales IP estándar, esto no es un problema. Cada nodo envía sus propios mensajes de control. Sin embargo, la compatibilidad con el modelo IP multicast estándar obliga a que los nodos MMARP además de sus propios mensajes de control (Ej. MMARP_SOURCE o) tengan que enviar los referentes a sus vecinos IP estándar. Por ejemplo, al recibir un menaje IGMP Report, un nodo MMARP enviaría un MMARP_JOIN para ese grupo multicast en represntación de su vecino. Uno de los problemas que puede aparecer es el hecho de que varios nodos MMARP reciban un mismo IGMP Report y creen diferentes caminos hacia una misma fuente.

El protocolo MMARP ha sido diseñado para evitar la generación innecesaria de caminos. Para ello, los mensajes MMARP incluyen un campo llamado “Generator IP Address” que indica el nodo IP estándar que provocó la generación de un determinado mensaje de control. De este modo, diferentes mensajes MMARP_SOURCE y MMARP_JOIN generados por un mismo mensaje IP estándar, se pueden detectar como duplicados, evitando así la generación innecesaria de caminos.

MMARP tiene unas particularidades que lo diferencian del resto de protocolos de encaminamiento multicast para redes ad hoc que lo hacen no directamente comparable en términos de rendimiento: el soporte de terminales estándar y la

interacción con la red fija. En [10] demostramos analíticamente que MMARP es capaza de ofrecer todas estas nuevas funcionalidades sin apenas sacrificar el rendimiento del protocolo. Para llegar a esta conclusión comparamos el protocolo MMARP con el protocolo ODMRP [4] que es famoso por se uno de los más eficientes para routing multicast en redes ad hoc hasta la fecha. En la siguiente sección analizaremos empíricamente el rendimiento de ambos protocolos.

4 Resultados Empíricos

Para el análisis experimental hemos preparado una maqueta consistente en 6 ordenadores portátiles con tarjetas IEEE 802.11b; cuatro de ellos actuarán como nodos MMARP y el resto como nodos IP estándar que establecerán una sesión IP multicast entre ellos. Las evaluaciones de rendimiento se han realizado con tres topologías diferentes: red ad hoc estática, movimiento de los nodos finales, y movimiento de los nodos MMARP.

En este artículo presentamos los resultados correspondientes al escenario más representativo, en el que un receptor multicast se va a ir desplazando por toda la extensión ad hoc. Los resultados se comparan con los obtenidos en [11] para el protocolo ODMRP en un escenario similar. Dado que la implementación de ODMRP no está disponible, haremos una configuración similar a la que el autor empleó para las pruebas de ODMRP para poder contrastar los resultados al menos de un modo cualitativo.

La información sobre la sobrecarga del protocolo MMARP se va a extraer directamente de los ficheros de registro generados por nuestra implementación de MMARP sobre Linux. Para la información de jitter, paquetes perdidos y demás, emplearemos una herramienta multimedia llamada ISABEL-lite que es una versión reducida para dispositivos ligeros de la herramienta ISABEL [12]. Los datos se extraerán de los logs de la sesión RTP que establece la aplicación.

4.1 Escenario con receptor móvil

La prueba consiste en una red ad hoc con la topología mostrada en la Fig. 5, en la que el receptor (R) se moverá según el patrón de movimiento marcado en la figura. La fuente (S) generará un flujo consistente en un video en formato MJPEG a 6 frames por segundo usando tamaño QCIF y un flujo de audio en formato GSM muestreado a 16 KHz con muestras de 8 bits. Los dos flujos se enviarán al mismo grupo multicast pero a diferentes puertos para el audio y el vídeo.

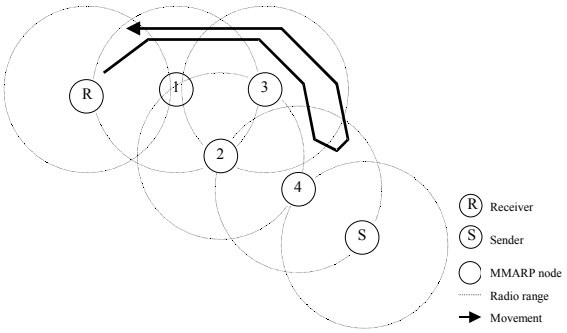


Figura 5. Topology changes in the end-node mobility scenario

Estos flujos generan aproximadamente un flujo continuo de unos 115 Kb/s. El receptor se unirá al grupo multicast al comenzar la prueba, para recibir estos datagramas.

Los datos inicialmente fluirán por el camino <S-4-2-1-R> que es el camino más corto entre la fuente y el destino. Esta situación se mantendrá hasta los 9 segundos de comenzar la prueba, momento en el cual el receptor se encuentra en el rango de cobertura tanto del nodo 1 como del nodo 3. El camino cambia pues a <S-4-2-3-R>. En algún momento entre los 9 y los 12 segundos, el enlace entre el receptor y el nodo 1 se rompe. Desde los 21 segundos hasta los 24 el nodo receptor estará en el radio enlace tanto del nodo 3 como del nodo 4. A los 24 segundos desde el inicio, el enlace entre el receptor y el node 3 se rompe y el camino cambia por tanto a <S-4-R>. A los 33 segundos del comienzo, el enlace entre R y el nodo 3 vuelve a estar disponible. Sin embargo, el camino más corto sigue siendo <S-4-R> por lo que este camino no cambiará hasta el segundo 36 en el que el enlace entre el receptor y el nodo 4 se rompe. A los 45 segundos desde el comienzo de la prueba el enlace R – 1 vuelve a estar disponible, y en el segundo 48, cuando se rompe el enlace R – 3, el mejor camino vuelve a ser <S-4-2-1-R>. Desde este instante hasta el final de la prueba este camino sigue siendo el óptimo.

Los resultados de las pruebas mostrados en la Fig. 6 soportan las mismas conclusiones que se extraen de las pruebas de ODMRP en [11]. Las pérdidas son una constante en las redes ad hoc, así como la variación en la capacidad de la red. Esto se aprecia especialmente en los puntos concretos de cambios de camino. También veremos como los resultados muestran la capacidad de MMARP para reestablecer los caminos rápidamente incluso con la movilidad de los nodos.

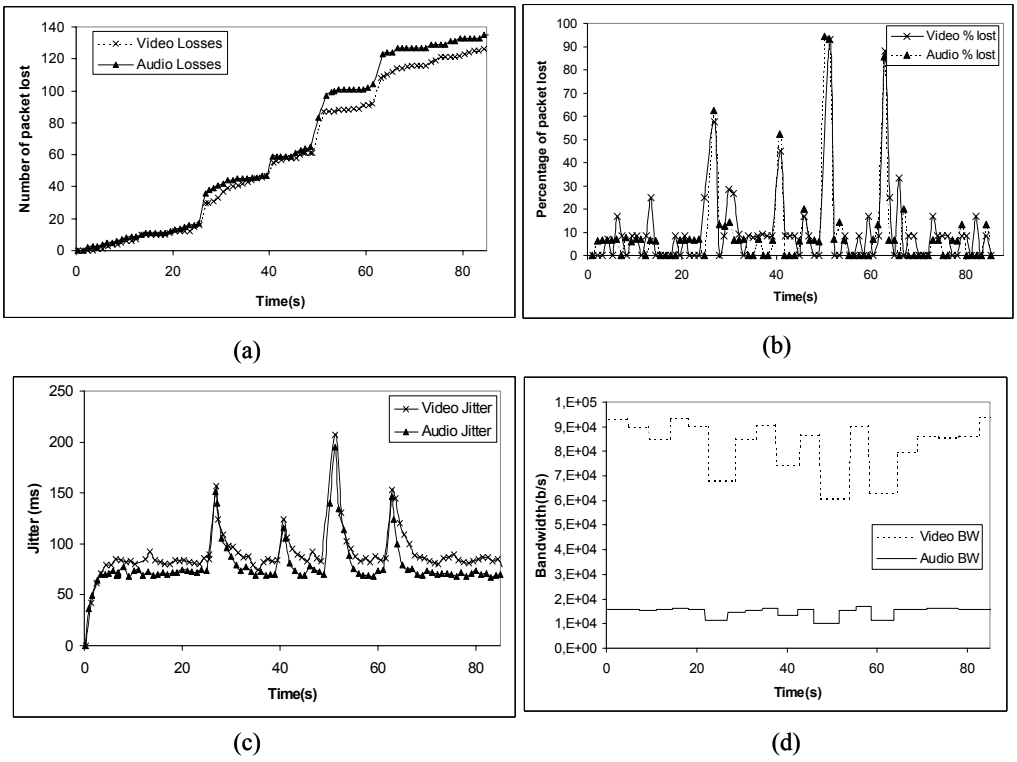


Figura. 6. Pérdidas totales (a), porcentaje de pérdidas (b), jitter (c) y ancho de banda (d)

La Fig. 6(a) muestra la pérdida de paquetes que se ha experimentado. Estas pérdidas corresponden aproximadamente a 1,5 paquetes por segundo tanto para el audio como para el vídeo. Los intervalos con altas pérdidas de paquetes corresponden a los cambios del camino más corto. Se aprecia como MMARP hace que estos periodos de altas pérdidas sean muy reducidos en el tiempo y en general el tiempo hasta crear un nuevo camino es menor a un segundo desde que se rompe el camino anterior. Esto queda claramente demostrado en los picos asociados a los porcentajes de pérdidas de la Fig. 6(b). Como se aprecia las pérdidas en esos instantes concretos son muy altas porque el nuevo nodo que pasa a formar parte del camino óptimo no tiene activo su MF:FLAG, y se ha de esperar al siguiente periodo de refresco hasta obtener un nuevo camino. Este es claramente el peor caso, y en una red de acceso con muchos más grupos activos y muchos más emisores y receptores la probabilidad de que se dé este peor caso es mucho más reducida y podrían llegar a no aparecer pérdidas en los handovers. En lo referente al jitter mostrado en la Fig. 6(c) se aprecia como en los periodos con baja tasa de pérdidas presenta cierta variabilidad propia de las redes ad hoc y la aleatoriedad en el acceso al medio compartido de 802.11b. Como era de esperar, en los periodos en los que se cambia de camino, el interarrival jitter aumenta debido a la pérdida de paquetes intermedios. Lo mismo se aprecia en la Fig. 6(d) en la que el ancho de banda se reduce justo cuando los caminos quedan invalidados por la rotura de los enlaces y hay que buscar nuevos caminos. Es destacable el hecho de que el porcentaje medio de pérdidas se reduce respecto a los resultados en [11] para un testbed real de MMARP. Mientras que en nuestro testbed conseguimos un 9% de pérdidas como media tanto para el audio como para el vídeo, las pruebas con ODMRP mostraron un 29% de pérdidas.

Hemos calculado la sobrecarga del protocolo comparando el consumo de ancho de banda debido a los mensajes de control de MMARP frente al consumo de ancho de banda debido a los mensajes de datos. Se ha definido la sobrecarga como:

$$\Omega = \frac{\sum_{j=1}^{N_{PC}} length(PC_j) * 100}{\sum_{i=1}^{N_{PD}} length(PD_i) + \sum_{j=1}^{N_{PC}} length(PC_j)} \qquad (1)$$

Donde N_{PC} y N_{PD} son el número de mensajes de control y de datos respectivamente que se han cursado, $length(PC_j)$ representa el tamaño del paquete de control j y $length(PD_i)$ representa el tamaño del paquete de datos i. Los resultados comparados extraídos de los logs de MMARP se han comparado con los disponibles para ODMRP y se muestran en la Tabla I.

Como se puede apreciar la implementación de MMARP tiene muy poca sobrecarga más que ODMRP y el rendimiento, en términos de tasa de datos conseguida, es similar. Esto no hace más que confirmar los resultados del modelo matemático de la sobrecarga de estos protocolos que elaboramos en [10]. Es importante destacar que estas diferencias en sobrecarga no son significativas, especialmente si tenemos en cuenta que MMARP aporta nueva funcionalidad en lo referente a soportar nodos IP

estándar e interacción con la red fija, que suponen una mayor sobrecarga.

TABLA I
SOBRECARGA DE MMARP Y ODMRP EN EL ESCENARIO PROPUESTO

	MMARP		ODMRP	
	Valor	% of BW	Valor	% of BW
Duración	85 s	N/A	66,76 s	N/A
Control	3,86 Kb/s	1,06	1,53 Kb/s	0,49
Datos	360,02 Kb/s	98,94	307,72 Kb/s	99,5

5 Conclusiones

Hemos presentado una nueva arquitectura multicast capaz de conseguir comunicaciones multicast eficientes en extensiones ad hoc a redes IP fijas. Además, se ha presentado la combinación de esta arquitectura con nuestra propuesta del protocolo MMARP, que por lo que sabemos es el primero en la literatura capaz de ofrecer comunicaciones multicast en extensiones ad hoc a redes IP fijas a la vez que se soporta el modelo IP multicast, terminales estándar e interacción con la red fija. Hemos ofrecido una análisis de la problemática de cumplir estos nuevos requisitos en un escenario de extensiones ad hoc y hemos demostrado que con un diseño apropiado un protocolo como MMARP es capaz de conseguir cumplir estos requisitos prácticamente sin penalizar el rendimiento (comparado a ODMRP [4]) a consta de algo de complejidad adicional en el protocolo.

Los resultados empíricos han demostrado que estas conclusiones, así como la idoneidad del protocolo para este tipo de entornos que formarán parte de las futuras redes inalámbricas de cuarta generación..

Como trabajo futuro, se están estudiando extensiones para soporte de tráfico unicast, así como el análisis de los mismos problemas usando como protocolo de red IPv6.

Referencias

[1] IST Project MIND (www.ist-mind.org)

[2] M. Ramalho, "Intra- and Interdomain Multicast Routing Protocols: A Survey and Taxonomy," IEEE Commun. Surveys and Tutorials, vol. 3, no. 1, Jan.-Mar. 2000, pp. 2-25.

[3] E.M. Royer and C.-K. Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks," IEEE Personal Communications, vol. 6, no 2, April 1999, pp. 46-55.

[4] Y. Yi, S.-J. Lee, "On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc networks." Internet-Draft, draft-ietf-manet-odmrp-04.txt. Work in progress, November 2002.

[5] H. Lei and C.E. Perkins. Ad Hoc Networking with Mobile IP. In Proceedings of the Second European Personal Mobile Communications Conference, October 1997, pp. 197-202.

[6] W. Fenner: "Internet Group Management Protocol, Version 2". *RFC 2236*, November 1997.

[7] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P., Sharma and L. Wei.: "Protocol Independent Multicast Sparse Mode (PIM-SM): Protocol Specification". RFC 2362, June 1998.

[8] S. Deering, "Host Extensions for IP Multicasting", Request For Comments (RFC) 1112. Internet Engineering Task Force (IETF), 1989

[9] Pedro M. Ruiz, Graeme Brown, Ian Groves, "Scalable communications for ad hoc extensions connected to Mobile IP networks". In proceedings of the Personal Indoor Mobile Radio Communications (PIMRC) Symposium. Lisbon, September 2002.

[10] Pedro M. Ruiz, David Larrabeiti and Antonio F. Gómez-Skarmeta. "Analytical Model for the Overhead Evaluation of Multicast Ad hoc Routing Protocols". Technical Report, University of Murcia. TR-DIIC 1/2002.

[11] S. J. Lee. "Routing and Multicasting Strategies in Wireless Mobile Ad hoc Networks". Ph. D. Dissertation. University of California. L.A. 2000.

[12] ISABEL CSCW Application. http://www.agora-2000.com.

Sesión 2B

Diseño e interconexión de redes

Experiencias con Redes Privadas Virtuales de Nivel 2 sobre una infraestructura óptica metropolitana de IP sobre DWDM

Carlos García[1], Luis M. Díaz[1], José L. Peña[2], Luis Bellido[3]
Francisco Valera[1], David Fernández[3], Arturo Azcorra[1], Julio Berrocal[3], Isidro Cabello[2], Rafael López[2]
[1]Universidad Carlos III de Madrid, [2]Telefónica I+D, [3]Universidad Politécnica de Madrid
Departamento de Ingeniería Telemática, Avda. de la Universidad 30, 28911 Leganés (Madrid)
Teléfono: 916248778 Fax: 916248749
Email: cgarcia@it.uc3m.es

Abstract*. PREAMBULO is a currently running research project, funded by the MCYT, and whose main objective is to install, configure and operate a metropolitan fiber optic research infrastructure, providing a data transport network using IP directly over DWDM. The first part of the article describes the build up of both the physical and the logical network at the three different participating nodes: Universidad Carlos III de Madrid, Universidad Politécnica de Madrid and Telefónica I+D. Over this infrastructure, different activities have been taking place: multi-video conferences, IPv6 experiences, tele-education experiences, etc. The second part of this article describes one of the technologies that are also being tested in the project as well as some of these experiences: level 2 VPN solutions.*

1 Introducción

Pese a que la existencia de infraestructuras de red basadas en fibra óptica no son hoy en día ninguna novedad, lo cierto es que en general las soluciones que se plantean para el transporte de datos sobre dichas infraestructuras vienen derivadas de arquitecturas de protocolos sustentadas normalmente por SONET o SDH.

El proyecto PREAMBULO (*Prototipo de red multiservicio de muy altas prestaciones basada en IPv4/IPv6 sobre multiplexación por longitud de onda*, [1]) es un proyecto perteneciente al plan nacional I+D+I 2000-2003 del MCyT, que plantea la instalación, configuración y operación de una red de investigación de fibra óptica en la Comunidad de Madrid, que proporcione un servicio de transporte de datos utilizando IP directamente sobre DWDM, entre los tres nodos de la red: la Universidad Carlos III de Madrid, la Universidad Politécnica de Madrid y Telefónica I+D.

En la sección dos de este artículo se describe el proyecto PREAMBULO, detallando por un lado la arquitectura que se ha puesto en funcionamiento (finales de 2002) y por otro lado las diferentes experiencias que se han realizado sobre dicha infraestructura y que se seguirán realizando hasta que termine el proyecto a finales del año 2003.

Además de las experiencias de alto nivel realizadas sobre PREAMBULO (multivideoconferencia, tele-educación, etc.), como infraestructura de red metropolitana que es, PREAMBULO se apuntó en su momento como el entorno perfecto para llevar a cabo también experiencias reales con equipamiento capaz de proporcionar soluciones de conectividad a bajo nivel.

Así, la sección tres, se centra en la descripción tecnológica de las diferentes soluciones de redes privadas virtuales que han aparecido durante estos últimos años. Pese a que se hará una breve mención a las soluciones que se plantean a nivel 3, la sección profundizará más en las soluciones de nivel 2 cuya evolución parece estar desarrollándose de forma muy activa de un tiempo a esta parte.

Por último, la sección cuatro comenta las principales experiencias de VPN de nivel 2 que se han realizado sobre la red de PREAMBULO y expone las conclusiones más importantes extraídas hasta el momento.

2 Proyecto PREAMBULO

2.1 Objetivos

La mayoría de las redes desplegadas en la actualidad que ofrecen servicios IP sobre fibras ópticas con WDM, no ofrecen el servicio IP directamente sobre WDM (2 capas), sino que tienen una arquitectura en

Figura 1. Entidades participantes en PREAMBULO [1]

3 capas, de modo que una parte de las funciones se realizan en la capa óptica (WDM), otra parte en la capa SDH (en cada longitud de onda se envían tramas SDH) y a continuación los datagramas IP vienen empaquetados en los contenedores virtuales SDH. También existen redes que ofrecen el servicio IP sobre un servicio ATM, que es ofrecido a su vez o bien sobre SDH (4 capas en total) o bien directamente sobre WDM (predominando sobre todo la primera opción). Pero el uso de IP directamente sobre WDM plantea ventajas (menos sobrecarga, ventajas económicas, gestión) que hacen interesante su desarrollo, a pesar de los temas que quedan todavía por investigar (control del tráfico, recuperación de caídas de enlaces de la red y calidad de servicio).

El proyecto PREAMBULO [1] se fundamenta en la previsión de que a medio plazo se va a producir una implantación masiva de infraestructuras de transmisión WDM que, además de soportar los servicios existentes actualmente, deberán ofrecer una respuesta eficiente en prestaciones y coste a un mercado de servicios dominado claramente por la tecnología IP.

El objetivo principal del proyecto PREAMBULO, es el de desplegar la mencionada infraestructura de red óptica y llevar a cabo experiencias avanzadas con servicios IP entre las que se encuentran:

- Tráfico multicast
- Calidad de servicio (QoS)
- IP sobre WDM
- IP versión 6 (IPv6)
- Ingeniería de tráfico
- Prestaciones de los *GigaSwitch Routers*
- Interoperabilidad con otras infraestructuras de red avanzadas

2.2 Arquitectura

La red de PREAMBULO se puede dividir por un lado en un núcleo de red, que proporciona la interconectividad necesaria para proporcionar un servicio de redes de área local virtuales (VLANs) entre los tres centros participantes, y la periferia de la red, compuesta por los distintos equipos de nivel 2 y 3 que se conectan en cada centro a este núcleo para dar un servicio IP o IPv6 a los distintos proyectos de investigación y experiencias que utilizan la infraestructura de PREAMBULO. En esta sección se detalla la arquitectura del núcleo de la red, cuya puesta en marcha ha constituido uno de los principales hitos del proyecto.

Núcleo de la red

El núcleo de la red de puede descomponer en el nivel físico, o de transmisión por fibra óptica, el nivel DWDM, y el nivel de enlace.

En el nivel físico, la red está soportada por dos pares de fibras monomodo: uno entre TID y UPM, y otro entre TID y UC3M. Sobre esta configuración "en línea", se ha establecido una red DWDM con una

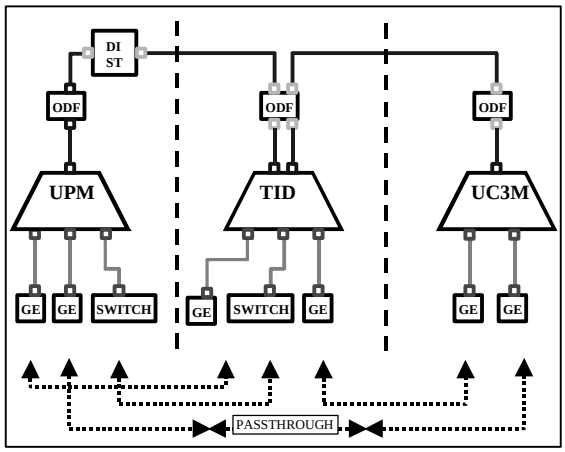


Figura 2: Esquema de la red DWDM

topología en triángulo, en la que los tres centros están conectados dos a dos. La configuración de red en el nivel DWDM se representa en la Figura 2, en la que se muestran los tres multiplexores DWDM conectados entre sí por los enlaces de fibra óptica.

Los equipos multiplexores son Nortel Optera Metro 5200. Los multiplexores en UPM y UC3M se han desplegado como terminales, de manera que todas las longitudes de onda utilizadas tienen su terminación en ellos. En cambio, el multiplexor en TID se ha desplegado como OADM (optical add/drop multiplexer), de manera que actúa como terminal para las longitudes de onda que soportan la comunicación hacia TID, y deja pasar las longitudes de onda para la comunicación UPM-UC3M, permitiendo, por tanto, una topología en triángulo entre los tres centros.

En cuanto a la conexión hacia la red de cliente, los equipos multiplexores proporcionan interfaces ópticas mediante las tarjetas denominadas OCI (optical-channel interface). En la red desplegada las interfaces utilizadas han sido Gigabit Ethernet-SX (850 nm) y ATM.

La instalación de las OCI adecuadas en cada multiplexor y el uso de tres canales o longitudes de onda distintas ha permitido desplegar una red en la que se dispone de enlaces Gigabit Ethernet (GE) dos a dos entre los tres centros y un enlace ATM adicional entre TID y UPM. Este último ha permitido continuar el servicio que ya se estaba dando sobre la fibra entre TID y UPM.

En cuanto al siguiente nivel, el nivel de enlace, el proyecto se ha centrado en proporcionar una infraestructura que proporciona un servicio de VLANs entre los tres centros participantes, utilizando los enlaces GE del nivel inferior. En un principio se evaluó la posibilidad de utilizar los enlaces GE para interconectar directamente routers IP de altas prestaciones, y proporcionar un servicio IP a los usuarios de la red PREAMBULO. Sin embargo, la utilización de conmutadores de nivel 2 ofrece las siguientes ventajas:

- Flexibilidad, versatilidad. Principalmente, la posibilidad de ejecutar diversos experimentos en paralelo y la asignación del ancho de banda disponible en fragmentos de 10/100/1000 Mbps
- Separación, independencia entre tráficos de distintos experimentos.
- Utilización de equipamiento más barato, tanto los conmutadores Ethernet, como las interfaces de nivel 2 a 100 Mbps para routers, sistemas finales (servidores) o conmutadores Ethernet adicionales.
- Posibilidad de conectar servidores directamente a la infraestructura de nivel 2.

En la decisión adoptada, también se consideró la posibilidad de reutilización de equipamiento ya existente y de los equipos adquiridos cuando el proyecto finalice.

Las características generales de la red de nivel 2 (**Figura 3**) son las siguientes:

- Topología en triangulo entre los conmutadores Ethernet de cada centro (uno o varios por centro), a través de los enlaces GE proporcionados por la red DWDM.
- Los enlaces troncales se configuran como enlaces inter-switch ("trunks"), de forma que transporten tráfico de todas las VLANs
- Cada puerto de los conmutadores Ethernet se configura como perteneciente a una determinada VLAN o como "trunk", en caso de conectar routers o servidores que pertenezcan a varias VLANs simultáneamente.

Sobre la infraestructura descrita se implementan distintos tipos de VLAN:

- Según número de participantes:
  - VLANs tipo 1: locales a un centro (sólo incluyen puertos correspondientes a un único conmutador y su tráfico nunca atraviesa el backbone).
  - VLANs tipo 2: en las que participan 2 centros (incluyen puertos en los conmutadores de dos centros y su tráfico se encamina por el enlace directo entre esos dos centros).
  - Tipo 3: VLANs en las que participan los tres centros.
- Según la velocidad de los puertos, de 10 Mbps, de 100 Mbps, de 1 Gbps o mixtas, con puertos a distintas velocidades.

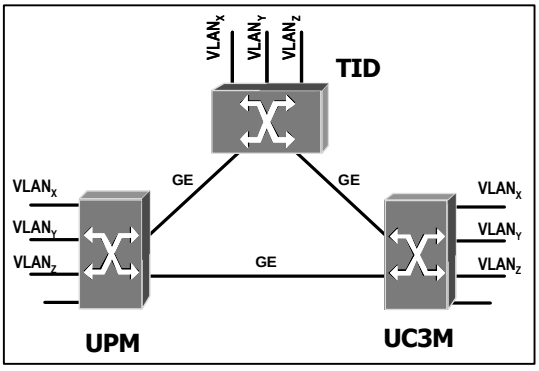


Figura 3: Red Lógica a nivel 2

Uno de los problemas que puede plantear esta topología en triángulo tiene que ver con el algoritmo de encaminamiento utilizado por los bridges: el algoritmo de *Spanning Tree (STP).*

En una topología con bucles, como es el triangulo que forman los conmutadores de la red de PREAMBULO, el algoritmo inhabilita uno de los enlaces para evitar que existan bucles, por lo que el tráfico entre dos de los centros no se encaminaría a través del enlace directo entre los mismos (ej, el tráfico entre UPM y UC3M no se encaminaría a través de su GE directa, o, peor todavía, el tráfico entre UPM y TID sería encaminado a través de UC3M).

Una posible solución a este problema sería el uso de conmutadores que ejecuten el algoritmo de Spanning-Tree por cada VLAN. Esta solución, estandarizada en [2] permitiría que, al menos para las VLAN de tipo 2, se pueda seleccionar mediante configuración (elección del nodo raíz) el enlace deshabilitado, eligiendo para cada VLAN el camino óptimo (enlace directo). Desafortunadamente, esta facilidad no está disponible en muchos de los conmutadores Ethernet que hay en el mercado

Otra alternativa, que finalmente ha sido la solución adoptada en el proyecto, consiste en deshabilitar el algoritmo STP y prohibir determinadas VLAN en cada enlace. Ambas opciones de configuración suelen ser ofrecidas por la mayoría de los fabricantes.

2.3 Experiencias

Después de proceder con la instalación de la infraestructura de red que se acaba de describir y de realizar las correspondientes pruebas de conectividad tanto a nivel físico como a nivel lógico entre las VLANs definidas, se dio por finalizado el proceso de implantación y se inició la fase de experiencias propiamente dicha (finales de 2002).

De entre las experiencias más importantes, realizadas hasta el momento, hay que destacar:

- Transporte de tráfico IPv6: una de las primeras experiencias que se realizaron, fue la migración de la maqueta IPv6 nativa implementada en el proyecto LONG [3], de tal forma que las conexiones entre UC3M, UPM y TID se han visto notablemente mejoradas.

- Vídeo-conferencias: la red de PREAMBULO se ha utilizado también para posibilitar la transmisión de eventos científico-tecnológicos (congresos, charlas, etc.). Las jornadas Telecom I+D 2002 [4], se hicieron llegar por ejemplo a las universidades utilizando PREAMBULO.

- Experiencias de tele-educación: la asignatura "*Redes de Banda Ancha*" perteneciente a la titulación de Ingeniería de Telecomunicación e impartida de forma distribuida ente las Universidades Politécnicas de Madrid, Valencia, Barcelona y la Universidad Carlos III, también se ha visto favorecida por el soporte dentro de las sedes de Madrid que se ha tenido por parte de PREAMBULO.

Además de estas experiencias que se han resaltado, se han llevado a cabo desde Febrero de 2003, diferentes pruebas de equipos y alternativas tecnológicas para proporcionar soluciones de redes privadas virtuales de nivel 2. Dichas experiencias son las que se describen en las siguientes secciones.

3 Redes Privadas Virtuales: VPN

Hoy en día, no es necesario resaltar ya la necesidad de las VPN en el entorno de red actual. Tradicionalmente, las empresas con diversas sedes han unido sus redes locales a través de líneas dedicadas punto a punto, unas veces reales, y otras veces mediante circuitos virtuales dedicados (típicamente, mediante FR/ATM). Esta situación ha perdurado mucho tiempo, pero está claro que estamos ante una solución sub-óptima, que ha provocado que los operadores deban desarrollar y mantener dos infraestructuras totalmente diferenciadas, la de tráfico de Internet (sobre una red puramente IP) y la de circuitos (sobre ATM/FR). Esta situación es, a la vez, más compleja, más cara, y más ineficiente, puesto que no se aprovechan al máximo los recursos de la red.

Actualmente, gracias a la aparición de MPLS [5], parece factible conseguir fusionar ambas infraestructuras en un único dominio administrativo, pudiendo dar el servicio de Internet, el de circuitos clásicos y el de VPN sobre la misma red. MPLS permite el aislamiento de la tecnología de nivel 2, y hacer converger desde el punto de vista de la conectividad lo mejor del mundo IP, con lo mejor del mundo de la conmutación de circuitos. Sin embargo, MPLS no resuelve por sí mismo los problemas asociados a las VPN.

3.1 VPN de nivel 3

En primer lugar, y dado que el tráfico mayoritario en la red es IP, parece obvio intentar soportar un servicio de VPN de nivel 3. Las soluciones planteadas surgen de dos grupos claramente diferenciados: el *IETF Provider Provisioned Virtual Private Networks* [6] y el *IETF Security Area* [7]. El primero se ha centrado en el uso de MPLS como solución para el soporte de VPNs de nivel 3, mientras que el otro se fundamenta en el uso de IPSec para la creación de dichas VPNs.

3.1.1 VPN-IPSec

El grupo de seguridad se centró en corregir y/o paliar las deficiencias de seguridad del protocolo IP, y crearon el protocolo IPSec [8]. Este protocolo permitía crear *Asociaciones Seguras* (SA) entre entidades, consiguiendo así confidencialidad y/o la autenticación de ambas partes. Básicamente, esta asociación segura puede verse como un túnel seguro entre dos entidades, donde ambas saben quién es realmente el otro (por ejemplo, mediante certificados X.501) y donde nadie puede escuchar ni interceptar la conversación (confidencialidad mediante cifrado). Visto así, se proporciona el soporte básico para poder crear una VPN sobre la infraestructura de Internet, pues basta con crear esta asociación segura (SA) o túnel seguro entre 2 entidades en distintas sedes de la misma VPN (típicamente, entre los cortafuegos de salida de dichas sedes) y enviar todo el tráfico IP destinado a la VPN a través de ese túnel IPSec-IP. Si se establece un mallado completo de túneles IPSec entre las distintas sedes, se crea una VPN de nivel 3, aunque esta solución no está exenta de problemas. En primer lugar, esta solución se coloca en el lado del cliente, lo cual no es del todo deseable, ya que muchos clientes prefieren contratar directamente el servicio y no tener que preocuparse de cómo ponerlo en marcha (y mucho menos mantenerlo). Además, la creación y mantenimiento de un mallado completo de túneles IPSec es una tarea compleja (hay que manejar gran cantidad de claves distintas, distribuirlas de forma segura, distribuir los certificados...), y la tarea de cifrar y descifrar exige unos recursos de computación elevados, por lo que la escalabilidad de esta solución es limitada. Finalmente, al ser una solución de cliente soportada sobre la red Internet, no tiene ningún mecanismo de QoS, al contrario de lo que ocurría en las VPN clásicas (ATM/FR).

3.1.2 BGP/MPLS

Por otro lado, aparece la solución propuesta por *Provider Provisioned Virtual Private Network Charter* [6], basada en BGP/MPLS [9]. Esta solución se basa en la construcción de túneles MPLS, mediante una doble indexación de etiquetas. El primer nivel (*inner label*) permite identificar un paquete como perteneciente a una VPN específica, mientras que la segunda etiqueta (*outer label*) permite que el paquete viaje por la red del proveedor desde el punto de entrada al de salida. BGP se usa como mecanismo de señalización de las *inner label* mientras que el mecanismo para señalizar las *outer label* es independiente de las VPN, y depende del mecanismo elegido para señalizar la red troncal del proveedor (normalmente, LDP o RSVP). Esta solución es claramente una solución de proveedor, donde el cliente sólo tiene que suministrar al proveedor el/los prefijos de red que es capaz de alcanzar. La seguridad se consigue gracias a la separación del tráfico en circuitos virtuales (igual que ocurría en ATM/FR), evitando el uso de cifrado, que es un proceso muy costoso. Además, se puede proporcionar QoS e ingeniería de tráfico a través de

los mecanismos básicos de MPLS (creación de túneles RSVP), lo cual es un servicio de valor añadido de incalculable utilidad en las redes actuales. Finalmente, la escalabilidad está asegurada gracias a esta doble indexación de etiquetas, ya que la troncal de la red no tiene que saber nada de VPNs, y los equipos frontera sólo tienen que almacenar la información relevante a las VPN's que cada uno maneje (ningún equipo de la red tiene que conocer todo sobre todas las VPNs).

Eso sí, esta solución también plantea problemas. En primer lugar, es necesario confiar en la información de encaminamiento que proporciona el cliente, lo cual no tiene por qué ser una buena idea. Además, no proporciona ningún soporte para IP Multicast, por lo que sería necesario algún mecanismo externo para soportarlo.

Finalmente, estas soluciones transportan tráfico de nivel 3, pero hace falta algún tipo de mecanismo adicional si lo que se desea es transportar de forma transparente tráfico de nivel 2.

3.3 VPNs de Nivel 2

3.3.1 Introducción

El envío de tráfico de nivel 2, tal como Ethernet, Frame Relay o ATM, sobre una red de transporte MPLS, está cobrando especial importancia en la actualidad, principalmente impulsado por el interés de los proveedores de servicio, y dirigido por determinados grupos de trabajo del IETF, así como por los principales fabricantes de equipos de transporte en la red.

Esta tecnología permitiría a los proveedores ofrecer el transporte de tráfico de los clientes mientras se continúa la migración a las redes IP de próxima generación. Es decir, seguir ofreciendo los tradicionales servicios a clientes, como enlaces Frame Relay, sobre un único núcleo IP, disminuyendo de esta forma los gastos de mantenimiento y gestión de red.

Aunque aún no se dispone de ningún estándar para ofrecer estos servicios, son muchos los fabricantes que ofrecen soporte para los borradores de Martini [10], [11] en el IETF. Como hemos comentado dentro del IETF existen dos grupos de trabajo dedicados a las redes privadas virtuales de nivel 2. Tanto el PPVPN, como el PWE3 [12], estudian la utilización de túneles basados en IP y L2TP, así como MPLS.

3.3.2 Objetivo

Como se ha comentado, una VPN es simplemente una forma de proporcionar comunicaciones privadas sobre una red pública. Tradicionalmente las empresas han contratado enlaces de nivel 2 a los proveedores de servicio, y han montado su propia infraestructura de nivel 3.

Las VPNs de nivel 2 son multiprotocolo por naturaleza, de forma que pueden soportar tanto tráfico de IP como de otros protocolos. De igual forma eliminan la participación del proveedor de servicio en las tareas de configuración de nivel 3 del cliente, beneficiando a ambos.

En la actualidad, la principal demanda de circuitos de nivel 2 se basa en tecnología Frame Relay, y poco a poco ganan terreno las VPN de nivel 2 basadas en Ethernet. En consecuencia, los principales beneficios de los proveedores de servicios vienen derivados de estas actividades. Sin embargo, supone un grave inconveniente el hecho de mantener diferentes infraestructuras para ofrecer diferentes servicios.

La nueva tecnología de VPNs de nivel 2 podría solucionar estos problemas manteniendo una única infraestructura basada en transporte sobre MPLS.

3.3.3 Túneles MPLS L2VPN

Los principales esfuerzos del IETF están dirigidos a la creación de VPNs de nivel 2 basadas en MPLS. De esta forma es posible crear túneles (LSP) basados en conmutación de etiquetas en lugar de usarIPSec. De la misma manera, es posible utilizar protocolos de control como LDP o BGP para el establecimiento de los circuitos virtuales (VCs) para el transporte de PDUs de nivel 2 a través de la red.

El borrador de Martini utiliza la técnica *label-stacking* de MPLS para permitir separar las etiquetas de circuitos virtuales (*VC labels*) y las etiquetas de túneles (*tunnel label*). La etiqueta de túnel identifica el camino que los paquetes tomaran a través de la red, mientras que la etiqueta de circuito virtual identifica la VPN en el destino (y el ingress node). En el núcleo de la red, los routers (LSRs) utilizan la etiqueta de túnel para el reenvío de paquetes, mientras los routers frontera (*egress LSRs*) usan la etiqueta de circuito virtual para determinar como procesar la trama.

Existen dos borradores de Martini, el primero de ellos [11] especifica como debe realizarse la encapsulación sobre circuitos virtuales para tecnologías como ATM, Ethernet, HDLC y PPP. Y define un campo *demultiplexor* para distinguir diferentes circuitos virtuales emulados sobre el mismo túnel. De igual forma se define una *palabra de control* (Control Word), cuya función es mantener el número de secuencia de las tramas, hacer relleno para paquetes pequeños según la tecnología de nivel 2, o llevar ciertos bits de control de este nivel. También permite la eliminación de la cabecera de nivel 2 y su reconstrucción en el router frontera.

El segundo borrador de Martini [10] define los procedimientos para la distribución de etiquetas, lo que permite el transporte de PDUs a través de una red MPLS. Aunque Martini especifica LDP para el establecimiento de túneles, otros grupos del. IETF

están estudiando el posible uso de otros protocolos (principalmente, BGP).

3.3.4 Extensiones al borrador de Martini

Si bien Martini establece la base para la creación de túneles sobre una red MPLS, debemos tener en cuenta que esto solamente proporciona túneles punto a punto. Una red privada virtual necesita de conectividad multipunto-multipunto para lo que se requiere una red mallada de túneles Martini.

En primer lugar el borrador de K. Kompella [13] especifica el uso de BGP para la distribución de bloques de etiquetas, y el mapeo de los identificadores de enlaces, en Frame Relay (DLCIs), ATM(VCIs), y otras tecnologías, sobre los circuitos virtuales.

Por otro lado Laserre [14] propone extensiones a Martini para el soporte de conectividad Ethernet multipunto-multipunto, permitiendo envío de tráfico broadcast y multicast a través de una VPN.

3.3.5 VPLS

VPLS (*Virtual Private LAN Service* [15]) identifica cómo un proveedor de servicios ofrece conectividad a nivel 2 a clientes con múltiples sedes de forma que sea transparente para el dispositivo frontera del cliente (*CE – customer edge*). El proveedor se encarga del transporte de las tramas de nivel 2 del cliente desde una sede hacia otra(s) a través del núcleo de la red. Para la provisión de este servicio se usa la tecnología de VPNs de nivel 2 basadas en MPLS.

Inicialmente la tecnología Ethernet resultaba una buena solución para las redes de área local, sin embargo, debido a su amplia difusión, en la actualidad esta tecnología está comenzando a aparecer como tecnología de acceso en redes MAN y WAN. Un puerto Ethernet permite conectar al cliente con el borde del proveedor (*PE – Provider Edge*), de manera que este tráfico se identificaría con una VPN de nivel 2 a través del identificador de puerto o de la etiqueta VLAN.

Para ofrecer este servicio, es necesario resolver el envío de paquetes en modo broadcast y multicast disponible en tecnología Ethernet, lo cual no se soporta de forma nativa en una red MPLS. Las diferentes sedes de un cliente, conectadas a través de una red MPLS, esperarán que su tráfico broadcast, multicast y unicast sea reenviado a los sitios apropiados. Esto implica ciertos requisitos en la red MPLS: aprendizaje de direcciones MAC, replicación de paquetes a través de túneles LSP para tráfico broadcast y multicast, e inundación para paquetes unicast con destinatario desconocido.

El objetivo principal de la tecnología VPLS es proporcionar de conectividad entre sitios de clientes geográficamente dispersos a través de redes MAN/WAN. De forma que el cliente perciba el mismo servicio como si estuviera conectado a través de una LAN (la red MAN/WAN se comportaría como un Bridge con aprendizaje de nivel 2).

4 Experiencias con VPLS

4.1 Introducción

Dentro del plan de pruebas propuesto en el marco del proyecto PREAMBULO, se encuentra la realización de experiencias con VPNs de nivel 2. A continuación se detalla la solución de Nortel Networks (*"Logical Provider Edge")* basada en DVPLS (Distributed VPLS, [16]), que proporciona como ya veremos ciertas mejoras sobre el mecanismo básico de VPLS visto anteriormente.

4.2 VPLS distribuido

Como se ha visto en el apartado anterior, el equipo clave para el correcto funcionamiento del servicio VPLS es el PE (*Provider Edge*). Este equipo (como mínimo) tiene que tener toda la información de las VPNs a la que pertenecen sus clientes, intercambiar las etiquetas de circuito virtual (*VC Labels)* con los otros PE (típicamente con BGP o LDP), intercambiar las etiquetas de túnel (*Tunnel Label)* con sus vecinos de la red troncal (mediante RSVP o LDP, dependiendo de si se desea hacer ingeniería de tráfico o no) y debe realizar el aprendizaje de direcciones MAC. Juntar toda esta funcionalidad en un único equipo plantea serios problemas.

En primer lugar, es evidente que un equipo de estas características será muy complejo, y por lo tanto muy caro. Además, teniendo en cuenta que el equipo tiene que hacer el aprendizaje de direcciones MAC de todas las máquinas asociadas a las VPNs de las que es miembro, la escalabilidad se ve limitada al número máximo de direcciones que sea capaz de almacenar en memoria. Desde estos puntos de vista es por tanto razonable plantearse que la solución con un único PE es ciertamente mejorable.

VPLS distribuido plantea una solución al problema de la escalabilidad y complejidad del PE. Esta solución se basa en desagregar la funcionalidad del PE en dos entidades: *PE-core* y *PE-edge*. La primera es la encargada de gestionar y mantener los túneles MPLS (*VC* y *Túnel Labels*) y de distribuir la información de VPNs al resto de PE, y la segunda entidad se encarga del aprendizaje de direcciones MAC y de la delimitación del servicio (cuando llega una trama MAC de cliente es capaz de identificar la VPN a la que pertenece y de saber el PE al que debe llegar dicha trama). Con esta arquitectura, separamos toda la complejidad asociada a los procesos MPLS de la mecánica propia del servicio VPLS en dos tipos de equipo diferentes, y como por cada *PE-core* tendremos varios *PE-edge*, se mejora sustancialmente la escalabilidad del conjunto sin aumentar la complejidad ni el coste de la solución.

4.3 Solución LPE

La solución planteada por Nortel Networks para la implementación de VPNs de nivel dos está basada en la arquitectura *"Logical Provider Edge"* (LPE). Esta arquitectura permite reducir la complejidad de los actuales PEs, desagregando sus funciones en varios equipos físicamente separados. Esta separación en entidades independientes permite obtener una solución mucho más rentable en costes y aumentar considerablemente el grado de escalabilidad y agregación de clientes.

Nortel Networks, dentro de su división *Optical Ethernet (OE),* propone dos equipos para realizar las funciones de un PE clásico: PE-edge y PE-core. Dentro de su gama de equipos, la funcionalidad del PE-edge viene implementada con sus equipos de la serie OM1000, mientras que para el PE-Core se utilizan los equipos de la serie OM8000.

El PE-edge es el encargado de separar la demarcación entre el cliente y el proveedor, realizar la agregación de los clientes, mantener las tablas de reenvío para cada servicio VPN definido, informar al PE-core de nuevas incorporaciones y aplicar políticas de calidad de servicio en caso que se necesiten.

Por otra parte, el PE-core es responsable de distribuir y mantener las etiquetas MPLS e información de VPNs entre los PE-core del backbone, realizar la encapsulación Martini, mantener la información de registro de los clientes de las VPNs y controlar los procesos de encaminamiento.

Los equipos de Nortel Networks poseen un mecanismo propietario para detectar de manera automática la incorporación de un nuevo cliente a un servicio VPN y propagar esa información por todo el backbone MPLS a todos los PE-cores. Para ello, el PE-edge utiliza un protocolo llamado OE-AD *(OE-Autodiscovery Protocol)* que informa al PE-core de la nueva incorporación en el PE-Edge. Esto permite que la provisión de nuevos clientes se realice de manera muy sencilla y rápida.

Los PE-edges independizan el tráfico de usuario frente al tráfico de proveedor demarcando el acceso del cliente con puertos UNI *(User Network Interfaces).* Estos puertos proporcionan la conexión con los servicios VPN de nivel 2 configurados en la red. Por esta razón, cualquier inestabilidad producida por el tráfico del cliente no afecta en absoluto a las características de la red del proveedor. Esto se consigue realizando una doble encapsulación del tráfico de cliente añadiendo una cabecera de nivel 2 *(Service Provider Ethernet Header)* y otra de nivel 3 *(OE/L2 Header).* Esta información permite que el paquete sea encaminado por una red de nivel 2 tradicional desde el PE-edge hasta el PE-core ya que las cabeceras introducidas respetan los campos definidos por el estándar 802.3 y de una cabecera IP normal.

Una vez que el tráfico de cliente llega al PE-core, éste analiza la cabecera OE-L2 para saber a que VPN pertenece y lo envía por los túneles MPLS de salida asociados al servicio. Cuando el tráfico llega a los extremos remotos es desencapsulado y vuelto a encapsular con las cabeceras Service Provider Ethernet y OE/L2 hasta que llega al PE-edge remoto. Una vez allí, se desencapsulan las cabeceras y se transmite por el puerto UNI de salida correspondiente del cliente configurado.

La **Figura 4** muestra el backbone MPLS utilizado entre los tres centros participantes.

Además de la implementación LPE, Nortel Networks posee un función propietaria para dar redundancia de equipo y enlace bastante eficiente llamada *Split Multilink Trunking (SMLT).* SMLT permite tener tiempos de convergencia entre 1 y 2 segundos como máximo muy inferiores a los conseguidos por STP. SMLT limita su funcionamiento a entornos de cliente "*dual homing*". Hace uso de varios switches donde se utilizan técnicas de agregación de enlace (MLT) y un protocolo de comunicación entre los switches SMLT llamado IST que es usado para intercambiarse las MACs aprendidas entre ellos.

Por lo tanto como conclusión, la solución de Nortel Networks ofrece:

- Facilidad y rapidez en la provisión de nuevos clientes.
- Una clara demarcación entre cliente y proveedor de servicio (UNI).
- Descubrimiento automático de clientes en toda la red gracias al OE-AD.
- Una gran capacidad de escalabilidad.
- Tiempos de recuperación ante fallos muy rápidos (S-MLT).

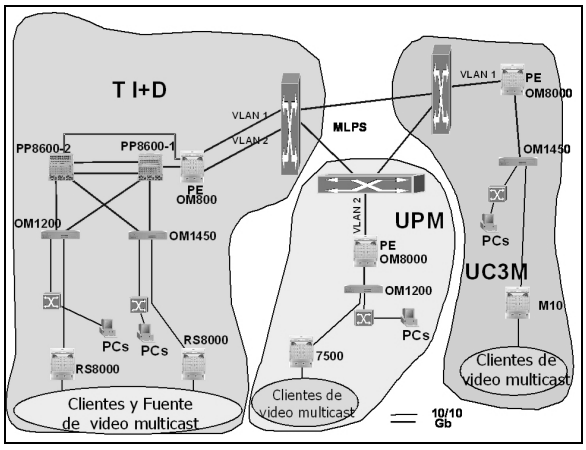


Figura 4. Maqueta LPE de Nortel

5 Conclusiones

En este artículo se ha presentado la infraestructura de red óptica para soporte de IP directamente sobre DWDM del proyecto PREAMBULO, describiéndola tanto a nivel físico como a nivel lógico y comentando las principales alternativas que se dieron antes de optar por una solución definitiva que proporcionase la conectividad deseada entre las tres sedes que participan en el proyecto.

Dicha solución se basa en mantener la conectividad a nivel 2 mediante conmutadores (con el algoritmo *Spanning Tree* deshabilitado), y utilizando VLANs para separar directamente a ese nivel el tráfico que va a circular por la red óptica.

Tras presentar las principales experiencias llevadas a cabo en el proyecto, el artículo se centra en una de ellas, las experiencias con soluciones de redes privadas de nivel 2.

Posteriormente se ha hecho una revisión de las principales líneas de investigación en lo referente a redes privadas virtuales (VPN), comentando la solución para VPNs de nivel 3, y haciendo especial énfasis en las soluciones de nivel 2.

Dichas soluciones se basan en tecnologías que se encuentra actualmente en desarrollo, y para las que han comenzado recientemente a aparecer las primeras implementaciones en equipos comerciales. Si bien casi todas las propuestas se basan en el uso de túneles Martini, existen diferentes propuestas para el establecimiento de los circuitos virtuales.

Finalmente se analiza la infraestructura que se ha probado dentro del proyecto PREAMBULO para dar soporte a una solución concreta VPN de nivel 2, Logical PE, propuesta por el fabricante Nortel Networks. De esta forma se demuestra la viabilidad de esta tecnología de última generación, comprobando las ventajas respecto a soluciones previas que presentaban ciertos problemas de escalabilidad.

Agradecimientos

Este artículo ha sido posible gracias a la financiación del proyecto PREAMBULO por parte del Ministerio de Ciencia y Tecnología, a través de su plan nacional I+D+I 2000-2003.

Las pruebas de VPN de nivel 2 han sido posibles gracias a Nortel Networks, que proporcionó tanto los equipos como la formación necesaria para llevarlas a cabo.

Referencias

[1] PREAMBULO. *Prototipo de red multiservicio de muy altas prestaciones basada en IPv4/IPv6 sobre multiplexación por longitud de onda* (TIC2000-0268-P4-C03-01). http://www.it.uc3m.es/preambulo [Abril 2003]

[2] IEEE 802.1s – Multiple Spannings Trees http://www.ieee802.org/1/pages/802.1s.html

[3] LONG. *Laboratories Over Next Generation Networks.* IST-1999-20393. http://long.ccaba.upc.es/

[4] Jornadas Telecom I+D. www.telecom-id.com

[5] IETF Multi-Protocol Label Switching Charter. http://www.ietf.org/html.charters/mpls-charter.html [Marzo 2003]

[6] IETF Provider Provisioned Virtual Private Networks Charter (ppvpn). http://www.ietf.org/html.charters/ppvpn-charter.html [Marzo 2003]

[7] IETF Security Area. http://sec.ietf.org

[8] IETF IP Security Protocol Charter (IPSec). http://www.ietf.org/html.charters/ipsec-charter.html [Enero 2003]

[9] RFC 2547bis: BGP/MPLS. http://www.ietf.org/rfc/rfc2547.txt [Enero 2003]

[10] Transport of Layer 2 Frames over MPLS. http://www.ietf.org/internet-drafts/draft-martini-l2circuit-trans-mpls-10.txt

[11] Encapsulation Methods for Transport of Layer 2 Frames Over IP and MPLS Networks. http://www.ietf.org/internet-drafts/draft-martini-l2circuit-encap-mpls-04.txt

[12] IETF Pseudo Wire Emulation Edge to Edge. http://www.ietf.org/html.charters/pwe3-charter.html [Marzo 2003]

[13] Layer 2 VPN Over Tunnels. http://www.ietf.org/internet-drafts/draft-kompella-ppvpn-l2vpn-02.txt

[14] Virtual Private LAN Service over MPLS. http://www.ietf.org/internet-drafts/draft-laserre-vkompella-ppvpn-vpls-03.txt

[15] Virtual Private LAN Service. http://www.ietf.org/internet-drafts/draft-kompella-ppvpn-vpls-01.txt

[16] Decoupled Virtual Private LAN Services http://ietf.org/intenet-drafts/draft-kompella-ppvpn-dtls-02.txt

Análisis del planificador de Asignación Jerárquica Paralela para conmutadores con colas virtuales a la salida

F.J. González Castaño[*], C. López Bravo[+], R. Asorey Cacheda[*], J. M. Pousada-Carballo[*]

[*]Departamento de Ingeniería Telemática, Universidad de Vigo
[+]Departamento de Tecnologías de la Información y las Comunicaciones,
Universidad Politécnica de Cartagena
[*]ETSI Telecomunicación, Campus, 36200 Vigo
[*]Teléfono: +34 986 813788, FAX: +34 986 812116
Email: javier@det.uvigo.es, cristina.lopez@upct.es, rasorey@det.uvigo.es

Abstract. *Input-queued packet switches are more scalable than output-queued ones. However, due to HOL blocking, their throughput is poor. Several schedulers for virtual-output-queued switches have been proposed to overcome this problem. Among them, the class of iterative maximal matching algorithms, with a first example being PIM (which uses random selection) and a second example being iSLIP (which uses round-robin selection and admits efficient practical implementations). iSLIP has several variants (with different pointer updating strategies) that exhibit a similar delay performance, like FIRM or -more recently- RDSRR (which outperforms the original description). In previous work, we proposed a new scheduler, Parallel Hierarchical Matching (PHM), which compares favorably to iSLIP-like algorithms. PHM can be considered the parallelization of previous high-performance sequential round-robin matching algorithms, like 2DRR or WWFA. In this paper, we estimate the average number of iterations required by PHM to achieve a maximal matching, and compare its delay performance with that of previous proposals for the same decision response-time.*

1 Introducción

Los conmutadores de paquetes con colas a la entrada son más escalables que los conmutadores con colas a la salida (ver referencia [1] y asociadas). También se ha observado que sufren una menor probabilidad de pérdida de paquetes cuando se les somete a un patrón de tráfico a ráfagas [2]. Sin embargo, su *throughput* es pobre, debido al *bloqueo en la cabeza de la cola* o *bloqueo HOL*. Incluso en las condiciones más favorables –por ejemplo, tráfico de entrada Bernoulli, uniformemente distribuido entre las salidas– el *throughput* se vería limitado al 58.6%. Hasta la fecha se han desarrollado diversas implementaciones con éxito [4],[5].

Un planificador del tipo *maximal size matching* es un dispositivo que, de un conjunto de colas virtuales de salida (*VOQ*, [6]), selecciona el mayor número posible de paquetes para transferirlos a las salidas del conmutador. La cola *VOQ(i,j)* forma parte de la cola de la entrada i y contiene paquetes dirigidos a la salida j, $i,j=1,...,N$, siendo N el número de entradas y salidas. Cuando el planificador selecciona la cola virtual *VOQ(i,j)*, se transfiere el paquete más antiguo (paquete *HOL*) de la entrada i *perteneciente a dicha cola* a la salida j. Para que el sistema sea escalable, como máximo se puede tomar un paquete de cada entrada i y como máximo se puede dirigir un paquete a cada salida j (de modo que el tiempo de acceso de las memorias sea independiente del tamaño del conmutador). Para caracterizar las colas *VOQ* y las transferencias entre entradas y salidas, en lo que sigue utilizaremos dos matrices R y S, tales que r_{ij} será igual a 1 si existe al menos un paquete en la cola *VOQ(i,j)* (en otro caso r_{ij}=0), y s_{ij} será igual a 1 cuando el paquete HOL de la cola virtual *VOQ(i,j)* sea seleccionado para su transferencia.

Los algoritmos para *VOQ* con mayor éxito han sido los pertenecientes a la clase *iterative maximal matching*. El primer algoritmo de esta clase fue PIM [8],[9], basado en mecanismos de selección aleatoria de *VOQ*s. Se ha demostrado que PIM alcanza un máximo local del problema de transferencia en $O(log_2 N)$ iteraciones, en media [8]. Sin embargo, la selección aleatoria es difícil de implementar en la práctica. Otro ejemplo de esta clase de algoritmos es iSLIP, que resuelve el problema mediante mecanismos *round-robin* y se ha utilizado en implementaciones reales de alta velocidad [10],[11]. iSLIP tiene muchas variantes, cada una de ellas con sus propios matices en la actualización de punteros *round-robin*, pero con unas prestaciones de retardo sólo ligeramente superiores (FIRM[12], RDSRR[13]). iSLIP se ha convertido en un estándar *de facto* en la investigación en planificación de *VOQ*s.

Otra clase de algoritmos, a los que nos referiremos como algoritmos de *asignación jerárquica secuencial* (AJS), se basa en "intentonas" de tamaño máximo. En este caso, las colas virtuales se ordenan en función de una matriz de jerarquías *H*, de tal forma que todas las *VOQs* con la misma jerarquía pertenecen a un grupo de tamaño máximo libre de conflictos. En un conmutador *N*×*N*, la matriz *H* está formada por *N* grupos de *N VOQ*s sin conflictos, es decir, que dentro de un grupo dos *VOQ*s cualesquiera no comparten ninguna fila o columna. En la Tabla I podemos ver un ejemplo de asignación jerárquica para un conmutador *4*×*4*. Los grupos se revisan secuencialmente y en orden jerárquico. Se selecciona una *VOQ* si, al analizar su grupo, no existen conflictos de entradas o salidas con otras *VOQ*s pertenecientes a grupos de mayor jerarquía. O, en otras palabras, si no hay conflictos con otras *VOQ*s de la misma fila o la misma columna, pertenecientes a grupos que ya han sido revisados.

Es necesario variar *H* periódicamente para evitar la inanición de alguna de las colas. La actualización se puede llevar a cabo de muchas maneras [15]. Por ejemplo, utilizando reglas aritméticas sencillas, como aumentar en una unidad (módulo N) la jerarquía de cada celda:

$$\forall i, j \quad h_{ij} \leftarrow (h_{ij} + 1)_N \qquad (1)$$

Naturalmente, es necesario generar una matriz inicial H^{init} válida. Una posibilidad es, a partir del elemento h^{init}_{11}, que podría tomar cualquier valor entre 0 y N–1, producir los demás elementos como sigue:

$$h^{init}_{1j} = \left(h^{init}_{1j-1} + 1\right)_N \forall j > 1 \text{ y, tras ello,}$$

$$h^{init}_{i1} = (h^{init}_{i-1N} + 1)_N \quad \forall i > 1 \text{ y}$$

$$h^{init}_{ij} = (h^{init}_{ij-1} + 1)_N \quad \forall i, j > 1.$$

Los algoritmos de esta clase (como por ejemplo 2DRR[15], WWFA[6] o HBRTNS[16]) necesitan *O(N)* pasos por *slot* de conmutación, y por tanto son poco escalables. Ahora bien, en conmutadores pequeños, en los que el número de pasos necesarios para llegar a la solución es reducido, estos algoritmos se benefician de la simplicidad de los pasos.

Tabla I

Ejemplo de matriz de jerarquías *H* para un conmutador *4*×*4*.

3	1	2	0
1	3	0	2
2	0	3	1
0	2	1	3

En trabajos anteriores [16], [17], propusimos un nuevo planificador de *asignación jerárquica paralela* (*Parallel Hierarchical Matching*, PHM), con prestaciones de retardo comparables a las de los algoritmos de tipo iSLIP. Por otra parte, PHM se puede ver como la *paralelización* de los algoritmos de asignación jerárquica secuencial, y se beneficia de su simplicidad estructural hasta cierto punto. Por tanto, PHM alcanza un compromiso entre las ventajas de las dos estrategias.

El resto de este artículo está organizado como sigue: en la sección 2 se repasa la descripción de los tres algoritmos representativos que vamos a comparar: PHM, 2DRR (un algoritmo de asignación jerárquica secuencial) y RDSRR (uno de los algoritmos de tipo iSLIP con mejores prestaciones) En la sección 3 se analiza la convergencia de PHM desde una perspectiva teórica. En la sección 4 se compara el retardo medio por paquete de los tres algoritmos representativos, para una misma duración de los intervalos de paquete. Finalmente, en la sección 5 se exponen las conclusiones del artículo.

2 Algoritmos

2.1 Algoritmo 2DRR

Los algoritmos de asignación jerárquica secuencial tienen N^2 unidades lógicas de decisión s_{ij}, $i,j=1,...,N$. Cada unidad s_{ij} se encarga de decidir si la cola virtual *VOQ(i,j)* se selecciona o no; es decir, si se transfiere un paquete de la entrada *i* a la salida *j*. Sea h_{ij} la jerarquía de la unidad s_{ij}.

Algoritmo 2DRR

$\forall ij \quad s_{ij} = 0$

Desde $l = N$ *hasta* 1:

Hacer en paralelo:

Si:

$r_{ij} = 1$ *y* $h_{ij} = l$ *y*

$\forall k \neq i \mid h_{kj} > h_{ij}, r_{ij} = 0$ *y*

$\forall k \neq j \mid h_{ik} > h_{ij}, r_{ik} = 0$

entonces $s_{ij} = 1$

Actualizar H

Entregar S y finalizar

2.2 PHM

En principio, PHM sigue la filosofía de los algoritmos de asignación jerárquica secuencial: tiene N^2 unidades lógicas de decisión s_{ij}, $i,j=1,...,N$, cada una de las cuales se encarga de decidir si se selecciona o no la cola virtual *VOQ(i,j)* correspondiente. De nuevo, sea h_{ij} la jerarquía o prioridad de la unidad s_{ij}. Sea *LIMN* el máximo número de iteraciones permitidas.

Algoritmo PHM

$\forall ij \quad s_{ij}^0 = 0, n = 0$

Iteración PHM:

i)

1) *Hacer en paralelo*:

Si:

$r_{ij} = 1$ *y*

$\forall k \neq i, s_{kj}^n = 0$ *y*

$\forall k \neq j, s_{ik}^n = 0$

entonces $t_{ij} = 1$,

en cualquier otro caso $t_{ij} = 0$

2) *Hacer en paralelo*:

Si:

$t_{ij} = 1$ *y*

$\forall k \neq i \mid h_{kj} > h_{ij}, t_{kij} = 0$ *y*

$\forall k \neq j \mid h_{ik} > h_{ij}, t_{ik} = 0$

entonces $s_{ij}^{n+1} = 1$

ii) $n \leftarrow n+1$. *Si* $n = LIMN$ *finalizar*.

Entregar S^n

Actualizar H

En una iteración cualquiera sólo se consideran las unidades libres de conflictos con decisiones tomadas en iteraciones previas (t_{ij}=1). A continuación, se filtran todas las unidades lógicas, excepto las que tengan mayor jerarquía en su fila y columna. Por tanto, en realidad, una iteración PHM es *más conservadora* que un ciclo completo de un algoritmo de asignación jerárquica secuencial, ya que una unidad *a* puede ser bloqueada temporalmente por otra unidad *b* de mayor prioridad, incluso si *b* es a su vez bloqueada por una tercera unidad *c* con mayor prioridad que *b* (pero que no bloquea a *a*).

Nos gustaría resaltar que tanto las variables t_{ij} como las variables s_{ij} se calculan en paralelo. Aunque incluimos las variables t_{ij} para clarificar la descripción, en realidad no serían necesarias para implementar el algoritmo: se puede prescindir de ellas sin más que sustituir t_{ij} en el paso (*i.2*) por la expresión equivalente en el paso (*i.1*).

Finalmente, debemos indicar que, en cualquier iteración intermedia, S^n puede ser subóptima, pero en cualquier caso está libre de conflictos. En consecuencia, el algoritmo se puede detener en cualquier instante, como en el caso de los algoritmos de tipo iSLIP.

2.3 Algoritmo RDSRR

La mecánica de este algoritmo podría resumirse como sigue [14]:

Iniciación. Los punteros que indican cuál es la entrada de mayor prioridad para cada salida en un instante dado se inician a cero. Los punteros equivalentes asociados a las salidas se inician a valores diferentes, entre 1 y *N*.

Fase 1 o *request*. Cada entrada envía una petición a todas aquellas salidas para las que tiene un paquete.

Fase 2 o *grant*. Cuando una salida recibe alguna petición, debe escoger una de ellas (*grant*). En concreto, elige aquella con mayor prioridad, en orden *round-robin*. En el caso particular de RDSRR, el sentido de la búsqueda es alternativamente horario y anti-horario, variando de una iteración a la siguiente. El puntero al elemento más prioritario se incrementa siempre en una unidad (modulo *N*), se genere o no un *grant*.

Fase 3 o *accept*. Si una entrada recibe algún *grant*, acepta el primero de ellos en orden *round-robin* de mayor a menor prioridad. En este caso el sentido de la búsqueda es siempre el mismo. El puntero al elemento más prioritario se incrementa siempre en una unidad (modulo *N*) haya o no *accept*.

3 Análisis de convergencia de PHM

En [17] se demuestra que, en el peor de los casos, el algoritmo PHM necesita $\lceil N/2 \rceil$ iteraciones para

conseguir un máximo local de la asignación de entradas-salidas (*maximal matching*). Sin embargo, a pesar del interés de este resultado, nos gustaría saber cuántas iteraciones PHM son necesarias *en media* para obtener un máximo local. Para responder a esta pregunta, supondremos que todas las *VOQs* tienen *la misma* probabilidad *q* de estar vacías al comienzo de cada intervalo de paquete. Esta condición se cumple, al menos, cuando el tráfico de entrada sigue una distribución Bernoulli uniforme y la actualización de la matriz *H* es aleatoria. Sea $p = 1 - q$.

Diremos que una *VOQ* con prioridad *k* se *resuelve* en una iteración determinada cuando, al finalizar dicha iteración, la *VOQ* no puede seguir participando en el algoritmo PHM. Esto ocurre en las siguientes circunstancias:

- Si la cola *VOQ* está vacía (obviamente, no puede ser seleccionada para transmitir).
- Si no está vacía pero todas sus competidoras de mayor prioridad lo están (la cola *VOQ* se selecciona para transmitir).
- Si no está vacía, pero al menos una de sus competidoras de mayor prioridad ha sido elegida para transmitir (en este caso la cola *VOQ* no puede transmitir en el intervalo de paquete en curso). La probabilidad de que la *VOQ* se bloquee de esta forma es:

$$1 - (1 - p \sum_{j=0}^{N-k-1} q^{2j})^2 \qquad (2)$$

Si sumamos las aportaciones de todas las circunstancias anteriores, la probabilidad de que una *VOQ* cualquiera se resuelva en la primera iteración vendría dada por:

$$P_{\psi_1}(N,p) = \frac{1}{N}\sum_{k=1}^{N}\left[q + pq^{2(N-k)} + p\left[1-(1-p\sum_{j=0}^{N-k-1}q^{2j})^2\right]\right] \qquad (3)$$

En la Fig. 1 se representa P_{ψ_1} . Si obtenemos el límite cuando $N \to \infty$, P_{ψ_1} tiende a una función convexa con un mínimo global en 94.3%; es decir, la mayor parte de las *VOQ*s parecen resolverse en la primera iteración. De hecho, en la Fig.1 se observa que el mínimo se estabiliza por encima de 94% incluso en conmutadores de tamaño medio.

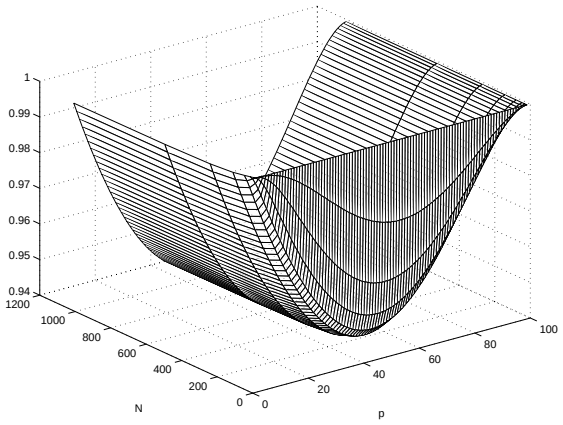


Figura 1: $P_{\psi 1}$

A continuación, deseamos determinar el número de iteraciones que se necesitan para conseguir un máximo local de la asignación. Para ello, podemos tener en cuenta que, siempre que existan *VOQ*s no vacías, PHM selecciona al menos una de ellas por iteración hasta que converge (en una iteración se seleccionan necesariamente las *VOQ*s no vacías de mayor prioridad que todavía no se hayan resuelto). Por lo tanto, encontrar el número de iteraciones necesario para conseguir un máximo local es equivalente a iterar hasta que no se puedan seleccionar más *VOQ*s para transmitir.

Una *VOQ* con prioridad *k* transmite un paquete en la primera iteración con probabilidad:

$$P\tau_1(N,k,p) = pq^{2(N-k)} \qquad (4)$$

Para valores de *n* mayores que 1, aproximaremos la probabilidad de transmisión antes de la iteración $n+1$ (evento τ_n) por:

$$P\tau_n(N,k,p) = p\left[\prod_{j=k+1}^{N}(q + p\sum_{l=j+1}^{N}P\tau_{n-1}(N,l,p)\right]^2 \qquad (5)$$

El evento τ_n tiene lugar cuando la *VOQ(i,j)* no está vacía y todas las *VOQs (l,j)* e *(i,m), l,m=1,...,N, l≠i, m≠j* que compiten con ella están vacías o, si no lo están, son bloqueadas por otras *VOQs* que ya han sido seleccionadas para transmitir pero no afectan a la *VOQ(i,j)*. Nótese que hablamos de *aproximación*, ya que suponemos que los bloqueos de las colas competidoras *VOQ(l,j)* y *VOQ(i,m)* son independientes. Esto no es cierto, porque podrían ser bloqueadas por la misma *VOQ(l,m)*. Sin embargo, después de la primera iteración la mayor parte de las *VOQs* habrán sido resueltas (P_{ψ_1} >94%) y, en consecuencia, las interdependencias serán poco probables.

Para comprobar la validez de la aproximación decidimos obtener mediante simulación el número medio de iteraciones necesarias para conseguir un *throughput* máximo, para tráfico Bernoulli uniforme y cargas de entrada variando del 0 al 100%. Para la simulación, por simplicidad, la matriz *H* se actualiza como se indica en (1). Tanto en este caso como en lo que resta del artículo, la longitud de las colas se fija de forma que no se produzcan pérdidas. Para cada valor de carga, el número de paquetes a simular se obtiene aplicando el método *Batch Means* [18], con un intervalo de tolerancia del 5% y un nivel de calidad del 95%. El resultado se muestra en la Fig. 2. En el peor de los casos, PHM necesita 3 iteraciones para alcanzar un máximo local en un conmutador *16×16*, y 4 iteraciones en un conmutador *32×32*.

A continuación, utilizando la expresión (5), estimamos $I_{PHM}(N)$, siendo $I_{PHM}(N)$ el mínimo número de iteraciones tal que:

$$max_p N \sum_{k=1}^{N} (P\tau_n(N,k,p) - P\tau_{n-1}(N,k,p)) < 1$$

Es decir, en todo el conmutador se elige menos de una *VOQ* para transmitir en la iteración $I_{PHM}(N)$. El resultado que se obtiene es $I_{PHM}(16)=3$ e $I_{PHM}(32)=4$, lo que coincide exactamente con los resultados de simulación. Además, $I_{PHM}(N)$ resultó ser menor o igual que $log_2 N$ en todos los casos considerados (hasta N=512). Esto sugiere que el número medio de iteraciones hasta la convergencia es similar al de los algoritmos de tipo iSLIP.

Los valores de $I_{PHM}(N)$ aquí obtenidos se utilizan en el apartado siguiente para calcular el tiempo necesario para finalizar una ejecución en un intervalo de paquete.

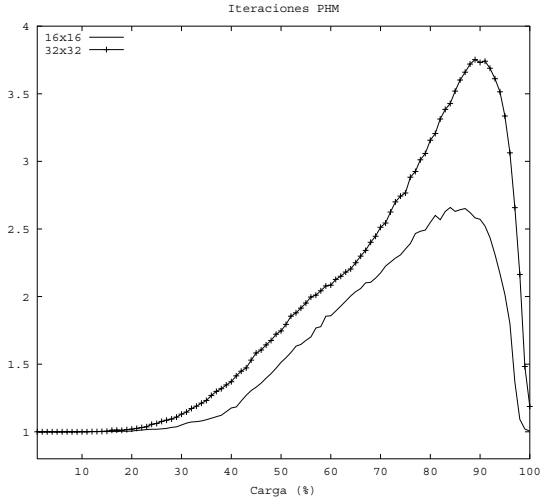


Figura 2: Número medio de iteraciones PHM para conseguir un máximo local de la asignación.

4 Retardo medio por paquete

En este apartado se comparan las prestaciones de retardo de los algoritmos descritos en el apartado 2, suponiendo que el plazo máximo (intervalo de paquete) es el mismo. El tiempo de respuesta exacto se ha obtenido a partir de implementaciones ASIC de los tres algoritmos, utilizando la librería *Ambit ASIC lca300k.alf* de *Ambit Physically Knowledgeable Sintesis 4.0 (Cadence Desing System).*

4.1 PHM frente a 2DRR

El tiempo de respuesta T_{2DRR_dec} del algoritmo de asignación jerárquica secuencial se puede formular como sigue:

$$T_{2DRR\_dec} = T_{2DRR\_aux} + N \times (T_{latch} + T_{2DRR\_stage}) \quad (6)$$

Donde T_{2DRR_stage} es el tiempo necesario para llevar a término cada paso del algoritmo (apartado 2.1), T_{latch} es el tiempo necesario para fijar resultados entre pasos y T_{2DRR_aux} es el tiempo que se necesita para recibir *R*, entregar *S* y actualizar *H*.

El tiempo de respuesta para tomar una decisión en PHM se puede formular de la siguiente manera:

$$T_{PHM\_dec} = T_{PHM\_aux} + I_{PHM} \times (T_{latch} + T_{PHM\_stage}) \quad (7)$$

Donde T_{PHM_stage} es el tiempo de respuesta de una iteración PHM (pasos 1 y 2, apartado 2.2), I_{PHM} es el número de iteraciones, T_{latch} es el tiempo que se necesita para fijar resultados entre iteraciones y T_{PHM_aux} es el tiempo necesario para recibir *R*, entregar *S* y actualizar *H*.

En la Tabla II se resume la comparación entre los algoritmos 2DRR y PHM.

Suponemos que $T_{latch} << min(T_{2DRR\_stage}, T_{PHM\text{-}stage})$ y que $T_{2DRR\_aux} \sim T_{PHM\_aux}$. Entonces, para un mismo tiempo de respuesta, el número de pasos 2DRR que nos podemos permitir $Ca \leq N$ es:

$$0 = Ca \times T_{2DRR\_stage} - I^*_{PHM} \times T_{PHM\_stage} \Rightarrow$$

$$Ca = min(N, I^*_{PHM}) \times (T_{PHM\_stage} / T_{2DRR\_satge}) \quad (8)$$

Donde I^*_{PHM} es el número mínimo de iteraciones necesarias para conseguir el menor retardo posible, cuando se utiliza PHM.

Hemos simulado PHM y 2DRR en conmutadores de tamaño *16×16* y *32×32*, para tráfico Bernoulli uniforme. Se actualiza *H* según la expresión (1).

Tabla II

Tiempo de respuesta: PHM frente a 2DRR

Tamaño	T_{2DRR_stage}	T_{PHM_stage}
16×16	1.55 ns	2.12 ns
32×32	1.83 ns	2.53 ns

Para cada tamaño de conmutador y para cada carga, el número de paquetes simulados se determinó utilizando el método *Batch Means* [18], con una calidad del 95% y un intervalo de tolerancia del 5%.

Teniendo en cuenta lo expuesto en el apartado 3, los valores escogidos para I^*_{PHM} fueron 3 en el caso *16×16* y 4 en el caso *32×32*. Las Figs. 3 y 4 muestran el retardo medio resultante por paquete. Podemos ver que PHM y 2DRR "completo" (*N* pasos) ofrecen prestaciones similares. Por el contrario, PHM es claramente superior *para un mismo tiempo de respuesta*. La diferencia aumenta con el tamaño del conmutador.

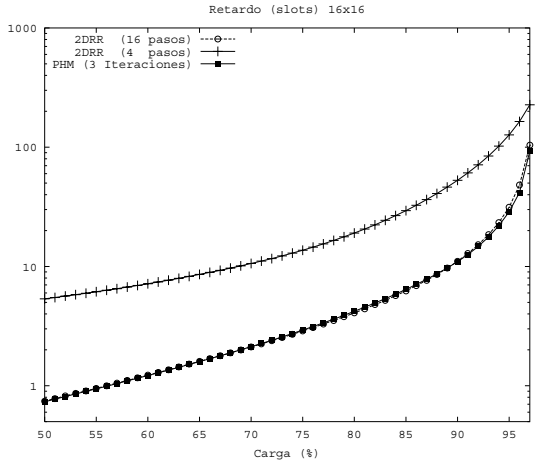


Figura 3: PHM frente a 2DRR, conmutador 16×16.

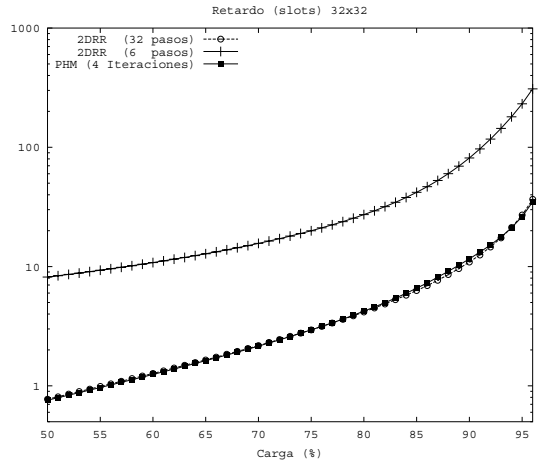


Figura 4: PHM frente a 2DRR, conmutador 32×32.

4.2 PHM frente a RDSRR

En las fases *grant* y *accept*, RDSRR utiliza *2N* árbitros *round-robin* para la toma de decisiones (uno por cada entrada y uno por cada salida). Una vez que se ha tomado una decisión, los punteros se actualizan tal y como se ha indicado en el apartado 2.

Para la implementación de RDSRR hemos seguido las indicaciones de [7] (figuras 3 y 11), basadas en las dos siguientes observaciones:

1) Desde el punto de vista de diseño de circuitos, un algoritmo de tipo iSLIP tiene dos etapas *idénticas*: *request*+*grant* y *accept*.

2) Los algoritmos de tipo iSLIP admiten una implementación solapada (*pipelined*). Es evidente que la fase *accept* de la iteración *i* se puede realizar en paralelo con la fase *request*+*grant* de la iteración *i*+1. Una vez que la fase *request*+*grant* de la iteración *i* ha finalizado, el planificador conoce el conjunto de entradas emparejadas y, por tanto, "no utilizables" en la iteración *i*+1 (aunque desconozca con qué salida se van a emparejar hasta que no termine la etapa *accept i*).

En consecuencia, el tiempo de respuesta de cada decisión se podría formular de la siguiente manera:

$$T_{RDSRR\_dec} =$$

$$T_{RDSRR\_aux} + (I_{RDSRR} + 1) \times (T_{latch} + T_{RDSRR\_stage}) \quad (9)$$

Donde $T_{RDSRR\text{-}stage}$ es el tiempo de respuesta de una etapa de tipo *request*+*grant* o *accept*, I_{RDSRR} es el número de iteraciones realizadas, T_{latch} es el tiempo para fijar los resultados entre etapas y T_{RDSRR_aux} es el tiempo necesario para recibir *R*, entregar *S* y actualizar los punteros.

La Tabla III resume los resultados que obtuvimos para los conmutadores *16×16* y *32×32*.

Para los tamaños de conmutador considerados, no es posible limitar el número de iteraciones de RDSRR para conseguir que $T_{RDSRR\_dec} \leq T_{PHM\_dec}$. Incluso en el caso de una única iteración RDSRR:

$$T_{RDSRR\_dec} - T_{PHM\_dec} =$$

$$= 2 \times T_{RDSRR\_stage} - I^*_{PHM} \times T_{PHM\_stage} =$$

6.62 ns (conmutador 16×16, I^*_{PHM}=3)

14.08 ns (conmutador 32×32, I^*_{PHM}=4)

En una hipotética implementación en la que la duración de T_{RDSRR_stage} fuese igual a la de T_{PHM_stage}, todavía podríamos permitirnos una iteración PHM adicional, debido a la falta de solapamiento de la primera etapa de RDSRR y la última. Teniendo todo esto en cuenta, en las Figs. 5 y 6 se representa el retardo medio por paquete para RDSRR con 1 iteración e I^*_{PHM}−1 iteraciones, y para PHM con I^*_{PHM} iteraciones. Podemos ver que el algoritmo PHM compite sin problemas con RDSRR, y es claramente mejor si intentamos obtener tiempos de respuesta similares.

Tabla III

Tiempo de respuesta: PHM frente a RDSRR.

Tamaño	T_{RDSRR_stage}	T_{PHM_stage}
16×16	6.49 ns	2.12 ns
32×32	12.1 ns	2.53 ns

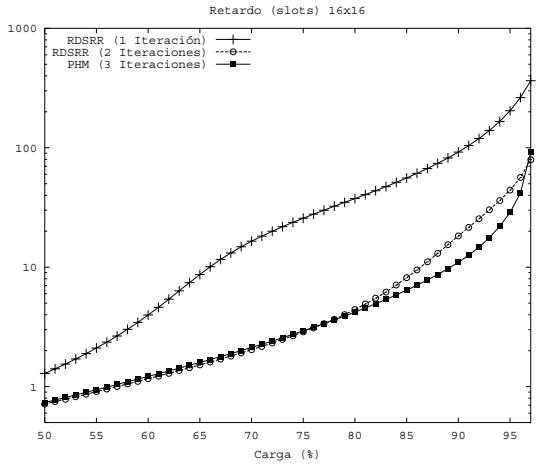


Figura 5: PHM frente a RDSRR, conm. *16×16*.

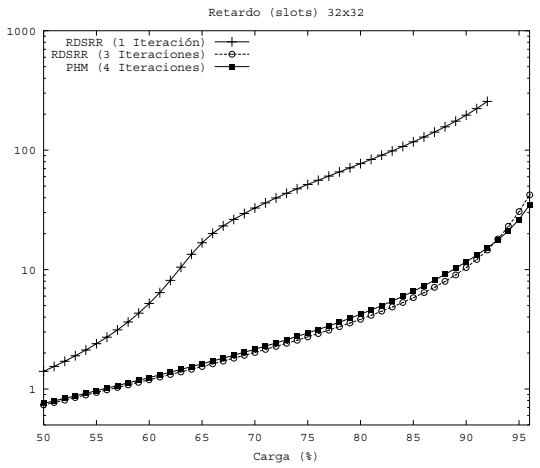


Figura 6: PHM frente a RDSRR, conm. *32×32*.

5 Conclusiones

El algoritmo de planificación PHM para conmutadores con colas virtuales a la salida es competitivo frente a los algoritmos de tipo *iterative maximal matching* y frente a los algoritmos de asignación jerárquica secuencial, tanto en términos de velocidad de convergencia (del orden de $log_2 N$), como en términos de retardo medio por paquete. Esto es especialmente patente si comparamos los retardos para una misma duración del intervalo de conmutación o paquete (en cuyo caso, podemos permitirnos más iteraciones PHM).

En un futuro inmediato, como continuación de este trabajo, nos proponemos obtener una formulación exacta de la probabilidad $P\tau_n$.

Agradecimientos

Los autores desean expresar su agradecimiento a SAIT-UPCT y al Centro de Supercomputación de Galicia -CESGA. Este trabajo ha sido financiado por el proyecto TIC2001-3339-C02-01/02.

Referencias

[1] R.Y. Awdeh y H.T. Mouftah, “Survey of ATM Switch Architectures”, Computer Networks and ISDN Systems, vol. 27, pp. 1567-1613, 1995.

[2] S.C. Liew, “Performance of Input-Buffered and Output-Buffered ATM Switches under Bursty Traffic: Simulation Study”, en Proc. IEEE Globecom, 1990, pp. 1919-1925.

[3] J. Peterson, “Throughput Limitation by Head-of-line Blocking”, en Proc. ITC-13, 1991, pp. 659-663.

[4] E. Leonardi, M. Mellia, M.A. Marsan y F. Neri, “Stability of Maximal Size Matching

Scheduling in Input-Queued Cell Switches", en Proc. IEEE ICC 2000, pp. 1758-1763.

[5] E. Leonardi, M. Mellia, F. Neri y M.A. Marsan, "On the Stability of Input-Queued Switches with Speed-Up", IEEE/ACM Trans. on Networking, vol. 9, pp. 104-118, 2001.

[6] Y. Tamir y H.-C. Chi, "Symmetric Crossbar Arbiters for VLSI Communication Switches", IEEE Trans. on Parallel and Distributed Sys., vol. 4, pp. 13-27, 1993.

[7] P. Gupta y N. McKeown, "Design and Implementation of a Fast Crossbar Scheduler," IEEE Micro Magazine, Ene.-Feb. 1999.

[8] T.E. Anderson, S.S. Owicki, J. B. Saxe, C. P. Thacker, "High Speed Switch Scheduling for Local-Area Networks", IEEE/ACM Trans. on Computer Systems, vol. 11, pp. 319-352, 1993.

[9] G. Nong, J.K. Muppala, y M. Hamdi, "Analysis of Nonblocking ATM Switches with Multiple Input Queues", IEEE/ACM Trans. on Networking, vol. 7, pp. 60-74, 1999.

[10] N. McKeown, "iSLIP: A Scheduling Algorithm for Input-Queued Switches", IEEE/ACM Trans. on Networking, vol. 7, n. 2, 1999.

[11] N. McKeown, M. Izzard, A. Mekkittikul, B. Ellersick y M. Horowitz, "Tiny tera: A packet switch core", IEEE Micro, vol. 17, pp. 26-33, 1997.

[12] D.N. Serpanos y P.I. Antoniadis, "FIRM: A Class of Distributed Scheduling Algorithms for High-Speed ATM Switches with Multiple Input Queues", en Proc. IEEE Infocom, pp. 548-555, 2000.

[13] Y. Jiang y M. Hamdi, "A Fully Desynchronized Round-Robin Matching Scheduler for a VOQ Packet Switching Architecture", IEEE Workshop on High Performance Switching and Routing 2001, pp. 407-411.

[14] J.M. Pousada-Carballo, F.J. González-Castaño, P.S. Rodríguez-Hernández y U.M. García-Palomares, "High Performance Real-Time Neural Scheduler for ATM Switches", IEEE Communications Letters, vol. 4, no. 11, pp. 372-374, Nov. 2000.

[15] R.O. LaMaire y D.N. Serpanos, "Two-Dimensional Round-Robin Schedulers for Packet Switches with Multiple Input Queues", IEEE/ACM Trans. on Networking, vol. 2, pp. 471-481, 1994.

[16] F.J. González-Castaño, C. López-Bravo, R. Asorey-Cacheda, J.M. Pousada Carballo y P.S. Rodríguez-Hernández, "Neuronal Parallel-Hierarchical-Matching Schedulers for Input-Buffered Packet Switches", IEEE Communications Letters, vol. 6, no. 5, pp. 220-222, 2002.

[17] R. Asorey-Cacheda, F.J. González-Castaño, C. López-Bravo, J.M. Pousada-Carballo y P.S. Rodríguez-Hernández, "On the Behavior of PHM Distributed Schedulers for Input Buffered Packet Switches", IEEE Transactions on Communications, aceptado y pendiente de publicación.

[18] A. M. Law and J. S. Carson, "A sequential procedure for determining the length of a steady-state simulation", Operations Research, vol. 27, pp. 1011-1025, 1979.

RRC-RED: Una solución simple al reparto equitativo de ancho de banda en RED

D. Teijeiro Ruiz, J.C López Ardao, Raúl F. Rodríguez Rubio, Manuel Fernández Veiga, Cándido López García

Dept. de Ingeniería Telemática. Universidad de Vigo
E.T.S.E. Telecomunicación. Campus Universitario Lagoas-Marcosende s/n.
36 200 Vigo, España
E-mail: Diego.Teijeiro@det.uvigo.es

Abstract *One weakness of the RED algorithm typical of routers in the current Internet is that it allows unfair bandwidth sharing when a mixture of traffic types shares a link. This unfairness is caused by the fact that at any given time RED imposes the same loss rate on all flows, regardless of their bandwidths. In this paper, we propose Random Rate-Control RED (RRC-RED), a modified version of RED. RRC-RED uses per-active-flow accounting to impose on each flow a loss rate than depends on the flow's own rate. This papers shows than RRC-RED provides better protection than RED and its variants to solve those problems (like FRED, CHOKe or RED-PD), and, moreover, it's easier to implement and lighter in complexity.*

1 Introducción

El espectacular crecimiento experimentado por Internet en los últimos años ha motivado que el control de la congestión se convierta en uno de los campos de mayor interés y actividad investigadora. La congestión se debe, fundamentalmente, a la escasez o infrautilización de tres recursos de vital importancia en los nodos: memoria, ancho de banda y capacidad de procesamiento.

Para prevenir la congestión, TCP ha utilizado tradicionalmente un mecanismo extremo a extremo en el cual las fuentes de tráfico ajustan las tasas de envío en función la información obtenida de la monitorización de sus propias conexiones.

Tradicionalmente, se han usado *routers drop-tail* (displicina que descarta siempre el último paquete que ha llegado a la cola), de forma que el descarte por desbordamiento de la memoria supone una notificación implícita de la existencia de congestión.Estos *routers* no tienen ninguna capacidad de previsión de la congestión; y cuando ésta se produce, suelen provocar una situación indeseable denominada **sincronización global**, en la que un número considerable de conexiones TCP, por poseer valores de RTT muy similares, tienden a sincronizarse en sus fases de incremento y decremento de las tasas de envío. Efectivamente, cuando las colas están llenas, los paquetes de dichas conexiones se descartan juntos. Así, todas la conexiones intentan recuperar esos paquetes y, tras un período de silencio, comienzan de nuevo una fase de envío, moviéndose las colas de los *routers* congestionados entre un estado lleno y otro vacío. De esta forma, la sincronización global puede acarrear una utilización de la red muy baja.

Una solución simple a este problema sería el uso del descarte aleatorio (*Random Drop*) [1]. No obstante, debería combinarse con mecanismos que permitiesen la notificación temprana del riesgo de congestión y la toma de medidas preventivas, y no paliativas. Es lo que se ha dado en denominar AQM (*Active Queue Management*).

Las técnicas AQM se basan en la detección temprana del riesgo de congestión por parte de los *routers*, que se lo notifican a los extremos con el objetivo de que éstos ajusten su tasa de transmisión antes de que lleguen a producirse los descartes de paquetes. AQM induce tamaños medio de cola menores y, por tanto, retardos extremo a extremo también menores.

Dentro de las técnicas AQM, las más populares son ECN (*Explicit Congestion Nofification*) [2] y RED (*Random Early Detection*) [3]. Nos centraremos en esta última por ser la más extendida actualmente, además de poseer la ventaja de no requerir modificación alguna en los extremos de la comunicación, dado que asume el funcionamiento tradicional de las fuentes tras la detección de un descarte.

El objetivo de RED es el descarte de paquetes de cada flujo proporcionalmente al ancho de banda usado por el flujo en el enlace de salida. Ello debiera implicar, por tanto, un reparto equitativo del ancho de banda disponible [3, 6].

Sin embargo, se han identificado situaciones en las que RED se comporta de forma inicua. Por ejemplo, en [7] se destaca el hecho de que, cuando varias conexiones TCP poseen diferente ocupación de memoria en un *router* RED, el descarte de paquetes proporcional no garantiza siempre un reparto equitativo del ancho de banda. El problema se puede ver agravado considerablemente en presencia de flujos UDP, pues no responden control de congestión alguno. También, en [5] se muestra que tanto *Drop-Tail* como RED son incapaces de garantizar la equidad del reparto, dándose situacio-

nes donde dos conexiones reciben más de la mitad del *throughput* total. Los propios autores apuntan que ello puede deberse al hecho de que RED incrementa la probabilidad de descarte de los paquetes, y si esto ocurre en la fase de *Slow-Start*, posteriormente resulta muy difícil recuperarse.

Por tales motivos, durante los últimos años se han propuesto algunas soluciones que intentan paliar este comportamiento indeseable de RED. Por ejemplo, en [7] se propone una modificación llamada *Flow* RED (FRED), que complementa a RED añadiendo cierta información de estado con el objetivo de dar preferencia a los flujos con un tamaño medio de cola menor, así como controlando a usuarios que no responden a las señales de congestión. Sin embargo, las mejoras de FRED no son predecibles y dependen bastante de los tiempos de llegada de los distintos flujos.

Al contrario que FRED,*Stochastic Fair Blue* (SFB) [8] no utiliza información de estado para los flujos, sino que utiliza distintos niveles de *hashing* para identificar a aquéllos que consumen mayor ancho de banda. Si bien esta solución funciona satisfactoriamente cuando hay pocos flujos compitiendo por el ancho de banda, a medida que aumenta el número de flujos, el algoritmo acaba penalizando a aquéllos que demandan menos ancho de banda.

Otra solución interesante es el algoritmo CHOKe [9], donde se propone la selección aleatoria de un paquete de la cola a la llegada de cada paquete. Si el paquete nuevo y el seleccionado pertenecen al mismo flujo, se descartan ambos; siendo aceptados ambos en caso contrario. El algoritmo se basa en la suposición de que los flujos más activos tendrán en media más paquetes en la cola. Sin embargo, CHOKe, al igual que SFB, no funciona bien cuando el número de flujos es elevado, ni tampoco en presencia de flujos UDP.

Por último, se encuentra la solución de Floyd, RED con Descarte Preferente (RED-PD) [10], que combina alguna de las ideas anteriores. Este mecanismo usa el historial de descartes de RED para identificar los flujos que consumen más de ancho de banda. Puesto que los descartes de RED son probabilísticos, y no un resultado del desbordamiento del búfer, pueden ser considerados como muestras aleatorias del tráfico entrante; si bien es muy probable que los flujos más activos tengan un mayor número de pérdidas, tal y como se ha demostrado en [11]. Con este dato, y una estimación del RTT, se identifican los flujos que envían a más tasa de la debida, penalizándolos mientras no reduzcan dicha tasa.

En este artículo se propone un nuevo algoritmo basado en RED, denominado *Random Control Rate - RED* (RRC-RED), que logra razonablemente bien la equidad en el reparto de ancho de banda, incluso mejor que algunas de las soluciones anteriores, pero con la ventaja de ser una solución muy simple, de inferior coste computacional y añadiendo una mínima información de estado en los *routers*.

El artículo está organizado de la siguiente forma: En la sección 2 se describe el funcionamiento de RED y se detallan los principales inconvenientes. La sección 3 describe el algoritmo propuesto, así como algunas cuestiones de implementación. Las prestaciones son analizadas en la sección 4. La sección 5 resume las principales conclusiones del trabajo.

2 El algoritmo RED

El algoritmo RED infiere una incipiente congestión del router cuando cierta estimación del tamaño medio de la cola (avg) supera cierto umbral (min_{th}). Para el cálculo de dicha estimación se usa un filtro paso-bajo basado en un estimador EWMA (*Exponential Weighted Moving Average*) sobre el tamaño instantáneo de dicha cola ($inst_{queue}$) cada vez que llega un paquete:

$$avg = avg * w_q + inst_{queue} * (1 - w_q) \qquad (1)$$

donde w_q es un parámetro configurable.

Si el tamaño medio de la cola está por debajo de un límite inferior (min_{th}), el paquete que acaba de llegar no se descarta. Cuando el tamaño medio de la cola sobrepasa min_{th}, pero es menor que un límite superior (max_{th}), RED descarta el paquete con cierta probabilidad P_{drop}. Esta probabilidad es proporcional al tamaño medio de la cola (avg), según la siguiente expresión:

$$P_{drop} = (avg - min_{th})/(max_{th} - min_{th}) * P_{max} \qquad (2)$$

Cuanto más próximo se halla avg a max_{th}, mayor es la probabilidad de descarte. Cuando el tamaño medio de la cola supera el límite máximo, max_{th}, el paquete entrante en ese momento es descartado con probabilidad 1. El parámetro P_{max} representa el límite superior de la probabilidad de descarte P_{drop}, siempre y cuando el tamaño medio de la cola no haya superado el límite máximo max_{th}, pues mientras esto ocurra, la probabilidad de descarte es 1, como se ha comentado con anterioridad.

De esta forma, RED puede detectar una congestión incipiente y persistente, además de permitir a los *routers* detectar ráfagas transitorias.

Finalmente, el descarte aleatorio de paquetes en RED evita el nada deseable efecto de la sincronización global.

2.1 Problemas con RED

Como ya se ha apuntado en la introducción, el objetivo de RED es que el descarte de paquetes de cada flujo sea proporcional al ancho de banda usado por el flujo en el enlace de salida. Ello se logra descartando cada paquete entrante con la misma probabilidad (suponiendo que el tamaño medio de la cola no cambie significativamente). De este modo, la conexión con la mayor tasa de entrada tendrá el mayor porcentaje total de paquetes descartados.

Según se explica en [7], si se supone que el tamaño medio de la cola no cambia durante un periodo corto de tiempo δ, RED descarta los paquetes entrantes con

una probabilidad fija p. Si se supone también que la tasa de una conexión i es λ_i (o $\lambda_i \delta$ paquetes por δ), el porcentaje de paquetes descartados en cada conexión i es:

$$\frac{\lambda_i p}{\sum \lambda_i p} = \frac{\lambda_i}{\sum \lambda_i} \quad (3)$$

Con una disciplina de servicio FCFS, la tasa de salida de una conexión i es proporcional a su ocupación de memoria, determinada ésta por el porcentaje de paquetes aceptados:

$$\frac{\lambda_i (1-p)}{\sum \lambda_i (1-p)} = \frac{\lambda_i}{\sum \lambda_i} \quad (4)$$

Es decir, ambas expresiones implican que RED descarta paquetes proporcionalmente al ancho de banda usado por cada conexión bajo disciplina FCFS.
Desde el punto de vista de una conexión, sin embargo, la tasa instantánea de descarte de paquetes durante un período pequeño δ es $\frac{\lambda p}{\lambda} = p$, que es independiente de la utilización de ancho de banda. Cuando existe riesgo de congestión, es decir, el tamaño medio de cola se sitúa por encima de min_{th}, la probabilidad de descarte tiene un valor mínimo no nulo para todas las conexiones, independientemente de su utilización del ancho de banda, lo que contribuye a una compartición del enlace de manera inicua. En [7] se apuntan las posibles causas:

- El hecho de que todas las conexiones vean la misma tasa de pérdidas implica que incluso una conexión que reciba mucho menos ancho de banda del que debiera experimentará pérdidas. Esto puede conducir a que un flujo TCP nunca reciba su parte proporcional, dado que cada pérdida puede suponer una reducción de su ventana a la mitad.

- La aceptación de un paquete de una conexión causa un aumento de la probabilidad de descarte para futuros paquetes de otras conexiones, incluso cuando éstas consuman menos ancho de banda. Esto causa descartes temporales indeseables que no mantienen la regla de proporcionalidad, incluso entre flujos idénticos.

- Una conexión que no responda a las señales de congestión puede llevar a una tasa de descarte muy alta para todas las conexiones. Ello contribuye a la incapacidad de RED para garantizar equidad en el reparto de ancho de banda a conexiones adaptativas en presencia de usuarios agresivos, incluso cuando la congestión no es severa.

3 El Algoritmo RRC-RED (*RED with Random Rate-Control*)

El algortimo que se propone en este artículo, RRC-RED, es un nuevo intento de mejorar la equidad en el reparto de ancho de banda cuando se usa RED. La principal novedad frente al resto de soluciones radica en un sencillo cálculo de la tasa media de cada flujo, de forma que cuando se deba descartar un paquete según RED, la elección recaiga con mayor probabilidad sobre los flujos que mayor ancho de banda están usando realmente.
El cálculo de esta tasa se realiza, cada vez que se extrae un paquete de la cola de un flujo i para ser enviado sobre el enlace, mediante un estimador EWMA:

$$th\_m_i = th\_m_i * w_{th} + th\_inst_i * (1 - w_{th}) \quad (5)$$

donde th_m_i es el *throughput* medio del flujo i, th_inst_i se refiere al *throughput* instantáneo del último paquete que llegó (calculado como el tamaño del paquete entre la diferencia entre los dos últimos instantes de llegada) y w_{th} es un parámetro de configuración. Por tanto, la diferencia fundamental entre RED y RRC-RED se sitúa en el momento del descarte. Mientras RED descarta el paquete entrante, RRC-RED introduce este paquete en la cola, para posteriormente elegir aleatoriamente *N* flujos de entre todos los que tienen en ese momento paquetes en la cola, descartando el último paquete que haya llegado del flujo que posea en ese momento la mayor tasa media. La elección del último paquete está pensada fundamentalmente para TCP, pues de este modo se evitan retransmisiones innecesarias.
Esta operación se puede implementar de una manera eficiente mediante un pequeño número de instrucciones de suma y desplazamiento en cada llegada de paquete. Al igual que en RED, estos cálculos se pueden realizar en paralelo con el envío de paquetes. De este modo, los cálculos necesarios no debieran influir significativamente sobre la capacidad de procesado del *router*.
Por otro lado, el uso de este estimador hace que RRC-RED, a diferencia de RED, otorgue una probabilidad de descarte muy baja para conexiones TCP que se hallan en la fase de *slow-start*.
A continuación se discute la elección del número *N* de flujos entre los que elegir el descarte, y en la sección 4 se presentan los resultados obtenidos con RRC-RED en distintos escenarios, en comparación con los de RED y las distintas soluciones propuestas en la literatura.

3.1 Elección del Número de Flujos

El algoritmo *RRC-RED*, a la hora de realizar el descarte aleatorio, elige la víctima entre *N* flujos cualesquiera. Tras muchas pruebas, se ha observado que el hecho de aumentar el número *N* de flujos sobre los que realizar la elección del descarte no mejora excesivamente los resultados obtenidos, mientras que se aumenta considerablemente la carga computacional del algoritmo. Es evidente que, al aumentar *N*, aumentará el número de comparaciones a realizar, lo cual puede introducir una carga computacional inaceptable. De hecho, tomando sólo dos flujos al azar, se logra mejorar RED con un incremento poco significativo de la carga computacional. Este es uno de los inconvenientes de las

Tabla 1: Influencia del Número de Flujos. Desv. Típica

Flujos Sel.	Total Flujos		
	8	20	80
2	8.35e+05	3.42e+05	1.18e+05
7	5.94e+05	5.38e+05	1.21e+05
15	-	9.21e+05	1.09e+05
70	-	-	1.19e+05
Mbps/Flujo	5.60	2.25	0.53

soluciones propuestas, pues no se había alcanzado un compromiso entre equidad y eficiencia.

Para calcular las diferencias que producen distintos valores de *N*, se han tomado los *throughputs* medios obtenidos por cada flujo, calculando posteriormente la desviación típica de éstos respecto al *throughput* medio de un flujo cualquiera. Este valor representa una medida de la dispersión de los flujos, de modo que valores pequeños implicarán que todos tienen una tasa media muy similar; todo lo contrario ocurre si el valor de la desviación típica es elevado. En la tabla 1 se muestran las diferencias en los resultados para distintos valores de *N*. A la vista de los resultados, se observa que el aumento del número de flujos disponibles para la elección no implica necesariamente un incremento en la equidad del *router*, tal y como se puede ver comparando las desviaciones típicas obtenidas tras promediar los flujos individuales en cada una de las simulaciones. Es cierto que, en determinadas circunstancias el aumento del número de flujos implica un mejor reparto del ancho de banda, pero a costa de aumentar significativamente la complejidad computacional del algoritmo. Dado que la mejora en la equidad apenas es relevante, sacrificamos ligeramente el rendimiento del algoritmo en favor de la eficiencia del mismo.

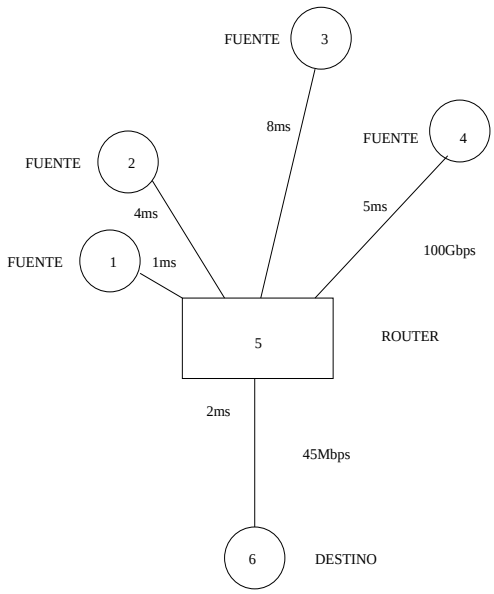
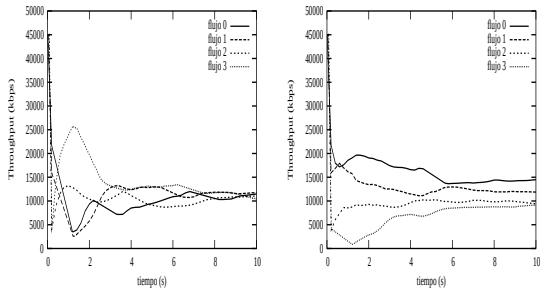


Figura 1: Escenario 1



(a) RRC-RED (b) RED

Figura 2: Comparación entre RRC-RED y RED

4 Evaluación de RRC-RED

En esta apartado se presentan los resultados obtenidos con el algoritmo propuesto sobre múltiples escenarios, comparando sus prestaciones con las de RED, y el resto de soluciones propuestas. Para ello se ha usado el **Network Simulator** [12].

Comenzamos comparando el funcionamiento de RRC-RED en un escenario idéntico al utilizado en [3], y que se muestra en la Figura 1. Se utilizan cuatro conexiones TCP con diferentes RTTs y diferentes instantes de comienzo. Los tamaños de las ventanas, fijados por la capacidad del enlace más pequeño y por el RTT, tendrán, respectivamente, unos valores de 33, 67, 112 y 78 paquetes. Las simulaciones se han realizado con un tamaño de paquete constante de 1000 bytes. El búfer del *router* es de 1000 paquetes, siendo $min_{th} = 30$ paquetes, $max_{th} = 80$, $w_q = 0{,}002$, y $max_p = 1/10$. Los instantes de comienzo de transmisión de cada fuente están separados 20 milisegundos, de una manera similar a como se hace en [3]. Las fuentes siempre tienen datos que enviar.

En la Figura 2 se muestra una comparación entre RED y RRC-RED. Puede observarse que el algoritmo RRC-RED consigue en pocos segundos que los cuatro flujos tengan tasas muy similares, al contrario de lo que ocurre con RED, incapaz de rebajar la tasa del primer flujo una vez que éste ya ha acomodado su ventana, forzando a los flujos nuevos a reducir la suya de manera radical. El funcionamiento de RED es claro: el primer flujo en llegar (flujo 0) dispone de todo el ancho de banda para él, y fija su ventana para poder aprovechar dicha capacidad; sin embargo, los flujos 1, 2 y 3, que aparecerán cada 20 ms, verán un ancho de banda mucho menor, pues en esos momentos es el flujo 0 el que monopoliza el enlace. Será entonces cuando entre en funcionamiento el mecanismo del descarte aleatorio, el cual tiene un efecto mucho más importante sobre los flujos que todavía se encuentran en la fase de *slow-start* y que aún se encuentran ajustando el tamaño de su ventana. Es obvio que cuando se llega a este punto, tal y como se puede ver en la gráfica, resulta difícil conseguir que el flujo más activo reduzca su ta-

sa, y viceversa, que el menos activo aumente su tasa, por lo que las diferencias de tasa entre todos los flujos son significativas. Nuestro algoritmo intenta evitar en la medida de lo posible este efecto indeseado, permitiendo alcanzar en poco tiempo la equidad en el reparto del ancho de banda. Esta velocidad de convergencia no discrimina entre conexiones de corta y larga duración, proporcionando un reparto equitativo en cualquiera de estas situaciones, algo que, como se puede observar, RED es incapaz de realizar de forma satisfactoria.

A continuación mostramos otra comparación que incluye las otras propuestas, RED-PD, FRED y CHOKe. Se usan ahora 8 flujos TCP en una simulación mucho más larga en el mismo escenario (se duplican las cuatro fuentes).

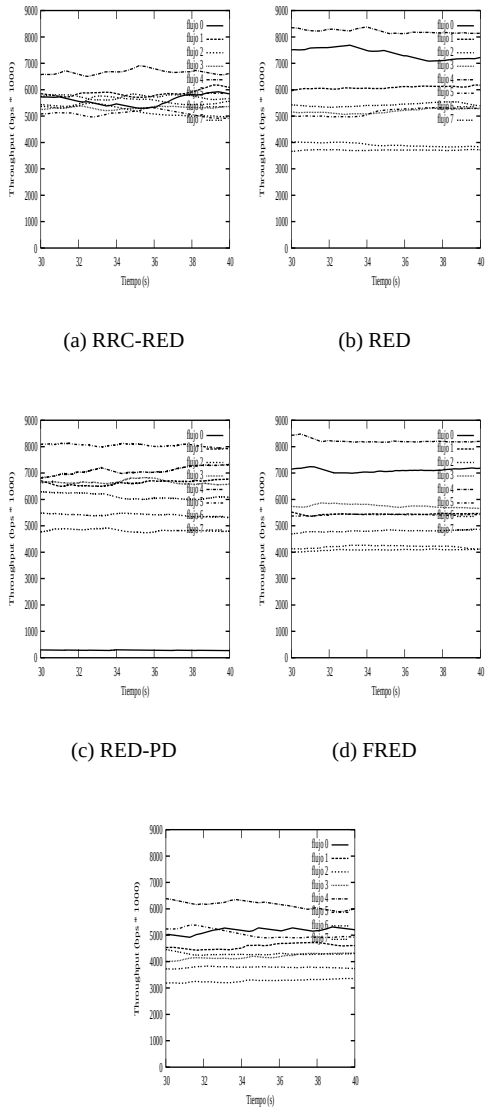
En la figura 3 se muestran los segundos finales de cada simulación, donde de nuevo se observa un correcto funcionamiento de RRC-RED. Si elegimos un flujo cualquiera y seguimos la evolución de su tasa a lo largo del tiempo, podemos observar los siguientes comportamientos: cuando su tasa es alta, es probable que tarde o temprano sea la víctima de los descartes, por lo que su tasa descenderá poco a poco. Tras haber reducido su tasa lo suficiente, el flujo dejará de ser el objetivo del algoritmo de descarte, por lo que paulatinamente su tasa irá aumentando de nuevo, mientras son otros flujos los que ven como la suya disminuye. Así, la tasa de cada flujo fluctuará ligeramente con el tiempo, dependiendo de en qué estado se encuentre.

Este control de la tasa de cada flujo permite mantener unos valores de *throughput* bastante parecidos para cada flujo, algo que no es capaz de realizar con tanta eficacia el resto de soluciones probadas. En las gráficas de la figura 3, se observa cómo el algoritmo RED mantiene los mismos problemas que en el caso anterior. Algo parecido le ocurre a RED-PD y a FRED. FRED apenas mejora, el comportamiento de RED, mientras que RED-PD, si bien consigue una cierta mejora, se olvida del flujo 0, al cual le otorga una ventana muy pequeña desde el principio. La única solución que funciona aceptablemente es CHOKe, aunque la equidad en el reparto es bastante peor que en RRC-RED.

Parece claro, pues, que el algoritmo propuesto funciona mejor, a la hora de proporcionar equidad a varios flujos, tanto para conexiones TCP cortas como largas.

Otro escenario interesante, donde se puede observar perfectamente el comportamiento de cada algoritmo, es aquél donde tenemos un número n (en este caso 8) de fuentes TCP idénticas. Situamos sus instantes de comienzo de forma equiespaciada en el tiempo 0.5 segs. Utilizamos el mismo escenario que los ejemplos anteriores, salvo que en este caso todas las ventanas son idénticas (100 paquetes), y el retardo que introduce cada enlace es siempre de 2 ms.

De nuevo, a la vista de la Figura 4, se observa la evolución favorable de RRC-RED en el tiempo. Mientras disciplinas como RED o RED-PD se encuentran ya en un régimen permanente, no ocurre lo mismo con RRC-RED, en donde las tasas de los flujos más favorecidos disminuyen paulatinamente, aumentando los menos favorecidos hasta entonces. Respecto al resto de



(a) RRC-RED (b) RED

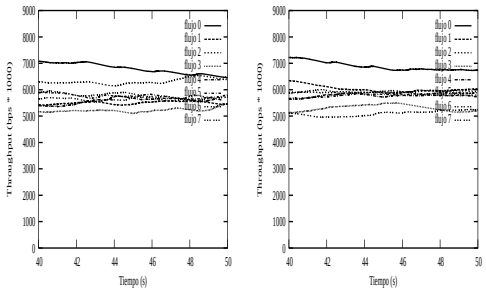
(c) RED-PD (d) FRED

(e) CHOKe

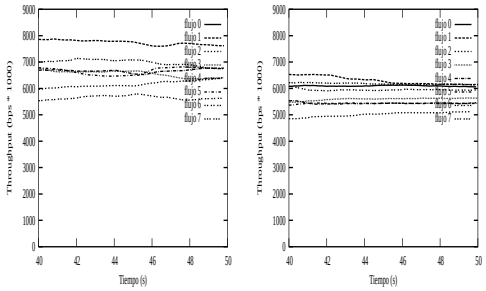
Figura 3: Comparación entre RRC-RED, RED, RED-PD, FRED y CHOKe

disciplinas, puede verse el fracaso de CHOKe a la hora de igualar las tasas, y el buen funcionamiento, en este caso concreto, de FRED, aunque ya hemos visto que no se comporta tan bien en el resto de escenarios.

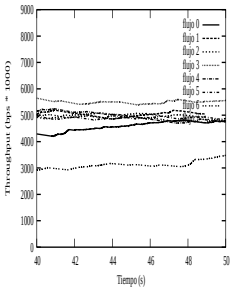
Para finalizar, compararemos la complejidad de los algoritmos analizados. RRC-RED tan sólo añade a RED tres comparaciones y una operación aritmética, con dos variables de flujo cada vez que se ejecuta el algoritmo. En cambio, FRED mantiene dos variables por cada flujo, realizando ocho comparaciones y 6 operaciones aritméticas más que RED, cada vez que un paquete es encolado y desencolado. RED-PD utiliza una lista donde se almacenan los últimos descartes de cada flujo en un intervalo determinado, lo que contribuye a aumentar bastante la información almacenada en el *router* cuando el número de flujos y descartes es elevado. Por último, CHOKe es la más simple de todas,



(a) RRC-RED (b) RED



(c) RED-PD (d) FRED



(e) CHOKe

Figura 4: Comparación con flujos idénticos

pues no necesita información de estado para los flujos y realiza sólo una comparación más que RED.

5 Conclusiones

El problema del control de la congestión de Internet ha sido investigado intensamente en los últimos años. Una de las soluciones con más éxito en este campo ha sido RED. Sin embargo, en numerosas situaciones RED se ha revelado como un mecanismo incapaz de garantizar un reparto equitativo del ancho de banda. Muchas han sido las propuestas aparecidas en la literatura para subsanar este inconveniente, aunque ninguna de ellas lo consigue de forma satisfactoria. En este artículo se presenta un nuevo algoritmo, RRC-RED, un mecanismo que utiliza una estimación de la tasa de los flujos entrantes en un *router* para detectar aquéllos que más ancho de banda consumen y, por tanto, descartar preferiblemente paquetes pertenecientes a dichos flujos con el fin de controlar el ancho de banda que éstos reciben y garantizar un reparto equitativo entre todos los flujos presentes. Se ha comprobado mediante simulación el funcionamiento del algoritmo propuesto, y se ha comparado con otras soluciones propuestas en la literatura.

6 Agradecimientos

Este trabajo ha sido subvencionado por el proyecto: "TIC2000-1126 del Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica".

Referencias

[1] A. Mankin, K. K. Ramakrishnam., editors for the IETF Performance and Congestion Control Working Group, Gateway Congestion Control Survey, RFC 1254, Agosto 1991, p. 21.

[2] S. Floyd. TCP and Explicit Congestion Notification. *ACM Computer Communication Review*, vol. 24, no. 5, pp. 10-23, Octubre 1994.

[3] S. Floyd y V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397-413, Agosto 1993.

[4] S. Floyd. RED: Discussions of Setting Parameters. `http://www.icir.org/floyd/REDparameters.txt`

[5] M. May, C. Diot, B. Lyles. Reasons not to Deploy RED, en *Proceedings of the IEEE/IFIP IWQoS*, Junio 1999.

[6] V. Rosolen, O. Bonaventure, G. Leduc. A RED discard strategy for ATM networks and its performance evaluation with TCP/IP traffic.*ACM Computer Communication Review*, Julio 1999.

[7] D. Lin and R. Morris, Dynamics of Random Early Detection, en *Proceeding of ACM Sigcomm* ACM, New York, 1997, pp. 127-137.

[8] W. Feng, D.D. Kandlur, D. Saha, K.G. Shin, "Stochastic Fair Blue: A Queue Management Algorithm for Enforcing Fairness", IEEE INFOCOM, April 2001.

[9] R. Pan, B. Prabhakar, and K. Psounis. CHOKe, A Stateless Active Queue Management Scheme for Approximating Fair Bandwidth Allocation. In IEEE INFOCOM, March 2000.

[10] Ratul Mahajan and Sally Floyd, RED with Preferential Dropping (RED-PD).E n Proc. ACM 9th International Conference on Network Protocols (ICNP), Nov. 2001.

[11] S. Floyd and K. Fall. Router mechanisms to support end-to-end congestion control, February 1997. LBL Technical Report.

[12] http://www.isi.edu/nsnam/ns

Algoritmo Óptimo de Selección de Longitud de Onda en Arquitecturas de Conmutación Óptica de Paquetes SCWP

Pablo Pavón Mariño, Joán García Haro, Josemaría Malgosa Sanahuja, Fernando Cerdán
Departamento de Tecnología de la Información y las Comunicaciones. Universidad Politécnica de Cartagena
Campus Muralla del Mar s/n
30202 Cartagena
Teléfono: 968 32 59 52 Fax: 968 32 53 38
E-mail: {Pablo.Pavon,Joang.Haro,Josem.Malgosa,Fernando.Cerdan}@upct.es

Abstract. *Optical Packet Switching (OPS) is far from commercial stage in backbone WDM networks due to hardware costs. Scattered Wavelength Path (SCWP) and Shared Wavelength Path (SHWP) are the two envisioned alternatives for network control and operation of this type of networks, proposed under the WASPNET project. SCWP offers the possibility to improve the statistical multiplexing of traffic in the links comparing to SHWP approach. For SCWP solutions, a wavelength selection mechanism which decides on packet output wavelength is required in each switching node. In this paper, an algorithm of this type is proposed, which provides an optimum performance when applied to output buffered OPS architectures. An analytical model for the algorithm evaluation is also presented. This scheme is applied in the study and discussion of the impact of SCWP and SHWP switching modes on performance and hardware costs for the KEOPS OPS switch fabric architecture. To achieve it, we conduct a necessary set of modifications for this architecture, which was not originally conceived with the SCWP/SHWP operational modes in mind.*

1 Introducción

1.1 Conmutación Óptica de Paquetes

El impresionante crecimiento del tráfico de Internet continúa estimulando el avance en tecnologías que permitan soportar la demanda de tráfico. Respecto a las llamadas redes troncales, uno de los progresos más impactantes en la última década ha sido la tecnología óptica de multiplexación por división en longitud de onda (*WDM, Wavelength Division Multiplexing*). La aplicación de esta técnica a las redes de fibra óptica instaladas ha multiplicado el ancho de banda de transmisión en estas redes troncales. El problema del reparto de este impresionante ancho de banda entre los demandantes, en la red troncal ha seguido tres estrategias distintas: Encaminamiento de Longitudes de Onda, Conmutación Óptica de Ráfagas y Conmutación Óptica de Paquetes (*Wavelength Routing*, *Optical Burst Switching* y *Optical Packet Switching*) [1].

Las redes WDM que utilizan la técnica de conmutación *Wavelength Routing (WR)* son la opción comercial más avanzada que se puede encontrar en el actual estado del arte de la tecnología fotónica. La arquitectura de este tipo de redes está basada en una topología de interconexión arbitraria de nodos WXC (*Wavelength Crossconnect Nodes*), también llamados encaminadores WDM. Una conexión de tráfico entre dos nodos de acceso (frontera) implica la reserva en modo conmutación de circuitos de un canal (longitud de onda) a lo largo de una serie de fibras atravesando la red troncal. El circuito establecido actúa como un medio de transmisión transparente, ahorrando costes respecto a las redes troncales JDS (Jerarquía Digital Síncrona) "convencionales", ya que no es necesario realizar un procesado electrónico de la señal en cada salto, sino únicamente en los nodos frontera. En este tipo de redes, el establecimiento de los circuitos está controlado por especificaciones como el *Multiprotocol Lambda Switching (MPλS)*. Sin embargo, la ineficiencia inherente de los mecanismos de conmutación de circuitos enfrentados a patrones de tráfico como el de Internet, hace que sea necesario sobredimensionar los nodos de conmutación para tener una probabilidad de bloqueo aceptable. Asimismo, es necesario aplicar técnicas de aglomeración de tráfico (*traffic grooming*) en los nodos frontera, con el gasto de gestión y procesamiento que conllevan. Este problema de la granularización de tráfico empeora a medida que la tecnología eleva las velocidades de transmisión en cada canal (existen dispositivos a *40* Gbps en estado comercial). Por todos estos motivos, la utilización del ancho de banda es muy pobre para las fluctuaciones de tráfico asociadas a redes como Internet.

La conmutación óptica por ráfagas, *Optical Burst Switching (OBS)* permite un control más fino del ancho de banda de transmisión. Esta técnica de conmutación establece que cuando una fuente desea transmitir una cierta cantidad de datos a un mismo nodo destino, debe encapsularlos en una ráfaga de tamaño variable, que es conmutada como un todo a lo largo de la red. Esta técnica permite una utilización más eficiente del canal, a costa de una mayor complejidad *hardware* de los dispositivos.

Sin embargo, la utilización más eficiente del canal debido a la multiplexación estadística del tráfico, y el mecanismo más directo para adaptarse a patrones de

tráfico de las redes de datos actuales se alcanza con la tercera alternativa: Conmutación Óptica de Paquetes (*Optical Packet Switching, OPS)*. En el dominio óptico, OPS es similar a las técnicas tradicionales de conmutación de paquetes bajo tecnología electrónica, excepto que la carga de datos (*payload*) del paquete permanece en estado óptico, mientras su cabecera es procesada electrónicamente. Este paradigma proporciona la mayor eficiencia en el uso del canal, al operar en la granularidad de un paquete. Sin embargo, la necesidad de realizar la función de conmutación paquete por paquete a las velocidades existentes en la red troncal implica las mayores exigencias para la tecnología fotónica. Por ello, no se prevé la construcción de ninguna red comercial OPS hasta el medio plazo.

Conceptualmente, la arquitectura de una red troncal OPS está basada en un conjunto de nodos conocidos como *Optical Cross-Connect Nodes (OXC)*, interconectados por enlaces WDM, como se muestra en la figura 1. Los nodos frontera de esta red troncal son el comienzo y final de paquetes ópticos, que en este artículo se asumirán de longitud fija. Los dispositivos que previsiblemente serán empleados en estos nodos frontera serán *routers* electrónicos trabajando a velocidades de terabits por segundo, con el necesario funcionamiento multiprotocolo, y una alta capacidad de almacenamiento [2]. Sin embargo, existen diversos aspectos de las redes troncales OPS que todavía están en fase de estudio, como el entramado de datagramas IP sobre este tipo de redes, o el tamaño óptimo de paquete. El mayor rendimiento se obtendría con segmentación en paquetes pequeños en los nodos frontera, estando este aspecto limitado por la pérdida de eficiencia en el *ratio* cabecera/datos, y por el tiempo de proceso de paquete. Un tamaño en tiempo realista de paquete, del orden de 1μs, transporta *5000* bytes en un canal a *40* Gbps, lo que está un orden de magnitud por encima de los valores habituales de tamaño máximo de segmentos de transporte TCP negociados en la red. Por lo tanto, presumiblemente, deberá ser implementado algún mecanismo de agregación de tráfico, de manera homóloga a lo especificado en la técnica de *Optical Burst Switching*.

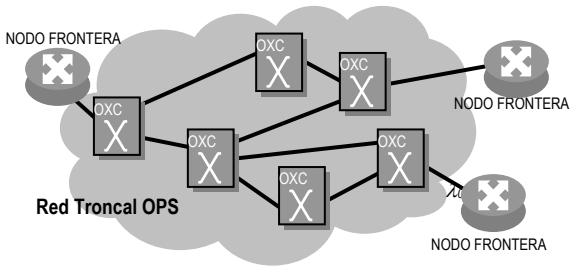


Fig. 1. Arquitectura de Red Troncal de Conmutación Óptica de Paquetes.

1.2 Modos de Operación de una Red Óptica de Paquetes

Está generalmente aceptado que las conexiones de tráfico entre nodos frontera (circuitos virtuales permanentes) serán implementadas mediante lo que se ha dado en llamar *Optical Packet Paths (OPP)*. La provisión de un OPP fija la secuencia de fibras a atravesar en cada salto desde el nodo de entrada hasta el de salida por los paquetes pertenecientes a ese OPP. La familia de especificaciones GMPLS (*Generalized Multi-Protocol Label Switching)* se conjetura como el mecanismo de control para el establecimiento de las conexiones y sus rutas. Esto permitirá el aprovechamiento de las investigaciones realizadas en distintos protocolos de señalización y control, para poder maximizar las ventajas en cuanto asignación de ancho de banda que permite OPS. Existen incluso varias propuestas para realizar parte del procesado de los paquetes (identificados por etiquetas) directamente bajo tecnología óptica [3].

En la frontera entre las redes WDM y la tecnología OPS, el modo de operación de la red establece la manera en la que las conexiones de tráfico (OPPs), son asociadas a las longitudes de onda en los enlaces (asumiendo que el número de OPPs que atraviesan un enlace es mucho mayor que el número de longitudes de onda). Este aspecto ha sido tratado en el proyecto WASPNET [4-6], donde se propusieron dos posibles metodologías: *Scattered Wavelenght Path (SCWP)*, y *Shared Wavelength Path (SHWP)*. La técnica SHWP especifica que en el camino óptico de cada OPP debe fijarse la fibra, y la longitud de onda de transmisión en cada salto. De esta manera, un paquete entrante en un conmutador demanda una fibra de salida y una longitud de onda de salida concretos, cuyos valores se leen de una tabla, tras ser establecidos durante la provisión de la conexión. La alternativa SCWP hace que la longitud de onda en cada salto no esté precisada. De esta manera, un conmutador de un nodo tiene libertad para elegir la longitud de onda de cada paquete dentro de la fibra de salida demandada, siguiendo criterios de eficiencia como el de la ocupación de las memorias.

SCWP y SHWP son los candidatos para el despliegue de una futura red troncal basada en conmutación óptica de paquetes. SCWP tiene la capacidad de ofrecer mejores prestaciones que SHWP debido a la flexibilidad para distribuir los paquetes en los canales de salida. La multiplexación estadística que se puede obtener, permite disminuir los requisitos de almacenamiento y el tiempo de espera en cola. Para ello, SCWP requiere una decisión de un planificador sobre la longitud de onda de salida, en vez de una consulta de un valor en una tabla (SHWP). De esta manera, se hace interesante el estudio de algoritmos de implementación sencilla que permitan explotar la flexibilidad en la asignación de canal de salida, potenciando el efecto de multiplexación estadística de OPPs en las fibras de salida.

En este artículo, se propone un algoritmo que cumple esas características, y que proporciona unas prestaciones *óptimas* cuando es aplicado a arquitecturas de conmutadores OPS que permitan emular colas a la salida. Se propone asimismo un modelo analítico que permite la evaluación de esta arquitectura, bajo condiciones de tráfico Bernouilli. A través de este modelo, se comparan las prestaciones de las alternativas SCWP y SHWP. El hilo conductor de la comparativa, es mostrar el ahorro en complejidad *hardware* que dicho algoritmo proporciona cuando es aplicado sobre una arquitectura OPS concreta con capacidad de emulación de colas a la salida, como es el conmutador KEOPS, de tipo difusión-selección (*broadcast-and-select*). Esta arquitectura, así como otras propuestas fuera del proyecto WASPNET, fueron diseñadas sin tener en cuenta la posible aplicación de los modos de operación SHWP/SCWP. Por ello, es necesario un proceso inicial de normalización de la arquitectura, para que estos modos de operación puedan ser aplicados y sus prestaciones comparadas.

El resto de este artículo está organizado de la siguiente manera. La segunda sección se centra en la descripción de la arquitectura KEOPS, y el proceso de adaptación requerido para la aplicación de los modos de operación SHWP/SCWP. A continuación, la sección 3 presenta el algoritmo de selección de longitud de onda de salida propuesto. La sección 4 presenta el modelo de análisis de las prestaciones de este algoritmo. La sección 5 aplica este análisis para la evaluación y comparativa de las alternativas SHWP/SCWP. Finalmente, la sección 6 concluye este artículo.

2 Arquitectura KEOPS

A mediados de los años 90, el Proyecto Europeo ACTS KEOPS (*KEys to Optical Packet Switching*), realizó una intenso trabajo en el campo de la conmutación óptica de paquetes, con el objetivo de definir la viabilidad de lo que se dio en llamar una Red Óptica Transparente de Paquetes (*Optical Transparent Packet Network -OTP-N)* [7]. De entre las arquitecturas de conmutación propuestas, la que ha recibido mayor atención ha sido el conmutador KEOPS de difusión-selección (*broadcast-and-select*). En este sentido, se construyó y probó satisfactoriamente un prototipo de *16x16* puertos operando a *10* Gbps, con un tamaño de trama de *1,646μs* (*1680* bytes a *10* Gbps).

En el conmutador KEOPS original, los paquetes entrantes por los distintos puertos de entrada son convertidos a una longitud de onda fija, y diferente para cada puerto. Posteriormente, se difunden por *M* líneas de retardo de longitudes ópticas de *0* a *M-1* ranuras temporales. Los paquetes entrantes al conmutador en las *M* últimas ranuras temporales son seleccionables en la sección de salida para ser transmitidas por cualquier puerto, mediante dos etapas de puertas ópticas basadas en Amplificadores Ópticos Semiconductores (AOS). La primera etapa de puertas ópticas selecciona uno entre *M* retardos (el tiempo de llegada del paquete), y la segunda etapa selecciona la longitud de onda (puerto de entrada del paquete). La arquitectura KEOPS ha sido objeto de gran interés debido a que no requiere convertidores de frecuencia sintonizables (sino a longitud de onda fija), permite emular el comportamiento de un conmutador con colas a la salida, la implementación de *multicast*, y la priorización de tráfico. Esta priorización es posible ya que la decisión sobre el momento de salida de un paquete puede realizarse con posterioridad a su llegada, algo no habitual en conmutadores basados en líneas de retardo. Su mayor desventaja es su difícil viabilidad para tamaños de conmutador mayores a *32x32* puertos [8], al tener asociadas unas pérdidas ópticas proporcionales a NM^2 y requerir $(NM+N^2)$ puertas ópticas en su construcción (donde *N* es en número de puertos de entrada y de salida) [7].

El conmutador KEOPS originalmente propuesto, fue diseñado contemplando puertos de entrada y salida mono-frecuencia (no WDM). Por ello, la aplicación de los modos de operación SHWP/SCWP requiere un proceso de adaptación con los siguientes pasos:

1) Adaptación de la arquitectura a puertos de entrada y salida WDM.

2) Especificación de un algoritmo de selección de retardo para el modo de operación SHWP, con el objetivo de maximizar las prestaciones del mismo ante distintos patrones de tráfico de entrada.

3) Especificación de un algoritmo de selección de longitud de onda de salida para el modo de operación SCWP.

Las modificaciones asociadas al primer punto comienzan con el añadido de una etapa de demultipexación óptica de los puertos de entrada WDM, en la sección de entrada del conmutador. Esta sección se conecta a un conmutador KEOPS de tamaño *nNxnN*, donde *N* es el número de fibras de entrada y salida, y donde *n* es el número de longitudes de onda por fibra. En la sección de salida, *n* puertos de salida sufren una conversión fija de longitud de onda y una multiplexación, para cada fibra de salida. En la figura 2, se puede observar un esquema de la arquitectura global.

El segundo punto del proceso de normalización decide sobre el retardo a asignar a los paquetes de entrada. El puerto de salida del conmutador KEOPS *nNxnN* subyacente vendrá definido por el OPP del paquete. Nuestro objetivo es encontrar un algoritmo que optimice el caudal (*throughput*) y retardo medio del conmutador, manteniendo el orden de los paquetes. En las redes WDM SHWP se preserva el orden de los paquetes extremo a extremo dentro del

mismo OPP, asegurando que cada uno de los nodos intermedios no genera desorden interno. Bajo estas consideraciones, la asignación de retardo *first-in-first-out (FIFO)* es la solución óptima obvia para un conmutador como el KEOPS capaz de emular colas a la salida. Esto se implementa fácilmente mediante un contador por puerto de salida que asigne incrementalmente el retardo a los paquetes.

La respuesta al tercer punto del proceso de normalización consiste en el algoritmo de selección de longitud de onda propuesto, que será descrito en la siguiente sección.

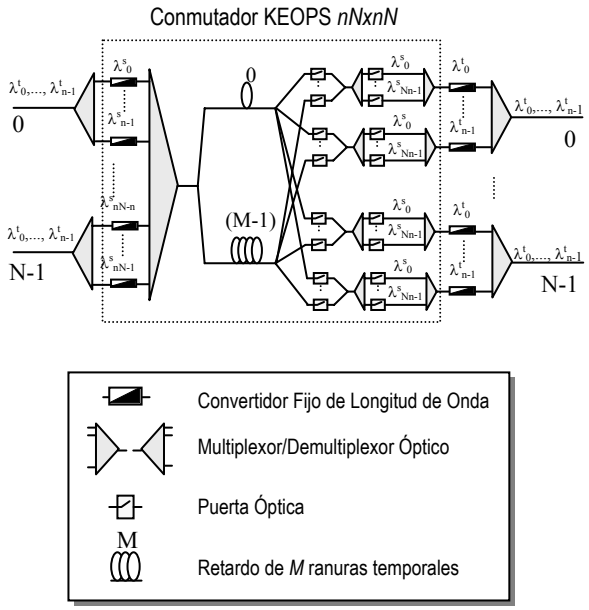


Fig. 2. Arquitectura KEOPS WDM.

3 Algoritmo de Selección de Longitud de Onda de Salida

En este apartado, se propone un algoritmo de selección de longitud de onda de salida, que ofrece prestaciones óptimas cuando se aplica a conmutadores OPS con capacidad de emulación de colas a la salida. En el caso del conmutador KEOPS, la elección de longitud de onda de salida se convierte en una selección entre los *n* puertos del conmutador *nNxnN* subyacente, correspondientes a la fibra de salida demandada por el paquete.

Los requisitos de un algoritmo de selección de longitud de onda son de nuevo los de optimización de *throughput*, retardo medio y conservación del orden en la entrega de los paquetes. Sin embargo, el problema de desorden extremo a extremo en redes SCWP WDM no es trivial. Esto se debe a que en estas redes, más de un paquete perteneciente al mismo OPP puede ser transmitido simultáneamente, en longitudes de onda distintas. Por ello, es necesario un mecanismo que asegure que un nodo sepa cuál es el orden correcto entre llegadas simultáneas de paquetes, y por lo tanto pueda mantener ese orden en el siguiente salto.

El desorden de paquetes en redes SCWP ha sido investigado en [6], y un algoritmo de secuenciación de paquetes sin el requisito de un contador global por OPP fue descrito para la versión retroalimentada (*feedback*) del conmutador WASPNET [4]. La necesidad de un contador global por OPP en la cabecera del paquete es una solución a evitar ya que aumenta el tamaño de la cabecera y añade complejidad al procedimiento. El método especificado en [6] se basa en la transmisión de los paquetes simultáneos dentro del mismo OPP, ordenados por su longitud de onda, de tal forma que los paquetes de orden más bajo deben transmitirse en longitudes de onda menores. Dados p_i, p_j, $i<j$, paquetes asociados al mismo OPP, el método requiere que: (1) p_i sea transmitido antes de p_j, o (2) p_i sea transmitido durante la misma ranura temporal que p_j, y $\lambda_i<\lambda_j$. El orden extremo a extremo dentro del mismo OPP se mantiene, si todos los nodos atravesados cumplen las condiciones anteriores.

El pseudocódigo del algoritmo propuesto en este artículo se describe en la figura 3, asumiendo un conmutador simétrico con *N* fibras de entrada y salida, *n* longitudes de onda por fibra y *M* retardos. Se trata de una simplificación del algoritmo presentado en [6], aplicado a conmutadores que emulen colas a la salida. El algoritmo se basa en una asignación cíclica (*round-robin*) de las longitudes de onda de salida desde λ_0 a λ_{n-1}, para paquetes destinados a la misma fibra de salida. Asimismo, el puntero *round-robin* es reiniciado cuando el sistema compuesto por las *n* colas esté vacío. De este modo, a un paquete se le asigna un retardo *d* y una longitud de onda *w* si y sólo si las colas *0..w-1* tienen *d* $(0\leq d\leq M-1)$ paquetes almacenados, y las colas *w..n-1* tienen *max(0,d-1)* paquetes almacenados. Bajo estas consideraciones, un paquete siempre sufre el mínimo retardo disponible. El *throughput* también se maximiza, ya que un paquete es descartado únicamente cuando las *n* colas correspondientes a las *n* longitudes de onda de su fibra de salida están ocupadas, y *Mn* paquetes están almacenados en el búfer. Además, se preserva el orden de los paquetes en los términos descritos en esta sección: los puertos de entrada son examinados secuencialmente, y las longitudes de onda más bajas son asignadas primero.

```
/* N = nº de fibras entrada/salida */
/* n = nº de long. de onda por fibra */
/* M = nº de retardos */

for input i = 0 to nN-1 do
   if (paquete p en entrada i) then
     f = fibra salida p ( = opp (p) )
     if (retardo [f] < M) then
        asociar retardo [f] a p
        asociar long. onda λ [f] a p
        /* λ [f] es un puntero RR */
        λ [f] ++
        if (λ [f] == n)
          λ [f] = 0
          retardo [f] ++
        endif
     endif
   endif
endfor

/* decrementar retardo[f], f=0..N-1
tras cada ranura temporal */

for fibra salida i=0 to N-1 do
  retardo [f] = max (0,retardo [f]-1)
  if (retardo [f] == 0)
     λ [f] = 0 /* reset puntero RR */
  endif
endfor
```

Fig. 3. Pseudocodigo del algoritmo de selección de longitud de onda presentado.

4 Modelo Analítico

En esta sección se presenta la evaluación del conmutador KEOPS bajo los modos de operación SHWP y SCWP. La comparación se basa en análisis por teoría de colas, validado a través de simulaciones.

El conmutador bajo estudio se asume como simétrico, con N fibras de entrada y de salida, n longitudes de onda $\lambda_0,..,\lambda_{n-1}$ por fibra. El análisis se realiza asumiendo tráfico de entrada Bernouilli uniformemente distribuido, de parámetro ρ_{in}.

El proceso de selección de retardo para la conmutación SHWP ofrece el tradicional comportamiento FIFO con colas a la salida de tamaño M. La evaluación de este modelo se basa en el estudio de una cola de salida fijada, alimentada por la agregación de Nn fuentes Bernouilli de carga ρ_{in}/N. Bajo estas consideraciones, la evaluación se elabora mediante el análisis tradicional de conmutadores con colas a la salida, que no será reproducido en este artículo (ver [9] para más detalles).

El proceso de control para los conmutadores SCWP, aplicando el algoritmo descrito en la sección 2, se basa en un distribuidor *round-robin* de los paquetes destinados a la misma fibra de salida. Esto provoca un reparto igualitario del tráfico entre las n colas de salida, de tamaño M (figura 4-a). El orden fijo de llenado designado por el algoritmo, y la operación con paquetes de tamaño fijo, permite el establecimiento de una identificación biunívoca entre la posición M_b, $0 \le b \le M-1$ en la cola de longitud de onda de salida λ_w, $0 \le w \le n-1$, y la posición de búfer M_{bn+w} de una cola equivalente con n servidores y Mn posiciones de memoria (ver figura 4-a,b). En los siguientes párrafos se expondrá el análisis desarrollado para la evaluación de esta cola multiservidor, alimentada por la agregación de nN fuentes independientes Bernouilli de carga ofrecida ρ_{in}/N.

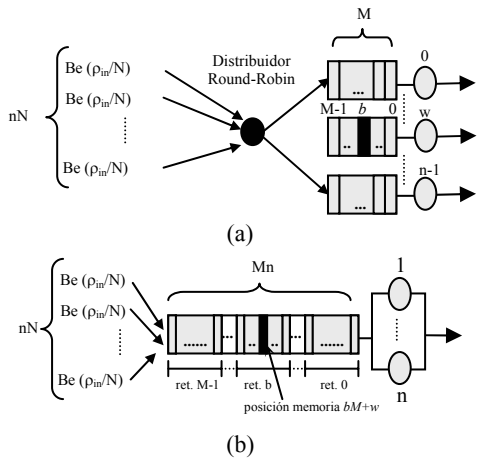


Fig. 4. (a) Modelo de colas para un conmutador $nNxnN$ para tráfico uniforme $Be(\rho_{in})$, (b) modelo de colas equivalente.

Inicialmente, se define la variable aleatoria A como el número de llegadas a la fibra de salida bajo estudio en una ranura temporal, obteniendo para $k=0,1,..,nN$

$$a_k = \Pr[A = k] = \binom{nN}{k} \left(\frac{\rho_{in}}{N}\right)^k \left(1 - \frac{\rho_{in}}{N}\right)^{nN-k} \quad (1)$$

que para $N = \infty$, $k=0,1,..\infty$ se convierte en

$$a_k = Pr[A = k] = \frac{\left(\frac{\rho_{in}}{n}\right)^k e^{-\frac{\rho_{in}}{n}}}{k!} \quad (2)$$

Denotando Q_m como el número de paquetes en la cola al final de la ranura temporal m, y A_m como el número de llegadas durante la ranura temporal m, se obtiene

$$Q_{m+1} = min\{max\{0, Q_m - n\} + A_m, n \cdot M\} \quad (3)$$

Q_m se modela como una cadena de Markov finita con probabilidades de transición

$P_{i,j} = \Pr[Q_{m+1} = j \,|\, Q_m = i]$ dadas por (4), cuando $M \geq N$.

$$P_{i,j} = \begin{cases} a_j & si\ i \leq n,\ j \leq nN \\ a_{j-i+n} & si\ n+1 \leq i \leq nM, \\ & i-n \leq j \leq min(nM-1, nN+i-n) \\ \sum_{s=nM+n-i}^{nN} a_s & si\ n(M-N+1) \leq i \leq nM, \\ & j = nM \end{cases} \quad (4)$$

Para $N \geq M$ tenemos

$$P_{i,j} = \begin{cases} a_j & si\ i \leq n,\ j < nM \\ \sum_{s=nM}^{nN} a_s & si\ i \leq n,\ j = nM \\ a_{j-i+n} & si\ n+1 \leq i \leq nM,\ i-n \leq j < nM \\ \sum_{s=nM+n-i}^{nN} a_s & si\ n+1 \leq i \leq nM,\ j = nM \end{cases} \quad (5)$$

Las probabilidades de estado en estado estacionario q_i, $i=0..nM$ pueden ser, por tanto, obtenidas directamente de las ecuaciones de balance de la cadena de Markov [10]. El *throughput* del sistema ρ_{out} se calcula observando el número de paquetes transmitidos durante una ranura temporal,

$$\rho_{out} = \sum_{s=1}^{nM} q_s \cdot min(s,n) \quad < n \quad (6)$$

La probabilidad de pérdida de paquete se calcula en función de los paquetes ofrecidos a la cola y los servidos por la misma (6).

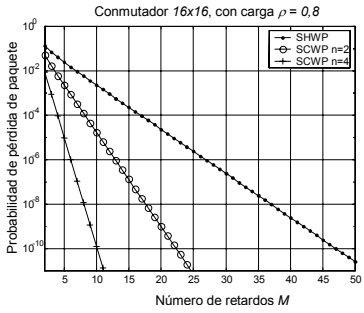
$$\Pr[\,packet\ loss\,] = 1 - \frac{\rho_{out}}{n\rho_{in}} \quad (7)$$

Finalmente, el tiempo medio de espera en cola $\overline{W}$ para un paquete puede ser calculado aplicando la ley de Little.

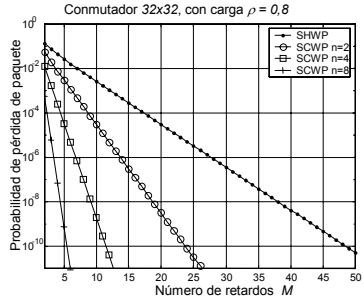
$$\overline{W} = \frac{\overline{Q}}{\rho_{out}} = \frac{\sum_{s=1}^{nM} s \cdot q_s}{\sum_{s=1}^{nM} min(s,n) \cdot q_s} \quad (8)$$

5 Evaluación de las arquitecturas

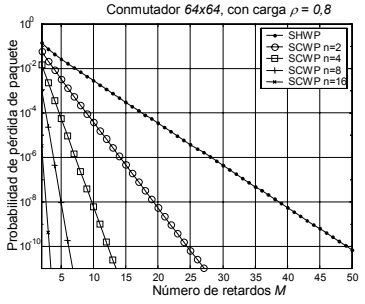
En esta sección, ambos procedimientos de análisis SHWP y SCWP serán empleados en la evaluación del conmutador KEOPS visto en la sección 2. La evaluación se centrará en el caso de conmutador simétrico, con N fibras de entrada y salida, n longitudes de onda por fibra y M líneas de retardo. En un primer paso, la influencia del modo de operación se ha estudiado en conmutadores de tamaño *16x16*, *32x32* y *64x64*, donde el número de puertos viene determinado por el parámetro *nN*. La figura 5-a,b,c, muestra la probabilidad de pérdida de paquete para el conmutador SHWP y SCWP en función del número de retardos *M,* para diferentes valores de *n,* y carga ofrecida $\rho_{in} = 0{,}8$. La conmutación SHWP no obtiene ningún aprovechamiento del número de longitudes de onda *n*, por lo que las prestaciones de un conmutador de este tipo son independientes de este parámetro, para un tamaño de conmutador fijo. Por otro lado, se observa un fuerte impacto del parámetro *n* en las gráficas para el conmutador KEOPS SCWP, que claramente mejora a la versión SHWP. Como ejemplo, bajo el modo de operación SHWP, en un conmutador *16x16* con *4* fibras y *4* longitudes de onda por fibra, son necesarios *42* retardos para obtener una probabilidad de pérdida de paquete $<10^{-1}$, mientras sólo se requieren *11* en la conmutación SCWP. Asimismo, la figura 5 muestra un leve incremento de la probabilidad de pérdida a medida que el tamaño del conmutador aumenta (en términos de número de fibras de entrada/salida). Para ambas versiones del conmutador, la curva para el valor $N=\infty$ puede considerarse una aproximación pesimista precisa cuando $nN>32$.



(a)



(b)



(c)

Fig. 5**.** Probabilidad de pérdida de paquete respecto a número de retardos, para un conmutador OPS *nNxnN* SHWP y SCWP con emulación de colas a la salida, carga de $\rho_{in} = 0{,}8$, $n = \{2,4,8,16\}$, y tamaños de conmutador *nN (a) 16x16, (b) 32x32, (c) 64x64.*

En la figura 6 se muestra la comparación del impacto en el retardo de los modos de operación SHWP y SCWP, normalizado en número de ranuras temporales, asumiendo un tamaño infinito del conmutador, y una probabilidad de pérdida de paquete despreciable. De nuevo, se observa una fuerte mejora en el conmutador SCWP respecto a la versión SHWP. Esta mejora se hace más evidente para mayores valores de *n*, incluso en carga altas.

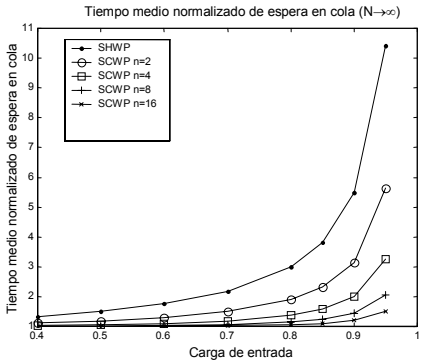


Fig. 6**.** Tiempo medio normalizado de espera en cola para distintas cargas de entrada, para un conmutador OPS *nNxnN* SHWP y SCWP con emulación de colas a la salida, en función del número de longitudes de onda por fibra *n* (asumiendo $N \rightarrow \infty$).

En la tabla I se calcula el número de retardos *M* para diferentes tamaños de conmutador, considerando un probabilidad de pérdida de paquete $< 10^{-9}$ con carga de entrada *0,8*. Como ejemplo, los valores obtenidos muestran un *50%* de ahorro en número de retardos en el conmutador SCWP, para un número de longitudes de onda por fibra igual a *2*. Valores mayores de *n*, conllevan requisitos de búfer decrecientes, que confirman los beneficios de este modo de operación. Por ejemplo, es necesario un tamaño de búfer de sólo *5* posiciones en un conmutador *32x32* con *8* longitudes de onda por fibra. La última columna muestra el impacto del parámetro *n*, en conmutadores de gran tamaño ($N \rightarrow \infty$), que sirven como una aproximación pesimista precisa.

La tabla I muestra también los cálculos en el número de puertas ópticas requeridas en la construcción del conmutador KEOPS que tiene asociada cada una de las alternativas, como medida de su complejidad *hardware*. Debido al presente estado de inmadurez de la tecnología de dispositivos fotónicos, el número de puertas ópticas necesario para la mayoría de alternativas está por encima de la viabilidad real con los actuales procesos de fabricación e integración [11]. Sin embargo, suponen una buena referencia como muestra del efecto del modo de operación SCWP en la simplificación de los requisitos *hardware*. Los datos en la tabla I manifiestan que una reducción del número de retardos necesario en la arquitectura KEOPS SCWP afecta fuertemente en el número de puertas ópticas necesarias. Asumiendo un tamaño constante de conmutador (*nN*), estos valores muestran un crecimiento lineal con el factor *M*. Por esta razón, el fuerte (y no lineal) decrecimiento observado en los requisitos de almacenamiento (*M*) en el modo de operación SCWP, es trasladado linealmente a la simplificación del número de componentes. Este ahorro se hace más evidente en conmutadores pequeños, cuando el sumando *MnN* (que decrece) es más apreciable frente al sumando n^2N^2. Como ejemplo, se consigue una reducción a menos del *35%* en el número de puertas ópticas con la alternativa SCWP para el valor *n=8, nN=16*.

6 Conclusiones y líneas futuras

El algoritmo de selección presentado en este artículo aprovecha de manera óptima la multiplexación estadística alcanzable con el modo de operación SCWP, cuando éste se aplica a conmutadores OPS con capacidad de emulación de colas a la salida. El algoritmo en cuestión es una simplificación del propuesto en [6], y ha sido diseñado para preservar el orden extremo a extremo de los paquetes en la red OPS SCWP. El análisis de prestaciones presentado también en este artículo permite estimar las ventajas en cuanto a retardo medio y reducción de requisitos de almacenamiento que pueden ser alcanzados. Como ejemplo de aplicación, se evalúa la reducción en equipamiento *hardware* que esta disminución conlleva para una adaptación realizada de la arquitectura OPS KEOPS. Este proceso de normalización es ineludible para aquellas arquitecturas diseñadas sin tener en cuenta los modos de operación SHWP/SCWP. En este sentido, es una línea de trabajo actual en nuestro grupo de investigación el realizar una comparativa de distintas arquitecturas OPS con capacidad de emular conmutación con colas a la salida, bajo el algoritmo de selección aquí propuesto, así como realizar la adaptación y evaluación de otras arquitecturas OPS sin esta capacidad en el marco SHWP/SCWP [12].

TABLA I
NÚMERO DE RETARDOS NECESARIOS, Y NÚMERO DE PUERTAS ÓPTICAS NECESARIAS, PARA UN CONMUTADOR KEOPS SHWP/SCWP, CON TRÁFICO DE ENTRADA BERNOUILLI Y CARGA 0,8, PARA TENER UNA PROBABILIDAD DE PÉRDIDA DE PAQUETE $<10^{-9}$.

		16x16		*32x32*		*64x64*		*N→∞*
	n	***M***	**P.Ópt.**	***M***	**P. Ópt.**	***M***	**P. Ópt.**	***M***
SHWP	**---**	42	928	44	2432	44	6912	45
SCWP	**2**	19	560	22	1728	22	5504	23
SCWP	**4**	10	416	11	1376	11	4800	12
SCWP	**8**	4	320	6	1216	6	4480	7
SCWP	**16**	***	***	3	1120	3	4288	4
SCWP	**32**	***	***	***	***	2	4224	3
SCWP	**64**	***	***	***	***	***	***	2

Agradecimientos

Este trabajo se enmarca dentro de los proyectos CICYT FAR-IP (TIC 2000-1734-C03-03) y MTCES (TIC 2001-3339-C02-02).

Referencias

[1] Murthy C., Gurusamy M., "WDM optical networks. Concepts, design and algorithms". Prentice Hall PTR, 2002.

[2] Yao S., Xue F., Mukherjee B., Yoo B., Dixit S., "Electrical ingress buffering and traffic aggregation for optical packet switching and their effect on TCP-level performance in optical mesh networks", IEEE Communications Magazine, vol. 40, no. 9, Sep. 2002, pp. 66-72.

[3] Murata M., Kitayama K., "A perspective on photonic multiprotocol label switching", *IEEE Network*, vol. 15, no. 4, July/August 2001, pp. 56-63.

[4] Hunter D., Nizam M., Chia M., Andonovic I., Guild K., Tzanakaki A., O'Mahony J., Bainbridge J., Stephens M., Penty R., White I., "WASPNET: A Wavelength Switched Packet Network", *IEEE Communications Magazine,* vol. 37, no. 3, March 1999, pp. 120-129.

[5] Chia M., Hunter D., Andonovic I., Ball P., Wright I., Ferguson S., Guild K., O'Mahony M., "Packet loss and delay performance of feedback and feed-forward arrayed-waveguide gratings-based optical packet switches with WDM inputs-outputs", *IEEE Journal of. Lightwave Technology*, vol. 19, no. 9, Sept. 2001, pp. 1241-1254.

[6] Nizam M.H.M., Hunter D.K., Andonovic I., "Designing an optimum WDM transport network: control architectures, node requirements and performance", *Proc. Soc. Photo-Optical Instrumentation Engineers (SPIE),* vol. 3531, Oct. 1998, pp. 244-255.

[7] Guillemot C., *et al.*, "Transparent optical packet switching: the European ACTS KEOPS project approach", *IEEE Journal of Lightwave Technology*, vol. 16, no. 12, Dec. 1998, pp. 2117-2134.

[8] Jacob J. B., *et al.*, "System Design and Evaluation of a Large Modular Photonic ATM Switch", *European Trans. on Telecomm.,* vol. 7, no. 6, Nov.-Dec. 1996, pp. 565-573.

[9] Hluchyj M., Karol M., "Queueing in high-performance packet switching", *IEEE Journal on Selected Areas in Communications.,* vol. 6, no. 9, Dec. 1988, pp. 1587-1597.

[10] Kleinrock L., "Queueing systems. Volume I: Theory", John Wiley & Sons, 1975.

[11] Pavon-Marino P., Garcia-Haro J., Malgosa-Sanahuja J., "Scaling strategies survey for envisaged backbone optical packet switches", Proc. of IASTED Comm. Systems and Networks (CSN 2002), Malaga, España, Sep. 2002, pp. 178-183.

[12] Pavon-Marino P., Garcia-Haro J., Malgosa-Sanahuja J., Cerdan F., "Maximal Matching Characterization of Optical Packet Input-Buffered Wavelength Routed Switches", aceptado para publicación en IEEE Workshop on High Performance Switching and Routing (HPSR 2003), Turín, Italia, Junio. 2003

Validación de mecanismos de vigilancia ante las nuevas necesidades en las redes de comunicaciones

F. D. Trujillo[*], A. J. Yuste[**], E. Casilari[*], A. Díaz Estrella[*] y F. Sandoval[*]

[*] Dpto. Tecnología Electrónica. E. T. S. I. Telecomunicación
Universidad de Málaga, Campus Universitario de Teatinos, s/n. 29071 Málaga
Teléfono: 952 13 14 24 Fax: 952 13 27 33
E-mail: trujillo@dte.uma.es

[**] Dpto. de Electrónica. E. U. P. de Linares
Universidad de Jaén, C/Alfonso X El Sabio, 28. 23700 Linares (Jaén)
Teléfono: 953 02 65 43 Fax: 953 02 65 08
E-mail: ajyuste@ujaen.es

Abstract. *Nowadays, Internet world is more and more used. The new broadband applications over Internet, like videoconference, require a certain Quality of Service (QoS). A methodology to achieve this needed QoS consists of implementing IP over ATM. In this case, an user can obtain this type of services by means of the Asymmetrical Digital Subscriber Line, ADSL. Moreover, two importants elements of the control in networks are the traffic shaping and the policing mechanism. The shaper enables the user to offer the network a traffic that can be easily policed; and the policer takes the necessary actions to enforce the complaince of a connection to a negotiated traffic contract. In this paper, we present an architecture that combines a shaper and a policer to control the traffic coming from a real user in an IP-ATM environment.*

1 Introducción

Con motivo de la gran demanda, que actualmente existe, de servicios multimedia de telecomunicaciones (voz, vídeo y datos), un usuario necesita disponer, en principio, de un gran número de redes de comunicaciones que den soporte y cabida a dichos servicios. Considerando este problema, surge la idea de utilizar, exclusivamente, una única infraestructura que integre todos los servicios actuales y que, además, sea lo suficientemente flexible, versátil y rápida para poder soportar las nuevas aplicaciones de comunicaciones que puedan aparecer en un futuro. De esta forma nació la Red Digital de Servicios Integrados de Banda Ancha (RDSI-BA), asentada sobre el Modo de Transferencia Asíncrono (ATM, *Asynchronous Transfer Mode*) [1], aprobado por la ITU-T (*International Telecommunication Union, Standardization Sector*), con el fin de satisfacer las demandas de velocidades elevadas, permitir un aprovechamiento óptimo de los recursos disponibles en la red mediante la multiplexación estadística y ofrecer una determinada calidad en el servicio solicitado (QoS, *Quality of Service*) por los usuarios.

De esta forma, ATM se ha convertido en el estándar para transporte, conmutación y multiplexación [1] en la RDSI-BA. Las características básicas que definen a ATM pueden resumirse en tres [2]: conmutación rápida de pequeños paquetes de longitud fija (células), servicios orientados a conexión (definición de canales y trayectos virtuales) y uso de técnicas de multiplexación estadística para mejorar la eficiencia en el compartimiento de recursos (ancho de banda, enlaces y colas) por parte de los usuarios. Gracias a esta multiplexación estadística, es posible conseguir una mejor utilización de los enlaces y de las colas; pero este mejor aprovechamiento de recursos tiene un coste traducido en la probabilidad de pérdida de células (CLR, *Cell Loss Rate*). La utilización de colas puede reducir esta CLR pero no eliminarla. Así que la única posibilidad es: o bien descartar información, con lo que los controles de flujo puede tener problemas debido a la alta velocidad de los enlaces en ATM, o bien limitar el número de conexiones establecidas mediante controles preventivos.

Por otra parte, también se debe indicar que Internet [3] ha desplazado hoy en día a las tradicionales redes de datos y ha llegado a convertirse en el modelo de red pública por excelencia. Uno de los factores de éxito se debe a la extensión de los protocolos TCP/IP (*Transfer Control Protocol/Internet Protocol*) como estándares para todo tipo de servicios y aplicaciones.

Hasta ahora, la importancia de Internet provenía más por el servicio de acceso y distribución de contenidos que por el servicio de transporte de datos, conocido como *best-effort*. Sin embargo, la rápida transformación de Internet en una infraestructura comercial ha provocado que la demanda de servicios de calidad como videoconferencia, voz sobre IP (VoIP), etc., se haya extendido vertiginosamente. Este tipo de servicios necesitan, igualmente, una garantía en cuanto al ancho de banda o al retardo máximo, aspectos que para poder ser proporcionados

será necesario dotar a IP de mecanismos de QoS que mejoren su servicio de transporte de datos [4]. De esta forma, el desarrollo y pruebas de nuevas aplicaciones con QoS requiere de nodos encaminadores que soporten los mecanismos de QoS en las que se basan estas aplicaciones.

Independientemente de la infraestructura utilizada, una vez que se ha establecido la conexión, es necesario proteger a la red de desviaciones de los parámetros de tráfico y QoS negociados que puedan afectar adversamente a la QoS de otras conexiones establecidas. Para llevar a cabo este cometido, se establece un control de vigilancia denominado UPC (*Usage Parameter Control*) [1].

En esta artículo se va a abordar la implementación del protocolo IP sobre ATM y su posterior vigilancia. Para ello en el siguiente apartado se explica detalladamente los procedimientos necesarios para la correcta transmisión de tráfico IP sobre la capa ATM. A continuación, en el apartado 3, se explican los aspectos relacionados con los mecanismos de vigilancia; para detallar, en la sección 4, el escenario de tráfico sobre el que se realizan las simulaciones y presentar los resultados de las mismas. Por último se extraen algunas conclusiones de los resultados obtenidos y se proponen algunas futuras líneas de trabajo.

2 IP sobre ATM

En la sección anterior, se ha comentado el extendido uso, cada vez más amplio de las tecnologías basadas en arquitecturas TCP/IP; además hay que tener en cuenta que la tecnología empleada en la RDSI-BA es como también se ha indicado, ATM. Por tanto, es importante conocer la forma de transmitir el protocolo IP sobre ATM. La tecnología ATM dispone de distintas capas de adaptación (AAL, *ATM Adaptation Layer*), para las aplicaciones que no soporten ATM de forma nativa. Estas capas, junto con su uso se pueden ver en la tabla 1 [5].

Tabla 1. Capas de adaptación al medio en ATM

Capa	Uso
AAL 1	Voz Emulación de circuitos
AAL 2	Voz y vídeo VBR
AAL ¾	Datos
AAL 5	LANE IP sobre ATM

Existen dos modelos para el transporte de datos sobre redes ATM [6]:

- La forma conocida como LANE (*LAN Emulation*) para redes de área local y estandarizada por el ATM Forum.

- Y la forma denominada multiprotocolos sobre ATM (MPOA, *Multiprotocol Over ATM*) para enlaces punto a punto, utilizando la capa AAL5. En este artículo, se va a considerar y estudiar el tráfico generado por esta forma de transmisión de paquetes IP sobre ATM.

Los servicios que se pueden solicitar a través de Internet, son de muy diversa índole como el correo (SMTP (*Simple Mail Transfer Protocol*), POP3 (*Post Office Protocol 3*)), la transferencia de archivos (FTP (*File Transfer Protocol*)), la mensajería instantánea (Messenger, IRC (*Internet Relay Chat*)) y, por supuesto, world wide web (HTTP (*HyperText Transfer Protocol*)). Todos estos servicios se deben transmitir por el enlace ATM, y es por ello que la fuente de tráfico que se va a estudiar y caracterizar se puede considerar como una fuente heterogénea [7], que incluye voz, datos de diversa índole, transmisión de vídeo, etc.

La topología empleada para la transmisión de IP sobre ATM, se puede apreciar en la figura 1.

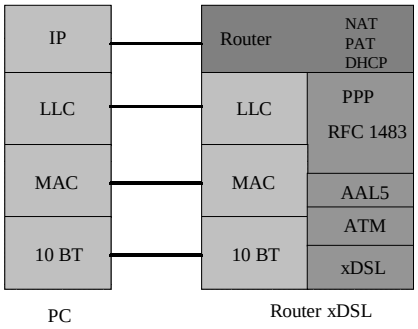


Figura 1. Torre de protocolos

La primera de las torres, que se muestra en la figura 1, es la correspondiente a la red de área local del abonado, así como al *router* de acceso a ATM. Actualmente, la forma más sencilla y económica para acceder a estos servicios, con un ancho de banda aceptable es utilizar las, llamadas, líneas de subscripción de abonado (DSL, *Digital Subscriber Line*). Concretamente, la vertiente asimétrica de DSL (ADSL, *Asymmetrical Digital Subscriber Line*) es la más adecuada para que el usuario acceda a servicios de banda ancha desde su domicilio. La ventaja de las técnicas DSL es que permiten utilizar el cableado existente hasta el domicilio del abonado para poder ofrecer servicios distintos a los de voz. El *router* de acceso a ATM se conecta a un *router* ATM del operador contratado y éste, a su vez, se une al proveedor de acceso a Internet del usuario, a través de una línea ATM o de tecnologías SDH (*Synchronous Digital Hierarchy*) o PDH (*Plesiochronous Digital Hierarchy)*.

Como se puede observar en la estructura de protocolos de la figura 1, los datos de usuario IP se transforman en células ATM, mediante la capa AAL5, encapsulando los datos a partir de PPP (*Point to Point Protocol*) [8] [9]. En [8] se establece la forma de encapsular datos sobre la capa de adaptación al medio AAL5, de dos formas distintas; esto es, multiplexando varios protocolos sobre un

canal virtual; o bien, asumiendo que cada protocolo viaja sobre un canal virtual propio. En [9] se establece la forma de transmitir múltiples protocolos sobre un enlace punto a punto de dos formas posibles y basándose en [8]:

- Multiplexado por canales virtuales: los paquetes se envían directamente a través de AAL5. Los distintos protocolos viajan por un canal virtual distinto.

- Encapsulado LLC (*Logic Link Control*): en este caso a cada paquete se le añade una nueva cabecera para poder discernir el protocolo de origen y, a continuación, se envía sobre AAL5.

En AAL5 la transmisión de los datos de usuario se divide en dos partes:

- La subcapa de convergencia (CS, *Convergence Sublayer*), la cual admite datos desde las capas superiores y los divide en paquetes de hasta 65535 bytes, añadiéndoles una cola.

- La subcapa de segmentación y ensamblado (SAR, *Segmentation and Reassembly Sublayer*) que divide los datos anteriores para conseguir una células ATM, con 48 bytes de datos y 5 de cabecera.

El proceso de encapsulado del paquete IP hasta la generación de las células ATM se resume en la figura 2.

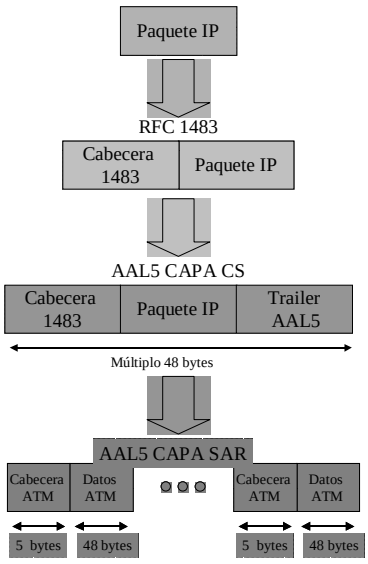


Figura 2. Encapsulado IP sobre ATM

3 Control de vigilancia

El motivo para la existencia de este tipo de mecanismo (UPC, *Usage Parameter Control*) se debe a varios motivos: la necesidad de monitorizar y vigilar el trafico de cada conexión [1] [2] a fin de evitar un uso fraudulento de la red por parte de los usuarios; la obligación de comprobar si el tráfico es conforme con los parámetros negociados en el contrato de tráfico; y la exigencia de proteger, de esta forma, a la red de una excesiva ocupación de recursos por parte de algún usuario, intencionada o involuntariamente, con la consiguiente degradación de la calidad ofrecida al resto de usuarios. La acción que se realiza sobre las conexiones no conformes consiste en el descarte de células de dichas conexiones.

El control de vigilancia va a estar ubicado en el nodo de acceso a la red y, además, debe existir un mecanismo de vigilancia para cada usuario, con lo que es de necesitado cumplimiento el hecho de que este tipo de control resulte económico en lo referente a costes. Asimismo, para un funcionamiento eficiente, los mecanismos de vigilancia deben cumplir los siguiente requisitos [1]:

- Detección de cualquier situación ilegal de tráfico y realización de las pertinentes acciones de control de forma rápida; es decir, actuación en tiempo real.

- Simplicidad en cuanto a implementación hardware y en costes.

- No realización de acciones (transparencia) para conexiones que respetan su contrato de tráfico e intervención para conexiones o usuarios que no se ajustan al contrato de tráfico.

- Rapidez, es decir, tiempo de reacción breve con la idea de evitar posibles degradaciones en la compartición de recursos.

3.1 Parámetros de calidad de funcionamiento

Un método para determinar la relación de células no aceptadas en una conexión con unas determinadas velocidades de pico y media, consiste en calcular el cociente entre el número de células que exceden el contrato de tráfico y el número total de células emitidas. A este cociente se le denomina tasa de pérdidas, cuyo significado es la probabilidad de que un mecanismo de vigilancia detecte una célula como excesiva y la descarte.

$$\text{tasa\_pérdidas} = \frac{\text{Núm. células eliminadas}}{\text{Núm. células emitidas}} \quad (1)$$

Por otra parte, la definición del comportamiento ideal, en cuanto a probabilidad de tasa de pérdidas, de un mecanismo de vigilancia se realiza en base a la siguiente expresión:

$$\text{tasa\_pérdidas} = \begin{cases} \dfrac{\sigma - 1}{\sigma} \text{ si } \sigma \geq 1 \\ 0 \text{ si } \sigma < 1 \end{cases} \quad (2)$$

donde el parámetro σ se define como el cociente entre la velocidad media real (Vmr) y la velocidad media contratada (Vmc), sirviendo las variaciones de este parámetro para medir la eficiencia de los controles de vigilancia.

De esta forma, se dice que la transparencia de un mecanismo de vigilancia consiste en la exactitud con la que éste se aproxima al comportamiento ideal: realizar acciones de control adecuadas en un flujo de células no conforme y evitar acciones de control incorrectas en un flujo de células conformes.

Otro parámetro es el tiempo de reacción o de respuesta, definido como el tiempo necesario para detectar una no conformidad con lo acordado en el contrato de tráfico.

3.2 Parámetros de tráfico a vigilar

Los parámetros de tráfico incluidos en el descriptor de tráfico ATM van a ser los parámetros que se tendrán que controlar. El valor de estos parámetros, junto con la QoS requerida se negocian durante la fase de establecimiento de la conexión.

En [1] se establece que la vigilancia de la velocidad de pico y de la velocidad media son obligatorias. La vigilancia de la velocidad de pico puede llevarse a cabo de forma sencilla y exacta mediante muchos algoritmos [2]. En cambio, la vigilancia de la velocidad media es la que plantea el mayor número de inconvenientes: un usuario puede estar transmitiendo tráfico a gran velocidad (incluso superior a la velocidad de pico) dentro de ráfagas, de forma que hagan superar la velocidad media en un momento determinado; y, sin embargo, este tráfico debe ser aceptado siempre y cuando el usuario no supere la velocidad media negociada a lo largo de la duración de la conexión. Por lo tanto, el problema reside en la dificultad a la hora de estimar la velocidad media con la conexión en curso y la elección de un adecuado periodo de muestreo de forma que no se incurra en un cálculo inexacto de esta velocidad media y no se retarde en demasía la actuación del control de vigilancia. El factor *C* de sobredimensionamiento, que se comenta posteriormente, persigue el objetivo de facilitar la vigilancia de la velocidad media.

3.3 Mecanismos para el control de vigilancia

Hasta el momento, han sido numerosos los mecanismos ideados para llevar a cabo la vigilancia del tráfico que se va a encaminar. De todos ellos, y a partir de estudios anteriores [10], para el caso que nos ocupa, se han elegido los algoritmos más característicos y los que mejores prestaciones han ofrecido. Según esto, los mecanismos de vigilancia empleados son:

- Algoritmo *leaky bucket* (LB): mecanismo basado en el algoritmo genérico de velocidad de célula (GCRA, *Generic Cell Rate Algorithm*) [1]. Consiste en un contador que se incrementa cada vez que llega una célula y se decrementa a velocidad constante.

- Algoritmo *jumping window* (JW): mecanismo basado en técnicas de ventana [11]. Su funcionamiento se basa en un contador que cuenta células en intervalos fijos de tiempo (ventanas) y que limita el número de células aceptadas de una fuente dentro de cada ventana a un número determinado.

- Algoritmo *exponentially weighted moving average* (EWMA): mecanismo también basado en ventanas [11] y, que al igual que el anterior, también usa ventanas consecutivas de tamaño fijo, pero el valor del número de células aceptadas en cada ventana no es invariable, sino que se actualiza dinámicamente mediante una suma exponencial.

- Algoritmos basados en lógica difusa (Catania, [12] y Kandel [13]). En este caso, se definen una serie de reglas difusas y conjuntos difusos que contemplan e identifican las posibles variaciones de los parámetros de tráfico que se van a vigilar.

Los parámetros de los mecanismos de vigilancia que se utilizan para caracterizarlos son: *N*, como el número máximo de células permitidas y *C*, como un factor de sobredimensionamiento necesario para facilitar la vigilancia de la velocidad media y evitar un desmesurado crecimiento en el dimensionamiento de los mecanismos de vigilancia [11].

4 Escenario de tráfico

En la simulaciones se van a utilizar fuentes reales de tráfico. Dichas fuentes se han obtenido a partir de capturas realizadas en la red de Internet RIUJA (Red Informática de la Universidad de Jaén) que se encuentra integrada dentro de otras redes científicas como son RICA (Red Informática Científica de Andalucía) y RedIris. El software utilizado en las capturas de trazas es 'Ethereal' [14], sobre una tarjeta Ethernet 10BaseT. Las trazas son muy variadas, incluyendo todo tipo de aplicaciones y protocolos. Las trazas reales se han dividido en dos conjuntos:

- El tráfico desde el usuario hacia la red

- El tráfico desde la red hacia el usuario

Como quiera que lo que se pretende es vigilar el tráfico de datos desde el usuario hacia la red, en las

pruebas y simulaciones se han utilizado únicamente las trazas que contienen estos datos.

Como se explicó en la sección anterior, los paquetes se convierten en células ATM a través de encapsulado LLC. Posteriormente, y antes de acceder a la red ATM, es necesario un proceso de conformado. La conformación de tráfico [2] trata de regular cada conexión con la finalidad de modelar el flujo de tráfico de dicha conexión para modificar su comportamiento y ajustarlo cuando sea necesario (fuente ilegal) a un patrón más estricto retardando para ello las células que se consideren oportunas.

El conformador empleado ha sido el basado en el algoritmo *leaky bucket* [15]. Para implementar este mecanismo es suficiente con modificar algunos aspectos del mismo mecanismo empleado, a su vez, para el control de vigilancia [16]; basta con variar su filosofía de funcionamiento: el conformador de tráfico debe retrasar células y no eliminarlas (como se explicará más adelante en el apartado de vigilancia) con el objeto de conseguir una reducción en la velocidad de la fuente. Por este motivo, debe incluir un mecanismo de almacenamiento de dichas células para su posterior servicio; es decir, que al mecanismo base utilizado en el control de vigilancia se le añadirá una cola donde se acumularán las células antes de ser servidas.

Como se ha comentado, el algoritmo utilizado para esta conformación de tráfico ha sido el *leaky bucket* (LB), que se ha explicado en el apartado anterior sobre mecanismos para el control de vigilancia. La única diferencia, es la necesidad de añadir una cola amén de modificar su funcionamiento para evitar que elimine células.

A la hora de realizar la conformación del tráfico, se deben elegir una serie de parámetros como velocidad de pico (PCR, *Peak Cell Rate*), velocidad media (SCR, *Sustainable Cell Rate*) y tamaño máximo de la ráfaga (MBS, *Maximum Burst Size*). En las simulaciones realizadas los parámetros usados son los correspondientes a una conexión tipo A para ADSL [17], según se muestra en la tabla 2, en donde la elección de MBS=32 células se debe a que son 32 células las necesarias para la transmisión del paquete IP de mayor longitud, sobre una MAC Ethernet.

Tabla 2. Parámetros de tráfico

Modalidad	PCR (Kbps)	SCR (Kbps)	MBS (células)
A	128	12.8	32

Es decir, los distintos servicios se efectúan según la velocidad binaria estadística (SBR, *Statistical Bit Rate*) de tipo 3 [1], que equivale al tráfico VBR (*Variable Bit Rate*) del ATM Forum [18].

5 Simulación y resultados

Esta sección se divide en tres partes. En la primera de ellas, se procederá a comprobar la validez de los algoritmos de vigilancia, comparando las probabilidades de descarte con las dadas por una vigilancia ideal y definida según la ecuación (2). A continuación, se procede a comprobar el comportamiento dinámico de los algoritmos: a partir de una fuente no conforme, se estudia como evoluciona la tasa de pérdidas en función del tiempo. Por último, se divide la simulación en tres intervalos temporales: en el primer y último intervalo se consideran fuentes no conformes, mientras que en la parte central, habrá una fuente de tráfico conforme; para posteriormente analizar cómo evoluciona el número de células pérdidas en función del tiempo.

Los parámetros que se van a usar para los distintos algoritmos se encuentran en la tabla 3.

Tabla 3. Parámetros de los UPC

Algoritmo	Parámetros
LB	N=32, C=1.1
JW	N=32, C=1.1
EWMA	N=32, C=1.1
Catania	N=32, C=1
Kandel	N=32, C=1

5.1 Vigilancia de fuentes no conformes

En este caso, se procede a conformar la fuente de tráfico con un coeficiente por encima del contratado y a vigilar este tráfico.

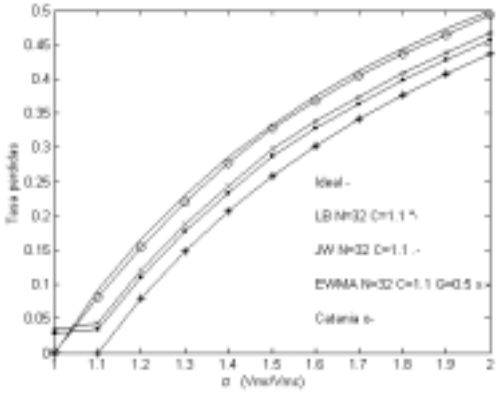


Figura 3. Tasa de pérdidas en función del parámetro sigma

En la figura 3 se puede determinar que el mejor de los algoritmos es el de Catania, seguido del EWMA y del LB. No existen diferencias significativas en cuanto a los algoritmos de Kandel y Catania; de hecho ambos coinciden en cuanto a comportamiento y están próximos al caso ideal.

5.2 Comportamiento temporal

En este caso se parte de una fuente no conforme, en la cual el parámetro σ toma un valor de 1.5 y se

procede a calcular las células eliminadas en función del tiempo para cada uno de los algoritmos.

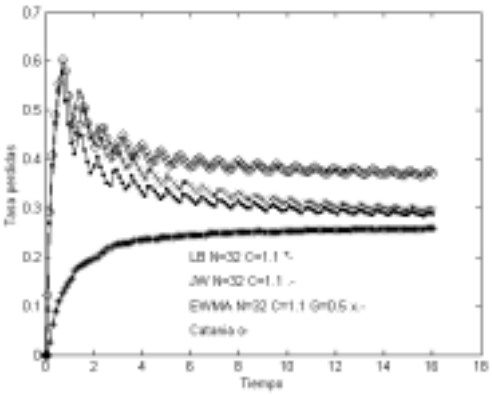


Figura 4. Evolución de las pérdidas en función del tiempo

En la figura 4 se observa que los algoritmos basados en ventana tienen un comportamiento muy similar, en cuanto al tiempo de respuesta (instante de tiempo en el cual empiezan a perder células); sin embargo, y como era de esperar, los algoritmos basados en lógica difusa son los de mejor comportamiento.

Las oscilaciones de los algoritmos basados en ventanas se deben a los cambios de ventana, puesto que, de un intervalo temporal al siguiente, el número de células aceptadas por ventana varía, por lo que el número de células que se eliminan es distinto, produciéndose la oscilación anteriormente comentada.

5.3 Comportamiento dinámico

Para este nuevo estudio, la fuente de tráfico que se estudia se divide en tres partes:

- En la primera parte del estudio, se tiene una fuente no conforme, con un valor de sigma de 1.5.
- En la segunda parte, se considera una fuente conforme con un valor de σ de 0.8
- Por último, una fuente no conforme con σ igual a 2.

Los resultados de la simulación se muestran en la figura 5.

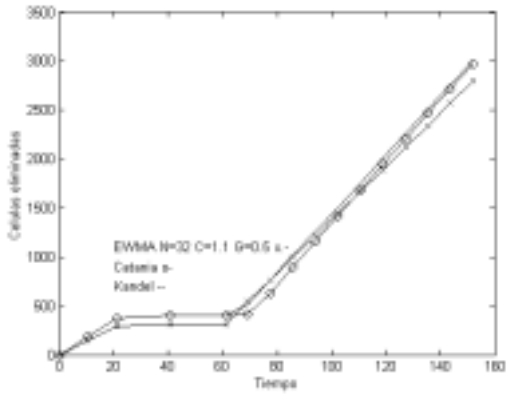


Figura 5. Células eliminadas

En la figura 5 se puede apreciar que en el último intervalo los algoritmos de lógica difusa no tienen un comportamiento eficiente:

- Catania ([12]), debido a su propio funcionamiento: cuando existen intervalos en los que la fuente es conforme, se concede excesivo crédito a este tipo de fuentes, por lo que existe un tiempo de retardo bastante considerable hasta que se vuelven a perder células, al pasar de nuevo a una fuente no conforme.
- Sin embargo, este tiempo es menor para Kandel ([13]). No obstante y al igual que [12], también existe retraso importante a la hora de llegar a una tasa de pérdidas óptima.

Para paliar estos hechos, se ha procedido a realizar una modificación en [12], de tal forma que a través de un contador, se va comprobando si una fuente es conforme o no. El contador se va incrementando, en cada ventana, si una fuente es conforme y se resetea en el caso de que no lo sea. Si este contador sobrepasa un umbral determinado, se restituyen los parámetros de [12] a sus valores iniciales. De esta forma, el crédito que se le asigna a las fuentes en el algoritmo original tiene una limitación temporal en función del umbral. Con esta modificación, denominada Catania modificado, y realizando una nueva simulación se obtiene la figura 6, en la que puede apreciarse que el algoritmo Catania modificado presenta tanto un buen comportamiento dinámico, como un apropiado valor de la tasa de pérdidas, en comparación con el resto de algoritmos.

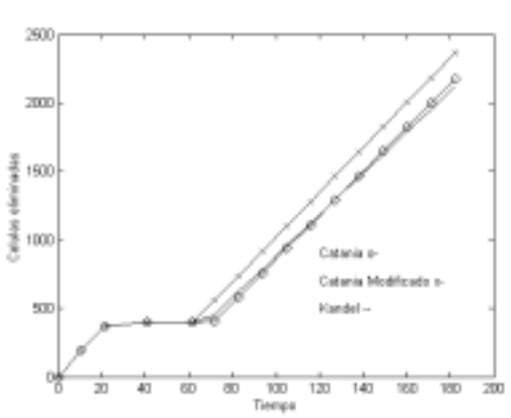


Figura 6. Células eliminadas (Catania modificado)

6 Conclusiones

En la actualidad la convergencia de las redes IP y ATM es una realidad, como queda demostrada con la aparición de ADSL. En este artículo queda probado que los algoritmos de control que se estaban utilizando hasta la fecha en todas la bibliografías, para el control de fuentes *on-off* homogéneas, también son válidos para una fuente tan heterogénea como la determinada por las propias características del tráfico de Internet.

Los problemas detectados, en algunos de los mecanismos empleados, han sido subsanados de forma sencilla y rápida, mediante la incorporación de un contador y la modificación del comportamiento de dichos algoritmos originales.

Se continuarán estudiando el comportamiento de estos algoritmos para distintas configuraciones de tráfico que tengan en cuenta un mayor número de fuentes (varias líneas ADSL). Así mismo, se contemplará la posibilidad de implementar nuevos algoritmos de conformación de tráfico basados en lógica difusa, para estudiar con detenimiento el comportamiento de la cola de dicho conformador y analizar las prestaciones del conjunto formado por conformador y mecanismo de vigilancia.

Agradecimientos

Este proyecto ha sido parcialmente financiado por la Comisión Interministerial de Ciencia y Tecnología (CICYT), proyecto número TEL99-0755.

Referencias

[1] ITU–T. “Recommendation I.371–Traffic control and congestion control in B-ISDN”. Ginebra (Suiza), 1996.

[2] M. Prycker. *Asynchronous transfer mode. Solution for broadband ISDN*. Prentice Hall International, 3ª edición. Estados Unidos (1995).

[3] J. Palet. “Situación Mundial de IPv6”. Mundo Internet. (2000)

[4] X. Xiao, L. Ni. “Internet QoS: A Big Picture”. IEEE Network, pp. 8-18, vol. 13, nº 2. (1999).

[5] W. Stallings. *Comunicaciones y redes de Computadores*. Prentice Hall International, 6ª edición. Madrid (2001).

[6] R. Montañana. “Redes Convergentes ATM-IP: ¿Por qué? ¿Cuándo? ¿Dónde? ¿Cómo?”. Seminario sobre ATM (IIR, Institute for International Research). Madrid (España), 16 marzo 1999.

[7] E. T. Saulnier, K. S. Vastola. “Modeling heterogeneous sources on multiple time scales”. Proceedings on The Conference on Computer Communications. INFOCOM’96. pp. 505-512, vol. 2. San Francisco (Estados Unidos), 24-28 marzo 1996.

[8] J. Heinanen. “Multiprotocol Encapsulation over ATM Adaptation Layer 5”. RFC 1483. IETF. Julio 1993.

[9] G. Gross, M. Kaycee, A. Lin, A. Malis, J. Stephens. “PPP over AAL5”. RFC 2364. IETF. Julio 1998.

[10] C. García Berdonés, F. D. Trujillo, A. Calisti, E. Casilari, A. Díaz Estrella, F. Sandoval. “Funciones UPC para tráfico VBR basadas en técnicas de inteligencia artificial”. Proceedings on Jornadas de Ingeniería Telemática. JITEL’97. pp. 119-128. Bilbao (España), 15-17 septiembre 1997.

[11] E. Rathgeb. “Modeling and performance comparison of policing mechanism for ATM networks”. IEEE Journal on selected areas in communications, pp. 325-334, vol. 9, nº 3. (1991).

[12] V. Catania, G. Ficili, S. Palazzo, D. Panno. “Using fuzzy logic in ATM sources traffic control: lessons and perspectives”. IEEE Communications magazine, pp. 70-81, vol. 34, nº 11. (1996).

[13] A. Kandel, O. Manor, Y. Klein, S. Fluss. “ATM traffic management and congestion control using fuzzy logic”. IEEE Transactions on Systems, Man and Cybernetics–Part C: Applications and Reviews, pp. 474-480, vol. 29, nº 3. (1999).

[14] Software de captura de trazas Ethereal: http://www.ethereal.com

[15] F. D. Trujillo, E. Casilari, A. Díaz Estrella, F. Sandoval. “A proposal of traffic shaping to

improve policing functions in communication networks". Proceedings on the IASTED International Conference on Communication Systems and Networks. CSN'02. pp. 371-375. Málaga (España), 9-12 septiembre 2002.

[16] D. Stiliadis, A. Varma. "A general methodology for designing efficient traffic scheduling and shaping algorithms". Proceedings on The Conference on Computer Communications. INFOCOM'97. pp. 326-335. Kobe (Japón), 7-11 abril 1997.

[17] Boletín Oficial del Estado. Orden Ministerial 26/3/1999. Ministerio de Fomento. 10 abril 1999.

[18] E. Perretti, F. Thepot. "ATM in Europe: The User Handbook." European Market Awareness Committee. ATM Forum. 1997.

Sesión 3A

Redes de Área Local Inalámbricas

Análisis de estabilidad en redes de comunicaciones que usan ALOHA ranurado DS CDMA
Loren Carrasco, Guillem Femenias

Evaluación de técnicas de espectro ensanchado para redes de sensores en canales ópticos no guiados difusos en interiores
Francisco Delgado, Jose A. Rabadán, Santiago T. Pérez, Rafael Pérez-Jiménez

Efecto combinado de técnicas de nivel de enlace independientes en el comportamiento de los protocolos de transporte de Internet sobre Redes de Area Local Inalámbricas
Ramón Agüero, Marta García, Luis Sánchez, Johnny Choque, Luis Muñoz

Propuestas para el despliegue de redes de acceso inalámbricas de bajo coste basadas en tecnología WLAN
Luis Sánchez, Verónica Gutiérrez, Ramón Agüero, Luis Muñoz

Análisis del protocolo IEEE 802.11b en un entorno celular
Mª Elena López-Aguilera, Jordi Casademont, Alfonso Rojas

Análisis experimental del comportamiento de TCP sobre IEEE 802.11b y del protocolo Snoop como mecanismo de mejora
Ramón Agüero, Luis Sánchez, Marta García, Johnny Choque, Luis Muñoz

Análisis de estabilidad en redes de comunicaciones que usan ALOHA ranurado DS CDMA

Loren Carrasco, Guillem Femenias
Universitat de les Illes Balears,
Departament de Ciencies Matemàtiques i Informàtica,
Ctra. Valldemossa km 7.5, 07122 Palma SPAIN,
email: loren@ipc4.uib.es, guillem.femenias@uib.es. *

Resumen

Slotted ALOHA DS CDMA schemes are increasingly used in wireless communications mainly as a component of many reservation protocols. We have investigated the performance of S-ALOHA CDMA systems comparing the DFT(Deferred First Transmission) and IFT(Immediate First Transmission) versions of the system finding that they have very similar characteristics. Moreover we present a stability analysis for both cases. Two stability regions are identified when using fixed retransmission probabilities(P_R), a region where the system is stable or unstable depending on the input load and another where the system is always stable but with bad throughput and delay figures. A dynamically changing ideal P_R scheme, maximizing throughput and delay is investigated.

1. Introducción

Los protocolos ALOHA Ranurados (R-ALOHA) están siendo ampliamente utilizados en redes inalámbricas por ejemplo en los canales de acceso aleatorio [1]. Un derivado del protocolo clásico R-ALOHA que actualmente despierta gran interés es un esquema DS-CDMA R-ALOHA (Direct Sequence Code Division Multiplexing ALOHA Ranurado) dado que es capaz de combinar las propiedades de DS-CDMA (p.e. la multiplexación estadística), y R-ALOHA (p.e. simplicidad y acceso aleatorio) para lograr una mayor eficiencia espectral. De hecho, sistemas móviles de 3ª generación como el UMTS incluyen un canal de acceso aleatorio R-ALOHA DS-CDMA. Además se espera que este canal de acceso trabaje con grandes volúmenes de tráfico dado que también se utiliza para la transmisión de paquetes de datos. En esas condiciones, pueden aparecer problemas de estabilidad que provocarían colapsos en dicho canal afectando a todo el sistema. De hecho cuando el número de usuarios del sistema es finito, este tipo de esquemas presentan problemas de bi-estabilidad. Una red biestable presenta dos estados estables y se sitúa de forma alternada alrededor de esos puntos. Típicamente un estado se corresponde con un throughput elevado y un bajo retardo, mientras que el otro corresponde a un estado de saturación. El comportamiento de la red se caracteriza entonces por periodos con buenas prestaciones intercalados con otros de colapso. Por estas razones, es preciso analizar en detalle la estabilidad de los sistemas R-ALOHA DS-CDMA diseñando estrategias adecuadas para combatir comportamientos no deseados. Diferentes artículos tratan la estabilidad del esquema R-ALOHA clásico controlando la probabilidad de transmisión de los paquetes [7][11][12][13], pero estos métodos no pueden ser directamente aplicados al caso que nos ocupa. Por otra parte, la estabilidad de protocolos R-ALOHA sobre redes de espectro ensanchado por salto de frecuencia se han investigado en [3] y [4]. De hecho utilizaremos un método similar al usado en [3] adaptado para sistemas DS-CDMA. Además para caracterizar el sistema nos basaremos en el modelo incluido en [8] y [10] lo que nos permitirá evaluar sus prestaciones. Así este artículo se organizará de la forma siguiente: la sección 2 incluye una descripción general del sistema, a continuación las secciones 3 y 4 presentan un modelo analítico del esquema R-ALOHA DS-CDMA. La sección 5 utiliza dicho modelo para analizar la estabilidad del sistema e identificar los factores que afectan a dicha estabilidad. Finalmente en 6 se presentan las conclusiones de este artículo y las futuras líneas de trabajo.

2. Descripción del sistema

En una red DS-CDMA-R-ALOHA se asigna a cada estación base un conjunto de códigos de ensanchamiento (spreading) cuyas identidades la estación difundirá a todos sus móviles. Además el tiempo se divide en ranuras iguales en las que puede transmitirse exactamente un paquete. Cada vez que un móvil desea enviar un paquete lo ensancha seleccionando aleatoriamente uno de los códigos y lo transmite en la siguiente ranura. Si un conjunto de móviles

*Este ha sido financiado en parte por el Ministerio de Ciencia y Tecnología y FEDER (Fondo Europeo de Desarrollo Regional) dentro del proyecto (TIC2001-0287).

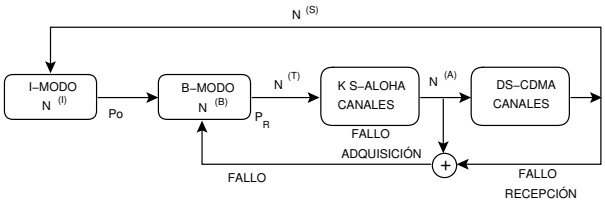


Figura 1: Modelo del sistema R-ALOHA DS-CDMA

transmiten en la misma ranura tendrán éxito si:

- primero, los móviles utilizan diferentes códigos de ensanchamiento y los receptores de la estación son capaces de adquirirlos correctamente,
- y segundo, que esos paquetes correctamente adquiridos no tengan errores (después de pasar por posibles códigos correctores) debidos generalmente al nivel de interferencia inherente a los sistemas DS-CDMA.

Los paquetes que no hayan tenido éxito deberán retransmitirse en una futura ranura.

3. Modelo del sistema

Consideramos la red R-ALOHA DS-CDMA descrita anteriormente. El modelo del canal queda completamente representado por la Fig.1 y los parámetros descritos en la tabla 3. Consideramos un sistema con N terminales activos. Se asume que cada móvil puede estar en uno de los dos modos al inicio de un intervalo de tiempo: modo-I (Vacío/*Idle*, sin un paquete preparado para su transmisión) o modo-B (Espera/*Backlogged*, con un paquete preparado para su transmisión). Un móvil que o bien 1) acaba de entrar en el sistema o 2) termina de conseguir la transmisión exitosa de un paquete se dice que se encuentra en modo-I. Se asume que un móvil en modo-I generará un paquete en el siguiente intervalo de tiempo y pasaría a modo-B con probabilidad P_0. Por otro lado, los móviles en modo-B son aquellos que tienen paquetes esperando para su transmisión porque: 1)estaban en modo-I y han generado un paquete o 2)han tenido una transmisión sin éxito y están esperando una retransmisión. Se asume que un móvil en modo-B transmitirá un paquete en el siguiente intervalo con probabilidad P_R. También asumimos que un terminal en modo-B no genera nuevos paquetes. Se dice que un terminal en modo-B que transmite/retransmite un paquete ha adquirido uno de los K códigos disponibles en el sistema si no hay otros terminales en modo-B con una transmisión en curso que en el mismo intervalo hayan seleccionado el mismo código.

Este modelo es la versión DFT (Primera transmisión diferida) de la versión IFT (Primera transmisión inmediata) descrita en [8]. El objetivo de la modificación consiste en unir los flujos procedentes de terminales en modo-I y en modo-B. Diferentes

Tabla 1: Descripción de los parámetros del sistema

Parámetro	Descripción
N	N° de móviles en el sistema
K	N° de pares receptor-código en la estación de base
$N_K^{(B)}$	N° de móviles en modo-B al inicio de la ranura k
$N_K^{(I)}$	N° de móviles en modo-I al inicio de la ranura k
$N_K^{(T)}$	N° total de móviles transmitiendo paquetes en la ranura k
$N_K^{(N)}$	N° de móviles en modo-I generando un paquete en la ranura k
$N_K^{(A)}$	N° de paquetes adquiridos en la ranura k
$N_K^{(S)}$	N° de paquetes correctamente recibidos en la ranura k
P_0	prob. de que un móvil en modo-I genere un paquete en una ranura
P_R	prob. de que un móvil en modo-B transmita un paquete en una ranura
$P_B(n)$	prob. media de error dadas n transmisiones simultaneas DSCDMA

autores [2] han estudiado las prestaciones de ambas versiones para el esquema R-ALOHA clásico, aquí determinaremos que ocurre en el caso de un sistema R-ALOHA DS-CDMA.

En el sistema DFT descrito anteriormente, como puede observarse en la Fig.1 se cumplen las siguientes relaciones entre los parámetros en un intervalo de tiempo k:

$$N = N_k^{(I)} + N_k^{(B)}, \tag{1}$$

$$N_{k+1}^{(B)} = N_k^{(B)} + N_k^{(N)} - N_k^{(S)}, \tag{2}$$

el valor de $N_k^{(N)}$ depende de $N_k^{(I)}$ y P_0 como

$$Pr\{N_k^{(N)} = \alpha | N_k^{(I)} = \beta\} = \begin{cases} \binom{\beta}{\alpha} P_0^{\alpha}(1-P_0)^{(\beta-\alpha)} & \text{si } \alpha \leq \beta \\ 0 & \text{en otro caso,} \end{cases} \tag{3}$$

y finalmente, $N_k^{(T)}$ es función solamente de $N_k^{(B)}$ y de la probabilidad P_R

$$Pr\{N_k^{(T)} = \alpha | N_k^{(B)} = \beta\} = \begin{cases} \binom{\beta}{\alpha} P_R^{\alpha}(1-P_R)^{(\beta-\alpha)} & \text{si } \alpha \leq \beta \\ 0 & \text{en otro caso.} \end{cases} \tag{4}$$

4. El modelo de cadena de Markov discreta

Al definir el estado del sistema como el número de terminales en espera al inicio del intervalo de tiempo k en [9] se demostró que $N_k^{(B)}$ es una cadena de Markov discreta sobre el espacio de estados $N_k^{(B)} \in \{n| \quad 0 \leq n \leq N\}$. La distribución en equilibrio se caracteriza por la matriz de transición entre estados,

$$P = [p_{ij}]_{(N+1)\times(N+1)}, \tag{5}$$

donde p_{ij} es la probabilidad de que se produzca una transición del estado i al estado j,

$$p_{ij} = Pr\{N_{k+1}^{(B)} = j | N_k^{(B)} = i\}, \quad 0 \leq (i,j) \leq N. \tag{6}$$

La distribución de equilibrio $\pi = [\pi_0, \pi_1, \ldots, \pi_N]$ puede ser determinada resolviendo el conjunto de ecuaciones lineales:

$$\pi = \pi P, \quad \sum_{k=0}^{N} \pi_k = 1. \tag{7}$$

Por tanto, el siguiente paso para describir completamente el sistema será obtener las probabilidades de transición y entonces usando (7) determinar la distribución de equilibrio.

Para obtener p_{ij} utilizamos la condición de partición,

$$\sum_{n=0}^{N} \sum_{s=0}^{\min(K,n)} Pr\{N_k^{(S)} = s, N_k^{(T)} = n | N_k^{(B)} = i\} = 1, \tag{8}$$

entonces, la probabilidad de transición entre estados definida en (6) puede ser expresada como

$$p_{ij} = \sum_{n=0}^{N} \sum_{s=0}^{\min(K,n)} Pr\{N_{k+1}^{(B)} = j, N_k^{(T)} = n, N_k^{(S)} = s | N_k^{(B)} = i\}. \tag{9}$$

Utilizando las relaciones entre parámetros (1) y (2), y dados

$$N_k^{(S)} = s, \quad N_k^{(B)} = i,$$
$$N_{k+1}^{(B)} = j, \quad N_k^{(T)} = n,$$

se obtienen

$$N_k^{(N)} = j - i + s, \tag{10}$$

y

$$N_k^{(I)} = N - i. \tag{11}$$

De esta forma p_{ij} se convierte en

$$\begin{aligned} p_{ij} &= \sum_{n=0}^{N} \sum_{s=0}^{\min(K,n)} Pr\{N_k^{(N)} = j - i + s, N_k^{(T)} = n, N_k^{(S)} = s | N_k^{(B)} = i\} \\ &= \sum_{n=0}^{N} \sum_{s=0}^{\min(K,n)} Pr\{N_k^{(N)} = j - i + s | N_k^{(T)} = n, N_k^{(S)} = s, N_k^{(B)} = i\} \\ &\quad \times Pr\{N_k^{(S)} = s | N_k^{(T)} = n, N_k^{(B)} = i\} \\ &\quad \times Pr\{N_k^{(T)} = n | N_k^{(B)} = i\}. \end{aligned} \tag{12}$$

Ahora considerando que $N_k^{(S)}$ depende solamente de $N_k^{(T)}$ y que $N_k^{(N)}$ solo depende de $N_k^{(I)}$ podemos reformular p_{ij} como

$$\begin{aligned} p_{ij} = \sum_{n=0}^{N} \sum_{s=0}^{\min(K,n)} & Pr\{N_k^{(N)} = j - i + s | N_k^{(I)} = N - i\} \\ & \times Pr\{N_k^{(S)} = s | N_k^{(T)} = n\} \\ & \times Pr\{N_k^{(T)} = n | N_k^{(B)} = i\}. \end{aligned} \tag{13}$$

Usando (3) y (4) y definiendo

$$f_{N^{(S)}|N^{(T)}=n}(s) \equiv Pr\{N_k^{(S)} = s | N_k^{(T)} = n\}, \tag{14}$$

p_{ij} se expresa como,

$$\begin{aligned} p_{ij} = \sum_{n=0}^{N} \sum_{s=0}^{\min(K,n)} & \binom{N-i}{j-i+s} P_0^{j-i+s} (1-P_0)^{(N-j-s)} \\ & \times \binom{i}{n} P_R^n (1-P_R)^{(i-N)} f_{N^{(S)}|N^{(T)}=n}(s), \end{aligned} \tag{15}$$

donde

$$\begin{aligned} & f_{N^{(S)}|N^{(T)}=n}(s) = \\ & \sum_{a=0}^{\min(K,n)} Pr\{N^{(S)} = s | N^{(A)} = a, N^{(T)} = n\} \\ & \quad \times Pr\{N^{(A)} = a | N^{(T)} = n\}. \end{aligned} \tag{16}$$

En [8] se obtuvieron dos expresiones para $Pr\{N^{(S)} = s | N^{(A)} = a, N^{(T)} = n\}$. Utilizaremos la correspondiente al caso en que no hay detección de una adquisición incorrecta y por tanto todos los usuarios que transmiten contribuyen a la interferencia CDMA

$$\begin{aligned} & Pr\{N^{(S)} = s | N^{(A)} = a, N^{(T)} = n\} \\ & = \begin{cases} \binom{a}{s} [1 - P_E(n)]^s P_E(n)^{a-s} & \text{si } 0 \leq s \leq a, \\ 0 & \text{en otro caso,} \end{cases} \end{aligned} \tag{17}$$

Hemos obtenido la siguiente expresión para $Pr\{N^{(A)} = a | N^{(T)} = n\}$,

$$Pr\{N^{(A)} = a | N^{(T)} = n\} = \frac{\binom{K}{a}\binom{n}{a} a! T_{(n-a,K-a)}}{K^n}, \tag{18}$$

donde

$$T_{(x,y)} = y^x - \left[\sum_{i=1}^{\min(x,y)} \binom{y}{i} \binom{x}{i} i! T_{(x-i,y-i)} \right]. \tag{19}$$

A partir de aquí, solo queda determinar la probabilidad de error en los paquetes $P_E(n)$. Asumiendo un control de potencia ideal y utilizando la hipótesis gaussiana para modelar la interferencia causada por otros usuarios, el canal puede modelarse como

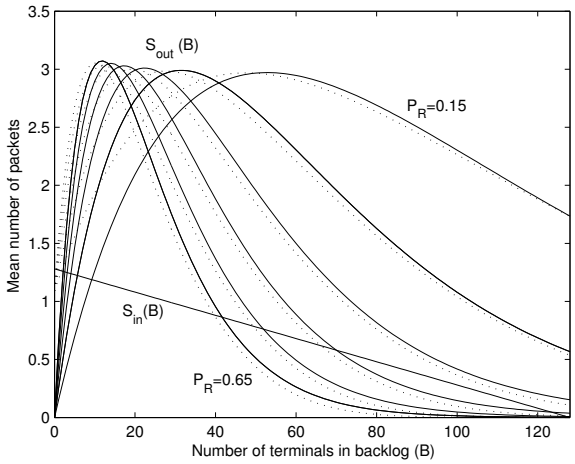


Figura 2: S_{in}para $P_0 = 0{,}01$ y comparación de DFT(líneas continuas) y IFT (líneas de puntos) S_{out} para $N = 128$, $K = 8$, $G_p = 128$, $L = 320$ y $P_R = (0{,}25, 0{,}35, 0{,}45, 0{,}55, 0{,}65)$.

AWGN y la probabilidad de bits erróneos expresarse como

$$P_b(n) = Q\left(\sqrt{\frac{3G_p}{(n-1)}}\right), \tag{20}$$

donde n es el número de paquetes transmitidos simultáneamente y G_p la ganancia de procesado o factor de ensanchamiento del espectro utilizado en sistema CDMA. Si asumimos además que la resistencia a los desvanecimientos de los sistemas DS-CDMA junto con el entrelazado harán que los errores se encuentren dispersos y aislados entre sí dentro de un paquete y que el sistema utiliza un código t corrector entonces la probabilidad de recibir un paquete de L bits con éxito es

$$P_C(n) \leq 1 - \left[\sum_{i=t+1}^{L} \binom{L}{i} P_b(n)^L (1 - P_b(n))^{L-i}\right]. \tag{21}$$

La utilización de la hipótesis gaussiana implica que en este estudio se ha considerado la existencia de un número considerable de usuarios N en el sistema. Si se deseara aplicar este estudio a sistemas con un número de usuarios más reducido seria suficiente con modificar la expresión para el cálculo del BER.

Una vez se ha caracterizado completamente el sistema compararemos las versiones DFT e IFT y analizaremos su estabilidad.

5. Análisis de estabilidad

En esta sección utilizaremos el principio de balance de flujo (*flow balance*)[3] para estudiar la dinámica del sistema R-ALOHA DS-CDMA tanto en el caso IFT como DFT. El principio de balance de flujo se basa en suponer que el sistema se encuentra en el estado B, lo que significa que el sistema se halla en el estado de la cadena de Markov descrita en la sección 4 donde el número de usuarios en espera es $N_k^{(B)} = B$. Supongamos ahora que $S_{in}(B)$ y $S_{out}(B)$ son, respectivamente, el número medio de paquetes netos que fluyen hacia el sistema ($N_k^{(N)}$ medio) y el número medio de paquetes que fluyen fuera del sistema ($N_k^{(S)}$ medio) en una ranura temporal. Si $S_{in}(B) > S_{out}(B)$, entonces el sistema tiende a desplazarse hacia un estado B mayor ($> B$). Si $S_{in}(B) < S_{out}(B)$, entonces el sistema tiende a desplazarse hacia un estado B menor($< B$). Si $S_{in}(B) = S_{out}(B)$, entonces el estado B es un estado de equilibrio. Un estado de equilibrio puede ser tanto estable como inestable. Si el número de estados de equilibrio es uno, se dice que el sistema es estable. En cualquier otro caso se dice que es inestable. Hay que subrayar que cuando el sistema alcanza un punto estable B_e el throughput medio coincide con $S_{out}(B_e)$.

El número medio de paquetes fluyendo hacia el sistema $S_{in}(B)$ es

$$S_{in}(B) = E\{N_k^{(N)} | N_k^{(B)} = B\} \tag{22}$$

Usando la relación (1) y desarrollando el promedio de $N_k^{(N)}$ condicionado al estado $N_k^{(B)} = B$ obtenemos

$$S_{in}(B) = \sum_{i=0}^{N-B} i Pr\{N_k^{(N)} = i | N_k^{(I)} = N - B\}. \tag{23}$$

Finalmente, usando (3) $S_{in}(B)$ puede escribirse como

$$S_{in}(B) = (N - B)P_0, \tag{24}$$

tanto en la versión DFT como IFT.

El número medio de paquetes fluyendo fuera del sistema $S_{out}(B)$ es

$$S_{out}(B) = E\{N_k^{(S)} | N_k^{(B)} = B\}. \tag{25}$$

Usando (8) $S_{out}(B)$ puede expresarse como $S_{out}(B) =$

$$\begin{aligned} &\sum_{n=0}^{N}\sum_{s=0}^{n} s Pr\{N_k^{(T)} = n, N_k^{(S)} = s | N_k^{(B)} = B\} \\ &= \sum_{n=0}^{N}\sum_{s=0}^{n} s Pr\{N_k^{(S)} = s | N_k^{(T)} = n, N_k^{(B)} = B\} \\ &\qquad \times Pr\{N_k^{(T)} = n | N_k^{(B)} = B\}. \end{aligned} \tag{26}$$

A partir del modelo tanto del caso IFT como DFT:

- $Pr\{N_k^{(S)} = s | N_k^{(T)} = n, N_k^{(B)} = B\} = Pr\{N_k^{(S)} = s | N_k^{(T)} = n\}$ y,
- $Pr\{N_k^{(S)} = s | N_k^{(T)} = n, N_k^{(B)} = B\} = 0$, si el número de transmisiones con éxito es superior al número de códigos disponibles o el número de paquetes transmitidos($s >$ mín(K, n)).

entonces, $S_{out}(B)$ puede escribirse de la forma siguiente

$$S_{out}(B) = \sum_{n=0}^{N} \sum_{s=0}^{\min(K,n)} sPr\{N_k^{(T)} = n | N_k^{(B)} = B\} \\ Pr\{N_k^{(S)} = s | N_k^{(T)} = n\}. \quad (27)$$

donde $Pr\{N_k^{(S)} = s|N_k^{(T)} = n\}$ ha sido obtenida previamente para ambos casos. Por otra parte, $Pr\{N_k^{(T)} = n|N_k^{(B)} = B\}$ es muy simple en el caso DFT

$$Pr\{N_k^{(T)} = n|N_k^{(B)} = B\} \\ = \begin{cases} \binom{B}{n} P_R^n (1-P_R)^{(B-n)} & \text{si } n \leq B \\ 0 & \text{en otro caso.} \end{cases} \quad (28)$$

mientras que para el sistema IFT, el término puede expresarse como $Pr\{N_k^{(N)} + N_k^{(R)} = n, |N_k^{(B)} = B\}$. Considerando que el éxito de una primera transmisión es un suceso independiente del éxito en las retransmisiones esta probabilidad puede calcularse como:

$$Pr\{N_k^{(N)} + N_k^{(R)} = n, |N_k^{(B)} = B\} \\ = \sum_{\alpha=0}^{n} Pr\{N_k^{(N)} = \alpha | N_k^{(B)} = B\} \\ \times Pr\{N_k^{(R)} = n - \alpha | N_k^{(B)} = B\} \\ = \sum_{\alpha=0}^{n} Pr\{N_k^{(N)} = \alpha | N_k^{(I)} = N - B\} \\ \times Pr\{N_k^{(R)} = n - \alpha | N_k^{(B)} = B\}, \quad (29)$$

donde,

$$Pr\{N_k^{(N)} = \alpha | N_k^{(I)} = N - B\} \\ = \begin{cases} \binom{N-B}{\alpha} P_0^{\alpha} (1-P_0)^{(N-B-\alpha)} & \text{si } \alpha \leq N - B \\ 0 & \text{en otro caso,} \end{cases} \quad (30)$$

$$Pr\{N_k^{(R)} = n - \alpha | N_k^{(B)} = B\} \\ = \begin{cases} \binom{B}{n-\alpha} P_R^{n-\alpha} (1-P_R)^{(B-n+\alpha)} & \text{si } \alpha \geq n - B \\ 0 & \text{en otro caso} \end{cases} \quad (31)$$

La fig.2 muestra $S_{in}(B)$ para $P_0 = 0{,}01$ y $S_{out}(B)$ tanto del caso IFT com DFT para diversas P_R. Como puede verse ambas versiones son prácticamente equivalentes. Además la figura muestra los cruces entre S_{in} y $S_{out}(B)$ para varias P_R mostrando que el comportamiento del sistema depende de esta probabilidad. De hecho para un sistema dado de N usuarios pueden distinguirse dos regiones:

- un rango de valores de P_R bajos donde el sistema es estable independientemente de P_0 (en la figura las curvas de $P_R = 0{,}15$ y $P_R = 0{,}2$ están en esta zona ya que cualquier pendiente en $S_{in}(B)$ no atraviesa más de una vez la curvas de $S_{out}(B)$),

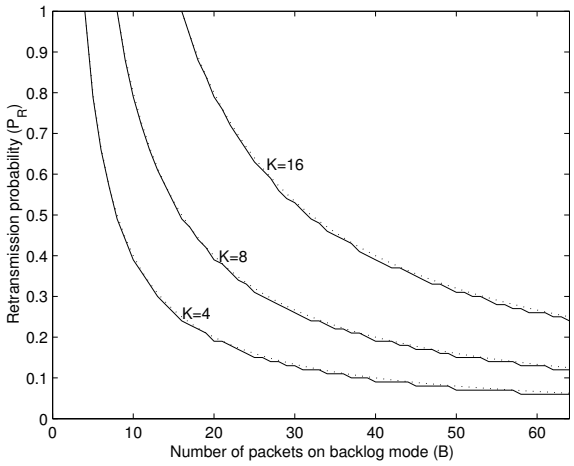


Figura 3: Efecto del número de códigos CDMA (K) sobre la P_R adaptativa para $N = 128$, $G_p = 128$, $L = 511$, $t = 2$. Hay dos curvas para cada valor de K una la obtenida de la simulación i la otra aplicando la expresión 33.

- un margen de valores de P_R donde el sistema es estable o inestable dependiendo de P_0 (en la figura las curvas de $S_{out}(B)$ para los demás valores de P_R ya que pueden dibujarse líneas $S_{in}(B)$ que cruzan más de una vez dichas curvas).

Por consiguiente, la mejor solución parece ser un valor bajo de P_R que conlleva un sistema siempre estable, sin embargo esta elección no optimiza el throughput puesto que los puntos de estabilidad (donde $S_{out}(B)$ representa el throughput) o cruce entre $S_{in}(B)$ y $S_{out}(B)$ se encuentra muy por debajo del máximo $S_{out}(B)$ posible.

Parece interesante encontrar una P_R que asegure la estabilidad y al mismo tiempo maximice el throughput del sistema.

5.1. Obtención de una P_R adaptativa

Para obtener esta P_R "óptima", será necesario calcular la P_R que maximiza (27). Vamos por tanto a derivar (27) con respecto a P_R y encontrar los ceros

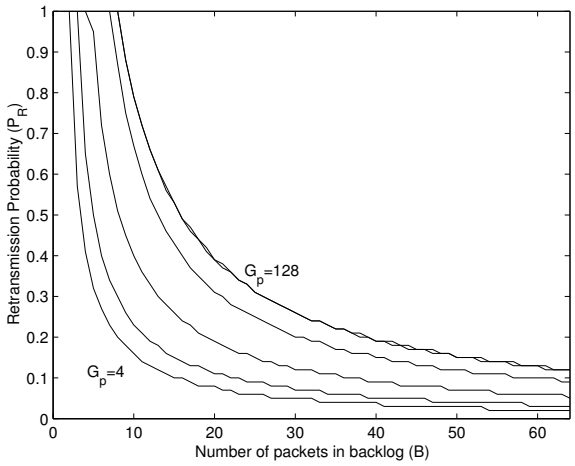


Figura 4: Efecto de la ganancia de procesado (G_p) sobre la P_R adaptativa para $N = 128$, $L = 511$, $t = 2$. Diferentes curvas corresponden a $G_p = 128, 64, 32, 16, 8, 4$.

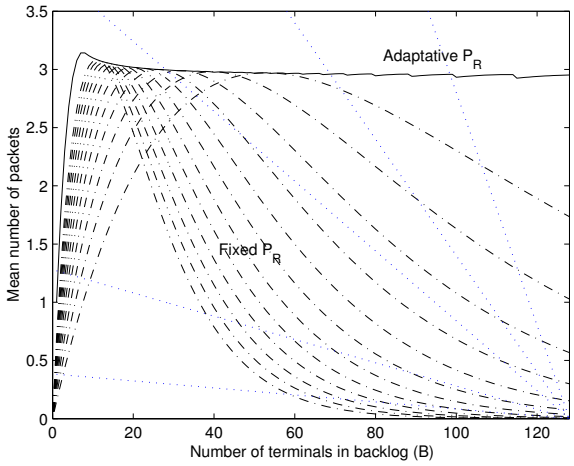


Figura 5: Curvas de S_{out} para diferentes P_R fijas (desde 0.15 a 0.7 con un paso de 0.05) y la P_R adaptativa con $N = 128$, $K = 8$, $L = 320$ y $G_p = 64$.

de la derivada:

$$\frac{\partial S_{out}(B)}{\partial P_R} = \sum_{n=0}^{N} \sum_{s=0}^{\min(K,n)} \frac{\partial Pr\{N_k^{(T)} = n | N_k^{(B)} = B\}}{\partial P_R} Pr\{N_k^{(S)} = s | N_k^{(T)} = n\}$$
$$= \sum_{n=0}^{N} \sum_{s=0}^{\min(K,n)} s[\binom{B}{n} n P_R^{(n-1)} (1-P_R)^{(B-n)} - \binom{B}{n} P_R^n (1-P_R)^{(B-n-1)} (B-n)] \times Pr\{N_k^{(S)} = s | N_k^{(T)} = n\}. \quad (32)$$

Para un sistema con N usuarios la expresión anterior depende del número de códigos K y de la probabilidad de éxito al transmitir un paquete $P_C(n)$ que a su vez depende la la ganancia de procesado G_p, la longitud del paquete L y la capacidad correctora del código corrector utilizado t. Aunque no es posible extraer una expresión cerrada para esta P_R adaptativa puede ser interesante hallar, si es posible, una expresión cerrada que se aproxime razonablemente a la P_R definida en (32).

Las Figs. 3 y 4 muestran la forma de esta P_R adaptativa y el efecto de los más importantes parámetros del sistema sobre ella.

Observando la Fig.3 puede extraerse una expresión muy aproximada para la P_R adaptativa en el caso de valores bajos de $P_C(n)$

$$P_R = \begin{cases} K/B & \text{si } B > K \\ 1 & \text{en otro caso,} \end{cases} \quad (33)$$

indica que la probabilidad "óptima" puede calcularse como la relación entre el número total de códigos disponibles K y el número de paquetes en estado de espera B. En la gráfica 3 la línea de puntos representa esta expresión aproximada, i por tanto puede verse su validez en el caso de tener un canal CDMA sin errores. De hecho, esta fórmula se corresponde a la probabilidad de retransmisión óptima descrita en [14] para estabilizar un esquema R-ALOHA clásico de un solo canal adaptado a un esquema con K códigos. Sin embargo como puede verse en la Fig.4 esta expresión (33) es válida sólo si el canal CDMA no introduce errores en la fase de recepción, esto es, cuando la probabilidad de éxito de transmisión de un paquete, una vez adquirido, se aproxima a 1 (en la fig.4 las curvas con $G_p = 64$ y $G_p = 128$ son prácticamente coincidentes con una curva del tipo K/B). Si el canal introduce un número significativo de errores debido a un nivel de interferencias excesivo (en la fig.4 la curvas con $G_p = 8$ y $G_p = 16$ se encuentran por debajo de la curva K/B) una expresión cerrada aproximada debe tener este factor en consideración. Como puede verse en la Fig. 5, la curva correspondiente a esta P_R variable se sitúa siempre por encima del resto de curvas y por tanto el punto de cruce en este caso entre $S_{in}(B)$ y $S_{out}(B)$ corresponderá siempre a un $S_{out}(B)$ máximo (máximo throughput) para cualquier carga de entrada.

Con respecto a la estabilidad del sistema puede señalarse que, aunque para buenos canales CDMA la P_R adaptativa implica un sistema estable para cualquier carga de entrada, como puede verse en la Fig.5 cualquier línea de $S_{in}(B)$ (es decir cualquier P_0) sólo tiene un punto de cruce con $S_{out}(B)$. Si el canal CDMA empeora puede existir más de un punto de cruce como puede verse en la Fig.6 donde se muestran las curvas de S_{out} para una ganancia de procesado baja así como algunas líneas de S_{in} como ejemplo. Sin embargo en este caso aunque el sistema no es estable todos los puntos de cruce corresponden a valores de throughput máximos y por tanto este sistema permanecerá siempre operativo.

Como ha sido mencionado anteriormente, un mal canal CDMA (con P_C bajos) tiene un efecto signifi-

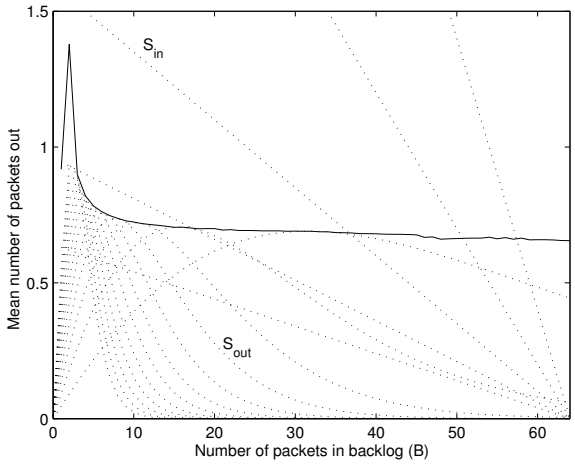


Figura 6: Curvas de S_{out} para diferentes P_R fijas (desde 0.15 a 0.7 con un paso de 0.05) y la P_R adaptativa con $N = 128$, $K = 8$, $L = 320$ y $G_p = 4$.

cativo sobre la P_R óptima (ver Fig. 4), debido a que en este caso el número máximo de paquetes transmitidos simultáneamente con éxito no está limitado por el número de códigos disponible sino por el nivel de interferencia. Este hecho esta representado en las curvas de las figuras 3 y 4 donde queda claro que un G_p bajo o un código corrector peor(parámetros L y t) tiene el mismo efecto que tomar un número inferior de códigos CDMA K. Este resultado sugiere una nueva formulación de (33) incluyendo el efecto del canal CDMA de la forma:

$$P_R = \begin{cases} \frac{\text{mín}(K,N_{ch})}{B} & \text{si } B > \text{mín}(K, N_{ch}) \\ 1 & \text{en otro caso.} \end{cases} \quad (34)$$

donde N_{ch} es el promedio del número de paquetes transmitidos con éxito a través del canal CDMA.

Hasta este momento para realizar este análisis teórico hemos supuesto que el número de terminales en espera B, y el número de N_{ch} es perfectamente conocido por todos los usuarios y evidentemente esta hipótesis no se cumple en un sistema distribuido. Diferentes autores [12][13][7][11] han propuesto métodos que tratan de estimar el número de usuarios en espera B aunque siempre para sistemas R-ALOHA clásicos (de un solo canal). Por tanto como complemento a este trabajo se ha realizado una adaptación de estos esquemas al caso de K códigos (canales) comparando sus prestaciones. Dicho estudio complementario queda fuera del marco del presente artículo.

6. Conclusiones

En esta ponencia hemos investigado las prestaciones de los sistemas CDMA con acceso ALOHA ranurado comparando las versiones DFT e IFT del sistema. Hemos hallado que ambas versiones tienen características muy similares. Además hemos presentado un análisis de estabilidad para ambos casos. Se han identificado dos regiones de estabilidad para el caso de P_R fijas, una región donde el sistema es estable independientemente de la carga de entrada y otra donde el sistema es estable o inestable dependiendo de la carga de entrada. Por consiguiente cuando se utilize un sistema de estas características resulta necesario seleccionar una P_R apropiada que garantice la estabilidad. Desgraciadamente el conjunto de P_R apropiadas es sub-óptimo en términos de throughput y retardo. En consecuencia se propone un esquema óptimo donde P_R cambia dinámicamente evaluándose además los principales factores que afectan a esta P_R adaptativa. Como se ha mencionado en la sección anterior partiendo del análisis teórico realizado en este artículo se ha llevado a cabo una evaluación de posibles métodos de estimación de la P_R óptima en un entorno distribuido. Actualmente se esta trabajando en variaciones del modelo original, como por ejemplo su adaptación en entornos sin control de potencia rápido donde deben tenerse en cuenta fenómenos como el efecto captura, o su utilización en sistemas con un número de usuarios N más reducido donde la hipótesis gaussiana para las interferencias deja de ser válida.

Referencias

[1] I. Akyildiz, J. McNair, L. Carrasco,R. Puigjaner, Y. Yesha "Medium Access Control Protocols for Multimedia Traffic in Wireless Networks ," *IEEE Network Magazine,* Vol.13, NO.4 July/August 1999.

[2] G.A. Cunningham,III "Delay Versus Throughput Comparisons for Stabilized Slotted ALOHA ," *IEEE Transactions on Communications,* Vol.38, NO.11 pp.1932-1934 November 1990.

[3] Sang Wu Kim "Stabilization of slotted ALOHA Spread-Spectrum Communications Network," *IEEE journal on Selected Areas on Communications,* Vol. 8, No. 4, pp. 555-561, May 1990.

[4] R. Muraly and B.L. Hughes,"Coding and stability in Frequency Hop Packet radio networks," *IEEE Transactions on Communications,* Vol.46, No.2, February 1998.

[5] Loren P. Clare, "Control Procedures for slotted ALOHA systems that achieve stability," *Proc. ACM SIGCOMM, 86 Symp. Commun. Arch. & Protocols*, pp.302-309 August 1986.

[6] M. Zorzi and Ramesh R. Rao., "Retransmission Control in Mobile Radio Slotted ALOHA," *IEEE ICC'94*, New Orleans May 1994.

[7] Ronald R. Rivest, "Network Control by Bayesian Broadcast" *IEEE Transactions on Information Theory,* VOL. IT-33, N° 3, pp.323-328, May 1987.

[8] Zhao Liu and Magda El Zarki, "Performance Analysis of DS-CDMA with Slotted ALOHA random access for packet PCNs" *Wireless Networks 1,* , pp.1-16, 1995.v

[9] L. Kleinrock and S.S.Lam, "Packet Switching in a multiaccess broadcast channel: Performance Evaluation" *IEEE Trans. Communications,* , COMM-23 pp.410-423, 1975.

[10] O. Sallent and R. Agustí, "Adaptative S-ALOHA CDMA as an Alternative Way of Integrating Services in Mobile Environments" *IEEE Trans. Veh. Technol,* , VOL.49, N°.3, pp.936-947, May 2000 .

[11] D. Geun Jeong and W. Sook Geon, "Performance of an Exponential Backoff Scheme for Slotted-ALOHA Protocol in Wireless Environment" *IEEE Trans. Veh. Technol,* , VOL.44, N°.3, pp.470-479, August 1995.

[12] B. Hajek and T. Van Loon, "Decentralized Dinamic Control of a Multiaccess Broadcast Channel" *IEEE Trans. Automat. Control,* VOL.AC-27, pp.559-569, June 1982.

[13] S.C.A. Thomopoulos, "A simple and Versatile decentralized Control for Slotted ALOHA, reservation ALOHA, and Local Area Networks " *IEEE Trans. Commun,* , VOL.COM-31, pp.763-774, June 1988.

[14] G. Fayolle,E. Gelenbe, and J. Labetoulle "Stability and Optimal Control of the Packet Switching Broadcast Channel " *J. Assoc. Comput. Machinery,* , VOL.24, N°.3, pp.375-386, July 1977.

[15] L. Kleinrock, "Queuing systems, Vol.1:Theory" *New York: Wiley-Interscience, 1975.*

Evaluación de Técnicas de Espectro Ensanchado para redes de sensores en canales ópticos no guiados difusos en interiores

F. Delgado, J. A. Rabadán, S. Pérez y R. Pérez-Jiménez

Grupo de Tecnología Fotónica y Comunicaciones
Departamento de Señales y Comunicaciones, Universidad de Las Palmas de Gran Canaria
Campus Universitario de Tafira, 35017 Las Palmas de Gran Canaria
Teléfono: 928457340 Fax: 928451243
E-mail: paco@agaete.teleco.ulpgc.es

Abstract. *In this work, the design and testing of a diffuse infrared, wireless fast-frequency hopping system for sensor networking is proposed. It is designed for synchronous transmission in order to reduce both complexity and cost of the terminal devices. We also show a network structure proposal and the prototype design and testing. We also give extensive operation results.*

1 Introducción

La saturación del espectro de radiofrecuencia en los últimos tiempos ha llevado a la búsqueda de soluciones alternativas en el campo de las redes inalámbricas. Una de estas alternativas es el uso de las comunicaciones ópticas en interiores. Sin embargo, el canal óptico presenta una serie de inconvenientes como la propagación multrayectoria o multicamino, o la gran presencia de interferencias presentes en la parte baja del espectro de frecuencias, bien producidas por la iluminación ambiente, por la iluminación artificial o por otros dispositivos ópticos.

En el presente artículo se presenta un sistema que aplica las técnicas de Espectro Ensanchado (SS), en concreto las de salto en frecuencia, como alternativa de sistema de comunicaciones que viene a paliar estos inconvenientes. Además el uso de estas técnicas permite el aprovechamiento de otras ventajas inherentes a ellas, como el uso de acceso múltiple al medio mediante código. El uso de estas técnicas está ampliamente difundido en sistemas de radio, pero no se han encontrado aplicaciones en el canal óptico.

En primer lugar, se describen brevemente los sistemas de espectro ensanchado de salto en frecuencia, a continuación se describe el prototipo de enlace desarrollado y, finalmente se presentan varias propuestas de aplicación del sistema en redes de datos de baja velocidad.

2 Sistemas FHSS en el canal óptico

Los sistemas de espectro ensanchado por salto de frecuencia, o *Frequency Hopping Spread Spectrum* (FHSS)[1], utilizan una señal de código para seleccionar una frecuencia de señal portadora de un conjunto de posibilidades. De esta forma, el ancho de banda de transmisión se amplía al conjunto de las nuevas portadoras y la señal de datos es transportada por una banda distinta cada vez que se cambia de frecuencia portadora.

Existen dos variantes de este esquema: *Fast* FHSS y *Slow* FHSS[1]. Un caso y otro se diferencian en la velocidad de salto de una portadora a otra. Si el cambio de portadora se realiza varias veces en un periodo de bit el sistema será *Fast* FHSS. Si por el contrario, el tiempo entre cambios de portadora permite transmitir varios bits en la misma portadora, nos encontraremos con un sistema *Slow* FHSS. La figura 1 ilustra un sistema FHSS.

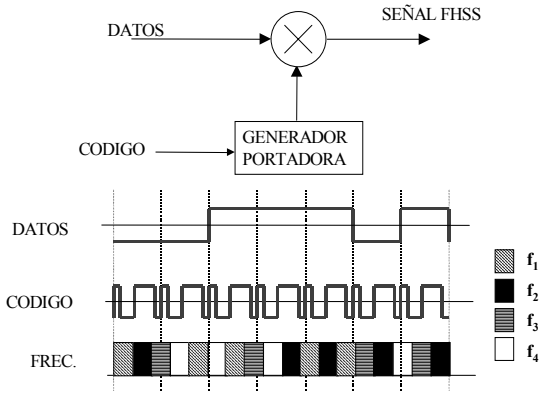


Fig 1. Sistema FHSS

Tanto en los sistemas FHSS como en los sistemas de espectro ensanchado en general, se justifica el aumento del ancho de banda utilizado por la obtención de nuevas prestaciones que mejoran considerablemente la calidad de la comunicación. La propiedad fundamental de los sistemas de espectro ensanchado es su elevado rechazo a las interferencias de banda estrecha [2], lo que les proporciona una mejora en la calidad de la señal recibida, denominada ganancia de proceso[3]. Por otra parte, mediante sistemas de espectro ensanchado es posible reducir el efecto de la propagación multicamino [4], de la que depende directamente la máxima tasa de transmisión del canal. Además, los sistemas de espectro ensanchado presentan la capacidad de acceso múltiple mediante división en código o CDMA, que permite que varios usuarios accedan y compartan el mismo canal al mismo tiempo y con las mismas

frecuencias. Este es uno de los esquemas de acceso al medio que más se utilizando últimamente por las redes de comunicación[5].

En este trabajo, se hace uso de los esquemas de modulación FHSS para dotar a las comunicaciones entre los nodos de una red óptica inalámbrica de las características mencionadas. Como veremos a continuación, hay ciertas propiedades del canal óptico no guiado que pueden ser mejoradas mediante el uso de enlaces con técnicas de espectro ensanchado.

El canal óptico presenta numerosas interferencias habituales[6], como son las distintas fuentes de iluminación: lámparas incandescentes, fluorescentes, lámparas alógenas, etc. Además existen otras fuentes de interferencias que pueden estar o no presentes en el medio, como los controles remotos, que producen ráfagas de datos cuando se utilizan, muy cortas y con velocidades muy pequeñas (cientos de bps); o los auriculares inalámbricos con señales de mayor ancho de banda (22-75 KHz) durante periodos de tiempo largos, en comparación con los anteriores. De mayor importancia, son las perturbaciones introducidas por otras redes ópticas no guiadas, tal es el caso de los enlaces IrDA (Infrared Data Association), con velocidades de hasta 16 Mbps, lo que supone interferencias de un ancho de banda considerables. Los sistemas de SS introducen mejoras ante estas interferencias reduciendo su efecto en la señal demodulada.

El otro fenómeno importante del canal óptico es la multipropagación, que se produce por las reflexiones de la radiación óptica en las paredes y los obstáculos del recinto[7]. Mediante el uso de espectro ensanchado se consigue una reducción drástica de las componentes de multipropación, y con ello, de la interferencia entre símbolos introducidas por el canal.

Además de lo anterior, la introducción de los esquemas de espectro ensanchado en los enlaces ópticos no guiado, permite que las redes basadas en estos enlaces puedan utilizar CDMA como técnica de acceso al medio.

3 Sistema Síncrono

La base de los sistemas de espectro ensanchado es la utilización de la señal de código para transmitir los datos y el uso de una réplica exacta de este código en recepción para poder recuperar los datos[1]. Por esta razón, el principal inconveniente de los sistemas de SS es la fuerte degradación del enlace frente a errores de sincronización entre transmisor y receptor, que impidan reproducir exactamente la señal de código en recepción. Por este motivo, la parte más complicada de los sistemas de espectro ensanchado se encuentra en los receptores; y es aquella que se encarga de mantener la sincronización entre la señal recibida y los códigos generados localmente en el receptor[3].

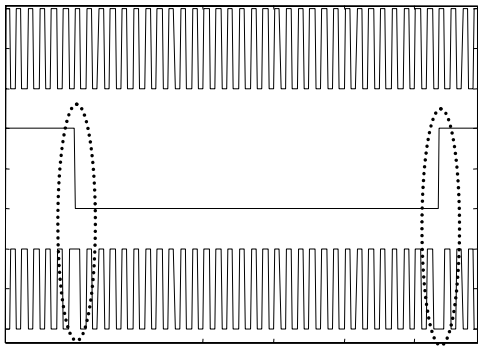


Fig 2. Señal de reloj (arriba), de duración de código (centro) y señal piloto final (abajo)

En este trabajo, se propone la incorporación de una señal piloto al sistema, con el fin de reducir la complejidad y coste de los circuitos de recuperación de sincronismo. De esa forma, se asegura una sincronización más robusta, a partir de la nueva referencia. Además, el hecho de que todos los elementos de la red se enganchen a la nueva señal, supone el paso a una estructura de red síncrona; con consecuencias positivas como se verá más adelante.

El piloto consiste en una portadora, a una frecuencia múltiplo de la del reloj de código, modulada en fase (BPSK) por una señal cuadrada, de un ancho de pulso igual a la duración de un periodo de la secuencia de código (señal de duración de código). Esta señal se muestra en la figura 2; podemos comprobar como con cada cambio de valor de la señal de duración de código, se produce un cambio de fase en la señal piloto.

En el receptor, será necesario extraer las dos señales antes mencionadas (reloj y duración de código) de la señal piloto presente en recepción. Para ello es necesario: por una parte, recuperar la portadora, de la que posteriormente se obtiene la frecuencia del reloj de *chip*; y por otra, demodular la señal BPSK y obtener la señal de duración de código para su posterior procesado. Con estas informaciones, se pueden simplificar los procesos de sincronización en el receptor. Básicamente el esquema de genereración y recuperación de la señal piloto es el de la figura 3.

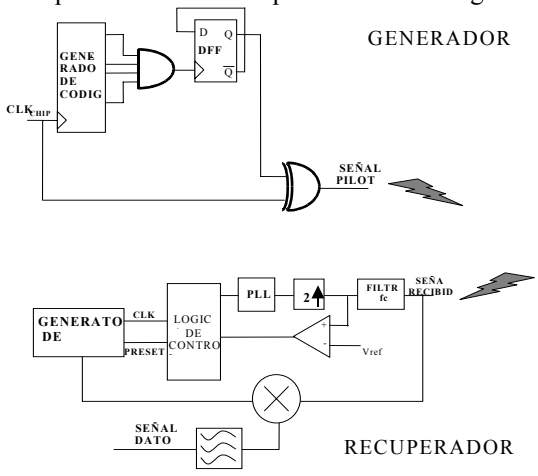


Fig 3. Sistema DSSS síncrono

4 Prototipo de enlace

Para comprobar las características de los sistemas propuestos, se ha diseñado e implementado un enlace de comunicaciones ópticas no guiadas con modulación *Fast FHSS* síncrono. El diagrama de bloques del sistema se muestra en la figura 4,

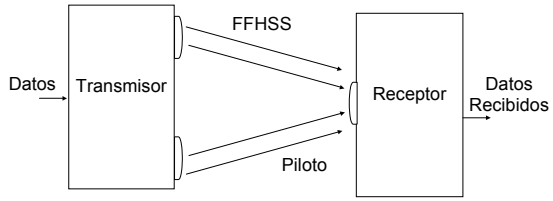


Fig 4. Esquema del sistema global.

Este enlace consta de un transmisor, que a su vez genera la señal de piloto de sincronismo y un receptor, que realiza la sincronización y recuperación de los datos. Un nodo de una red implementada con esta tecnología debería incorporar tanto el transmisor como el receptor, pero para las comprobaciones de funcionamiento y prestaciones el esquema propuesto es suficiente. Las caracteristicas básicas de este sistema son las siguientes:

- Tasa binaria de 512 kbits por segundo.
- Tasa de 1,536 Mchips por segundo.Esto supone utilizar tres portadoras distintas para transmitir cada bit de datos implicando redundancia en la transmisión.
- 32 frecuencias portadoras. Que están uniformemente distribuidas desde 24,384 MHz hasta 72 MHz en pasos de 1,536 MHz.
- Frecuencia de la portadora de la señal piloto de 9,216 MHz. Que ha sido seleccionada para que no interfiera con la señal FHSS. En la figura 5 se muestra la densidad espectral de potencia de la señal recibida (FHSS más piloto), y se puede apreciar la distancia entre piloto y la modulación.

Para la implementación de este prototipo se ha utilizado dispositivos de lógica programables y sintetizadores de generación directa digital (DDS), principalmente. En la figura 6 se muestra una foto del prototipo diseñado. Como ambos módulos (transmisor y receptor) se encuentran en la misma situación, se uitilizará un espejo para llevar la señal del transmisor al receptor.

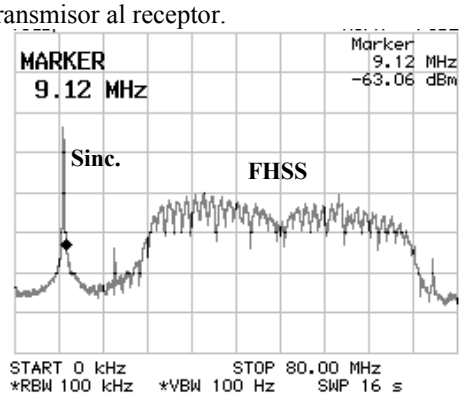


Fig 5: DEP de la señal FHSS y de la señal de sincronismo

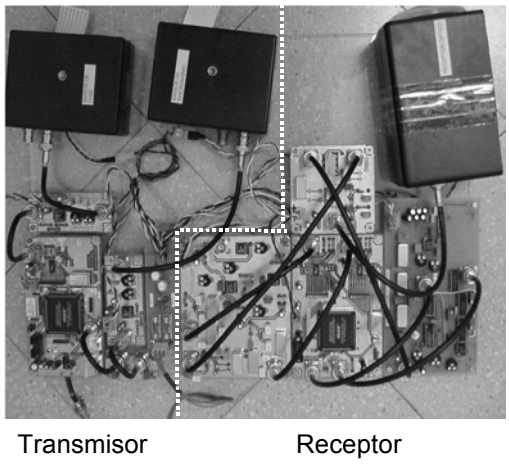


Fig 6. Sistema diseñado

3 Optimización del acceso al medio mediante CDMA

El desarrollo de sistemas FHSS introduce la posibilidad de incorporar protocolos de control de acceso al medio mediante códigos. Este es la mejor manera de maximizar la capacidad del canal en estos sistemas que ocupan un elevado ancho de banda. Generalmente, se asocia un código a cada uno de los pares emisor, receptor, sólo conocido por ellos y que se encuentra perfectamente sincronizado por ambos. Estos códigos son los encargados de generar las secuencias pseudoaleatorias de frecuencias en el enlace FHSS.

En los sistemas FHSS tradicionales, en cada enlace la fase de estos códigos es diferente, es decir, en cada par emisor-receptor la secuencia de código empieza y termina en un instante de tiempo distinto. En estas circunstancias, la forma de garantizar el menor número de coincidencias en la ocupación del mismo sector del espectro de frecuencia, entre dos enlaces distintos, es escoger códigos con propiedades determinadas. Normalmente se emplean códigos con baja correlación cruzada entre ellos. Con esta característica, óptima para los sistemas de Secuencia Directa, no se garantiza que no pueda haber un pequeño número de coincidencias en los sistemas de salto en frecuencia. Esto es debido a que, en estos últimos, las secuencias de código generan la palabra de código que indica al sintetizador de frecuencias cual es la portadora a emplear en cada instante. La interferencia entre dos enlaces FHSS que utilizan dos códigos distintos se produce cuando los dos transmisores utilizan la misma frecuencia simultáneamente. Por ello, la distancia Hamming entre los símbolos generados por las dos secuencias es el parámetro en función del cual se deben diseñar los códigos para el entorno CDMA. Una opción usualmente empleada, es la utilización de códigos Reed-Solomon y realizando un adecuado mapeado para el paso de la secuencia de códigos a la palabra de control del sintetizador digital de frecuencias.

3.1. Ventajas del sistema síncrono en el empleo de CDMA

Con las técnicas expuestas antes, se puede garantizar que todos los códigos comienzan y terminan en el mismo instante de tiempo. Esto implica que el diseño de redes que utilizan control de acceso múltiple al medio mediante código se simplifica considerablemente. En este caso si es posible garantizar que nunca exista una coincidencia de frecuencia entre dos transmisores distintos dentro de una misma red. Ya no se depende de parámetros estadísticos para este fin, además se pueden garantizar distancias adecuadas entre las portadoras empleadas por los distintos emisores. Otra posibilidad que se presenta es el diseño de los códigos a partir de las secuencias de frecuencias deseadas.

La opción más inmediata es la de poder emplear el mismo código en dos transmisores distintos con la particularidad de que en cada uno de ellos, el generador de código se inicialice en momentos distintos. Como una de las características de los códigos pseudoaleatorios es su baja autocorrelación, el hecho de que en cada enlace uno siempre se encuentre desplazado con respecto al otro es una medida de la mínima probabilidad de coincidencia entre dos transmisores en la selección de la frecuencia. La figura 7 representa la frecuencia escogida por dos transmisores distintos con relación al tiempo.

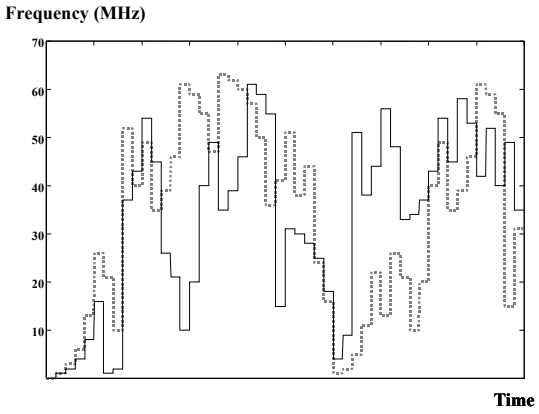


Fig 7. Evolución de la frecuencia con el tiempo en dos transmisores con mismo código desplazado

Otra posibilidad es el utilizar códigos de diseño propio a partir de la secuencia de frecuencias. Si se conoce el número máximo de transmisores distintos a emplear en la misma red y el número total de frecuencias, se pueden diseñar las secuencias de portadoras usadas por cada transmisor de forma que nunca dos transmisores empleen la misma simultáneamente. Esto es sólo posible en este caso gracias a que todas las secuencias empiezan y terminan al mismo tiempo. Para ello se genera una matriz como la de la Tab. 1, correspondiente a tres enlaces y ocho frecuencias disponibles, donde se representan en las columnas los intervalos de tiempo correspondientes a un periodo de la secuencia y en las filas el transmisor. Los valores del interior de la tabla corresponden a la frecuencia empleada por cada transmisor en cada intervalo.

Tabla 1

	T1	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**
Tx1	F5	F2	F7	F4	F1	F3	F6	F8
Tx2	F2	F5	F1	F8	F7	F6	F4	F3
Tx3	F8	F6	F5	F1	F3	F4	F2	F7

Al aumentar la cantidad de transmisores y de frecuencias disponibles, se hace complicado el sistema, por lo que existen algoritmos que generan los códigos partiendo de la misma idea [8].

La ventaja del primer sistema es que todas las propiedades básicas de los códigos empleados en espectro ensanchado se mantienen, así como su carácter pseudoaleatorio. En resumen, el empleo del sistema síncrono FHSS redunda en una considerable reducción de las interferencias producidas en un canal debido a otro, dentro de la misma red CDMA. Además, con una adecuada planificación de los códigos se pueden evitar zonas del espectro de frecuencias que estén afectadas por alguna señal interferente de banda estrecha.

4. Aplicaciones. Topologías de red

Una vez analizada la estructura de cada nodo individual, se expondrán las posibles estructuras de redes donde es viable su aplicación. Como ya se ha dicho, estos sistemas están pensados para entornos cerrados confinados por superficies con un determinado coeficiente de reflexión. El hecho de que todos los nodos utilicen la misma señal piloto de sincronismo, limita la distancia máxima de los enlaces, debido a los retardos producidos.

En primer lugar, existe un nodo principal que se diferencia de los demás en que es el encargado de transmitir el sincronismo de toda la red. La señal de sincronismo, como se ha visto es una portadora de frecuencia 9,216MHz modulada en fase por una moduladora, que es la señal que marca los principios y finales de las tramas de código. Por sus características en cuanto a frecuencia y ancho de banda, este piloto no se verá seriamente afectado por el efecto de la propagación multitrayecto. Además, ha de llegar a todos los nodos de la red (dentro del recinto), por lo que el nivel de potencia con el que se transmite, será más alto que el empleado para las señales de datos. Así pues, el nodo principal debe estar situado en un lugar visible desde las posiciones de todos los demás.

El primer ejemplo de aplicación es el de una red de actuadores, en la que existe un nodo principal con un transmisor óptico para la señal de sincronismo y otro con el que se transmiten, sumadas eléctricamente, todas las señales de datos FHSS. El resto de los nodos, simplemente serán receptores, llevan

incorporado un recuperador de sincronismo y receptores FHSS. Es el ejemplo de la figura 8.

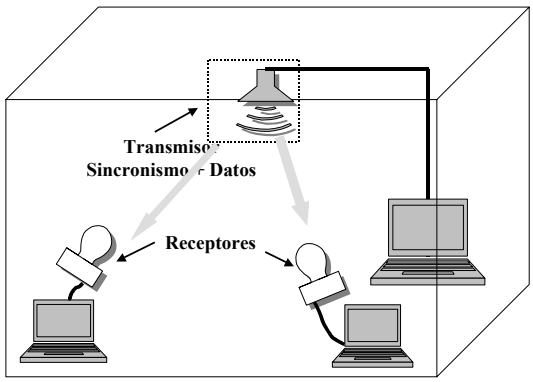


Fig 8. Ejemplo de disposición. Red de actuadores con un nodo principal emisor y múltiples receptores

Al considerarse igual el retardo sufrido por la señal de sincronismo y las de datos, el retardo no es crítico puesto que cada receptor (nodo secundario) recibe el sincronismo y su señal de datos con el mismo retardo.

Otra disposición es aquella en la que existe un nodo que sólo genera el sincronismo para todos los demás, que son los que se comunican entre ellos. En este caso, cada nodo recibe el sincronismo con un retardo determinado. Si existen dos nodos N1 y N2, cada uno de ellos genera su señal FH con su propia señal de sincronismo, pero ambos deben estar entre ellos relativamente sincronizados. Por este motivo es importante que la el desfase entre los pilotos de N1 Y N2 sea despreciable. Mediante simulaciones, se ha comprobado que un desfase en esta señal del 20% de su periodo, no afecta a la tasa de error del sistema. La figura 9 muestra esta disposición.

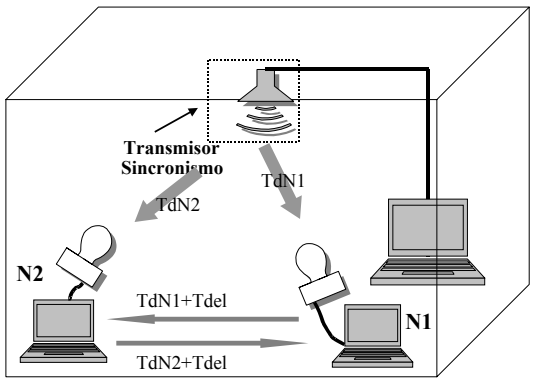


Fig 9. Disposición con un emisor dedicado al sincronismo u nodos secndarios con comunicación bidireccional.

En la figura 9 se refleja el retardo sufrido por la señal de sincronismo entre el nodo principal y cada uno de los secundarios (N1 y N2). Además se ve el retardo implicado en la transmisión de datos entre un nodo y el otro. Por ejemplo, para la correcta transmisión entre en nodo N1 hacia el nodo N2, el desfase total entre el código local de N2 y el de la señal recibida procedente de N1, será TdN1+Tdel-TdN2, que debe ser inferior al 20% del periodo de salto de la señal FH. Esto limita este tipo de topología a distancias de unos pocos metros, para la tasa de salto empleada en el presente sistema de 1,536MHz.

Otra opción es que todos los nodos reciban y emitan hacia un nodo central, que implementa todos los posibles enlaces y se encarga de la gestión del tráfico de la red y del sincronismo. El límite vendría dado por la distancia entre los nodos secundarios y el principal. En este caso la tasa de salto de la señal recibida en el nodo principal, procedente de uno secundario, vendrá retardada una cantidad $2Tdel$ respecto a la señal de código generada localmente en el primero.

En todas estas disposiciones vistas, cada equipo se comunica con uno sólo que posea su mismo código. Se trata por tanto de enlaces punto a punto en cada instante. Aunque es posible controlar mediante software el que cada nodo pueda cambiar en momentos determinados de código para comunicarse con otro.

5. Conclusiones

En este trabajo se ha demostrado la posibilidad del empleo de técnicas de salto en frecuencia en el canal óptico no guiado, presentando considerables mejoras en cuanto a probabilidad de error en presencia de interferencias de banda estrecha, así como del efecto de propagación multitrayecto. Además, se ha diseñado un prototipo de sistema de sincronismo que simplifica considerablemente la implementación de los receptores FH. Asimismo, la introducción de esta mejora permite utilizar códigos CDMA diseñados a medida o trabajar con el mismo código en varios enlaces con diferentes enlaces. Este sistema de sincronismo resulta más sencillo de implementar que los habitualmente empleados en técnicas de espectro ensanchado

Por último, se ha presentado una propuesta de aplicación novedosa de estas técnicas, en redes de datos de baja velocidad en interiores, como las de comunicación entre sensores, actuadores o equipos móviles mediante enlaces ópticos.

6. Agradecimientos

Este trabajo se ha realizado con la financiación de la Comunidad Autónoma de Canarias mediante el proyecto de investigación PI2001/109

Referencias

[1] R. C. Dixon, "Spread Spectrum Systems", John Wiley & Sons, 1984.

[2] R. Iqbal and J.S. Bedi, “Performance Analysis of Interference Rejection Techniques in Spread-Spectrum Communication”, Proceedings of TENCON´91.

[3] L.B. Milstein, "Interference Rejection Techniques in Spread-Spectrum Communications", Proceedings of the IEEE, vol. 76, no. 6, June 1988.

[4] Sklar, “Digital Communications”, Prentice-Hall, 1988.

[5] A. S. Tanenbaum, "Computer Networks", Prentice Hall,1996

[6] J. M. Kahn, J. R. Barry, "Wireless Infrared Communications", Proceedings of the IEEE, vol. 85, no. 2, February 1997.

[7] R. Perez-Jimenez, V.M.Melián, M.J. Betancor. "Analysis of Multipath Impulse Response of Diffuse and Quasi-Diffuse Optical Links for IR-WLAN" Proc. INFOCOM'95, pp. 7d.3.1-7d.3.7. Boston, MA., USA. abril 1995

[8] J. Salehi. “Code Division multiple acces techniques in Optical-Fiber Networks” . IEEE Transactions on Communications, Vol 37, nº8, pp 824-830, Aug. 1989.

Efecto Combinado de Técnicas de Nivel de Enlace Independientes en el Comportamiento de los Protocolos de Transporte de Internet sobre Redes de Área Local Inalámbricas

R. Agüero, M. García, L. Sánchez, J. Choque, L. Muñoz
Departamento de Ingeniería de Comunicaciones
Grupo de Ingeniería Telemática. Universidad de Cantabria
39005 Santander
E-mail: [ramon,marta,lsanchez,jchoque,luis]@tlmat.unican.es

Abstract*. This paper evidences an important enhancement of Internet transport protocols behavior over the IEEE 802.11b Wireless LAN if a Forward Error Correction (FEC) scheme is added to the idle RQ mechanism natively used by such technology. The FEC scheme has been enhanced with some smart cross-layer optimizations, and its operation adapts to channel conditions. Although providing a combined enhancement effect, both techniques do not follow a collaborative scheme, working on an independent fashion. A complete experimental approach has been followed, and exhaustive measurement campaigns have been carried out over a real platform to derive a large number of performance parameters. The FEC has been implemented as a module belonging to a generic layer two Performance Enhancing Proxy (PEP), which was the main outcome of the Wireless Internet NEtworks (WINE) IST project.*

1 Introducción y Planteamiento del Problema

A pesar de la reciente e importante expansión de las tecnologías inalámbricas, no sólo en el entorno de oficinas, sino también como método para establecer infraestructuras de acceso poco convencionales ("hot spots" y zonas rurales), queda todavía un gran conjunto de aspectos, mayoritariamente técnicos, que es necesario resolver. Uno de los más estudiados es el deficiente rendimiento que presentan los protocolos de Internet al operar sobre este tipo de arquitecturas, debido, principalmente, a que su diseño se realizó teniendo en cuenta las características intrínsecas de las redes tradicionales cableadas y, por tanto, el efecto de las particularidades del medio radio sobre su comportamiento puede llegar a ser muy negativo.

Entre las tecnologías de redes de área local inalámbricas (WLAN, Wireless Local Area Network) que han ido surgiendo en la últimos años, la recomendación IEEE 802.11b es, sin ninguna duda, la más extendida en la actualidad, existiendo una importante corriente investigadora que se centra en el comportamiento de los protocolos de la pila IP sobre dicha interfaz, abordándose este análisis desde una doble perspectiva, según se trate de tráfico orientado a la conexión (transporte TCP) o no (transporte UDP) [1], [2].

Por otro lado, también es interesante diferenciar entre el rendimiento que se alcanza cuando la presencia de errores debidos al canal inalámbrico es despreciable, y el que se obtiene cuando los errores aparecen de manera relevante y sobre todo, a ráfagas. En el primero de los casos se observa una pérdida considerable de la capacidad bruta, debido a la gran sobrecarga que introducen las capas física y MAC del estándar IEEE 802.11b. De esta manera, y cuando se emplea la tasa de binaria de 11 Mbps, alrededor del 50% de la capacidad bruta se pierde, alcanzándose un rendimiento máximo en torno a 6 Mbps con tráfico UDP. La sobrecarga tiene todavía un impacto mayor al utilizar TCP, obteniéndose 5 Mbps, debido a los reconocimientos generados por la entidad receptora.

La diferencia entre los rendimientos que se alcanzan con ambos protocolos de transporte cobra más importancia cuando se trabaja sobre canales caracterizados por una relación señal a ruido (SNR, Signal to Noise Ratio) baja. UDP, como protocolo no orientado a la conexión, continúa generando tráfico, sin tener en cuenta la aparición de errores y la pérdida de datagramas correspondiente. Por otra parte, TCP se diseñó para reaccionar de manera apropiada ante situaciones de congestión en la red, causa principal de la pérdida de segmentos en las redes cableadas tradicionales. En este sentido, un transmisor TCP reduce la tasa de generación de datos al recibir indicaciones de congestión en la red, siendo ésta una política poco adecuada en redes inalámbricas. Se ha demostrado que la presencia de periodos de inactividad elevados en la entidad transmisora es la principal causa del bajo rendimiento cuando TCP se utiliza sobre canales inalámbricos caracterizados por una baja SNR [3].

Adicionalmente, debe considerarse otro aspecto diferenciador entre la caracterización basada en UDP y TCP. Mientras que este último sigue un esquema orientado a la conexión y, por tanto, asegura la correcta entrega de todos los segmentos de datos en el receptor (utilizando, si fuera preciso, mecanismos

de retransmisión), UDP es un protocolo no orientado a la conexión y, por tanto, si las condiciones del canal radio son malas, hay un número de datagramas que se pierden y no se recuperan posteriormente. Es, por tanto, necesario analizar además del rendimiento, como en el caso de TCP, la pérdida sufrida por el nivel de aplicación.

La Tabla 1 recoge los resultados obtenidos en un canal 802.11b con una SNR baja, utilizando tanto tráfico UDP como TCP. Aunque posteriormente se proporcionará una definición más detallada, es importante recalcar la diferencia entre la tasa de error de trama (FER, Frame Error Rate) y la pérdida MAC residual. La primera estadística representa el porcentaje de tramas erróneas frente al total de tramas que llegan al receptor, mientras que la segunda se refiere a la pérdida sufrida a nivel IP. Se observa que, para una FER similar, la pérdida MAC residual es apreciablemente mayor para tráfico UDP. La explicación a este fenómeno está en el hecho de que TCP reacciona ante las condiciones deficientes del canal radio reduciendo la tasa de envío de segmentos a la red.

Es importante, asimismo, mencionar que todas las medidas anteriores se han realizado en un entorno real, en el que la distancia entre ambos terminales era alrededor de 15 metros, con la presencia adicional de obstáculos metálicos y personas en movimiento en el canal de propagación. La Fig. 1 muestra la distribución de SNR que caracteriza dicho escenario en particular.

En la literatura se han propuesto algunos métodos para minimizar estas deficiencias; mientras que unos enfocan el problema a través de la modificación de los protocolos de transporte (como ocurre en [4]), otros se basan en soluciones locales, transparentes para las capas superiores [5]. Aunque tradicionalmente se ha puesto mayor atención en la mejora de TCP, algunas técnicas tradicionales, como la corrección de errores mediante FEC, pueden ser beneficiosas tanto para TCP como para UDP. Este trabajo pone de manifiesto una importante mejora en el comportamiento de ambos protocolos si se emplea dicha técnica.

Tabla 1. Parámetros de rendimiento de los protocolos TCP y UDP en un canal IEEE 802.11b (11 Mbps) con baja SNR

Protocolo	Medida	Rendimiento (Mbps)	FER	Pérdida MAC Residual
UDP	1	2.32	52.98%	14.57%
	2	3.58	33.11%	5.83%
	3	4.03	26.06%	5%
	4	4.78	16.26%	2.45%
	5	5.49	6.93%	1.18%
	6	5.96	1.40%	0.19%
TCP	1	0.55	41.85%	7.2%
	2	0.67	36.01%	3.5%
	3	1.19	29.17%	4.2%
	4	2.39	25.53%	2.9%
	5	3.23	15.30%	1.3%
	6	4.36	5.21%	0%

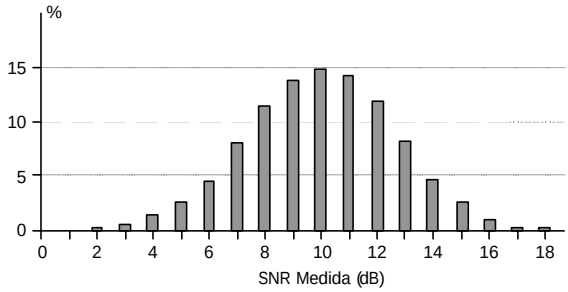


Figura 1. Distribución de la SNR en el escenario de medida

2 Técnicas de nivel de enlace independientes

2.1 ARQ "propietario"

El estándar IEEE 802.11 define un mecanismo ARQ simple para hacer frente tanto a las colisiones como a los errores que introduce el canal radio. Cuando una estación recibe una trama, chequea el Código de Redundancia Cíclica (CRC, Cyclic Redundancy Check), para comprobar si la trama llegó correctamente. En caso afirmativo, mandará una confirmación a la estación transmisora; en caso contrario, será ésta la que, tras un intervalo definido, retransmita la trama. Las tarjetas comerciales IEEE 802.11b de Avaya (antes Lucent, Orinoco y Agere) fijan en su "firmware" el número máximo de retransmisiones por trama a tres, siendo dicho parámetro imposible de modificar desde el controlador de la tarjeta, es decir, no existe ninguna forma de poder controlar las retransmisiones de 802.11 [6]. De esta manera, la pérdida de un datagrama IP implica la recepción de cuatro tramas erróneas de manera consecutiva y, por tanto, la FER no coincide con la pérdida MAC residual o pérdida IP, en este caso.

2.2 FEC Adaptativo

El principal objetivo de un esquema FEC a nivel de enlace es la corrección del mayor número posible de las tramas erróneas que llegan a un receptor. Su diseño necesita una exhaustiva campaña de medidas previa, con la que caracterizar la distribución de los bits erróneos en las tramas. Tras realizar esta tarea en el escenario caracterizado por la distribución de SNR de la Fig. 1, se eligió un código Reed-Solomon, con una capacidad correctora de t=15 símbolos. Teniendo en cuenta la distribución del número de bits erróneos por trama y el tamaño medio de la ráfaga de bits incorrectos, valores obtenidos a través de las medidas realizadas, este código debería corregir alrededor del 70% de las tramas erróneas en el peor de los casos (trabajando con un tamaño de trama de 1500 bytes). El código fue definido sobre el cuerpo de Galois $GF(2^{16})$, por lo que la longitud de cada símbolo es de 16 bits y, por tanto, la sobrecarga añadida por el FEC (2t) es de 480 bits por trama.

Con el fin de minimizar la sobrecarga computacional introducida por el FEC, se siguieron dos estrategias. En primer lugar, el proceso de decodificación aprovecha la capacidad de detección implementada a través del CRC de IEEE 802.11b, información accesible a través del controlador del interfaz inalámbrico. De esta manera, si al llegar una trama al receptor se recibe sin error, el FEC simplemente suprimirá sus bits de redundancia, sin que la función de decodificación procese la trama. Por otro lado, para evitar que las tramas sean codificadas cuando las condiciones del canal son buenas, la información acerca de la SNR, exportada por la infraestructura subyacente, se utiliza como métrica para decidir si proteger una determinada trama o no. Esta funcionalidad se consigue gracias al marco en el que se incluye el esquema FEC, que se describe en la siguiente sección y que incluye un conjunto de funcionalidades, entre las que se encuentra la de medir periódicamente la calidad del enlace. Tras una caracterización exhaustiva del enlace radio, en la que se derivó el porcentaje de tramas recibidas erróneas para cada valor de SNR, se eligió un umbral de codificación de 15 dB, tal y como se muestra en la Fig. 2. Cuando la SNR observada se encuentra por debajo de este nivel, las tramas transmitidas sí llevarán protección FEC pero, sin embargo, cuando las condiciones del canal son mejores, el codificador FEC no añade sobrecarga alguna a las tramas transmitidas. Adicionalmente, para evitar oscilaciones continuas en torno al umbral, se incluyó un sencillo mecanismo de histéresis en la implementación.

Es importante destacar que las dos técnicas de nivel de enlace que se han descrito anteriormente operan de forma independiente. En este sentido, no se puede decir que esta propuesta represente una solución FEC/ARQ híbrida. A pesar de que el FEC sea capaz de corregir los errores de una trama en particular, el esquema ARQ no será consciente de esta situación y, por tanto, continuará con su operación normal, retransmitiendo la trama hasta que reciba una confirmación o alcance el número máximo de cuatro transmisiones por trama. De esta manera, un número considerable de retransmisiones son inútilmente decodificadas por el FEC, ya que tienen que descartarse, provocando una pérdida relevante de rendimiento. Para evitar que datagramas duplicados continúen su camino a través de la pila de protocolos, se empleó el campo identificador de la cabecera IP para poder descartarlos.

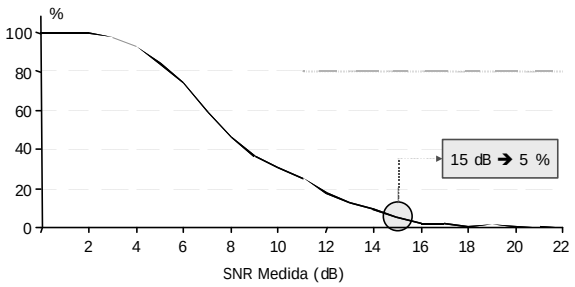


Figura 2. Porcentaje de tramas erróneas en función de la SNR

3 Marco de desarrollo

Como se ha mencionado anteriormente, el esquema FEC se ha implementado como un módulo perteneciente a un PEP de capa 2 genérico, denominado Capa de Adaptación Inalámbrica (WAL, Wireless Adaptation Layer), que fue el resultado principal del proyecto europeo WINE perteneciente al programa IST [7]. La capa WAL compensa el escaso rendimiento de los protocolos TCP/IP cuando operan sobre redes de área local inalámbricas con baja SNR, ocultando las deficiencias del canal radio a las capas superiores. En lugar de modificar los protocolos de Internet para adecuarlos a las características del medio radio, la adaptación se lleva a cabo a través de esta nueva entidad, situada entre la capa IP y las tecnologías de acceso subyacentes. Uno de los aspectos clave de la capa WAL es su flexibilidad, ya que está formada por un conjunto de módulos, apropiados para diferentes tipos de tráfico, siendo el FEC ejemplo de uno de ellos.

La capa WAL añade una cabecera propietaria, con una longitud de 2 bytes, y el FEC protege todo el paquete WAL, a excepción de la misma. De esta manera, los errores que se encuentren en la cabecera WAL serán irrecuperables.

Como ya se ha comentado con anterioridad, el FEC adapta su operación a la calidad del enlace inalámbrico, que se mide en términos de su SNR. El cambio de la capacidad correctora del FEC en una comunicación en curso entre dos terminales se realiza tras el establecimiento de un nuevo canal lógico, denominado “asociación” en el marco del proyecto WINE, que depende tanto del tipo de tráfico como de las condiciones del canal radio (la SNR asociada al mismo). Este cambio entre asociaciones se realiza tras el intercambio de un conjunto de paquetes propietarios WAL, sin imponer dicho intercambio una sobrecarga relevante.

4 Plataforma de medidas y definición de parámetros

Para validar la arquitectura propuesta, se llevó a cabo una completa campaña de medidas en una plataforma experimental. Esta sección resume los aspectos principales de la misma. Adicionalmente se definirán los parámetros que se emplearán más adelante para verificar las mejoras de la propuesta.

La plataforma experimental consta de dos máquinas Pentium III, con sistema operativo Red Hat Linux 7.1 (kernel actualizado a la versión 2.4.9). Una de ellas actúa como Punto de Acceso (AP, Access Point) y la otra como Terminal Móvil (MT, Mobile Terminal). Tanto el AP como el MT incorporan tarjetas PCMCIA IEEE 802.11b de Orinoco configuradas en modo ad hoc. La WAL, con el módulo FEC incorporado, se carga como dispositivo de red virtual en ambos terminales. Para adaptar la operación del

FEC a la calidad del enlace radio, el AP manda un mensaje propietario (denominado "beacon") cada 5 segundos, al que el MT responderá, reportando la SNR medida.

Para cuantificar la mejora introducida por el esquema propuesto se emplearán distintos parámetros. Dado que se han utilizado dos protocolos de transporte contrapuestos, algunos parámetros serán asimismo diferentes para cada uno de ellos. A continuación se describen brevemente:

Rendimiento o throughput: número de bits útiles recibidos dividido entre la duración del experimento.

FER: cociente entre el número de tramas MAC recibidas con error y el total de tramas recibidas.

Longitud Media de Ráfagas Erróneas (EFB, Erroneous Frame Burst): longitud media de las ráfagas de tramas erróneas a lo largo de la medida.

Longitud máxima de Ráfagas Erróneas: longitud de la mayor de las ráfagas de tramas erróneas recibidas a lo largo de la medida.

Pérdida MAC Residual: número de datagramas IP no recuperados por el mecanismo ARQ definido en el estándar IEEE 802.11. Se corresponde con el porcentaje de datagramas IP que se habrían perdido si el FEC no hubiera estado presente.

Pérdida IP Residual: porcentaje de datagramas no recuperados por las técnicas de nivel de enlace (ARQ y FEC).

Para el caso de TCP son necesarios algunos parámetros adicionales para enriquecer el análisis. Todos cubren estadísticas específicas de la operación de TCP y son proporcionadas por las herramientas de análisis empleada a lo largo de la campaña de medidas (*tcpdump y tcptrace*).

Retransmisiones: número total de segmentos de datos TCP retransmitidos por la entidad transmisora.

Máximo número de retransmisiones: número máximo de veces que un mismo segmento de datos se retransmite a lo largo de toda la transferencia.

Tiempo de inactividad máximo: máximo periodo de tiempo durante el que la entidad transmisora no envía ningún segmento.

Reconocimientos triplicados: número de reconocimientos que llegan por tercera vez al transmisor.

Adicionalmente, el módulo FEC proporciona un conjunto de métricas necesarias para poder analizar su comportamiento. A pesar de que no se recogen en las tablas de la siguiente sección, es interesante definirlas brevemente, ya que dan una idea de todos las posibles situaciones que se pueden dar cuando una trama atraviesa el módulo FEC:

- Número de tramas recibidas por el módulo.
- Número de tramas protegidas recibidas sin error.
- Número de tramas protegidas recibidas con error, tanto las corregidas como las no corregidas por el FEC.
- Número de tramas protegidas recibidas con error, pero que no tuvieron que ser corregidas, por estar su error situado en la cabecera IEEE 802.11 (datagrama IP correcto).
- Número de tramas retransmitidas que tuvieron que ser descartadas, tanto las recibidas sin error como las corregidas por el FEC.
- Número de tramas no protegidas recibidas sin error.
- Número de tramas no protegidas recibidas con error, que serán descartadas.

A partir de los parámetros anteriores, se puede calcular el número de datagramas que llegan a la capa IP. De esta manera se puede comprobar la validez de todo el procedimiento, comparando este valor con los resultados dados por la herramienta de medida. Además, es posible estimar la manera en la que se comportó el FEC, calculando las siguientes métricas:

Capacidad correctora: porcentaje de tramas corregidas frente al número total de tramas erróneas y protegidas que llegan al módulo.

Ganancia FEC: porcentaje de datagramas IP que el FEC ha sido capaz de recuperar, frente al número de pérdidas que se habría producido si el FEC no hubiera estado presente.

Por último, también es interesante conocer el número de tramas (en términos relativos) que no llegaron al FEC por presentar errores en la cabecera WAL.

5 Resultados experimentales

En el caso de UDP, los experimentos se realizaron empleando la herramienta *nttcp*. El AP manda 10.000 paquetes UDP/IP hacia el MT. Ya que la sobrecarga conjunta de la WAL y el módulo FEC es de 63 octetos, la longitud del campo de datos de UDP se fija a 1409 bytes, para mantener el mismo tamaño de la trama empleado anteriormente. El experimento se repitió un total de 15 veces en la situación caracterizada por la SNR de la Fig. 1. Como se mencionó en la Sección 1, este entorno presenta una gran variabilidad en las condiciones del canal. La Tabla 2 recoge los resultados obtenidos, ordenados de

mayor a menor FER. A continuación se resaltan las principales conclusiones que se pueden extraer:

- Salvo en las medidas 3 y 4, la pérdida IP residual, medida después de que el FEC haya procesado las tramas, permanece por debajo del 1%, a pesar de que la FER alcance valores cercanos al 50%. Además, comparando los valores de las Tablas 1 y 2, se puede observar que, en condiciones similares de FER, esta importante mejora en la pérdida IP residual se produce sin degradación alguna en el rendimiento final.

- Con la excepción de las medidas 3, 4, 8 y 14, el FEC es capaz de recuperar más del 90% de los datagramas IP que se habrían perdido si no se hubiera utilizado.

- Teniendo en cuenta la SNR que caracteriza el entorno de medidas (ver Fig. 1) y el umbral anteriormente mencionado, la mayoría de las tramas están protegidas por la redundancia FEC, ya que la SNR reportada por el MT suele ser menor de 15 dB.

- La capacidad correctora del FEC es, salvo en la medida 3, mayor del 70%, siendo por tanto dicha capacidad mejor que la fijada en la fase de diseño, como se recoge en la Sección 3. La escasa tasa correctora alcanzada en la medida 3 se puede explicar atendiendo a la longitud de las ráfagas de tramas erróneas, mucho mayor que en el resto de casos.

- Cuando la tasa de errores no es demasiado alta, puede darse el caso de que, a pesar de haber corregido la mayoría de las tramas con errores no se consiga una ganancia FEC apreciable. Este aspecto se debe a que, en estos casos, el efecto de descartar tramas tiene un impacto mayor y, además, el mecanismo ARQ puede ser suficiente para solventar la aislada ocurrencia de errores. La medida 14 es un claro ejemplo de ello. En este caso, adicionalmente, la adaptación FEC no fue demasiado fina, por lo que el 78% de las tramas erróneas no llevaron protección FEC; esto junto con el hecho de que todas las tramas corregidas tuvieron que ser descartadas, resultó, finalmente, en una ganancia FEC nula.

- El número de tramas erróneas que no alcanzan el módulo FEC es relativamente pequeño. En todas las ocasiones, salvo en la medida 3, el porcentaje frente al total de tramas erróneas permanece por debajo del 10%. Como se ha comentado anteriormente, la longitud de las ráfagas de errores en la medida 3, supera con mucho, a las obtenidas en el resto de las situaciones por lo que parece lógico que se hayan dado más casos de tramas con bits erróneos en la cabecera WAL. Este aspecto también contribuye a la baja ganancia FEC observada en esa medida.

En los experimentos TCP, se empleó la aplicación *NcFTP* para transferir un fichero de 10 Mbytes. Durante toda la campaña de medidas se utilizó la versión TCP Reno, con las opciones de "Timestamp" y reconocimiento selectivo activadas. Tal y como sucedía en el caso de UDP, con el fin de adaptar el tamaño de la trama a los 63 bytes de sobrecarga introducidos por la WAL y el FEC, el tamaño máximo de segmento (MSS, Maximum Segment Size) se redujo de 1448 a 1385 bytes. El FEC se emplea en ambos terminales por lo que protege tanto los segmentos de datos generados por el transmisor como los reconocimientos devueltos por el receptor.

La Tabla 3 compara los resultados obtenidos sin el FEC (recogidos anteriormente en la Tabla 1), con los que se obtuvieron tras activar dicho módulo. En este caso, la mejora introducida se puede apreciar en todos los parámetros de caracterización del comportamiento de TCP obtenidos para cada una de las medidas. En las peores situaciones de FER, el rendimiento alcanzado en la transferencia del fichero

Tabla 2. Resultados con tráfico UDP, utilizando el módulo FEC sobre un canal IEEE 802.11b con baja SNR

#Test	Tput UDP (Mbps)	FER	EFB Med	EFB Max	Pérdida MAC Residual	Pérdida IP Residual	Ganancia FEC	Capacidad Corrección	Tramas erróneas no proteg	Tramas erróneas no llegan al FEC
1	2.76	48.86%	3.17	194	11.58%	0.08%	99.31%	95.58%	0%	0.66%
2	2.93	43.02%	1.9	16	4.39%	0%	100%	99.47%	0%	0.50%
3	3.14	34.96%	5.33	1288	9.24%	5.20%	43.72%	56.66%	0%	13.99%
4	3.50	30.58%	3.36	397	6.21%	3.30%	46.86%	75.35%	24.11%	5.63%
5	3.65	28.59%	2.22	38	3.37%	0.07%	97.92%	98.25%	0%	0.57%
6	3.85	27.12%	2.47	47	4.03%	0.22%	94.54%	93.30%	0%	1.59%
7	3.91	23.23%	2.74	206	3.71%	0.14%	96.23%	91.21%	0%	0.68%
8	4.45	19.61%	2.08	465	2.21%	0.99%	55.20%	81.72%	13.44%	3.84%
9	4.39	15.66%	2.30	62	2%	0.26%	87%	87.25%	0%	2.41%
10	4.48	14.89%	1.43	16	0.61%	0%	100%	96.05%	0%	0.69%
11	4.61	12.89%	2.59	127	1.89%	0.16%	91.53%	85.72%	0%	1.99%
12	5.13	9.17%	2.66	150	1.94%	0.17%	91.24%	84.07%	3.44%	1.30%
13	4.84	3.33%	1.22	10	0.04%	0%	100%	97.62%	0%	2.02%
14	5.59	2.13%	2.51	60	0.28%	0.28%	0%	95.65%	78.3%	2.75%
15	5.22	1.99%	2.10	39	0.20%	0.04%	80%	80%	0%	3.43%

Tabla 3. Resultados con tráfico TCP con y sin el módulo FEC sobre un canal IEEE 802.11b con baja SNR

FEC	#Test	Tput TCP (Mbps)	FER	Pérdida MAC Residual	EFB Med	EFB Max	Retx TCP	Max Retx	Inact Max (seg)	Triple ACK´s	Capacidad Corrección	Ganancia FEC
No	1	0.55	41.85%	7.2%	2.43	104	620	9	28.16	114		
	2	0.67	36.01%	3.5%	2.08	109	264	9	39.68	103		
	3	1.19	29.17%	4.2%	2.27	89	314	6	8.32	68		
	4	2.39	25.53%	2.9%	2.08	43	217	5	1.28	61		
	5	3.23	15.3%	1.3%	1.80	48	103	4	2.24	36		
	6	4.36	5.21%	0%	1.3	6	1	1	0.20	1		
Yes	1	2.56	43.5%	11.2%	4.05	220	137	2	0.29	24	74.69%	84.93%
	2	2.67	42.9%	9.2%	2.68	78	50	2	0.43	6	92.14%	94.73%
	3	2.64	35.8%	4.6%	2.9	224	114	3	0.44	13	83.83%	96.31%
	4	2.55	32.3%	7.3%	4	193	138	3	1.24	19	67.8%	76.11%
	5	3.72	15.9%	2.6%	2.95	141	54	3	0.29	12	76.63%	77.5%
	6	4.15	8.4%	1.4%	3.35	214	4	1	0.17	1	79.28%	96.23%

se multiplica por cinco. Asimismo, el número de retransmisiones de TCP se reduce considerablemente debido, principalmente, a la elevada ganancia FEC que se observa en todos los casos (mayor del 76%). Adicionalmente, ningún segmento TCP ha sido retransmitido en más de tres ocasiones. Por último, el periodo de inactividad máximo en el transmisor es siempre menor de 1,5 segundos y, por tanto, las ráfagas de tramas erróneas observadas son mayores que en el caso de no utilizar FEC, siendo más cercanas a las obtenidas con tráfico UDP, en el que el transmisor no se detiene a la espera de ningún reconocimiento.

6 Conclusiones

Se ha demostrado que el FEC es un método muy eficaz para mejorar el comportamiento, tanto de TCP como de UDP, en aquellas situaciones en las que el esquema ARQ del estándar IEEE 802.11b no es suficiente para recuperar los errores a ráfagas producidos por el canal radio. Para el tráfico UDP, el FEC reduce notablemente la pérdida de datagramas que sufriría la aplicación receptora, mejorando la calidad subjetiva percibida por el usuario. Por otra parte, en el caso de TCP, para tasas de error de trama entre el 30% y el 45%, el FEC evita que el rendimiento caiga por debajo de 2.5 Mbps, estabilizándolo en torno a dicho valor.

Esta importante mejora se ha logrado a pesar de que las dos técnicas de nivel de enlace operan de manera independiente. Es más que probable que si se pudiera emplear una interacción real entre ambas, se alcanzaría un rendimiento más optimizado. Teniendo en cuenta que están apareciendo productos comerciales más abiertos, que permitirán una colaboración más estrecha entre ambas técnicas, se acometerá en el futuro la integración del esquema FEC con esas nuevas interfaces.

Referencias

[1] G. Xylomenos, G. Polyzos, P. Mähönen, M. Saaranen, "TCP Performance Issues over Wireless Links", IEEE Communications Magazine, Abril 2001, pp 52-58.

[2] M. García, R. Agüero, L. Muñoz, P. Mähönen, "Behavior of UDP-Based Applications over IEEE 802.11 Wireless Networks", 12th IEEE International Symposium on Personal Indoor and Mobile Radio Communication, PIMRC 2001, San Diego (USA), Octubre 2001. Vol II, pp 72-77.

[3] L. Muñoz, M. García, J. Choque, R. Agüero, P. Mähönen, "Optimizing Internet Flows over IEEE 802.11b Wireless Local Area Networks: A Performance Enhancing Proxy Based on Forward Error Correction", IEEE Communications Magazine, Vol 39, Nº 12, Diciembre 2001, pp 60-67.

[4] A. Bakre, B. R. Badrinath, "I-TCP: Indirect TCP for Mobile Hosts", 15th International Conference on Distributed Computing Systems, Mayo 1995

[5] D.C. Feldmeier, A.J. McAuley, J.M. Smith, D.S. Bakin, W.S. Marcus, and T.M. Raleigh, "Protocol Boosters," IEEE Journal on Selected Areas in Communications, Vol 16, Nº 3, Abril 1998, pp. 437-444.

[6] D. Gibson, "Wireless Networking with Linux and IEEE 802.11b," 8th International Linux-Kongress, Enschede, Holanda, Noviembre 2001.

[7] P. Mähönen et al, "Platform-Independent IP Transmission over Wireless Networks: The WINE Approach," IEEE Personal Communications, Vol 8, Nº 6, Diciembre 2001, pp. 32-40.

Propuestas para el despliegue de redes de acceso inalámbricas de bajo coste basadas en tecnología WLAN

L. Sánchez, V. Gutiérrez, R. Agüero, L. Muñoz
Departamento de Ingeniería de Comunicaciones.
ETSII y Telecomunicaciones – Universidad de Cantabria
Avda. Los Castros s/n, 39005 SANTANDER
Teléfono: 942 201392 Ext:14 Fax: 942 201488
E-mail: {lsanchez,verónica,ramon,luis}@tlmat.unican.es

Abstract. *People are always searching for faster and cheaper access solutions to the Internet. As we have witnessed with voice services over the years, the benefits that mobility adds to a solution generates great appeal for users. Providing users with access to the Internet while they are on the move or away from their desk or office promises to deliver many benefits for corporate entities and consumers. The first generation of always-on data solutions is called General Packet Radio Service (GPRS). The next evolutionary step is known as 3rd Generation mobile technology (3G). These networks will provide faster and more transparent access to data and multimedia services like the Internet. Wireless Local Area Networks (WLAN) is a technology that has been around for some time. At present most WLAN deployments have been part of internal company wireless intranet solutions. However, there has been a great deal of effort amongst telecommunication suppliers to provide an integrated public WLAN solution by allowing an operator to deploy WLAN as a complement to their existing Wide Area Networks. This article will depict two different scenarios for these public WLANs as it will also describe a tool developed by this group which tackle with all the management needs of this kind of networks.*

1 Introducción

En la actualidad, la mayoría de los organismos de estandarización, fabricantes y operadores están centrando sus esfuerzos en el despliegue de las redes de 3G. Paralelamente, las WLAN están emergiendo como una nueva forma de proveer servicios de datos; en lugares como pueden ser aeropuertos, centros de convenciones, estaciones de tren, hoteles, etc, aparecen como una alternativa a las redes existentes mientras que en zonas poco pobladas, típicamente áreas rurales, surgen como complemento a las redes fijas convencionales y/o redes de 3G, puesto que el despliegue de éstas resulta económicamente inviable dado el bajo factor de penetración de los servicios de datos en estas zonas. La madurez y el bajo coste relativo de la tecnología basada en el estándar IEEE 802.11b [1,2,3], así como el hecho de que opere en la banda industrial científico médica (ISM, *Industrial Scientific Medical*) de 2.4GHz, hacen de ella una de las opciones más atractivas y viables a la hora de implementar soluciones para los escenarios anteriormente citados.

El contenido de este artículo se estructura en tres partes. En primer lugar se describen las posibles aplicaciones de las WLAN para el despliegue de redes de acceso públicas en dos escenarios típicos como son un Hot-Spot y un área rural. Posteriormente, se describe una herramienta de gestión de redes de acceso públicas desarrollada por el Grupo de Ingeniería Telemática de la Universidad de Cantabria. Se trata de una solución software que proporciona los mecanismos necesarios para la gestión del acceso a Internet a través de redes de acceso inalámbricas. El artículo finaliza con una serie de conclusiones fruto del trabajo desarrollado con este tipo de redes y la herramienta de gestión y con las posibilidades que abren este tipo de alternativas.

2 Aplicaciones de las WLAN en el despliegue de redes de acceso públicas

Las WLAN aparecen a finales de los 90, destinándose básicamente a la implantación de redes corporativas en grandes compañías. La tecnología con más aceptación y más utilizada dentro de las WLAN es la que se basa en el estándar IEEE 802.11 en su extensión de alta velocidad IEEE 802.11b, que, utilizando técnicas de espectro ensanchado por secuencia directa, es capaz de llegar a regímenes binarios brutos de 11 Mbps. Sin embargo, estudios realizados [4] concluyen que el rendimiento neto se ve reducido a causa del protocolo de acceso al medio hasta alcanzar un máximo de unos 6 Mbps. La Tabla 1 muestra algunas de las características más importantes de 802.11b.

El crecimiento experimentado en el uso de las WLAN hace que hayan aparecido nuevos escenarios en los cuales se pueden aprovechar las características de banda ancha, bajo coste y fácil despliegue de éstas. En base a la experiencia adquirida con esta tecnología dentro de este grupo de investigación, se propone el uso de las WLAN para el despliegue de

Tabla 1: Características de la tecnología IEEE 802.11b

	IEEE 802.11b
Fecha de aprobación	Sept. 1999
Ancho de banda disponible	83.5 MHz
Frecuencia de trabajo	2.4–2.4835 GHz
Número de canales no interferentes	4
Tasa binaria por canal	1, 2, 5.5, 11 Mbps
Modulación	DSSS

redes de acceso en dos escenarios bien distintos. El primero de ellos se trata del despliegue de un Hot-Spot y el segundo se corresponde con el despliegue de una red de acceso en un área rural como extensión a una red de acceso convencional.

2.1 Despliegue de un Hot-Spot

En general, un Hot-Spot puede definirse como una red de acceso público que se despliega sobre espacios por los que pasan una gran cantidad de usuarios, los cuales generan un alto volumen de tráfico agregado. Pero sin duda, la mejor forma de definir un Hot-Spot es mediante un ejemplo. Hoteles, centros de convenciones, estaciones de tren, aeropuertos, etc son los escenarios típicos en los que se dispone generalmente de enlaces de datos de gran capacidad, ya sea a través de cable, xDSL (x Digital Subscriber Line) o enlaces E1. Mediante una red cableada se lleva la red a diversos lugares estratégicos en los que se colocan un conjunto de Puntos de Acceso (PA), dotando de esta forma de cobertura radio a todo el área que se desea cubrir. En la Fig. 1 se ilustra el despliegue en vertical de un Hot-Spot, por ejemplo un hotel.

Existen herramientas software que permiten planificar redes inalámbricas en interiores bastante costosas por lo general. Sin embargo, estos estudios pueden realizarse de forma sencilla considerando dos aspectos principalmente: el área de cobertura radio y los requerimientos en cuanto a calidad de servicio (i.e. ancho de banda disponible,...) que se va a ofertar a los usuarios.

Para el primero de los aspectos, es necesario realizar un estudio de la estructura del edificio a partir de sus planos, identificando a priori los puntos donde se van

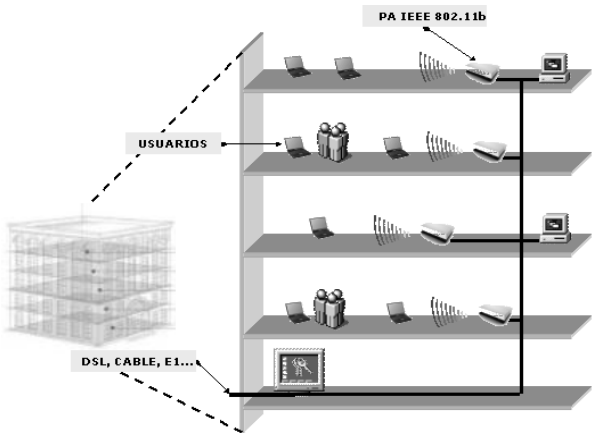


Figura 1: Arquitectura de un Hot-Spot (ejemplo de un edificio)

a colocar los PA, de forma que se consiga la cobertura deseada. Con objeto de verificar esta primera aproximación, se realiza posteriormente una campaña de medidas. Para ello, se colocan los PA en los lugares seleccionados y se comprueba que la relación señal a ruido (SNR, *Signal to Noise Ratio*) medida en cada punto del área a cubrir sea suficientemente elevada, para que no se degrade el rendimiento de las comunicaciones debido a los errores del canal radio. En cuanto a lo que se refiere a la planificación de la calidad de servicio, es necesario tener en cuenta el número de usuarios potenciales que se conectarán a un único PA. Si el número es elevado, puede ser necesario duplicar las capacidades colocando nuevos PA. Para ello, cada PA puede utilizar un canal diferente, utilizando aquellos mutuamente no interferentes que, como se indica en la Tabla 1, son cuatro. No podemos olvidar que, en general, la capacidad de nuestro Hot-Spot viene dada por el ancho de banda que se tiene en el enlace de datos, ya que el ancho de banda por usuario disponible en la red de acceso será normalmente superior al que se tendrá en el tubo de salida.

2.2 Planificación de una red de acceso en un área rural

Un gran número de operadores de cable están considerando usar equipos en la banda ISM de 2.4 GHz para proporcionar acceso a servicios de datos en áreas rurales, donde los núcleos de población se encuentran dispersos y el factor de penetración de estos servicios es muy bajo, haciendo inviable el cableado de fibra óptica o el uso de tecnologías como LMDS *(Local Multipoint Distribution System)*. Si a estos condicionantes se le añaden, como se ha dicho anteriormente, la posibilidad de operar sin necesidad de licencia en la citada banda y el bajo coste de los equipos de usuario, IEEE802.11b se convierte en la opción tecnológica más atractiva para las redes de acceso de estas características.

En este apartado se van a tratar los temas más relevantes cuando se desea planificar una red de acceso en un área rural. Se comienza con una introducción de conceptos básicos de propagación, que son los que se van a tener en cuenta en la planificación, puesto que se trata de una red de acceso radio. A continuación, se propone una arquitectura de red flexible y escalable, capaz de adaptarse a cualquier entorno rural. Por último, se presentan los resultados obtenidos en las medidas de rendimiento, interferencias, etc, que se hicieron en una maqueta de pruebas que se montó en la Universidad de Cantabria con objeto de probar la viabilidad de la solución propuesta y poder así contrastar los resultados obtenidos. En la actualidad, esta solución ha sido probada en entornos reales y su funcionamiento es el esperado.

Conceptos de propagación

El hecho de que en la banda de 2.4 GHz no se necesite licencia para operar no implica que no

existan unas limitaciones de espectro y potencia. El rango de frecuencias asignado para dicha banda se extiende desde 2.400 GHz hasta 2.485GHz. Los equipos que pueden adquirirse en España cumplen esta normativa, por tanto, no se hará hincapié en ella. No ocurre lo mismo con la limitación de potencia, 10 mW de Potencia Isotrópica Radiada Equivalente (PIRE), que, si bien la potencia transmitida por los equipos no sobrepasa dicho límite, el uso de antenas de elevada ganancia hace que en muchas ocasiones se sobrepase la PIRE indicada. Esta limitación en potencia hace que no se pueda pensar en zonas de cobertura ilimitada.

Teniendo en cuenta las especiales características de este tipo de entornos, se asume el modelo de propagación de tierra plana como modelo válido para el cálculo de atenuaciones. En él, sólo se tienen en cuenta las pérdidas de propagación en espacio libre, asegurándose que en todos los puntos entre el emisor y el receptor, se salva el radio de la primera zona de Fresnel. De este modo se evitan las posibles pérdidas por difracción.

La atenuación de la señal debida a las pérdidas de propagación en la banda de 2.4 GHz se obtiene utilizando la siguiente fórmula:

$$Atenuación\ (dB) = 100\,dB + 20\log(d_{km})$$

Sustituyendo el valor de las pérdidas de propagación y la potencia recibida por la sensibilidad del receptor en la ecuación de balance del enlace, se puede encontrar el límite superior para la distancia entre un transmisor y un receptor.

$$Pot_{RX}(dBm) = PIRE(dBm) - Atenuación(dB) + G_{RX}(dBi)$$

Arquitectura de la red

Dependiendo de la tecnología empleada para la interconexión con la red troncal, pueden distinguirse dos casos diferentes: (1) cuando se realiza mediante satélite, se emplaza una estación VSAT (*Very Small Aperture Terminal)* en un punto estratégico del casco urbano desde donde se despliega la subred de acceso que da cobertura a los usuarios del mismo; y (2) cuando mediante enlaces LMDS o radioenlaces digitales se llega a puntos situados fuera de los núcleos urbanos. Desde estos emplazamientos, es necesario realizar el despliegue de una subred de distribución para llegar hasta los núcleos de población. Como ocurre en el primer caso, en los puntos de terminación de la subred de distribución se realiza el despliegue de las subredes de acceso. La arquitectura de red que se describe en este artículo se corresponde con este último caso, por ser el más completo de ambos.

La topología de la red propuesta se ilustra en la Fig. 2. En ella se distinguen dos clases de subredes: la de distribución y la de acceso. La primera está formada por equipos del fabricante canadiense Wi-LAN [5], y

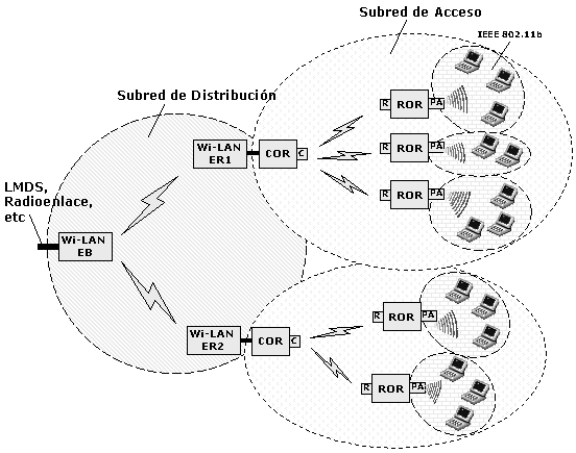


Figura 2: Arquitectura de Red de Acceso en un entorno rural

la segunda por equipos de AVAYA [6]. Todos ellos son equipos de bajo coste que trabajan en la banda de 2.4GHz, lo que asegura un rápido despliegue de red. A continuación se describen detalladamente los elementos y parámetros de configuración más relevantes de cada una de ellas.

La subred de distribución utiliza los equipos Wi-LAN Hopper Plus 120-24. Uno de ellos configurado para que trabaje como Estación Base (EB) y el resto como Estaciones Remotas (ER), trabajando todos en la misma frecuencia central. Esta frecuencia se elige seleccionando uno de los siete canales de 33 MHz definidos por el fabricante (véase Tabla 2). Esta canalización no coincide con la especificada en IEEE 802.11b, hecho que debe considerarse al realizar la elección de frecuencias de los equipos de AVAYA, para evitar las posibles interferencias.

Con objeto de que el haz de radiación se superponga lo menos posible con el generado por el equipamiento de AVAYA que se encuentra situado junto a las ER, la antena de la EB se elige con sectorización de 120º. Para las ER se eligen antenas que sean directivas a fin de confinar la potencia radiada, minimizando así la interferencia cocanal con los equipos de AVAYA que se implantan en estos puntos y que forman parte de la subred de acceso. Por la misma razón, se utilizan polarizaciones ortogonales entre ambas infraestructuras, planificando el equipamiento de la subred de distribución con polarización horizontal y el de la subred de acceso con polarización vertical.

La subred de acceso está formada por el sistema de Routers de Exteriores (OR, *Outdoor Routers*) inalámbricos de AVAYA. Existen tres tipos de OR: el Router Central (COR, *Central Outdoor Router*), el

Tabla 2: Tabla de frecuencias centrales de los equipos Wi-LAN

Canal	Frecuencia
1	2.4258 GHz
2	2.4302 GHz
3	2.4345 GHz
4	4.4400 GHz
5	2.4455 GHz
6	2.4498 GHz
7	2.4542 GHz

Router Remoto (ROR, *Remote Outdoor Router*) y el Router Cliente (ORC, *Outdoor Router Client*). El COR es el elemento central, como su propio nombre indica, y a él pueden conectarse tanto los RORs como los ORCs, utilizando un protocolo propietario de AVAYA. Los RORs son los elementos encargados de extender la cobertura a puntos remotos de la subred de acceso, distribuyendo la señal del COR directamente o actuando como repetidores inalámbricos para llevar dicha señal a puntos que no son alcanzables mediante un enlace COR-ROR. Por último, los ORC son terminales de usuario que utilizan tarjetas IEEE 802.11b, las cuales mediante software emulan el protocolo propietario que les permite conectarse directamente al COR.

Tanto los COR como los ROR tienen tres interfaces: una Ethernet y las otras dos PCMCIA en las que se insertan sendas tarjetas IEEE 802.11b. En un COR, estas interfaces inalámbricas pueden configurarse como COR o como PA. En un ROR, éstas pueden configurarse con funcionalidad de ROR maestro o esclavo (cuando se configura como repetidor), o de PA.

En la arquitectura propuesta, cada una de las subredes de acceso está formada por un COR y varios ROR que se conectan directamente al COR. Cada ROR tiene configurada la otra interfaz PCMCIA como PA, dando cobertura de esta forma a los usuarios de la zona y posibilitándoles el *roaming* entre los distintos PA. Todos los PA cumplen con las especificaciones WiFi (*Wireless Fidelity*) [7], por lo que cualquier usuario con un cliente IEEE802.11b puede conectarse a la red.

Debido al hecho de que las ER de Wi-LAN y los COR de AVAYA se ubican en el mismo emplazamiento (conectados entre si mediante la interfaz Ethernet de cada uno), es necesario hacer una planificación de frecuencias de forma que se minimicen los efectos de interferencia que se producen. Por ello, se elige para las ER el canal 1 de Wi-LAN (véase Tabla 2) y el canal 13 de IEEE 802.11b (véase Tabla 3) para los COR, consiguiéndose una separación entre canales adyacentes de 18.7 MHz. En estas condiciones y desplazando los planos de ambas antenas se ha comprobado que la probabilidad de error (P_b) se reduce por debajo de 10^{-4}.

Tabla 3: Conjunto de canales para IEEE 802.11b DDSS

	Europa	Francia	Japón	EE.UU
Canal	Frec. (Ghz)	Frec. (Ghz)	Frec. (Ghz)	Frec. (Ghz)
1	2.412	-	2.412	2.412
2	2.417	-	2.417	2.417
3	2.422	-	2.422	2.422
4	2.427	-	2.427	2.427
5	2.432	-	2.432	2.432
6	2.437	-	2.437	2.437
7	2.442	-	2.442	2.442
8	2.447	-	2.447	2.447
9	2.452	-	2.452	2.452
10	2.457	2.457	2.457	2.457
11	2.462	2.462	2.462	2.462
12	2.467	2.467	2.467	-
13	2.472	2.472	2.472	-

Cada COR está equipado con una antena omnidireccional, de ganancia típica 7dBi, y cada ROR con una antena directiva, de ganancia de 14 dBi. La potencia de la tarjeta de AVAYA situada en el COR es de 15 dBm, y la potencia de las tarjetas que se insertan en cada ROR es de 8 dBm. La conexión de las tarjetas con las antenas se realiza mediante un *pigtail*, un extremo de éste se conecta a las tarjetas y en el otro se suele colocar un *surge arrestor* para evitar daños en las tarjetas debido a las descargas. La antena se une con el *pigtail* mediante cable coaxial, en nuestro caso se utiliza el LMR400. Teniendo en cuenta las atenuaciones introducidas por los cables y los conectores, nunca se sobrepasará el límite de 20 dBm de PIRE.

La sensibilidad de las tarjetas depende de la tasa binaria del medio radio, siendo, -83dBm/-87dBm/-91dBm/-94dBm para regímenes de 11Mbps/5.5Mbps/2Mbps/1Mbps respectivamente, con P_b inferior a 10^{-5} . El margen de desvanecimiento recomendado para estos equipos, a fin de no degradar la P_b, es del orden de los 10 dB. En la Tabla 4 se muestran las distancias típicas de separación entre un COR, con antena omnidireccional, y los ROR, con antenas directivas, para las distintas regulaciones y ganancias de antena.

No se recomienda que haya más de 6 RORs por cada COR, puesto que, como veremos posteriormente en las medidas realizadas, el caudal eficaz a nivel FTP con un sólo enlace COR-ROR es de 3.2 Mbps, para velocidades de las tarjetas de 11Mbps.

En base a medidas de interferencias realizadas con las tarjetas IEE802.11b y como se muestra en la Tabla 1, el número máximo de canales no interferentes son cuatro, los canales 1, 5, 9 y 13. Para los enlaces COR-ROR se había elegido el canal 13, por ello, las interfaces con funcionalidad de PA deben configurarse en canales del 1 al 9. En PA con bastante proximidad geográfica hay que volver a tenerlo en cuenta, evitando de esta forma las posibles interferencias en las zonas de solapamiento. En aquellas zonas que se encuentren lejanas, pueden reasignarse frecuencias para los PA.

Plataforma de pruebas

En el campus de la Universidad de Cantabria se montó una maqueta de pruebas con objeto de verificar las capacidades de los equipos seleccionados. La topología de la misma se ilustra en la Fig. 3 y consiste en una EB basada en equipo de

Tabla 4: Distancias alcanzables en exteriores

			FCC		*ETSI*
Velocidad	**Sensibilidad**	**Velocidad**	**14 dBi**	**24 dBi**	**14 dBi**
11 Mbps	-83 dBm	11 Mbps	3.5 Km	8.5 Km	1.2 Km
5.5 Mbps	-87 dBm	5.5 Mbps	5 Km	10 Km	1.9 Km
2 Mbps	-91 dBm	2 Mbps	6.5 Km	12 Km	2.5 Km
1 Mbps	-94 dBm	1 Mbps	8 Km	14 Km	3.7 Km

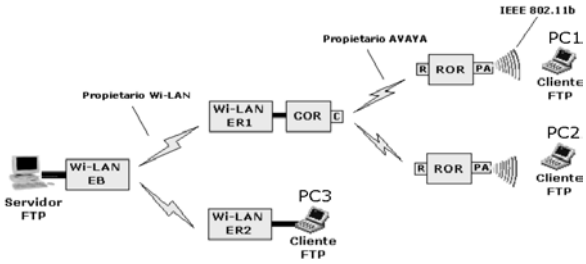


Figura 3: Maqueta de pruebas de la Universidad de Cantabria

WiLAN y dos ER del mismo fabricante, emplazadas todas ellas en edificios distintos. Una de las ER, la ER1, está interconectada con un COR de AVAYA, emulando de este modo un punto remoto de la subred de distribución. Para emular la subred de acceso se emplean dos ROR a los que se conectan sendos PCs a través de sus correspondientes PA. La otra ER, la ER2, se conecta a otro PC. Por último, la EB está conectada a otro PC que actúa como servidor FTP de los distintos PC remotos (PC1, PC2 y PC3).

Inicialmente se llevó a cabo una campaña de medidas para caracterizar el comportamiento de los equipos de la subred de distribución. Ésta se realizó con una EB y varias ER. Cabe destacar que, entre los parámetros de configuración de los Hopper de Wi-LAN que trabajan como ER, existe uno que se denomina *throttle* que permite fijar el caudal máximo para el enlace con la EB. Modificando dicho parámetro es posible distribuir el ancho de banda entre los distintos enlaces en caso de que existan diferentes necesidades en alguno de ellos.

En la Tabla 5 se recogen los resultados que se derivan del conjunto de medidas realizadas en una conexión punto a punto, en las que un PC que se conecta a una ER establece una sesión FTP (*File Transfer Protocol*) con el servidor conectado a la EB, para distintos valores de *throttle*.

Siguiendo las mismas pautas que en la medida anterior, se realizaron otras medidas con una EB y varias ER, comprobándose que el caudal eficaz se repartía entre ambas cuando el *throttle* estaba fijado al máximo.

Posteriormente se realizó otra campaña para caracterizar los equipos de la subred de acceso. La razón de estas medidas es principalmente conocer la cota máxima del caudal esperado y, de este modo, poder evaluar la potencial reducción de dicho caudal, que puede derivarse del hecho de concatenar posteriormente la infraestructura de Wi-LAN. La primera prueba se realizó comunicando un PC conectado a un ROR con otro conectado a la interfaz Ethernet de un COR y realizando una transferencia de un archivo del uno al otro. Configurando las tarjetas

Tabla 5: Caudal eficaz obtenido con configuración punto a punto

Throttle	Caudal
Máximo	7.68 Mbps
128kbps	128 Kbps
1.28Mbps	1.28 Mbps

AVAYA tanto en el ROR como en el COR para una tasa binaria de 2 Mbps (estándar) se obtuvo un rendimiento neto de 1.4 Mbps.

Para validar este resultado, se realizaron el mismo tipo de medidas con una configuración ROR-COR-ROR, esto es, dos PCs conectados con sus correspondientes ROR que se comunican a través del COR obteniéndose un rendimiento de 634,8 Kbps. La velocidad se ha reducido prácticamente a la mitad, como cabía esperar, puesto que ahora son dos los ROR que comparten el recurso radio.

Indicar que se realizaron medidas similares a las anteriormente descritas, configurando las tarjetas de COR y ROR a 11Mbps alcanzándose 3.2 Mbps como caudal máximo de la transferencia FTP.

Utilizando la topología de pruebas de la Figura 3, se realizaron varias medidas considerando distintos parámetros de *throttle* en los equipos de Wi-LAN y de AVAYA. Al igual que en las medidas anteriores, los valores de caudal reflejados en la Tabla 6 son los que se obtuvieron al establecer diferentes sesiones FTP entre los PC (PC1, PC2 y PC3) y el servidor FTP conectado a la EB.

A tenor de los resultados obtenidos en las campañas de medidas realizadas, podemos afirmar que nos encontramos ante una red de acceso inalámbrica bastante flexible y escalable. Que si bien no está diseñada para dar cobertura a un gran número de usuarios, es ideal para entornos rurales, donde los núcleos de población se encuentran bastante dispersos y donde el factor de penetración es generalmente muy bajo. Este tipo de soluciones representan una alternativa a tener en cuenta por muchos de los operadores de cable que en su día se comprometieron a dar cobertura en estas zonas.

3 Herramienta para la gestión de WLAN públicas

Hasta el momento se han abordado aspectos relativos al diseño y planificación de redes de acceso públicas basadas en tecnología WLAN, en las que cualquier entidad o persona puede ofertar servicios de acceso a Internet. No puede olvidarse que, por el hecho de tratarse de una red pública, es necesario controlar y gestionar el uso de la misma, esto es, control de acceso de usuarios, tarificación, etc. Por ello, con objeto de poder presentar una solución WLAN completa para el despliegue de estas redes, es necesario ofrecer una herramienta que permita

Tabla 6: Medidas de rendimiento Avaya + Wi-LAN

Clientes	Throttle Wilan	Throttle Avaya	Caudal
PC1	Máximo	Máximo	1,4 Mbps
PC1	1.536 Mbps	Máximo	1,4 Mbps
PC1	1.536 Mbps	64 Kbps	62'260 Kbps
PC1	1.536 Mbps	128 Kbps	126 Kbps
PC1, PC2	2 Mbps	64 Kbps	62/62 Kbps
PC1, PC2	2 Mbps	512 Kbps	504/504 Kbps
PC1,PC2,PC3	1.536 Mbps	510Kbps	380/485/1530 kbps

realizar dicha gestión. En este sentido, dentro de nuestro grupo se ha desarrollado una herramienta software conocida como Hot-Spots de la Universidad de Cantabria (HSUC), que ofrece todas las facilidades necesarias para la provisión de servicios de Internet a través de redes de acceso públicas.

En este apartado, se enumeran los funcionalidades funcionales que debe implementar una herramienta de este tipo, se propone la arquitectura utilizada por la HSUC y, finalmente, se describe su funcionamiento.

3.1 Funcionalidades de la HSUC

Los requerimientos funcionales de una herramienta que permita gestionar redes de acceso públicas son diversos. Con objeto de poderlos estructurar de alguna forma, los dividimos en dos grupos: los relacionados con el control de acceso de usuarios y los relacionadas con la gestión puramente económica. Siguiendo este criterio, las funcionalidades que implementa la herramienta HSUC son las siguientes:

Funcionalidades para el control de acceso

- Arquitectura abierta y escalable
- Acceso a la red independiente de la tecnología radio empleada (IEEE 802.11b, bluetooth, etc) e incluso posibilidad de acceso mediante Ethernet.
- Protección contra intrusos y restricción de acceso a Internet a usuarios no permitidos.
- Usuarios de contrato y de prepago.
- Identificación de usuarios en el sistema basada en web protegida mediante SSL (*Secure Socket Layer*).
- Autenticación mediante RADIUS (*Remote Authentication Dial In User Service)* .
- Asignación dinámica de direcciones IP *(Internet Protocol)* mediante DHCP *(Dynamic Host Configuration Protocol).*

Funcionalidades para la gestión del sistema

- Sistema de facturación centralizada que contabiliza el uso de los recursos de red que ha hecho cada cliente.
- Generación automática de facturas con su correspondiente detalle de consumos.
- Consulta *on-line* del consumo actual y de facturas anteriormente emitidas.
- Planes de precios para los distintos perfiles de usuarios de contrato.
- Gestión de planes de precios: creación, modificación y borrado.
- Planes basados en una cuota mensual y facturación por volumen de tráfico. Posibilidad de discriminar entre UDP *(User Datagram Protocol)* y TCP *(Transmission Control Protocol).*
- Dar de alta/baja a usuarios de contrato en el sistema.
- Modificar datos de usuario *on-line*.
- Crear bonos con límite de tiempo para usuarios de prepago.
- Impresión de bonos de prepago.
- Control del estado de los diferentes bonos.
- Gestión de contraseñas.
- Estadísticas y gráficos del tráfico en la red
- Sincronización automática del reloj del sistema
- Independiente de la tecnología de red

3.2 Arquitectura de la HSUC

En este apartado se propone una arquitectura de bajo coste y rápido despliegue válida para todo tipo de redes de acceso inalámbricas. En la Fig.4 se muestra el ejemplo de un Hot-Spot, pero esta propuesta es igualmente viable para cualquier red de acceso que haya a la izquierda del Servidor de Acceso a la Red (SAR). Como puede observarse, la inteligencia de la herramienta HSUC reside en dos servidores: el SAR y el Servidor de Gestión (SG). El primero se encarga de permitir el acceso a Internet a aquellos usuarios dados de alta en el sistema, mientras que el segundo es la entidad central, en la que residen la mayoría de funcionalidades de gestión de la herramienta: autenticación, tarificación, gestión del uso de la red, ...

La HSUC es una solución software desarrollada en máquinas con sistema operativo Linux, por tanto, una plataforma abierta que hace que el coste de licencias y mantenimiento sea prácticamente nulo.

A continuación se detallan las principales características de los dos servidores:

Características del SAR

- La red de acceso puede estar formada por PA de cualquier tecnología radio (Bluetooth, IEEE802.11b) permitiendo incluso Ethernet.
- Direccionamiento mediante servidor DHCP.
- Funciones de *proxy* mediante servidor NAT *(Network Address Translation).*
- Total transparencia para el usuario final, esto es, éste puede acceder al sistema sin necesidad de instalar ningún software.
- Servidor HTTPS *(Secure HyperText Transfer Protocol)* para la identificación de usuarios.

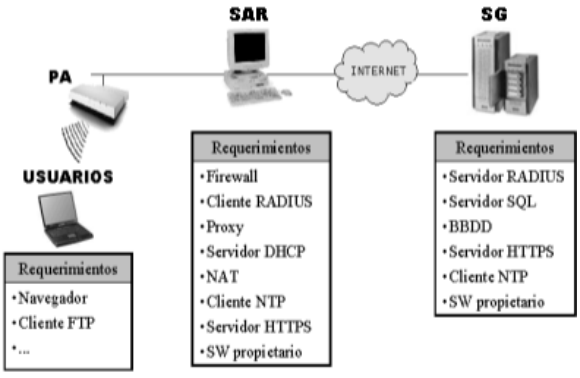


Figura 4: Arquitectura y requerimientos de la herramienta HSUC

- Redireccionamiento automático a la página de inicio, una vez iniciada la sesión.
- Servidor de Nombres de Dominio DNS.
- Funciones de *firewall* para proteger al sistema frente a intrusos y accesos no deseados
- Cliente RADIUS para la AAA (*Authentication, Authorization & Accounting*).
- SNMP *(Simple Network Management Protocol)* para la gestión y monitorización de los PA.

Características del Servidor de Gestión

- Servidor HTTPS.
- Servidor de Bases de Datos (BBDD).
- Servidor RADIUS para la AAA.
- *Firewall* de protección contra intrusos y restricción de acceso a Internet a usuarios que no estén registrados. Posibilidad de poner filtros en función de los perfiles de usuario.
- Aplicación de gestión controlada mediante interfaz web que permite tener acceso al sistema de facturación (generación y chequeo de facturas), al sistema de gestión de clientes (crear, modificar y eliminar cuentas de usuarios tanto de prepago como de contrato) y a las estadísticas de uso de la red (de cada usuario o del total de los usuarios).

3.3 Funcionamiento de la HSUC

En este apartado se describe el funcionamiento de la herramienta HSUC mediante un ejemplo en el que un usuario accede a Internet a través de un Hot-Spot gestionado con dicha herramienta.

Acceso a la red
Cuando el usuario enciende su equipo (portátil, agenda portatil, ...), éste se asocia al PA 802.11b con el que tiene mejor SNR. La interfaz de red inalámbrica debe estar configurada para obtener la dirección IP mediante el protocolo DHCP. De esta forma, el SAR le asigna dinámicamente una dirección disponible dentro del rango establecido en el mismo. Este proceso se realiza automáticamente de forma totalmente transparente al usuario, permitiendo a partir de este momento que el usuario pueda identificarse en el sistema.

Inicialmente cualquier petición de conexión al exterior de la red de acceso está bloqueada por el SAR. De esta forma, un usuario no puede establecer ninguna comunicación con servidores situados fuera de la red de acceso sin que esté identificado en el sistema. Sin embargo, existe la posibilidad de colocar enlaces a páginas blancas (ej. publicidad, servicios del hotel, etc.) en el servidor web del SAR, que son accesibles de forma totalmente gratuita. Estos enlaces se encuentran accesibles desde la página de *login*.

Proceso de Login
Este proceso se realiza a través de una aplicación basada en Web, sin necesidad de que el usuario tenga instalado ningún software especial. El único requerimiento para éste es que disponga de un navegador en su equipo. Con objeto de simplificar el proceso y evitar que el usuario tenga que recordar e introducir la dirección de la página de *login*, cada vez que desee conectarse, se le redirige automáticamente a ésta cuando realice la primera petición de una página web. En ella, el usuario debe introducir los datos de nombre de usuario y contraseña en un formulario y pulsar el botón de envío. Para dotar de seguridad al envío se cifra mediante un *hash* MD5 la contraseña y la transferencia de datos se realiza mediante HTTPS. El SAR recibe los datos y los envía al SG utilizando primitivas del protocolo RADIUS[8]. El SG comprueba en su BBDD si el usuario está registrado en el sistema y en caso de que el resultado de la comprobación sea positivo, se desencadena el proceso de inicio de sesión. En caso contrario, se le devuelve un mensaje de error al usuario.

Inicio de sesión
Una vez que el SG ha comprobado que el usuario puede acceder al sistema, le envía los permisos que tiene asignados él usuario al SAR, el cual almacena el nombre, la dirección MAC *(Médium Access Control)* y la dirección IP en un nuevo registro de la tabla de usuarios conectados. De esta forma se evita que dos usuarios con el mismo identificador puedan estar conectados a la vez, ya que, cada vez que un usuario nuevo intenta acceder al sistema, el SAR comprueba esta tabla para ver si está conectado. El inicio de la sesión continúa con la eliminación por parte del SAR del bloqueo de conexiones hacia el exterior y con la redirección a la página solicitada originariamente por el usuario. Finalmente, en el lado de éste se descarga una aplicación basada en tecnología Java que permite al usuario ver la información relativa a su sesión, es decir, se le mantiene periódicamente informado del tiempo de conexión y del volumen de tráfico que ha generado y recibido. De esta forma, un usuario puede controlar el gasto de su sesión. Junto a esta información, la aplicación tiene dos botones adicionales, uno para actualizar los datos de sesión en un momento puntual y el otro para solicitar el fin de la sesión.

Mantenimiento y cierre de sesión.
Finalizado el proceso anterior, el usuario ya puede disfrutar de su acceso a Internet y es en este momento cuando el SAR y el SG comienzan a realizar el control del uso de la red por parte de éste. El control se realiza en dos niveles: por un lado se contabiliza el tiempo de conexión y por otro el tráfico enviado y recibido. Esto permite que existan dos tipos de usuarios en el sistema: los de prepago y los de contrato. Los primeros son usuarios que adquieren un bono de un determinado tiempo mientras que los segundos son usuarios de contrato que reciben una factura del volumen de datos generados/recibidos. En ella se puede comprobar los detalles de consumo y tiempo de cada conexión realizada. Para ello, el SAR se encarga de mantener el control de los usuarios que están conectados a la red en todo momento. Cuando un usuario finaliza su sesión, el tiempo de conexión y

la cantidad, en bytes, de datos transmitidos y recibidos (se diferencia entre tráfico TCP y UDP) se almacenan en la BBDD del SG.

El cierre de una sesión puede deberse a tres opciones: que el usuario cierre su sesión, que el sistema le desconecte cuando ha estado durante un tiempo inactivo o que finalice el tiempo del bono contratado. La primera de las opciones es la más general, en ella el usuario utiliza la aplicación JAVA que refleja la información de la sesión para desconectarse. La segunda se produce cuando un usuario abandona la red en el transcurso de una sesión sin haber solicitado explícitamente el cierre de la misma (ej. apaga su equipo sin haber pulsado el botón de fin de sesión). Pasado un tiempo, el SAR detecta que este equipo ha abandonado la red y procede a cerrar su sesión automáticamente. La última es aplicable sólo en el caso de usuarios de prepago. En este caso, cuando el SAR detecta que ha finalizando el tiempo que un usuario tiene en su bono, le ofrece la posibilidad de introducir los datos de uno nuevo, o en su defecto, cuando expire su tiempo lo desconecta automáticamente.

En cualquiera de los tres casos anteriores el proceso de cierre es el mismo: el SAR bloquea nuevamente el acceso al exterior para ese usuario y envía los parámetros de la sesión mediante primitivas RADIUS al SG, que lo almacena en su BBDD.

4 Conclusiones

Cada vez más, los usuarios de las tecnologías de la información necesitan trabajar fuera de sus lugares habituales. El ancho de banda que ofrecen las redes móviles de 2G a estos usuarios es limitado y teniendo en cuenta el retraso que llevan la implantación de las redes de 3G, es necesario buscar soluciones competitivas que permitan a los usuarios seguir realizando su trabajo.

La madurez de la tecnología basada en el estándar IEEE 802.11b, hace que sea una de las más atractivas para ofrecer este tipo de soluciones. El hecho de que en la banda de 2.4GHz no se necesite licencia para operar hace que éstas sean soluciones de bajo coste y rápido despliegue. Esto también presenta inconvenientes, pues la banda de trabajo es limitada y, si varios operadores intentan ofrecer estos servicios en la misma zona, pueden aparecer problemas de interferencias.

Un ejemplo de aplicación de las WLAN son los Hot-Spot en los que se ofrece servicio de acceso público a Internet a usuarios móviles. Escenarios de este tipo los podemos ver en las salas VIP de los grandes aeropuertos o en hoteles, donde el servicio se ofrece a través de PA IEEE 802.11b. Puesto que estas soluciones están experimentando un gran auge, se ha intentado clarificar cómo planificar una red de acceso en un escenario de este tipo.

Los operadores de cable están considerando utilizar tecnologías WLAN para ofrecer servicios de acceso a Internet en zonas rurales, ya que, dadas las especiales características de estos entornos, otras soluciones resultan económicamente inviables. Además de las consideraciones puramente económicas, en zonas que pueden ser consideradas patrimonio histórico-artístico, esta tecnología ofrece un impacto realmente reducido debido al menor tamaño del equipamiento necesario en comparación con otras tecnologías que se barajan para la solución de este problema.

Una característica común a ambas aplicaciones de las WLAN es que se trata de redes de acceso públicas. Con objeto de poder ofrecer una solución completa para esta clase de redes, es necesario introducir una herramienta software que permita explotarlas económicamente, esto es, controlar el acceso de usuarios y cobrar por su uso de forma centralizada. Esta es la labor que permite realizar la herramienta HSUC desarrollada en el seno de nuestro grupo de investigación. Gracias a la flexibilidad y escalabilidad de dicha herramienta es posible controlar con ella cualquier conjunto heterogéneo de redes de acceso públicas.

Agradecimientos

Los trabajos expuestos en este artículo han sido realizados por varias personas del Grupo de Ingeniería Telemática de la Universidad de Cantabria. Por ello, los Autores quieren agradecer el esfuerzo a todos aquellos que lo han hecho posible.

Referencias

[1] IEEE 802.11, “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification,” 1997.

[2] IEEE 802.11b, “Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification: High-Speed Physical Layer Extension in the 2.4 GHz Band,” 1999.

[3] R. van Nee, G. Awater, M. Morikura, H. Takanashi, M. Webster and K. Halford, “New High-Rate Wireless LAN Standard,” *IEEE Communications Magazine,* Diciembre 1999, pp. 82-88.

[4] M. García, R. Agüero, L. Muñoz, P. Mähönen, “Behavior of UDP-Based Applications over IEEE 802.11 Wireless Networks”, PIMRC 2001, San Diego (USA), Octubre 2001. Vol II, pp 72-77.

[5] Wi-LAN: http://www.wilan.com

[6] AVAYA: http://www.avaya.com

[7] The Wi-Fi Alliance: www.wirelessethernet.com

[8] RFC 2865 “Remote Authentication Dial in User Service (RADIUS)” C. Rigney, S. Willens Livingston, A. Rubens Merit, W. Simpson Daydreamer. Junio 2000.

Análisis del protocolo IEEE 802.11b en un entorno celular

Elena Lopez-Aguilera, Jordi Casademont, Alfonso Rojas
elopez@entel.upc.es, jordi.casademont@entel.upc.es, alfonso@entel.upc.es
Departamento de Ingeniería Telemática
Universidad Politécnica de Cataluña

Abstract*. In this paper we present an analysis of the protocol IEEE 802.11b in different environments. Its performance in an indoor environment is evaluated, in terms of throughput and transmission delay, and the influence of the beacon period is analyzed. On the other hand, a possible technique to improve the throughput and transmission delay is exposed, and the importance of the contention windows value is studied to improve performance in presence of asymmetric data traffic. Finally the protocols characteristics in outdoors environments are explained and its performance is studied, in terms of throughput and transmission delay.*

1 Introducción

En 1999 IEEE (The Institute of Electrical and Electronics Engineers) definió el primer estándar, IEEE 802.11 [1], para la regulación de las redes de área local inalámbricas. Este estándar proporciona velocidades de transmisión entre 1 y 2 Mbps, en la banda de los 2,4 GHz, y soporta diferentes medios de transmisión: por infrarrojos y por radiofrecuencia. En este último caso se tiene dos tipos de transmisiones por espectro ensanchado: Frequency Hopping Spread Spectrum (FHSS) y Direct Sequence Spread Spectrum (DSSS). Desde su nacimiento, dicho estándar ha evolucionado dando lugar a diferentes estándares. Por un lado se encuentra el IEEE 802.11b [2], que permite velocidades de transmisión de 1, 2, 5,5 y 11 Mbps en la misma banda de frecuencias, utilizando radiofrecuencia con tecnología DSSS. Por otro lado apareció IEEE 802.11a [3], que alcanza velocidades de transmisión de 54 Mbps, pero en la banda de frecuencia de 5 GHz. Sin embargo, IEEE 802.11g pretende también conseguir velocidades de hasta 54 Mbps, pero en la banda de los 2,4 GHz. Finalmente IEEE 802.11e supone la incorporación de calidad de servicio en el protocolo de acceso al medio, por parte del grupo de trabajo IEEE 802.11.

El estándar soporta dos tipos de topologías: redes con infraestructura y redes ad-hoc.

En las redes con infraestructura se tiene una estación que actúa como punto de acceso, coordinando el comportamiento de la red, ya que todas las comunicaciones entre estaciones deben pasar por él. Esta topología puede funcionar bajo dos mecanismos: en modo DCF (Distributed Coordination Function) o PCF (Point Coordination Function). El primero de ellos se basa en el protocolo CSMA/CA (Carrier Sense Medium Access/Collision Avoidance), y presenta dos técnicas para la transmisión de paquetes: el modo básico y el RTS/CTS. Por otro lado, el mecanismo PCF utiliza una estación de la red como coordinadora de las demás estaciones. Dicha estación será la encargada de dar permisos a las demás, para que éstas puedan transmitir datos.

Finalmente, las redes ad-hoc están formadas por un grupo de estaciones inalámbricas, que comparten una misma área de cobertura y operan en modo DCF. Estas estaciones, a través de sus tarjetas de red inalámbrica, forman una red de área local. Estas redes se caracterizan por su gran simplicidad, pero presentan algunas desventajas, como son su limitada cobertura y el hecho de tratarse de redes aisladas, sin posibilidad de comunicación con otras.

Hasta el momento todos los estudios realizados en torno a IEEE 802.11 están dirigidos a entornos interiores unicelulares. Sin embargo la idea de diseñar una red celular basada en el estándar IEEE 802.11 resulta muy atractiva, dado el bajo coste de la tecnología requerida y la utilización de la banda de 2.4 GHz. Además IEEE 802.11b proporciona una velocidad de transmisión de hasta 11 Mbps, muy por encima de la ofrecida por EDGE (Enhanced Data Rates for GSM Evolution) y W-CDMA (Wide Code Division Multiple Access).

En este artículo limitamos nuestra investigación a las redes con infraestructura, operando concretamente bajo el esquema DCF en modo básico. Además de exponer y analizar la problemática presente en un escenario unicelular, también se estudia el comportamiento del protocolo de acceso al medio IEEE 802.11b en un escenario de múltiples celdas.

El artículo se distribuye como se explica a continuación. En la sección 2 se realiza un repaso del esquema DCF operando en modo básico. En la 3, se describen los parámetros considerados en las simulaciones realizadas para el análisis de la capa de acceso al medio. A continuación en la sección 4 se exponen los resultados obtenidos considerando diferentes escenarios. Finalmente las conclusiones se exponen en la sección 5.

2 IEEE 802.11 DCF

Dentro del protocolo IEEE 802.11, el mecanismo fundamental es el DCF, operando en modo básico.

En este caso, cuando una estación tiene un paquete listo para ser transmitido, únicamente puede transmitir si el canal se encuentra libre durante un período de tiempo DIFS (DCF Interframe Space). En caso contrario, la estación continua escuchando el canal hasta que lo encuentre libre durante DIFS, y a continuación se espera un intervalo aleatorio de backoff antes de iniciar la transmisión. La inclusión de dicho tiempo de backoff es un mecanismo muy útil para minimizar la probabilidad de colisión entre transmisiones procedentes de diferentes estaciones. También es importante destacar, que este tiempo de backoff además es utilizado para evitar que una única estación acapare el medio, perjudicando a las demás. De esta manera una estación deberá esperar un intervalo aleatorio de backoff, además de un DIFS, entre dos transmisiones de paquetes consecutivas.

El tiempo de backoff responde a un comportamiento exponencial. Su valor se escoge de forma aleatoria en el intervalo *(0, W-1)*. *W* recibe el nombre de ventana de contención, y su valor depende del número de intentos de transmisión fallidos. De esta manera, la primera vez que falla la transmisión de un paquete, la ventana de contención toma su valor mínimo CW_{min}, y por cada nueva transmisión fallida *W* dobla su valor anterior, hasta llegar a un límite $CW_{max}=2^{m}CW_{min}$. Los valores CW_{min} y CW_{max}, están estandarizados y dependen del medio de transmisión (infrarrojo o radiofrecuencia) y en el caso de la radiofrecuencia también de la tecnología utilizada (FHSS o DSSS). En la Tabla 1 se pueden observar los valores CW_{min} y CW_{max} para cada caso.

El tiempo de backoff se decrementa mientras el medio se encuentra libre. En el momento en que éste pasa de nuevo a estar ocupado por la transmisión de un paquete procedente de otra estación, dicho decremento se paraliza, reanudándose de nuevo cuando el medio vuelve a estar libre.

IEEE 802.11 utiliza un sistema de reconocimientos positivos, ya que el esquema CSMA/CA no ofrece detección de colisiones. Cuando el paquete se recibe con éxito, la estación receptora envía el ACK correspondiente después de esperar un tiempo SIFS (Short Interframe Space). Este tiempo SIFS es inferior a DIFS, y de esta manera se da prioridad a los paquetes de reconocimiento sobre los de datos.

3 Entorno de simulación

Para el análisis del protocolo de acceso al medio IEEE 802.11b se ha utilizado un simulador, realizado en el lenguaje de programación C++, que opera sobre plataformas Windows.

Para la realización de las simulaciones, que se detallarán en la siguiente sección, se ha elegido como

Tabla 1. Valores mínimos y máximos de la ventana de contención

PHY	CW_{min}	CW_{max}
FHSS	16	1024
DSSS	32	1024
IR	64	1024

medio de transmisión la radiofrecuencia, con tecnología DSSS y velocidad de transmisión de 1 Mbps.

Tanto para el escenario unicelular, como para el compuesto por múltiples celdas, se han tomado celdas con características comunes. Estas celdas se han considerado de forma hexagonal y radio de 150 metros. Para este tamaño de celda, el retardo extremo-extremo alcanza 1 μs, que según lo especificado en [1] es el valor máximo que este retardo debe alcanzar.

En el escenario unicelular dicha celda está formada por un único punto de acceso, y 10 estaciones de usuario. Cada una de estas estaciones genera paquetes de longitud determinista y tamaño 1023 bytes. Por otro lado, el punto de acceso no genera tráfico hacia las demás estaciones. Finalmente no se permiten las retransmisiones de paquetes y se considera una tasa de error de bit (BER) nula.

El escenario de múltiples celdas se compone de 100 celdas, formando un rectángulo de 10x10. Aunque para la obtención de estadísticas se tomarán únicamente las 36 celdas centrales. En esta área de cobertura, se utiliza un cluster de tamaño 3, como patrón de reuso de frecuencias. Se considera que la elección de las frecuencias en la banda de 2,4 GHz se realizará de forma que la interferencia por canal adyacente sea inapreciable.

Cada una de las celdas contiene un punto de acceso. En esta área de cobertura se distribuyen 1000 estaciones de usuario de forma uniforme. Como en el caso unicelular, cada estación de usuario genera paquetes de longitud determinista y tamaño 1023 bytes, y además el punto de acceso también envía tráfico a cada una de las estaciones. El número máximo de retransmisiones posibles por paquete es de 10. Por último mencionar, que se comprobará si durante algún intervalo del tiempo de transmisión de un paquete, la relación señal a interferente (SIR) cae por debajo del valor umbral. Como umbral se ha tomado 0 dB según se especifica en [4]. Si la SIR en algún intervalo toma un valor inferior al especificado en el umbral, el paquete recibido se considerará erróneo.

En ambos escenarios, se ha considerado que tanto los puntos de acceso como el resto de las estaciones carecen de movilidad y emiten con una potencia de 0 dBm, utilizando antenas omnidireccionales. Para el cálculo de las potencias recibidas se considera que la

atenuación depende de la distancia según d^g con $g = 3,5$.

Finalmente, los valores de los parámetros que definen el acceso al medio del IEEE 802.11b, considerados en las simulaciones, se muestran en la Tabla 2. Tanto los tamaños de las diferentes cabeceras y tipos de paquete, como los valores de los tiempos involucrados, se corresponden con los especificados en [1].

El escenario unicelular presentado es altamente relevante en los entornos de comunicaciones reales actuales. Dicho escenario se corresponde con el que encontraríamos, por ejemplo, en las diferentes salas de los aeropuertos, si se quisiera ofrecer al cliente acceso a Internet. De esta manera los clientes podrían tener acceso a su correo electrónico y a páginas Web, mientras esperan su vuelo. Por los mismos motivos el escenario también sería aplicable a las salas de espera de las estaciones de tren.

Por otro lado, el escenario multicelular expuesto también es significativo en los entornos de comunicaciones reales actuales. Éste se correspondería con el que se encontraría en un área exterior extensa, segmentada en diferentes celdas, para que los clientes puedan acceder a su correo electrónico, a páginas Web y también puedan intercambiar datos entre ellos.

4 Resultados

El análisis del protocolo se ha divido en dos partes. Primero se ha estudiado su eficiencia en entornos interiores unicelulares, para finalmente continuar con el estudio del protocolo IEEE 802.11b en entornos exteriores celulares.

Tabla 2. Parámetros utilizados para las simulaciones

Cabecera MAC	34 bytes
Cabecera PHY	192 bytes
ACK	14 bytes
Beacon	362 bytes
Velocidad de transmisión	1 Mbps
Máximo retardo extremo-extremo	1 μs
Tiempo de Slot	21 μs
SIFS	10 μs
DIFS	52 μs
Ventana de backoff	CW_{min}=32 CW_{max}=1024

4.1 Escenario de una celda

En un escenario unicelular se tiene una estación actuando como punto de acceso y múltiples estaciones de usuario, todas ellas compartiendo un mismo medio. El uso de intervalos de backoff y de tiempos de espera DIFS y SIFS, permite que todas las estaciones puedan acceder al medio de forma equitativa y aleatoria.

Por el medio no compiten únicamente los paquetes de datos y los de reconocimiento, sino que también entran en juego los paquetes de *beacon*. El punto de acceso los envía de forma periódica a las demás estaciones de la celda. Estos son necesarios para la correcta sincronización de las estaciones de usuario.

La presencia de paquetes de beacon en el medio afecta a la transmisión de los paquetes de información. Contra menor sea el período de beacon, mayor tráfico de señalización se está incorporando en el medio, y esto provocará el consiguiente aumento de los tiempos de espera en cola (Wq) y del tiempo de transmisión de un paquete de información (T_{tx}). Por este motivo se debe llegar a un compromiso a la hora de escoger el período de beacon, ya que si éste es pequeño afectará negativamente a la transmisión de la información, pero por el contrario, si su valor es muy elevado, la sincronización de las estaciones se verá afectada.

Se ha analizado la influencia del período de beacon en el comportamiento del protocolo de acceso al medio, estudiando el throughput y el tiempo medio de transmisión de un paquete en la celda. Bajo el término tiempo medio de transmisión se entiende el tiempo transcurrido desde que el paquete ha salido de la cola de su estación correspondiente y está esperando a poder acceder al medio, hasta que la estación transmisora recibe el paquete ACK correspondiente. Este tiempo incluye el tiempo que tarda un paquete en ganar el medio, conocido como retardo de MAC. Dicho retardo tiene en cuenta los intervalos de backoff y el tiempo de espera DIFS.

En las Fig. 1 y 2 se puede observar el throughput medio en la celda y el tiempo medio de transmisión de un paquete, respectivamente. Se considera que todas las estaciones del sistema tienen siempre un paquete disponible para enviar, por lo tanto el throughput obtenido recibe el nombre de *throughput de saturación* [5]. Efectivamente se puede comprobar que el throughput incrementa a medida que los paquetes de beacon están más espaciados en el tiempo, y por el contrario el tiempo medio de transmisión se ve decrementado. De las gráficas se extrae que un período de beacon adecuado sería 100 ms. Si se compara el throughput de saturación obtenido para períodos de beacon elevados, con los resultados expuestos en [5], considerando 10 estaciones generando tráfico, se observa que los resultados obtenidos son coherentes con los mostrados en el artículo citado.

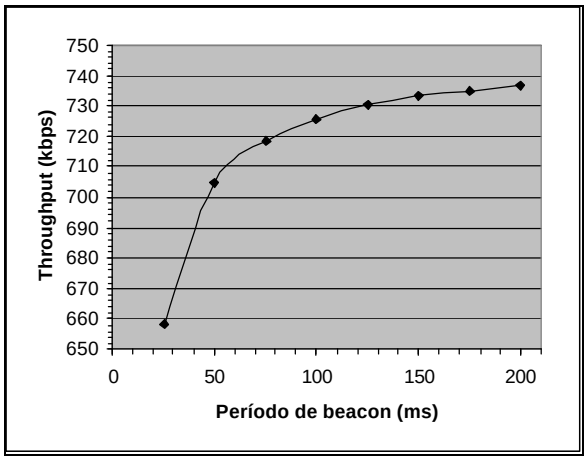


Figura 1. Throughput en función del período de beacon

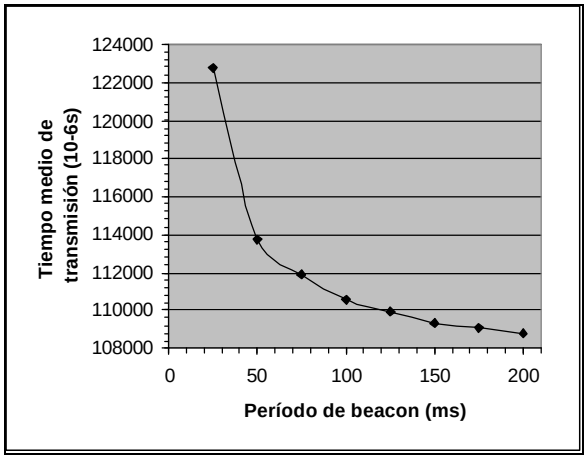


Figura 2. Tiempo medio de transmisión en función del período de beacon

Hasta el momento se ha contabilizado el retardo de MAC considerando que cada estación debe esperar siempre un tiempo DIFS, como mínimo, a partir del momento exacto en el que tiene un paquete listo para ser transmitido, aunque el medio ya lleve desocupado un tiempo DIFS o superior a éste. En lugar de obligar siempre a la estación a esperar un DIFS, se podría tener en cuenta la memoria de la capa MAC, y considerar que si el canal ya lleva libre un tiempo DIFS o superior, la estación puede acceder al medio sin necesidad de esperar de nuevo este tiempo de acceso. Bajo estas suposiciones el retardo medio de transmisión de un paquete disminuiría.

Se considera que el tráfico ofrecido a la celda es de 200 kbps. En la Tabla 3, se presentan las diferencias (ΔT_{tx}) obtenidas en cuanto a retardo medio de transmisión, si se tiene en cuenta la memoria del MAC. Se puede observar una ligera disminución en el tiempo de transmisión. De esta forma se comprueba que las diferencias entre ambos criterios no son significativas, y que tanto la elección de uno como de otro nos lleva a comportamientos de la capa MAC muy cercanos.

Hasta la fecha, se han realizado múltiples estudios con el fin de optimizar la eficiencia de la capa MAC, en cuanto a throughput y tiempo de transmisión se refiere [5] [6] [7]. Se observó que un parámetro importante a la hora de incrementar dicha eficiencia es la ventana de backoff.

Por este motivo se ha analizado la influencia de dicha ventana de backoff, en el caso de tener tráfico asimétrico en la œlda. Para este caso se ha podido comprobar que la elección de los parámetros CW_{min} y CW_{max} es determinante.

Se considera que el tráfico es asimétrico cuando el flujo de datos incorporado a la red por las estaciones de usuario y dirigido hacia el punto de acceso, es inferior al tráfico que el punto de acceso proporciona a la red. En el escenario que se ha analizado hasta el momento, se tienen 10 estaciones transmitiendo tráfico, y un punto de acceso que no envía información a ninguna de las otras estaciones. Para analizar el escenario unicelular en presencia de tráfico asimétrico, se considera que el punto de acceso envía información a cada una de las estaciones presentes en la celda. De esta manera, dicho punto de acceso se encuentra 10 veces más cargado que el resto de las estaciones. Si éste compite por el medio en las mismas condiciones que el resto de estaciones, el tiempo medio de espera en cola de un paquete de información será muy superior para el punto de acceso que para el resto de estaciones de usuario. Por este motivo se ha decidido dar prioridad al punto de acceso, eligiendo nuevos valores de CW_{min} y CW_{max} para éste, disminuyendo siempre CW_{min} respecto a su valor anterior, y manteniendo la ventana de backoff original para las estaciones de usuario.

Inicialmente se analiza el comportamiento del protocolo en presencia de tráfico asimétrico para un sistema poco cargado, al que se le ofrece 400 kbps. En la Tabla 4 se observa la reducción del tiempo medio de espera en cola en el punto de acceso, en función del período de beacon, cuando reducimos el valor de CW_{min} del punto de acceso. En este caso se observa que las diferencias entre los dos resultados no son significativas, al encontrarnos en un sistema poco cargado. Por último se deduce que la influencia del paquete de beacon tampoco es significativa.

A continuación se repite el análisis anterior, esta vez considerando que todas las estaciones tienen siempre al menos un paquete en la cola esperando a ser transmitido. En la Tabla 5 se observa la reducción considerable del tiempo medio de espera en cola en el punto de acceso. En este caso, además, se puede observar la influencia negativa de la presencia de paquetes de beacon en el tiempo medio de espera en cola.

La elección de los valores máximos y mínimos de la ventana de backoff también afecta al throughput de la celda. Si se reduce CW_{min} el comportamiento del tiempo medio de espera en cola mejora notablemente, pero, como contrapartida, el del throughput se ve empeorado, según se expone en [5]. Dicho comportamiento es independiente del período de beacon utilizado, y se observa en la Tabla 6.

Tabla 3. Tiempo de transmisión en función del período de beacon y la memoria de MAC

Período beacon (ms)	T_{tx} sin memoria MAC (ms)	T_{tx} con memoria MAC (ms)	ΔT_{tx} (ms)	ΔT_{tx} (%)
25	11005,4	10960	45,4	0,41
50	10623,9	10585,8	38,1	0,36
75	10524	10475,3	48,7	0,46
100	10462,1	10418,3	43,8	0,42
125	10429,5	10377,2	52,3	0,5
150	10405	10362,6	42,4	0,41
175	10386,1	10347,3	38,8	0,37
200	10368	10323,4	44,6	0,43

Tabla 4. Wq para el punto de acceso según el período de beacon y la ventana de backoff, para un sistema poco cargado

Período de beacon (ms)	Wq (ms) CW_{min}=32 CW_{max}=1024	Wq (ms) CW_{min}=8 CW_{max}=32	Diferencia Wq (ms)
25	4,34707	3,71032	0,63675
50	3,8193	3,23648	0,58282
75	3,76583	3,02449	0,74134
100	3,51441	3,07765	0,43676
125	3,55321	2,9487	0,60451
150	3,40113	2,91752	0,48361
175	3,4637	2,86767	0,59603
200	3,39777	2,87348	0,52429

Tabla 5. Wq para el punto de acceso según el período de beacon y la ventana de backoff, para un sistema cargado

Período de beacon (ms)	Wq (ms) CW_{min}=32 CW_{max}=1024	Wq (ms) CW_{min}=8 CW_{max}=32	Diferencia Wq (ms)
25	245659	134,429	245524,571
50	51974,4	235,926	51738,474
75	9461,84	325,642	9136,198
100	2428,73	330,606	2098,124
125	686,357	374,878	311,479
150	570,538	431,109	139,429
175	499,039	399,848	99,191
200	464,134	448,756	45,378

Tabla 6. Throughput en función del período de beacon y de la ventana de backoff

Período de beacon (ms)	Throughput CW_{min}=32 CW_{max}=1024 (kbps)	Throughput CW_{min}=8 CW_{max}=32 (kbps)	Diferencia Throughput (kbps)
25	703,297	622,031	81,266
50	772,303	647,438	124,865
75	792,758	655,265	107,493
100	799,365	662,561	136,804
125	795,599	664,516	131,083
150	799,566	665,535	134,031
175	797,602	664,111	133,491
200	800	666,363	133,637

Por lo tanto, a la hora de escoger los tamaños máximos y mínimos de la ventana de backoff, se deberá tener en cuenta tanto el comportamiento del throughput como el del tiempo de espera de cola, y elegir los valores CW_{min} y CW_{max} más adecuados para ambos.

Si se compara el comportamiento de una celda con tráfico asimétrico con otra con tráfico simétrico, ambas con el mismo tráfico ofrecido, se observa que el comportamiento del primer caso es mejor, en cuanto a throughput se refiere. En la Tabla 7 se muestra el throughput y el número medio de veces que un paquete entra en backoff antes de ser transmitido, para ambos tipos de tráfico. Se observa como para el caso de tráfico asimétrico el número de backoffs por paquete es inferior al obtenido para tráfico simétrico. En el caso de tener tráfico asimétrico se tiene una estación, el punto de acceso, más cargada que el resto. De esta manera, la cola en dicha estación estará siempre llena, mientras que las otras estaciones mantendrán su cola desocupada durante más tiempo. Esto provoca que mientras los paquetes generados en el punto de acceso se encuentren en backoff, los generados en las demás estaciones se transmitirán de forma inmediata, siempre y cuando encuentren su cola vacía. Por el contrario, cuando el tráfico es simétrico, tanto el punto de acceso como las demás estaciones de usuario mantienen siempre su cola ocupada. En este caso todas las estaciones tienen la misma

probabilidad de entrar en backoff, y por consiguiente el número de backoffs por paquete aumenta, provocando una caída del throughput.

4.2 Escenario de múltiples celdas

Si se considera un escenario de múltiples celdas, a la problemática presentada en el escenario anterior, se le debe añadir la influencia de la interferencia cocanal, presente entre celdas que operan a la misma frecuencia. Como se mencionó en la sección 3, en este estudio se considerará que la interferencia por canal adyacente es nula.

Se ha estudiado el comportamiento del protocolo IEEE 802.11b en una red celular, bajo diferentes condiciones de carga. Para evaluar su comportamiento se han analizado el throughput, el tiempo medio de transmisión, el porcentaje de paquetes erróneos (PER) y la SIR por celda, variando el tráfico ofrecido en dicha celda y en ausencia de paquetes de beacon. Los resultados de dicho análisis se pueden observar en las Fig. 3, 4, 5 y 6.

En la Fig. 3, se observa el throughput, y se obtiene el throughput saturación por celda del sistema multicelular estudiado. Como se esperaba, de la Fig. 4 se deduce que a medida que el tráfico ofrecido a la celda aumenta, el tiempo medio de transmisión de un paquete también se ve incrementado, ya que el número de paquetes que competirán por el medio será mayor, y por consiguiente aumentará el número de paquetes que entrarán en backoff. Por los mismos motivos, en la Fig. 5 se observa como la PER aumenta a medida que se incrementa el tráfico ofrecido, y en consecuencia la SIR disminuirá, como se deduce de la Fig. 6.

5 Conclusiones

En el artículo presentado, se han expuesto las características principales del protocolo de acceso al medio IEEE 802.11b. En primer lugar se ha caracterizado el protocolo para entornos interiores. Se ha analizado su comportamiento para diferentes períodos de beacon, y se ha podido observar que la elección adecuada de este parámetro es fundamental para conseguir un comportamiento adecuado del protocolo. También se ha intentado proporcionar soluciones para conseguir mejorar las prestaciones del protocolo, en cuanto a throughput y tiempo de transmisión se refiere. Para ello se han analizado las mejoras obtenidas en el caso de tener en cuenta la memoria de la capa MAC, para de esta forma no tener que esperar tiempos DIFS adicionales e innecesarios. Además se ha estudiado cómo mejorar el comportamiento de los puntos de acceso en el caso de tener tráfico asimétrico, disminuyendo el valor de su ventana mínima de backoff. Por último se ha analizado el comportamiento del protocolo en un entorno exterior, compuesto por múltiples celdas, donde la interferencia cocanal juega un papel importante.

6 Reconocimientos

Este trabajo ha sido parcialmente financiado por el proyecto CICYT TIC2000-1041-C03-1.

Tabla 7. Throughput y nº de backoffs por paquete en función del período de beacon y del tipo de tráfico para CW_{min}=32 y CW_{max}=1024

Período de beacon (ms)	Tráfico simétrico		Tráfico asimétrico	
	Nº backoffs	Throughput (kbps)	Nº backoffs	Throughput (kbps)
25	1,7824	643,573	1,47717	703,297
50	1,78644	682,625	1,40086	772,303
75	1,78751	696,747	1,3865	792,758
100	1,78803	703,306	1,37137	799,365
125	1,78891	707,032	1,34134	795,599
150	1,79086	709,778	1,33179	799,566
175	1,79088	710,93	1,32924	797,602
200	1,78656	713,65	1,33381	800

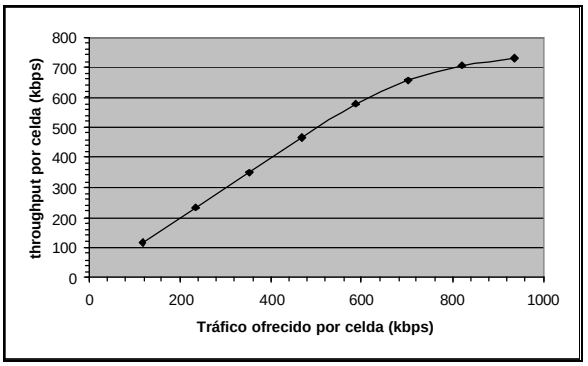


Figura 3. Throughput por celda en función del tráfico ofrecido

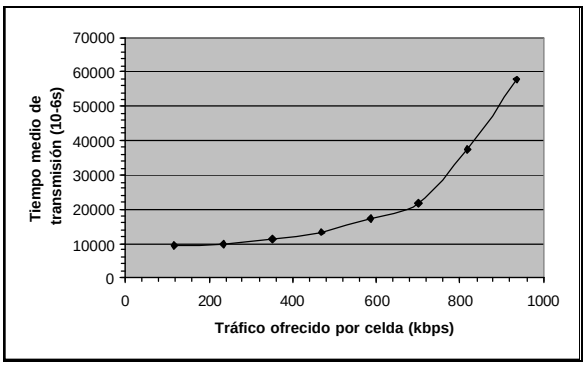


Figura 4. Tiempo medio de transmisión por celda en función del tráfico ofrecido

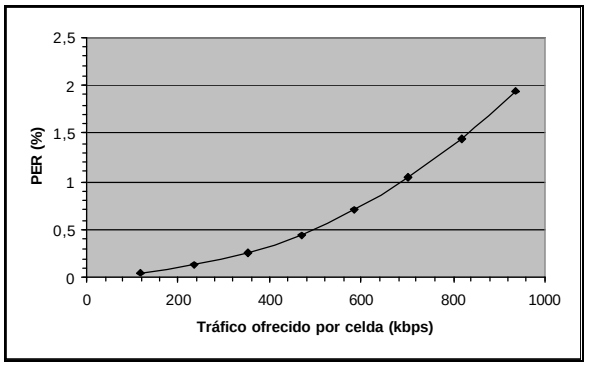


Figura 5. PER por celda en función del tráfico ofrecido

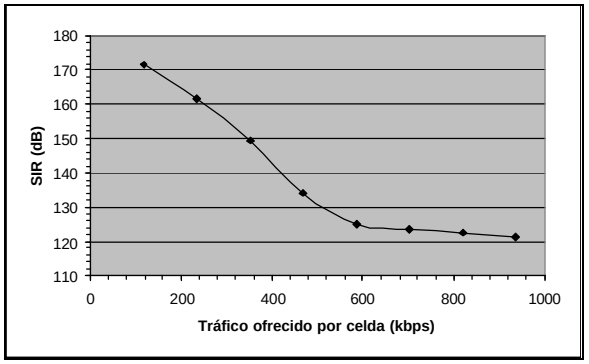


Figura 6. SIR por celda en función del tráfico ofrecido

7 Referencias

[1] IEEE, "Reference number ISO/IEC 8802-11:1999(E) IEEE Std 802.11, 1999 edition. International Standard [for] Information Technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications", 1999.

[2] IEEE, "IEEE Std 802.11b - 1999 Supplement to IEEE Std 802.11, 1999 Edition. Supplement to IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High–Speed Physical Layer Extension in the 2.4 GHz Band", 1999.

[3] IEEE, "IEEE Std 802.11a - 1999 Supplement to IEEE Std 802.11, 1999 Edition. Supplement to IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High–Speed Physical Layer Extension in the 5 GHz Band", 1999.

[4] K. K. Leung, B. McNair, L. J. Cimini Jr., J. H. Winters, "Outdoor IEEE 802.11 Cellular Networks: MAC Protocol Design and Performance", International Conference on Communications, vol. 1, pp. 595 – 599, Abril – Mayo 2002.

[5] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function", IEEE Journal on selected areas in communications", Vol. 18, No. 3, pp. 535 – 547, Marzo 2000.

[6] Chuang Heng Foh, Moshe Zukerman, "Performance Analysis of the IEEE 802.11 MAC Protocol", Proceedings of the EW 2002 Conference, pp. 184-190, Febrero 2002.

[7] H. Wu, S. Cheng, Y. Peng, K. Long, J. Ma, "IEEE 802.11 Distributed Coordination Function (DCF): Analysis and Enhancement", International Conference on Communications, vol. 1, pp. 605 – 609, Abril – Mayo 2002.

Análisis Experimental del Comportamiento de TCP sobre IEEE 802.11b y del protocolo Snoop como Mecanismo de Mejora

R. Agüero, L. Sánchez, M. García, J. Choque, L. Muñoz
Departamento de Ingeniería de Comunicaciones
Grupo de Ingeniería Telemática. Universidad de Cantabria
39005 Santander
E-mail: [ramon,lsanchez,marta,jchoque,luis]@tlmat.unican.es

***Abstract**. The combined effect of the advent of Wireless Local Area Networks (WLAN) and the poor behaviour exhibited by Internet protocols when operated over such infrastructures has opened a large number of researching interests, which work towards the improvement of such bad performance. This article belongs to this movement and goes even further, as it provides both an in-depth characterization of how badly TCP behaves over IEEE 802.11b and a full experimental analysis of one of the solutions that have been designed to overcome this problem, the Snoop agent. This work has been done within the context of the Wireless Internet Networks (WINE) project, corresponding to the 5th framework of the European IST programme, addressed to enhance the performance of the TCP-UDP/IP protocol stack over wireless infrastructures such as IEEE 802.11, Bluetooth and HIPERLAN/2. The main outcome of this project was a Performance Enhancing Proxy (PEP), so-called Wireless Adaptation Layer (WAL), targeted at Internet protocols when they operate over wireless shared access LANs. The Snoop agent has been implemented as a module belonging to this PEP.*

1 Introducción

El creciente interés que las tecnologías de redes de área local inalámbricas han venido generando recientemente ha causado que su presencia en la vida cotidiana sea cada vez mayor. Además, la aparición de nuevos escenarios y aplicaciones para este tipo de redes también ha sufrido un aumento espectacular. Al margen de un interés comercial innegable, las WLAN también han acaparado el esfuerzo de las comunidades investigadoras, pues se prevé que sean un elemento integrante de lo que se denomina como 4G. Destaca la multitud de propuestas que han surgido para tratar de ocultar las deficiencias del canal radio a las capas superiores, en especial a la pila de protocolos de Internet, por ser la más extendida en la actualidad. Los componentes de esta arquitectura no fueron diseñados teniendo en cuenta las características de los entornos inalámbricos y su comportamiento sobre este tipo de tecnologías de red dista mucho de ser adecuado. En ese sentido, destaca sobremanera el protocolo de transporte TCP, pues los complejos mecanismos de control de congestión que incorpora han sido optimizados para su utilización en redes cableadas, en las que la pérdida de segmentos se debe generalmente a la congestión de los nodos intermedios de la red. Al recibir indicaciones de que una situación de ese tipo estuviera ocurriendo, un transmisor TCP reaccionaría reduciendo el ritmo de generación de segmentos. Esta política es muy perjudicial en entornos inalámbricos, pues la pérdida de segmentos tiene una naturaleza completamente diferente, pudiéndose atribuir, prácticamente en su totalidad, a las condiciones hostiles del canal radio. Uno de los objetivos de este artículo es profundizar en este comportamiento deficiente, ofreciendo una exhaustiva caracterización del rendimiento de TCP sobre la WLAN IEEE 802.11b, a través del estudio del impacto de los errores causados por el medio inalámbrico.

Se han propuesto diversos mecanismos para mejorar este comportamiento deficiente, siguiendo enfoques claramente diferenciados. Mientras un conjunto trata de solucionar el problema modificando el propio protocolo de transporte TCP, hay otros que buscan una aproximación local (transparente a los protocolos IP). A este último grupo pertenece el mecanismo que se evaluará experimentalmente. Se trata del agente Snoop, desarrollado por la Universidad de Berkeley, que es una de las opciones que más interés ha suscitado (el IETF lo ha incluido recientemente en uno de sus RFC [1]). Se complementa el trabajo reflejado en [2], pues se realiza la caracterización en un entorno completamente real, empleando además la extensión de alta velocidad del estándar, IEEE 802.11b. Para ello se llevó a cabo una exhaustiva campaña de medidas, en la que se obtuvo un conjunto considerable de parámetros que posteriormente fue analizado.

El artículo se estructura en las secciones siguientes: primeramente se analizará en detalle cuál es el comportamiento del protocolo TCP sobre IEEE 802.11b, en un principio sobre un canal ideal, con el objeto de conocer cuál es el máximo rendimiento que se puede alcanzar, para después analizar la influencia de los errores originados por el canal radio en su comportamiento. A continuación, se dará una breve descripción de las principales características del agente Snoop, para posteriormente pasar a analizar cuál es el efecto de su utilización. Por último, se

finalizará con un conjunto de conclusiones y de líneas de trabajo que se abren.

2 Comportamiento nativo de TCP sobre IEEE 802.11b

Para poder validar la mejora introducida por el agente Snoop es necesario tener claro cuál es el comportamiento nativo del protocolo TCP sobre redes inalámbricas basadas en el estándar IEEE 802.11. Para ello se dispuso de una plataforma de medidas experimental, en la que dos terminales comparten el canal inalámbrico, actuando uno de ellos como transmisor TCP y el otro como receptor puro, por lo que únicamente genera reconocimientos. El canal inalámbrico que comparten se trata de un enlace bidireccional no simultáneo, pues debido al método de acceso al medio (MAC, Medium Access Control) empleado no se puede tener información viajando en ambos sentidos a la vez. En ese sentido, y a pesar de que el tráfico generado por el receptor TCP sea pequeño, el rendimiento obtenido debería ser inferior al que ofrece el mismo canal a aplicaciones que utilicen UDP como protocolo de transporte [3].

2.1 Canal ideal

La Tabla 1 recoge los rendimientos que se obtuvieron en una situación en la que ambos terminales estaban situados lo suficientemente cerca como para considerar que la presencia de errores debidos al canal inalámbrico era despreciable. Adicionalmente, se ha representado ese rendimiento como porcentaje frente a la tasa binaria de trabajo y al rendimiento nominal (aquel que se ofrece al nivel IP).

Se puede observar que aproximadamente el 10% del rendimiento nominal se pierde debido al mecanismo de reconocimiento implementado a nivel 4. Este aspecto se corrobora en la Fig. 1, que representa el reparto de la tasa binaria global entre todos los mecanismos que intervienen en la comunicación. Esta gráfica se ha obtenido a través de un estudio analítico en el que se han tenido que hacer unas estimaciones acerca del tiempo medio de espera por trama según el mecanismo de backoff que se define en el estándar IEEE 802.11b, y que se sale del alcance del presente artículo. Estas suposiciones, que se basan en el carácter claramente asimétrico del tráfico, consisten en asumir que las tramas generadas por la entidad TCP receptora no sufren ninguna espera, mientras que una distribución uniforme (similar a la que se midió en el caso de tráfico unidireccional [3]) se aplicará a los segmentos de

Tabla 1. Rendimientos TCP sobre canal IEEE 802.11b ideal

Rb (Mbps)	Rendimiento Nominal		Rendimiento TCP		
	Mbps	% Rb	Mbps	% Rb	% Nom
11	6.2	56	5.0	45	80
5.5	3.9	71	3.3	60	85
2	1.7	85	1.5	75	88
1	0.9	90	0.8	80	89

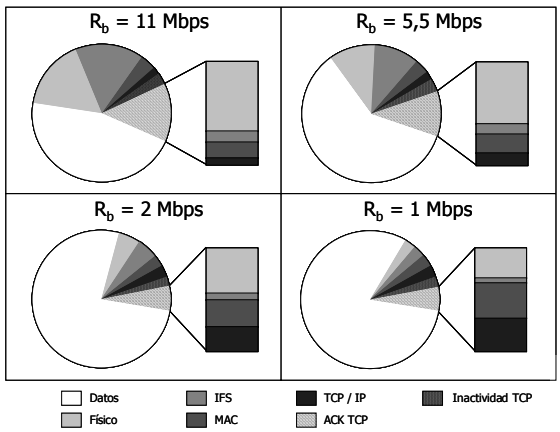


Figura 1. Reparto de la tasa binaria bruta en condiciones ideales

datos que genere el transmisor. El efecto global de esta estimación es similar al que se deriva en [4].

2.2 Influencia de los errores del canal radio en TCP

Siguiendo con la caracterización anterior, y con el fin de determinar la influencia de los errores debidos al canal radio en el rendimiento de TCP, los dos terminales se situaron en una posición que presenta una relación señal a ruido (SNR, Signal to Noise Ratio) lo suficientemente baja (como se muestra en la Fig. 2) como para asegurar la presencia de errores en el canal inalámbrico.

La Tabla 2 muestra un conjunto de los parámetros más relevantes de la conexión TCP establecida entre los dos terminales. Sólo se ha empleado la tasa binaria máxima permitida (11 Mbps), ya que el efecto de los errores es más significativo. Uno de los primeros aspectos a destacar es la gran variabilidad de los rendimientos obtenidos en las medidas, lo que refleja una situación real, en la que las condiciones del enlace radio son muy cambiantes. Resaltar, por otro lado, que las herramientas de medida empleadas no son apropiadas para obtener medidas del tiempo de ida y vuelta (RTT, Round Trip Time), por lo que se empleó un programa propietario para su caracterización.

Además del rendimiento, la tabla muestra un conjunto de métricas adicionales, que ayudan a obtener una visión más completa del comportamiento de la conexión TCP. Estas aparecen definidas a continuación:

- Retransmisiones: número total de segmentos de datos retransmitidos por la entidad TCP transmisora.
- Número máximo de retransmisiones: máximo número de veces que un mismo segmento TCP de datos es retransmitido.
- Periodo de inactividad: tiempo máximo que el transmisor ha permanecido inactivo.
- Reconocimientos duplicados: número total de reconocimientos duplicados recibidos en el transmisor.

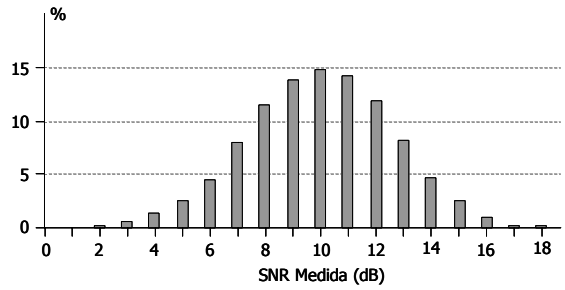


Figura 2. Relación señal a ruido en el escenario de medida

- Reconocimientos triplicados: número total de reconocimientos triplicados recibidos en el transmisor.

Como se puede observar en la Tabla 2, el factor más determinante en la pérdida de rendimiento se asocia a la presencia de periodos de inactividad. Es fundamental identificar las causas concretas de la aparición de estos periodos de inactividad, con el fin de determinar la manera más apropiada para mejorar el comportamiento de TCP en entornos radio.

En las capturas analizadas se han identificado dos tipos de inactividad diferentes; el primero se produce cuando un único segmento se retransmite en repetidas ocasiones por expiración del intervalo de retransmisión (RTO, Retransmission TimeOut), en las que se aplica el procedimiento de backoff definido en TCP. Si un segmento se retransmite N veces, y la primera retransmisión se disparó con un $RTO_{inicial}$, el tiempo de espera global se puede calcular a partir de la siguiente expresión:

$$Inactividad\ total = RTO_{inicial}\left(2^{N}-1\right)$$

El segundo de los tipos, que se ha mostrado como el más relevante, se producía tras la retransmisión única de un segmento, con un valor elevado del RTO. En alguna ocasión se ha observado el máximo RTO permitido por la implementación de TCP que se ha empleado (2 minutos). Es necesario, por tanto, determinar la concatenación de hechos que causan dicha situación. Para ello se debe realizar un análisis exhaustivo del código que implementa el protocolo TCP dentro del Sistema Operativo (SO) empleado durante la campaña de medidas, Linux.

Antes de describir las funciones en las que realmente se implementan los diferentes mecanismos empleados en TCP, es necesario introducir un conjunto de variables que usan dichas funciones, relevantes en el análisis posterior.

- packets_out: segmentos TCP que han sido enviados y todavía no han sido reconocidos por parte del receptor.
- retransmits: número de retransmisiones disparadas por expiración del RTO.
- srtt: RTT suavizado (escalado por ocho, para facilitar la implementación).
- mdev: desviación estándar del RTT (escalada por cuatro).
- rto: valor actual del intervalo a aplicar en las retransmisiones.
- backoff: empleado en el caso de que un segmento se haya de retransmitir más de una vez.
- snd_cwnd: valor actual de la ventana de congestión (en segmentos).
- snd_ssthresh: umbral que se maneja en el procedimiento slow start.

Las siguientes funciones son las que se emplean para la actualización de las variables temporales que la implementación de TCP utiliza para gestionar las retransmisiones. A lo largo de todas las medidas se ha utilizado la opción de Timestamp [5], lo que determina el uso de unas funciones u otras.

- tcp_ack_saw_tstamp: si la opción Timestamp está activa, esta función se emplea para la actualización del RTT y el RTO.
- tcp_rtt_estimator: gestiona los valores de RTT a lo largo de una conexión TCP, basándose en el trabajo de Van Jacobson [6].
- tcp_set_rto: calcula, a partir de los valores actuales de RTT, el RTO.
- tcp_bound_rto: limita el RTO entre dos valores establecidos en la implementación (0,2 y 120 segundos en la versión de TCP empleada).

Las Fig. 3 y 4 muestran los diagramas de flujo de dichas funciones.

Uno de los casos más sorprendentes entre todos los analizados es el segundo de los que aparecen en la Tabla 2, en el que el rendimiento efectivo obtenido se situó en los 0,707 Mbps (frente a los 5 Mbps que se obtuvieron en el caso ideal). En esta conexión, como se puede observar en la Fig. 5 se produjo una parada de 57 segundos, suponiendo un gasto de aproximadamente el 50% de la tasa binaria bruta de trabajo.

En la Fig. 6 se puede ver el intercambio de segmentos que dio lugar a dicha inactividad. Con el fin de facilitar su lectura se ha empleado una numeración relativa de los segmentos que asimismo se han

Tabla 2. Parámetros de conexiones TCP sobre un enlace IEEE 802.11b con baja SNR

Rb (Mbps)	Medida	Rend (Mbps)	Retx	Max Retx	RTT Medio (ms)	Desv Std RTT (ms)	Inactividad Max (seg)	Inactividad Total (seg)	ACK Dup	Triple ACK
11	1	2.91	89	5	42,11	16,95	1,6	10,73	267	20
	2	0.71	178	8	41,83	35,6	57,4	102,23	290	30
	3	4.49	2	1	50,05	11,35	0,17	1,15	51	2
	4	0.50	112	6	45,86	44,74	120	152,83	141	11
	5	4.59	3	1	47,43	11,09	0,4	1,66	75	4

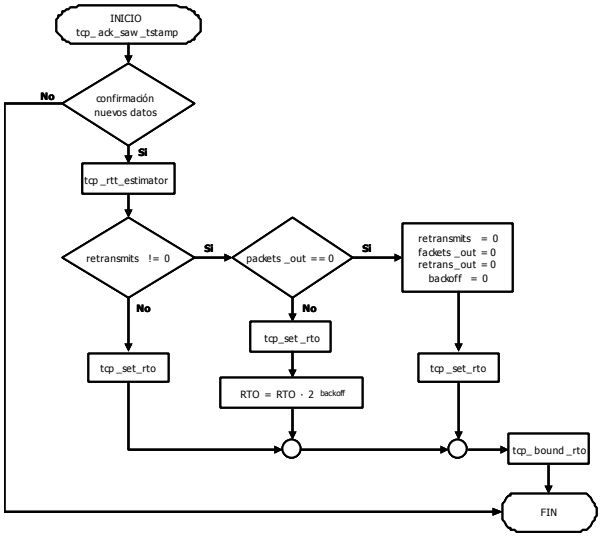


Figura 3. Diagrama de flujo de tcp_ack_saw_tstamp

separado teniendo en cuenta su dirección. Por otra parte, es importante recordar que, debido al mecanismo de petición automática de repetición (ARQ, Automatic Repeat reQuest) de la recomendación IEEE 802.11, la pérdida de un segmento TCP supone, realmente, la recepción de cuatro tramas MAC corruptas de manera consecutiva. La Tabla 3 muestra la variación de las variables descritas anteriormente a lo largo de la captura, siendo la columna de Situación la que enlaza la información en dicha tabla con la Fig. 6. El valor de la ventana de congestión al comienzo de la captura es de ocho, ya que se observa que el transmisor TCP tiene ese número de segmentos en el aire sin haber sido confirmados. Tras la retransmisión, por expiración del timeout, del segmento uno, se invoca el procedimiento de slow start, pasando snd_cwnd a valor 1, y el umbral de slow start a la mitad de la ventana de congestión previa (4). Las sucesivas retransmisiones (hasta tres) del mismo segmento, consecuencia de la no recepción de confirmación alguna, causan que el valor del backoff y el de la variable retransmits aumenten paulatinamente. Al recibir el reconocimiento de los segmentos 1, 2 y 3

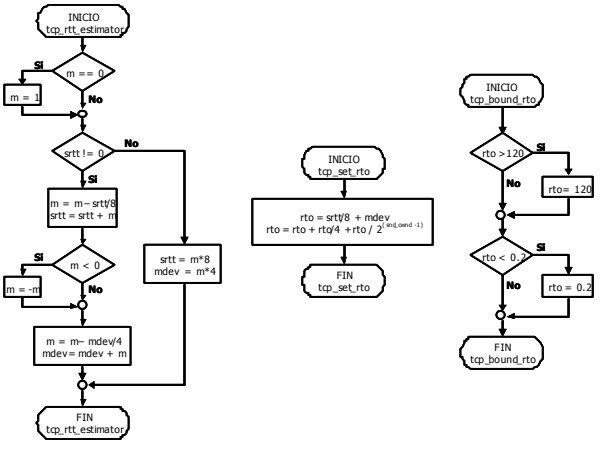


Figura 4. Diagramas de flujo de las funciones tcp_rtt_estimator, tcp_set_rto y tcp_bound_rto

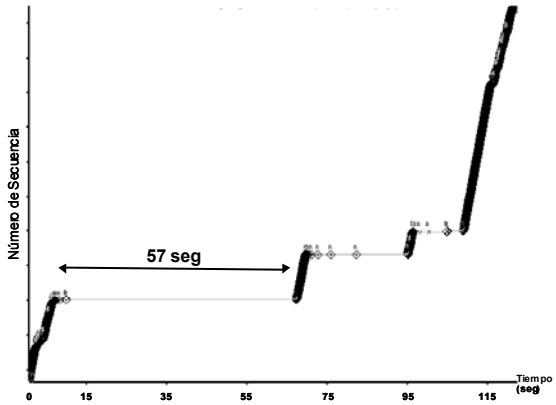


Figura 5. Inactividad de 57 segundos en una conexión TCP

(ack4) además del 5 (mediante una confirmación selectiva [7]), el transmisor no borra el estado de retransmisión (variable retransmits), ya que packets_out no es nulo, pues aún quedan segmentos por confirmar; de la misma manera, la variable backoff tampoco sufre variación y se emplea (según el algoritmo de Karn [8]) para sucesivas retransmisiones. Por último, como se ve en el diagrama de flujo de la Fig. 3, cada vez que se confirmen nuevos datos, las variables RTT se actualizan paulatinamente, como se observa en la Tabla 3.

En esta situación, el transmisor TCP retransmite segmentos que se encuentren en la cola correspondiente, pero no puede enviar datos nuevos, ya que el estado de retransmisión continúa activo [9]. Cuando se pierde el reconocimiento para el octavo segmento, el transmisor espera durante 57 segundos antes de proceder a su retransmisión. El RTO aplicado se puede obtener directamente a partir de los valores que aparecen en la tabla y de la expresión

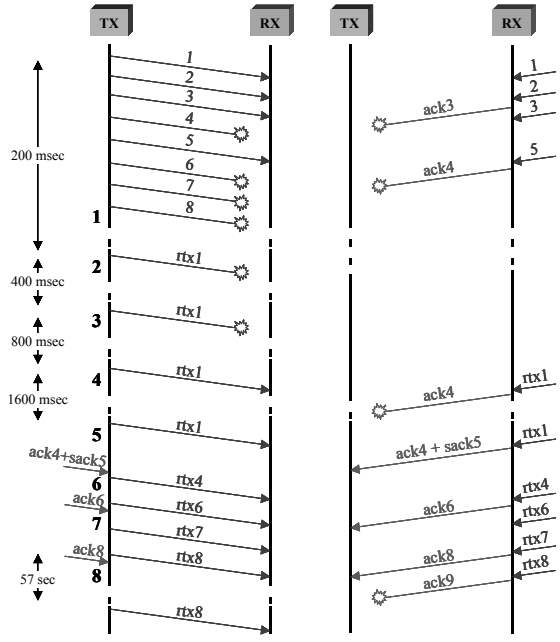


Figura 6. Intercambio de segmentos en la conexión TCP

Tabla 3. Evolución de las variables causante de una inactividad de 57 segundos en una conexión TCP

Situación	Evento	srtt (ms)	mdev (ms)	snd_cwnd	snd_ssthresh	retransmits	backoff	packets_out
1	Start	30	10	8	?	0	0	8
2	Rtx 1 (rto)	30	10	1	4	1	1	8
3	Rtx 1 (rto)	30	10	1	4	2	2	8
4	Rtx 1 (rto)	30	10	1	4	3	3	8
5	Rtx 1 (rto)	30	10	1	4	4	4	8
6	Rx ack(4)	400	750	2	4	4	4	5
7	Rx ack(6)	350	660	3	4	4	4	3
8	Rx ack(8)	310	580	4	4	4	4	1

correspondiente de la función tcp_send_rto.

$$RTO = (srtt + 4mdev)\left(1 + 0{,}25 + \frac{1}{2^{snd\_cwnd-1}}\right)2^{backoff} =$$

$$= (310 + 4 \cdot 580)\left(1 + 0{,}25 + \frac{1}{2^3}\right)2^4 = 57860\, mseg$$

Como no excede los límites establecidos en la función tcp_bound_rto, este es el valor aplicado, que coincide con el observado (Fig. 5).

3 Agente Snoop

Se trata de una solución específica para la mejora de TCP en plataformas inalámbricas desarrollada por la Universidad de Berkeley. El agente Snoop se introduce en un terminal al que se denomina Punto de Acceso (PA), que actúa como intermediario entre el Terminal Móvil (TM) y la red cableada, como muestra la Fig. 7. Este agente monitoriza cada uno de los segmentos que atraviesan el PA en cualquier dirección, manteniendo en todo momento el estado de las conexiones TCP. El comportamiento del agente Snoop varía en función tanto del tipo de tráfico (datos o reconocimientos) como de su dirección (tráfico ascendente – desde el TM – o descendente – hacia el TF –).

En una transferencia de datos desde un Terminal Fijo (TF) al TM, el agente Snoop situado en el PA mantendrá una copia de todos los segmentos de datos generados por el TF, eliminando los ya confirmados , tras analizar los reconocimientos que envía el TM. Más concretamente, su labor consistirá en:

- Retransmitir segmentos de datos con anterioridad a que lo haga el TF, bien mediante el uso de temporizadores o tras la recepción de un reconocimiento duplicado (ya que este hecho indica la pérdida de un segmento), .
- Eliminar los reconocimientos duplicados generados por el TM de forma que se eviten los mecanismos de control de congestión en el TF.

Para ilustrar el funcionamiento del agente Snoop se plantea el siguiente ejemplo: el TF envía los siguientes segmentos de datos TCP hacia el TM: A, B, C, D, E (en este orden); el PA, tras guardarlos en su memoria local, los reenvía hacia el TM. Se asume que el segmento B no llega correctamente debido a los errores en el enlace inalámbrico. Por tanto, el TM recibirá únicamente los segmentos A, C, D, E (en este orden). La recepción de C, D y E antes de B provocará la transmisión de reconocimientos duplicados por parte del TM. Si el agente Snoop no estuviera presente, el TF, al recibirlos, iniciaría el mecanismo de fast retransmit, provocando una retransmisión y la reducción de la ventana de congestión. El agente Snoop retransmitiría B localmente, eliminando los reconocimientos duplicados en su camino hacia el TF, y evitando, de ese modo, la pérdida de rendimiento provocada por el fast retransmit y el control de congestión en el TF.

Como ya se ha mencionado anteriormente, el agente Snoop también retransmite segmentos tras la expiración de un temporizador local. Los límites de RTO manejados por la implementación de TCP no se adaptan al entorno inalámbrico, demostrándose además como una de las causas más relevantes de la pérdida de rendimiento. El agente Snoop utilizará unos valores más adecuados para reducir el efecto negativo de los periodos de inactividad.

En el caso de una transferencia de datos desde el TM hacia el TF, el agente Snoop detecta la pérdida de segmentos en el enlace inalámbrico chequeando su orden de llegada e informa al TM de que no son consecuencia de la congestión de la red, evitando que emplee los mecanismos habituales de control de congestión. Para ello, sería necesario utilizar ciertas modificaciones en el propio protocolo TCP en el TM; por ejemplo, Reconocimientos Negativos [2] o Notificaciones Explícitas de Pérdida [10]. Este caso no se ha considerado en el análisis en que se basa este artículo, para respetar la filosofía de solución local de la WAL.

4 Influencia del módulo Snoop en el comportamiento de TCP

Para introducirlo en el marco del proyecto WINE, la implementación del agente Snoop de la Universidad de Berkeley, realizada para el SO FreeBSD, fue portada a Linux, para proceder, posteriormente, a su integración dentro de la arquitectura de la WAL [11] El agente Snoop es un módulo asimétrico, esto es, sólo reside en el PA; además, como ya se ha mencionado anteriormente, sólo se ha contemplado el caso en el que la transferencia de datos sea desde el TF hacia el TM (tráfico descendente). Como se muestra en la Fig. 7, la plataforma experimental sobre la que se llevó a cabo la campaña de medidas para

evaluar las ventajas del agente Snoop consistía en tres equipos con procesadores Pentium III soportando Linux Red Hat 7.1 (kernel actualizado a la versión 2.4.9) como SO. El TM, actuando como cliente FTP, accede al TF, que hace las veces de servidor de ficheros, a través del PA. Tanto el TM como el PA disponían de sendas tarjetas PCMCIA IEEE 802.11b Orinoco dispuestas en modo ad hoc.

Este apartado se centra en el análisis del comportamiento del agente Snoop cuando las condiciones del enlace radio son malas y, por tanto, la presencia de errores es apreciable. Al utilizarlo en un canal ideal, su efecto es despreciable, y no se produce pérdida alguna de rendimiento (siendo este el comportamiento deseable). En ese sentido, y para asegurar la presencia de errores en este enlace, se situó al TM en un punto en el que la SNR fuese lo suficientemente baja (ver Fig. 1). En este caso se ha añadido un conjunto adicional de estadísticas al análisis, para facilitar la comparación entre los resultados obtenidos con y sin la presencia del agente. Estas aparecen definidas a continuación:

- Segmentos perdidos: segmentos TCP erróneos observados durante la transmisión.
- Ráfaga media de Paquetes Erróneos (EPB media, Erroneous Packet Burst): longitud media de las ráfagas de segmentos TCP erróneos.
- Ráfaga máxima de Paquetes Erróneos (EPB máxima): longitud máxima de las ráfagas de segmentos TCP erróneos.

Como se ha mencionado anteriormente, el estándar IEEE 802.11b especifica el uso de un mecanismo ARQ para la retransmisión de tramas. En las tarjetas Orinoco IEEE 802.11b, una trama puede ser transmitida hasta en cuatro ocasiones, por lo que la pérdida de un datagrama IP implica la recepción de cuatro tramas MAC erróneas de manera consecutiva. Para obtener las estadísticas a nivel de paquete TCP es necesario procesar las correspondientes a nivel de trama, a las que se tiene acceso tras realizar un conjunto de modificaciones en el driver de la tarjeta inalámbrica.

Los resultados de la Tabla 4 sólo recogen los valores obtenidos al emplear la tasa binaria máxima, pues, como ya se ha mencionado, el efecto de los errores es más apreciable, aunque se ha comprobado que el efecto del agente Snoop es igualmente beneficioso para el resto de regímenes binarios. Las malas condiciones del canal radio acarrean una pérdida elevada del rendimiento, como se puede ver en las estadísticas que se obtienen de pérdida de paquetes (Perd pqt) y de longitudes media y máxima de las ráfagas erróneas. Aunque éstas afectan mayoritariamente a los segmentos de datos, también se observa cierta tasa de pérdida en los reconocimientos enviados por el TM. Como se ha mencionado anteriormente, todas estas pérdidas activan los mecanismos de control de congestión implementados en TCP, que reaccionan frente a algo para lo que no fueron inicialmente diseñados. La consecuencia directa es que, sin la presencia del agente Snoop, el rendimiento que se obtiene varía en un rango muy amplio, pudiéndose llegar a observar valores por debajo de los 700 Kbps. Sin embargo, con el módulo Snoop cargado, la variabilidad se reduce notablemente y el rendimiento global se estabiliza entre 1.5 y 2.5 Mbps, incluso siendo las condiciones del canal más severas, como lo refleja el hecho de que la tanto la pérdida de paquetes como las longitudes de las ráfagas de errores sean mayores, tal y como se muestra en la Fig. 8.

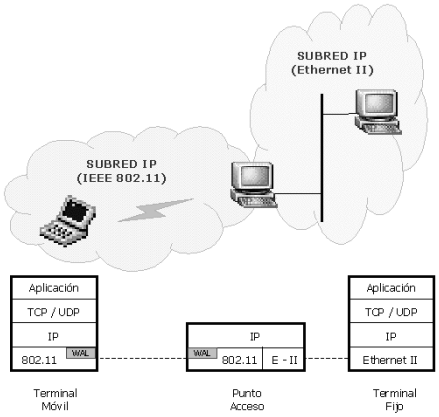


Figura 7. Plataforma de medidas experimental

La Fig. 9 demuestra que las prestaciones del agente Snoop son asimismo beneficiosas para otras tasas binarias. Se puede ver que en este caso (trabajando a 2 Mbps) el rendimiento se mantiene estable en torno a 1 Mbps y que, además, en la mayoría de las ocasiones, la presencia del agente causa un aumento en la eficiencia, a pesar de que las condiciones del canal sean peores.

Además, como se puede ver en la Tabla 5, el agente Snoop reduce el número de reconocimientos duplicados y triplicados que llegan al TF, eliminándolos en el PA. De esta manera no se activan los mecanismos de control de congestión en el TF, observándose una reducción considerable del número total de retransmisiones así como del número máximo de veces que un mismo segmento es retransmitido. Como se ha visto anteriormente, estos mecanismos de control de congestión provocan largos periodos de inactividad en el transmisor, siendo la causa principal de la pérdida de rendimiento del protocolo TCP sobre enlaces IEEE 802.11b caracterizados por una baja SNR. El agente Snoop reduce estos periodos de inactividad mediante la

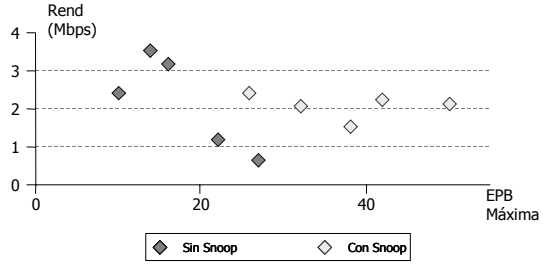


Figura 8. Rendimientos con y sin Snoop frente a la EPB Máxima trabajando a 11 Mbps

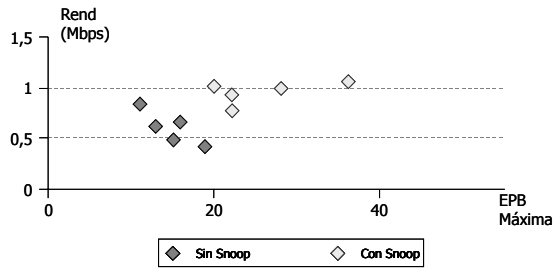


Figura 9. Rendimientos con y sin Snoop frente a la EPB Máxima trabajando a 2 Mbps

retransmisión local de los paquetes erróneos, como se ve en la Tabla 5, permitiendo que el rendimiento global de la transferencia aumente y se mantenga relativamente constante independientemente de las condiciones del canal radio.

5 Conclusiones

Se ha realizado un exhaustivo análisis del comportamiento del protocolo TCP sobre la WLAN IEEE 802.11b, observándose pérdidas de rendimiento muy elevadas, principalmente debidas a la presencia de grandes periodos de inactividad, cuando el canal se caracteriza por una SNR baja. Se ha demostrado, asimismo, que la aparición de dichos periodos es consecuencia directa de los mecanismos de control de congestión que se han incorporado a las implementaciones de TCP, y que no fueron diseñados teniendo en cuenta las características de los entornos inalámbricos.

Esta caracterización sirve de base para afrontar el diseño e implementación de mecanismos que ayuden a mejorar las deficiencias observadas. En este artículo se han evaluado las prestaciones el agente Snoop como ejemplo de dichos mecanismos. Como una extensión al trabajo de Balakrishnan, el análisis se ha realizado sobre la plataforma IEEE 802.11b y de una manera totalmente experimental, al contrario que otros estudios previos, en los que el canal se trataba de emular mediante el uso de generadores de patrones de error. El agente Snoop se ha integrado como módulo dentro de un PEP genérico, denominado WAL, principal resultado del proyecto WINE.

Tabla 5. Parámetros de operación del agente Snoop

Medida	Segmentos Retx Snoop	ACK suprimidos Snoop
1	620	2084
2	377	655
3	361	1233
4	386	859
5	342	809

Tras una exhaustiva campaña de medidas sobre una plataforma experimental se concluye que cuando las condiciones del canal son buenas, el efecto del agente Snoop sobre el rendimiento alcanzado es despreciable, pues no introduce sobrecarga alguna. Sin embargo, en un canal caracterizado por una SNR baja y por la presencia de un elevado número de paquetes erróneos, normalmente dispuestos en ráfagas, el agente Snoop mejora el rendimiento de TCP, ocultando las pérdidas al emisor y estabilizando el rendimiento (entre 1,5 y 2,5 Mbps para un régimen binario de trabajo de 11 Mbps).

Como extensión a este trabajo, y aprovechando la capacidad ofrecida por la WAL, se caracterizará el efecto conjunto de diversas técnicas de nivel de enlace, analizando la interacción del agente Snoop con otros mecanismos, como por ejemplo un módulo corrector de errores (FEC, Forward Error Correction).

Agradecimientos

Este trabajo ha sido realizado en el marco del proyecto IST-1999-10028 “Wireless Internet Networks” (WINE), financiado por la Unión Europea dentro del contexto del programa IST (Information Society Technologies). Los autores agradecen las contribuciones de los integrantes del consorcio: VTT Electronics (Finlandia), Philips Research Monza (Italia), Universidad de Roma “La Sapienza” (Italia), AQL (Francia), Cefriel (Italia), Intracom S.A. (Grecia), Universidad de Atenas (Grecia), Acorde (España), Universidad de Cantabria (España) y la Universidad de Queen de Belfast (Irlanda).

Tabla 4. Parámetros de conexiones TCP sobre un enlace IEEE 802.11b con baja SNR

Rb (Mbps)	Snoop	Medida	Rend (Mbps)	Perd Pqt (DATOS)	EPB Media (DATOS)	EPB Max (DATOS)	Perd Pqt (ACK)	EPB Media (ACK)	EPB Max (ACK)	Retx	Max Retx	Inact Max (seg)	ACK Dup	Triple ACK
		1	3,55	174	2,28	14	20	1,25	4	177	3	0,64	341	30
		2	3,16	81	1,88	16	64	1,45	4	84	4	2,08	313	23
11	No	3	2,39	215	1,69	10	0	0	0	217	5	1,28	577	61
		4	0,67	255	1,53	27	1	1	1	264	9	39,68	732	103
		5	1,19	309	2,06	22	0	0	0	314	6	8,32	586	68
		1	1,52	622	1,59	38	1	1	1	97	3	1,71	203	2
		2	2,07	451	2,05	32	1	1	1	112	3	6	112	3
11	Si	3	2,41	419	3,22	26	1	1	1	229	3	0,66	49	3
		4	2,23	413	3,27	42	14	1,27	3	143	3	1,34	59	3
		5	2,09	469	3,23	50	16	1,06	2	218	2	1,7	62	3

Referencias

[1] J. Border, M. Kojo, J. Griner, G. Montenegro, Z. Shelby, "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations", RFC 3135, Junio 2001.

[2] H. Balakrishnan, S. Seshan, R. H. Katz, "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks", ACM Wireless Networks, Vol. 1, Diciembre 1995.

[3] M. García, R. Agüero, L. Muñoz, P. Mähönen, "Behavior of UDP-Based Applications over IEEE 802.11 Wireless Networks", 12th IEEE International Symposium on Personal Indoor and Mobile Radio Communication, PIMRC 2001, San Diego (USA), October 2001. Vol II, pp 72-77.

[4] A. Kamerman. and G. Aben. "Net Throughput with IEEE 802.11 Wireless LANs", Wireless Communications and Networking Conference, Chicago, Sep. 2000.

[5] V. Jacobson, R. Braden, D. Borman, "TCP Extensions for High Performance", Request For Comments RFC 1323, mayo 1992.

[6] V. Jacobson, M. J. Karels, "Congestion Avoidance and Control", Proceedings of ACM SiGCOMM, noviembre 1999.

[7] M. Mathis, J. Mahlavi, S. Floyd, A. Romanow, "TCP Selective Acknowledgement Options", Request For Comments RFC 2018, octubre 1996.

[8] P. Karn, C. Partridge, "Improving Round-Trip Time Estimates in Reliable Transport Protocols", Proceedings of ACM SiGCOMM, 1987.

[9] R. Braden, "Requirements for Internet-Hosts-Communications Layers", Request For Comments, RFC 1122, octubre 1989.

[10] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," in Proceedings of ACM/SIGCOMM '96, Stanford, CA, agosto 1996.

[11] L. Bechetti, F. Delli Priscoli, T. Inzirelli, P. Mähönen, L. Muñoz, "Enhancing IP Service Provisión over Heterogeneous Wireless Networks: A Path towards 4G", IEEE Communications Magazine, agosto 2001, Vol 39, Nº 8, pp 74-81.

Sesión 3B

Agentes y redes programables

Integración de Servicios Web mediante un Modelo Composicional de Agentes Software*

M. Amor, L. Fuentes y J.M Troya
Departamento de Lenguajes y Ciencias de la Computación. Universidad de Málaga
Campus de Teatinos, s/n. E.T.S.I. Informática
29071, Málaga
Telephone: 952 13 28 10 Fax: 952 13 13 97
{pinilla, lff, troya}@lcc.uma.es

***Abstract**: Web services are the newest trend in information technology, being considered the most used alternative for building distributed open systems. Although Web services currently involve a single client–server access, the market is demanding cooperative Web services to provide a global solution. Recently software agents appear as a good option that can cope with the control of Web services composition, obtaining an integral solution. This paper presents an approach to integrate Web services and software agent technologies. The basis of our approach is the use of component technology for the development of adaptive software agents. Our compositional software agent performs automated software composition, which makes possible to plug Web services into the agent functionality, and compose them during agent interaction.*

1 Introducción

Hoy en día los servicios Web (*Web Services*) pueden verse como una tecnología emergente para el desarrollo de aplicaciones distribuidas sobre Internet. La clave del rápido avance de esta tecnología es que ofrecen una plataforma para la integración A2A (*Application to Application*) y el desarrollo de aplicaciones B2B (*Bussines to Bussines*). Aunque es reciente y se encuentra aún en periodo de evolución, esta tecnología integra tres áreas importantes del desarrollo tecnológico como son la Web, las plataformas distribuidas y las transacciones electrónicas, que conforman la base de desarrollo de los servicios Web. Actualmente, los protocolos de comunicación de aplicaciones distribuidas CORBA-IIOP o Java-RMI, aunque son fundamentales, no son candidatos a convertirse en el único elemento de comunicación A2A. Realmente es la tecnología de servicios Web la que se perfila como protocolo único de comunicación, definiendo estándares ampliamente aceptados para especificar las capacidades y funcionalidades principales de estas plataformas.

Aunque el concepto de un Servicio Web está en evolución, existen ciertas características que debe presentar. Estos servicios deben ser accesibles a través de protocolos conocidos y ampliamente desarrollados, y deben de permitir el envío y recepción de mensajes codificados en XML[1].

En la actualidad el mercado demanda soluciones integrales que aumentan la complejidad de las aplicaciones sobre la Web. A pesar de ofrecer una mayor interoperabilidad con respecto a otras tecnologías, la infraestructura de los servicios Web carece de mecanismos y herramientas de composición que permitan la coordinación e integración de diferentes servicios más simples en una sola aplicación. Hasta ahora, la única propuesta para componer servicios Web es la definición de lenguajes de coreografía, que sirven de base para coordinar distintos servicios Web[2,3].

Por otra parte, existen ciertos requisitos que pueden limitar su aplicación en Internet. Los servicios Web son componentes pasivos que no pueden adaptarse o dar aviso de nuevos contenidos o cambios en su entorno para ofrecer un mejor servicio. Esta es una restricción importante teniendo que cuenta que Internet es un entorno abierto, y que las fuentes de información y los enlaces de comunicación pueden aparecer y desaparecer con bastante frecuencia.

Para salvar las limitaciones que presentan los Servicios Web se propone el uso de la Tecnología de Agentes. Un agente es un componente software autónomo y activo que es capaz de interactuar mediante un protocolo de comunicación. Por lo tanto, un agente software es capaz de acceder a un Servicio Web en nombre de un cliente, y podría alertar o notificar cuando hay alguna actualización. Pero sobre todo, un agente software es capaz de integrar y controlar la composición de diferentes servicios Web de forma transparente para proporcionar una solución global.

Sobre la interacción de agentes y servicios Web existen algunas propuestas. FIPA (*Foundation for Intelligent Physical Agents*)[4], organización de estandarización para sistemas de agentes, ofrece dos propuestas. Por un lado define una

* Esta investigación ha sido financiada en parte por el proyecto CITYT TIC 2002-04309-C02-02.

especificación para la integración de sistemas software en general, como componentes COTS (Commercial off-the-shelf) desarrollados en CORBA, Java, o servicios Web, en sistemas multi-agente a nivel de aplicación. Y por otro propone una integración de más bajo nivel, que permita a los agentes software utilizar la infraestructura de los servicios Web (por ejemplo, el servicio de transporte de mensajes). A este nivel también existe una propuesta de AgentCities para integrar la arquitectura de servicios Web dentro de un marco de trabajo propio [5].

Sin embargo, a pesar de estos esfuerzos, no es fácil lograr la integración de ambas tecnologías. Las plataformas de agentes actuales carecen de la flexibilidad necesaria para abarcar la amplia oferta de estándares relacionados con los servicios Web. Y adaptar un agente software para que soporte un nuevo protocolo o un nuevo lenguaje de descripción supone volver a programar el agente y reemplazarlo.

Por lo tanto, una posible solución sería diseñar y construir agentes más flexibles y adaptables a todas estas nuevas tecnologías, estándares, lenguajes y plataformas. El elemento clave de nuestra propuesta es una arquitectura basada en componentes para diseñar y desarrollar agentes software [6, 7], donde el comportamiento del agente es ofrecido por diferentes componentes. Nuestra propuesta se beneficia de la flexibilidad de los sistemas orientados a componentes, donde es más fácil incorporar, sustituir o reutilizar componentes sin que el resto del sistema se vea afectado. Para conseguir una mejor descomposición funcional del agente software también hemos aplicado la tecnología de separación de aspectos [8, 9], de forma que el comportamiento y la coordinación de un agente software están separados internamente en entidades diferentes dentro de la arquitectura. Esta separación permite al agente acceder a servicios Web como parte de su funcionalidad, y coordinar la interacción, por ejemplo, con distintos servicios en una misma conversación.

El artículo está organizado de la siguiente forma: En la sección 2 se muestra de forma general el estado del arte de los servicios Web y los modelos de referencia propuestos por FIPA; en la sección 3 se describe nuestra arquitectura composicional de agentes software, para introducir y mostrar, en la sección 4, nuestra propuesta de integración de agentes software y servicios Web, y finalizar con las conclusiones de nuestro trabajo.

2 Estado del Arte

2.1 Los Servicios Web

Un Servicio Web es un componente software pasivo que acepta peticiones de servicio, realiza algún proceso, y devuelve una respuesta. Para acceder a un servicio sólo es necesario conocer los métodos que ofrece el servicio Web, e invocarlos usando los protocolos estándar de Internet, codificando los parámetros en formato XML, y recibir la respuesta también en formato XML.

Por tanto XML proporciona el lenguaje común para intercambiar información. Constituye la base para desarrollar los protocolos, lenguajes de descripción y comunicación de acuerdo a las definiciones del consorcio W3C. Los más relevantes se detallan a continuación:

– SOAP (*Simple Object Access Protocol*) proporciona el protocolo de comunicación de servicios Web. Es fundamental para la comunicación mediante paso de mensajes, sin embargo más que definir un nuevo protocolo de transporte, SOAP trabaja sobre protocolos de transporte existentes como HTTP, SMTP o MIME.

– La descripción de un servicio se efectúa en WSDL (*Web Service Description Language*). Una descripción WSDL es un documento XML en el que se identifica el servicio y se facilita el esquema para poder invocarlo, incluyendo información sobre los protocolos que es posible utilizar y el formato de los mensajes.

– Estas descripciones residen en directorios de páginas amarillas donde los servicios Web son registrados. UDDI (*Universal Description, Discovery and Integration registry*) [10] es un directorio de servicios a nivel mundial en el que los proveedores de servicios podrán registrar sus propios servicios. Los clientes podrán acceder al directorio mediante UDDI para consultar la descripción de un servicio, la información sobre el proveedor y todo lo necesario para contactar con él.

Además del uso de estos estándares, los usuarios y desarrolladores de Servicios Web deben acordar los términos que van a usar cuando describen un servicio o el contenido de un mensaje. Para establecer un acuerdo semántico sobre estos términos (operaciones, parámetros de los mensajes) se usan ontologías. Una ontología permite describir el vocabulario adoptado en la descripción de un servicio Web, sus propiedades y relaciones. RDF[11] y DAML+OIL[12] son ejemplos de lenguajes estándar para describir ontologías.

Coreografiar se puede definir como describir las relaciones y patrones de uso entre servicios Web. Para coreografiar servicios Web es necesario describir estados, conversaciones, gestionar el ciclo de vida, o definir el comportamiento que muestra un servicio durante un intercambio de mensajes. Ante la necesidad de componer servicios Web diferentes compañías definieron lenguajes de coreografía propios, como WSFL de IBM, XLANG de Microsoft, o BPEL4WS desarrollado por IBM, Microsoft y BEA. Posteriormente, W3C propuso los lenguajes estándar de descripción de coreografías WSCI (*Web Service Choreography Interface*) [3] y WSCL (*Web Service Conversation Language*) [2], que todavía se encuentran en proceso de definición.

WSCL surge del sector del B2B, donde es necesario llevar a cabo transacciones, que consisten en varias peticiones sobre un servicio. En este caso no es suficiente con especificar que documentos XML recibe o envía un servicio Web, sino que también es necesario describir en que orden se deben de intercambiar estos documentos dentro de una misma transacción. Por ejemplo, para realizar una compra a través de un servicio Web, es necesario: primero registrarse (iniciar una conexión en una cuenta de usuario); en segundo lugar elegir el producto; y finalmente efectuar el reembolso una vez se recibe la factura. Una descripción en WSCL incluye varias secuencias válidas de intercambio de mensajes, y las descripciones del formato de los documentos XML que el servicio Web puede aceptar o transmitir.

WSCI es más complejo que WSCL, ya que no solamente describe el orden de las operaciones, sino que además permite describir relaciones entre el contenido de los mensajes de entrada y de salida, las reglas que gobiernan el intercambio de mensajes, y que operaciones se llevan a cabo dentro de una misma transacción o sesión. Una descripción WSCI podría considerarse como la descripción de un caso de uso del servicio Web.

WSCL y WSCI no describen de forma completa un servicio Web, ya que ofrecen sólo una descripción dinámica de servicio que debe completarse con una descripción estática, por ejemplo en WSDL. Esta descripción estática da la información necesaria para acceder al servicio (protocolo de transporte, localizadores). De esta forma, una vez que el cliente determina, consultando una descripción en WSCL o WSCI, la secuencia de mensajes que debe intercambiar con el servicio Web, recurre a la descripción WSDL para acceder al servicio.

2.2 Los Agentes FIPA

La organización de estandarización para sistemas de agentes FIPA ha propuesto un modelo de referencia general que define los componentes o entidades que deben existir en un plataforma de agentes FIPA. Estas entidades, mostradas en la Figura 1, son:

- Los agentes, procesos computacionales autónomos que ofrecen uno o varios servicios. Un agente es identificado de forma unívoca por un *Agent Identifier* (AID). Los agentes se comunican usando un lenguaje de comunicación de agentes (ACL).
- El componente *Directory Facilitator* (DF), ofrece un servicio de directorio. En este directorio los agentes pueden publicar sus servicios, o pueden consultar los servicios que ofrecen otros agentes.
- El componente *Agent Management System* (AMS) controla el acceso y uso de la plataforma. Este componente mantiene un directorio de AIDs de agentes registrados que permite buscar agentes por su identificador mediante un servicio de directorio. Cada componente debe registrarse en el componente AMS para obtener un AID válido.
- El Servicio de Transporte de Mensajes (MTS) establece el mecanismo de comunicación entre agentes.
- Los componentes software describen cualquier sistema software que sea accesible para un agente, aunque no forme parte de él.

Además de este modelo de referencia, FIPA define una especificación para la integración de sistemas software en general, como componentes COTS (Commercial off-the-shelf) en CORBA o Java, o servicios Web, en sistemas multi-agente [13]. Esta especificación permite que un agente pueda acceder a los servicios de un componente aunque no disponga de la funcionalidad necesaria para realizar este acceso, y además facilita la publicación y gestión dinámica de estos nuevos recursos. La integración se lleva a cabo extendiendo el modelo de referencia para incluir dos nuevos roles de agentes, el agente ARB (*Agent Resource Broker*) y el agente envoltorio (*wrapper*) (también mostrados en la Figura 1). Realmente, no constituyen dos nuevos agentes, sino un conjunto de funciones o servicios que debe presentar un agente para tener ese rol, lo cuales están definidos en una ontología. Un agente ARB contiene descripciones de los servicios ofrecidos por sistemas software, y ofrece a otros agentes un servicio de directorio para localizar estos sistemas software, identificados por sus descripciones.

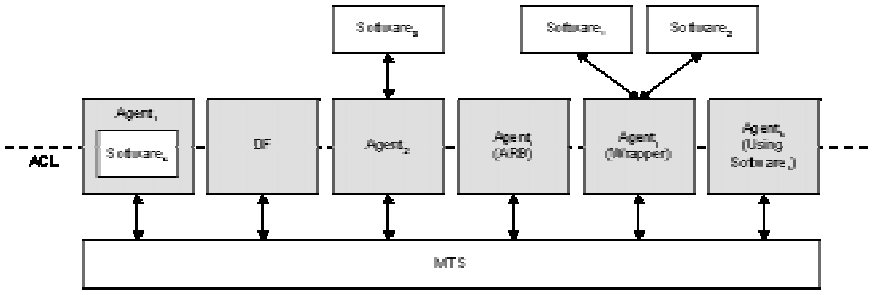


Figura 1. Modelo de Referencia para la Integración de Software en Agentes (imagen extraída de [13]).

Un agente ARB debe registrarse en el agente DF como proveedor de un servicio de directorio de recursos descritos mediante una lista de propiedades, incluyendo su localización. Un agente que proporcione este servicio debe soportar la ontología FIPA-ARB, que define acciones y predicados para registrar y buscar servicios software.

Un agente envoltorio es el encargado de interactuar de forma dinámica con un sistema software. Este agente ofrece un servicio de acceso al sistema software, y permite a los agentes cliente invocar acciones sobre el sistema software en cuestión. Estas acciones estarán contenidas en mensajes descritos en ACL. Un agente envoltorio debe soportar la ontología FIPA-Wrapper, que define acciones y predicados para inicializar y emitir peticiones a sistemas software, y registrarse en el DF para ofrecer sus servicios. De la implementación de un agente envoltorio y su comunicación con el sistema software se ocupan los desarrolladores del agente y los proveedores de herramientas de soporte del sistema software.

Según esta especificación, para acceder a un servicio Web, un agente software cliente debe llevar a cabo las siguientes acciones:

- El agente cliente debe buscar en el DF agentes que ofrezcan un servicio ARB para consultar acerca de la existencia de un sistema software que cumpla una descripción y unas necesidades específicas. Como resultado de esta consulta, el agente ARB devuelve la localización de un sistema software que cumple estas restricciones.
- Si el agente cliente no puede interactuar directamente con el sistema software, debe acudir a un agente envoltorio. De nuevo el agente cliente deberá consultar el DF para ver si hay un agente que ofrezca el servicio de envoltorio para el sistema software deseado. El agente cliente inicia la interacción con el sistema software a través de los servicios del agente envoltorio. Éste es el encargado de invocar las operaciones sobre el sistema software en respuesta a las peticiones enviadas por el agente cliente mediante mensajes convencionales de FIPA.

La principal diferencia entre un agente envoltorio y un agente que interactúa con un servicio software directamente, es que el primero es capaz de contactar con nuevos servicios de forma dinámica. Esta diferencia también se ve reflejada en el DF.

3 Arquitectura Composicional de Agentes Software

En esta sección presentamos un modelo composicional para el diseño de agentes software. Los agentes son sistemas que son capaces de llevar a cabo tareas y acciones de forma autónoma para conseguir una serie de objetivos. La principal característica de nuestro modelo es que la funcionalidad y comportamiento del agente se encapsula en diferentes componentes software.

La parte más crítica del diseño de un sistema basado en componentes es su descomposición funcional. Normalmente dentro de un mismo componente encontramos, además de su comportamiento interno, código relativo a la comunicación con el resto de componentes. Esto hace difícil reutilizar sólo la parte funcional de un componente cuando varían los componentes con los cuales se coordina. Para mejorar la descomposición funcional del agente aplicamos el principio de separación de aspectos, y modelamos la coordinación entre componentes en una nueva entidad, a la que llamaremos conector. Entre las ventajas de esta separación está el aumento de la reutilización, tanto de la funcionalidad del componente que ya no incluye código específico para la comunicación con otros componentes, como de los conectores que determinan los patrones de interacción. La separación del aspecto de coordinación en una entidad independiente ya ha sido utilizada en trabajos anteriores con éxito[14].

La Figura 2 muestra el diagrama de clases UML de la arquitectura propuesta para el diseño de agentes software basada en componentes y conectores. Usamos estereotipos UML para modelar las entidades de nuestro modelo, denominadas <<Mediador>>, <<Interfaz>>, <<Componente>>, y <<Conector>>.

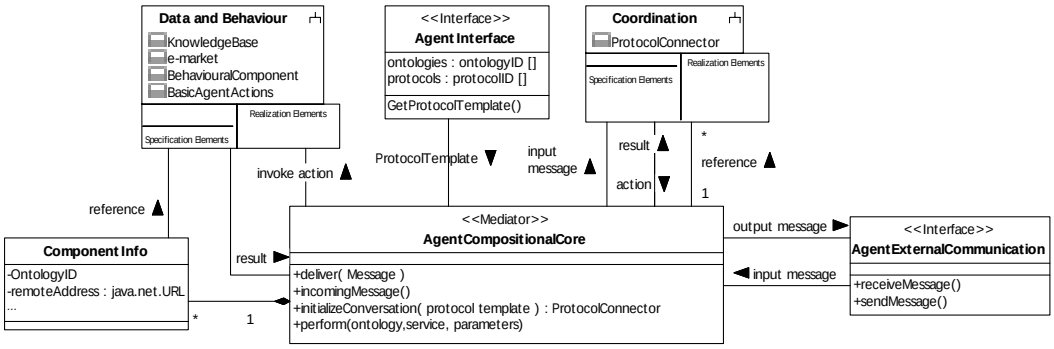


Figura 2. Arquitectura Composicional de Agentes Software.

Por simplicidad, la Figura 2 no incluye la descomposición en clases del subsistema "*Data and Behaviour*" que contiene las clases que representan datos y comportamiento. Igualmente el subsistema "*Coordination*" aparece en forma de paquete, que incluye a la clase conector y otras clases relacionadas ([7] describe de forma detallada el diagrama de clases UML).

Los conectores coordinan las diferentes interacciones en las que participa el agente. Un agente puede participar en más de una conversación de forma simultánea, siendo cada una controlada por un conector diferente que encapsula un protocolo de interacción. Los conectores sólo se diferencian en el protocolo que coordinan, que es especificado por una plantilla que se carga durante su inicialización. De esta forma es posible agregar nuevas plantillas de protocolos descritas en XML, e incluso soportar nuevos protocolos en tiempo de ejecución [7].

Los componentes encapsulan datos y comportamiento. Algunos componentes de comportamiento están siempre presentes en la arquitectura porque ofrecen la funcionalidad básica del agente, como por ejemplo enviar un mensaje a otro agente (por ejemplo, el componente *BasicAgentActions*), o almacenar datos (en el componente *KnowledgeBase*). El comportamiento que es específico de un dominio de aplicación como por ejemplo la compra y venta dentro de un mercado electrónico, también son facilitadas por componentes (Ej.: el componente *e-market*). Los componentes son añadidos a la funcionalidad del agente cuando ésta se necesita, y pueden buscarse o reemplazarse durante el ciclo de vida del agente si aparecen nuevas necesidades. Este comportamiento específico de un dominio de aplicación también puede ser facilitado por componentes COTS o servicios Web.

Los componentes de interfaz controlan las interacciones del agente con su entorno. El componente *AgentExternalCommunication* (AEC) recibe los mensajes de entrada enviados por otros agentes, y envía mensajes a una plataforma de agentes para su entrega al agente destinatario. Los mensajes de entrada son procesados por este componente, que podría descartar aquellos cuya sintaxis no sea correcta según el ACL utilizado. El componente *AgentInterface* (AgInt) contiene la interfaz pública del agente, que es una extensión de los IDLs tradicionales de los componentes software. Por ejemplo, el componente AgInt incluye, además de una descripción de los servicios ofrecidos por el agente, una lista de los protocolos de interacción soportados, las plantillas que describen estos protocolos, y las ontologías que describen los servicios de los componentes registrados.

La función principal del componente mediador *AgenteCompositionalCore* (ACC) es la de llevar a cabo la composición dinámica de los componentes y conectores durante una interacción. El componente ACC recibe los mensajes de entrada procesados y aceptados por el componente AEC, pertenecientes a interacciones activas del agente, y los remite al conector que controla esa conversación. Si el mensaje pertenece a una nueva interacción, este componente es el encargado de iniciar un nuevo conector adecuado que controle la conversación a partir de la plantilla de protocolo adecuada. Para llevar a cabo la composición dinámica, este componente mantiene información sobre los componentes que proporcionan la funcionalidad en el agente. Cada componente registrado es representado por una instancia de la clase *ComponentInfo*, e identificado por una especie de nombre de rol, que será utilizado posteriormente durante la composición para identificar el componente.

Como el papel que un componente desempeña dentro del agente viene determinado por la ontología que describe los servicios que ofrece, usamos el identificador de la ontología para identificarlo. Además, es necesario guardar información sobre la localización del

componente. Si el componente es una instancia contenida en el agente, la clase *ComponentInfo* contendrá una referencia local al componente. En otro caso, es decir, si el componente es un componente COTS externo o un servicio Web, el atributo *remoteAddress* contendrá la URI que localiza al componente.

La composición dinámica relaciona en tiempo de ejecución los conectores de coordinación con los componentes que proporcionan el comportamiento. Una característica relevante de nuestro modelo es que los componentes y los conectores no mantienen referencias directas entre ellos, aumentando su independencia, y por tanto su reutilización. Además, la composición dinámica permite incluir nuevos componentes o actualizar los ya registrados en el agente de forma dinámica, sin necesidad de recompilar y reemplazar el agente. Esta característica hace posible agregar fácilmente nuevos componentes y servicios Web como parte de la funcionalidad del agente. También permite, en tiempo de ejecución, reemplazar el proveedor registrado para un servicio Web por otro que ofrezca un servicio más eficiente, o con un menor coste. Por ejemplo, un agente puede examinar los diferentes proveedores de un mismo servicio Web, y decidir en tiempo de ejecución recurrir al que ofrezca un menor tiempo de respuesta o se encuentre más cercano a la localización actual del agente.

4 Integración de Servicios Web

En nuestro modelo la integración de servicios Web en un sistema de agentes se realiza dentro del marco de una interacción o conversación con otro agente o recurso en la Web. En nuestro caso el agente es capaz de invocar directamente las operaciones del servicio Web, sin necesidad de buscar un agente envoltorio que acceda al servicio por él. Nuestra propuesta podría considerarse una solución intermedia entre los agentes que interactúan directamente con un servicio Web, y un agente envoltorio que es capaz de acceder a partir su descripción. Una de las ventajas de nuestra propuesta frente a un agente envoltorio es que éste no es capaz de resolver el contenido semántico de la descripción de un servicio, ya que simplemente se encarga de gestionar y ofrecer el acceso. Igualmente en nuestra propuesta el agente puede incluir criterios propios para decidir cambiar el proveedor del servicio Web, adaptándose de forma personalizada al entorno. Por otro lado, la interacción del agente con componentes COTS es homogénea, pudiéndose incluso sustituir componentes COTS por servicios Web y viceversa fácilmente. Además, reducimos el número de agentes involucrados, y el coste de comunicación (tiempo, ancho de banda).

4.1 Descripción de los Servicios ofrecidos por un Agente

Nuestros agentes son capaces de interpretar la descripción de un protocolo y ejecutarlo. Esta descripción incluye no sólo el formato de los mensajes, y las reglas de intercambio, sino que también especifica que acciones debe llevar a cabo el agente durante su participación en la interacción. Durante la ejecución, se establece una correspondencia entre estas acciones, y los servicios que ofrecen los componentes registrados en el agente.

Para establecer esta correspondencia los componentes deben ofrecer sus servicios a través de interfaces bien definidas. Aunque tradicionalmente, la interfaz pública presenta una descripción sintáctica de los servicios que ofrece el componente software, también es necesario establecer una descripción semántica sobre los términos utilizados en la interfaz.

Por eso nuestra propuesta considera el uso de ontologías para describir la interfaz pública de un componente. Las ontologías permiten establecer un acuerdo semántico sobre términos y objetos comunes, y describir relaciones entre ellos, por lo que resultan una potente herramienta para poder alcanzar un mayor grado de interoperabilidad semántica. El lenguaje DAML+OIL, definido sobre XML y RDF ofrece un amplio conjunto de constructores con los que crear ontologías y marcar información para que pueda ser interpretada por un agente. DAML-S [15] es una de las ontologías definida sobre DAML+OIL para describir las propiedades y capacidades de un servicio Web. Una descripción DAML-S complementa las descripciones tradicionales de servicios Web, por ejemplo en WSDL, definiendo lo que un servicio puede hacer. Una descripción en DAML-S contiene: la interfaz ofrecida y también la requerida por el componente; un modelo computacional del servicio; y una descripción de cómo utilizarlo, que se hace corresponder con la interfaz pública tradicional.

Por tanto es posible extender el uso de DAML-S para describir cualquier componente software, adecuando la descripción sobre su uso a la interfaz ofrecida por el componente, que suele depender de la implementación (por ejemplo, si es un componente CORBA, esta descripción será una IDL, a partir de la cual es posible invocar un servicio). Esta descripción será utilizada en tiempo de ejecución para invocar el servicio sobre el componente. Así es posible definir una interfaz homogénea para cualquier

componente software, que al ser declarativa puede ser interpretada por un agente antes de decidir invocar un servicio, y también puede ser utilizada para validar la invocación de dicho servicio en función de la descripción de un protocolo. Además, esta descripción formará parte de la interfaz del agente como parte de los servicios que éste ofrece. Por ejemplo, si un agente registra un servicio Web para realizar transacciones bancarias, éste es capaz de ofrecer el saldo o realizar una transferencia para otros agentes durante una conversación.

4.2 Invocación de Servicios Web

A continuación mostraremos como se invoca un servicio Web durante la participación del agente en una interacción.

Supongamos un agente software que está participando en una subasta electrónica en nombre de un usuario. Nuestro agente está dotado de la coordinación necesaria para seguir el protocolo de negociación que rige la subasta y de la funcionalidad adecuada para generar pujas con el objeto de conseguir el artículo subastado. Además, se establece la necesidad de que el ganador de una subasta haga efectivo el pago inmediatamente después de haber sido proclamado ganador, ya que si no dispone de efectivo en ese momento se elige como ganador al agente que realizó la segunda puja más alta. Por tanto, el agente debe ser capaz de realizar esta transferencia bancaria sin demora. Por cuestiones de seguridad, las entidades bancarias ofrecen sus servicios *online* a través de aplicaciones seguras, minimizando la distribución de código e información que permita un acceso no permitido al servicio. Por tanto, no es natural que esta funcionalidad sea ofrecida por un componente interno del agente, sino que por el contrario el agente recurrirá al servicio Web de la entidad bancaria donde residen los fondos. En consecuencia, es necesario que el agente, durante su participación en la subasta, realice una transferencia bancaria, invocando el servicio Web correspondiente.

La Figura 3 muestra como colaboran los componentes de la arquitectura del agente para llevar a cabo la transferencia bancaria ofrecida por el servicio Web. La participación del agente en la subasta es controlada por el conector *Auction*. Cuando el agente recibe el mensaje del subastador informándole que ha resultado ganador de la subasta, el conector solicita que se lleve a cabo la transferencia de la cantidad pujada a una cuenta bancaria dada. La acción de invocar el servicio de transferencia es ofrecido por el componente *Bank* (mensaje 1 en la Figura 3). Sin embargo no es el conector que se encarga de invocar directamente este servicio sobre el componente registrado, sino que es el componente Mediador el que recoge esta petición. El componente Mediador se encarga de reenviar esta petición al componente que ofrece ese servicio, y de enviar el resultado de vuelta al conector que lo solicitó.

Obsérvese que el componente Mediador es el que lleva la cabo la composición dinámica entre componentes y conectores, gestionando las invocaciones de servicios realizadas por los conectores y delegándolas sobre los componentes adecuados. Como comentamos anteriormente los componentes registrados son direccionados por el identificador de la ontología que los describe, por lo que el componente Mediador sólo tiene que recuperar la referencia del componente de la clase *ComponenteInfo* que ofrezca esa funcionalidad (en el ejemplo, la ontología es *Bank*, y describe los servicios que ofrece una entidad bancaria). Este paso esta reflejado en el mensaje 2 y 3 de la Figura 3. Como puede haber más de un componente registrado para una misma ontología (el usuario puede tener cuenta en varios bancos), el componente Mediador elegirá el componente que mejor se ajuste a una determinada regla. Por ejemplo, se puede elegir el servicio Web que se encuentre más cerca al agente en el momento de realizar la petición, o aquel que dado el tipo de operación cobre la comisión más baja. Ésta es una característica de nuestro modelo que nos permite construir agentes mas adaptables, debido al mínimo acoplamiento entre los componentes internos del agente.

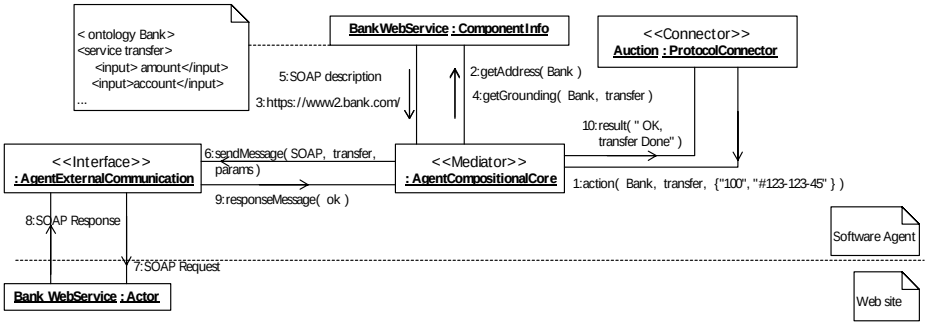


Figura 3. Diagrama de Colaboración de la Invocación de un Servicio Web.

Volviendo a nuestro ejemplo, una vez que el componente Mediador tiene la localización del servicio Web bancario adecuado establece la correspondencia entre la petición general emitida por el conector y la llamada al servicio correcta (éste es el *grounding* en DAML-S),contenida en la descripción DAML-S del componente. Para este caso concreto, el componente construye un mensaje SOAP que será enviado a través del componente AEC al servicio Web concreto (pasos 4 y 5 en la Figura 3). El componente AEC, como responsable de la comunicación del agente con el entorno, es capaz de enviar cualquier tipo de mensaje a través de cualquier tipo de protocolo de transporte o plataforma de componentes. Finalmente, el componente Mediador enviará el resultado de la operación, recibido por el componente AEC (pasos 6, 7, 8, 9 y 10 en la Figura 3). Nos gustaría resaltar que la invocación de un servicio sobre un componente es homogénea sea cual sea la implementación del servicio, algo que simplifica enormemente la programación de este tipo de agentes.

5 Conclusiones

En este trabajo proponemos una arquitectura basada en componentes para construir agentes software que combina componentes y separación de aspectos. En este modelo dos entidades distintas modelan la funcionalidad y la coordinación del agente. Los componentes que modelan el comportamiento (servicios Web, o componentes COTS en general) pueden ser agregados en tiempo de ejecución sin necesidad de reconfigurar el agente, debido a que los componentes son enlazados mediante composición dinámica, y no existen referencias directas entre ellos.

En relación a las otras propuestas mencionadas en este trabajo para la integración y coordinación de servicios Web tenemos que:

– El modelo de referencia propuesto por FIPA para la integración de sistemas software nos parece una solución válida para aquellos agentes que no tienen capacidad para acceder a determinados sistemas software, y que necesitan de la ayuda de un agente envolvente para que medie en su interacción con el servicio Web. En nuestra propuesta los agentes pueden interactuar directamente con un servicio Web, reduciendo así el coste que supone establecer una comunicación adicional con el agente intermediario. Por supuesto, desde el punto de vista de la reutilización, nuestra propuesta permite utilizar nuestros agentes como agentes envoltorio para otras plataformas de agentes.

– Nuestra propuesta permite coreografiar servicios Web como parte del comportamiento de un agente software. Un agente puede coordinar la invocación de diferentes servicios web, como parte de una transición (un conjunto de acciones) en un protocolo. Estas acciones están descritas en XML utilizando las construcciones proporcionadas por DAML-S. Es posible, por tanto, que el agente pueda llevar a cabo una coreografía, por ejemplo, en WSCI incluyendo el intérprete adecuado.

Actualmente estamos trabajando sobre un prototipo de agente software sobre la arquitectura composicional mostrada, compatible con la especificación dada por FIPA, para la cual hemos utilizado la familia de productos J2EE, incluyendo el paquete JAXB y JAXP de procesamiento XML.

6 Referencias

1. Extensible Markup Language (XML). http://www.w3c.org/XML/
2. W3 Consortium. Web Services Conversation Language. http://www.w3.org/TR/wscl10/
3 W3 Consortium. Web Services Conversation Language. http://www.w3.org/TR/wscl10/
4. The Foundation for Intelligent Physical Agents. http://www.fipa.org/
5. AgentCities Web Services Working Group. http://www.agentcities.org/Activities/
6. M. Amor, M. Pinto, y L. Fuentes, J.M. Troya, “Putting Together Web Services and Compositional Software Agents”, ICWE 2003. Oviedo, Spain 2003.
7. M. Amor, L. Fuentes, y J.M. Troya, “ Training Compositional Agents on Negotiation Protocols”. Iberagents 2002, Proceedings of the Fourth Iberoamerican Workshop on Multi-agent Systems, 2002.
8. G. Kiczales et al. “Aspect-Oriented Programming”, Proceedings of ECOOP´97, LNCS 1241, Springer-Verlag, 1998.
9. Aspect-Oriented Software Development. http://www.aosd.net
10. UDDI, Universal Description, Discovery and Integration of Web Services. The UDDI Whitepaper. http://www.uddi.org.
11. Resource Description Framework. http://www.w3c.org/RDF/
12. The DARPA Agent Markup Language. http://www.daml.org/language/
13. FIPA Agent Software Integration Specification. http://www.fipa.org/specs/
14. L.Fuentes y J.M.Troya, “A Java Framework for Web-based Multimedia and Collaborative Applications”, *IEEE Internet Computing*, vol 3, nº2 págs. 55-64. 1999
15. DARPA. DAML-S. http://www.daml.org

Diseño de una plataforma de agentes compatible con FIPA para dispositivos limitados

Guillermo Díez-Andino Sancho, Rosa M García Rioja, Mª Celeste Campo Vázquez
Departamento de Ingeniería Telemática
Universidad Carlos III de Madrid.
Avd. Universidad 30 28911 Leganés

e-mail:{gdandino, rgrioja, celeste}@it.uc3m.es

Abstract *Recent advances in micro-electronic and wireless technologies have fostered the proliferation of small devices with limited communication and processing power. We think that agent technology will be of great help in pervasive systems development. Pervasive systems are inherently dynamic, with devices continually coming and going. Agents are autonomous software entities that can interact with their environment, and therefore they adapt well to such frequent changes. However, the use of Multi-Agent Systems (MAS) in pervasive environments poses important challenges, and it is necessary to adapt their design to meet these challenges.*

The first part of this paper describes the design of a FIPA-compliant agent platform, adapted to the limited devices that work in an ad-hoc network. The authors have presented their proposals to the Working Group FIPA Ad-Hoc, and have been included in the white paper elaborated by this group previous to the standardization. The second part describes the agent platform implementation in real devices, using the Java 2 Micro Edition technology.

1. Introducción

En los últimos años los avances realizados en los campos de la microlectrónica y de los protocolos inalámbricos, han permitido la proliferación de un gran número de pequeños dispositivos con capacidad de cómputo y comunicación. Los más visibles son los nuevos dispositivos personales, como PDAs o teléfonos móviles, pero hay otros muchos que se embeben en el entorno que nos rodea, de forma invisible, pero que al comunicarse con nuestros dispositivos personales, permiten que el entorno físico que nos rodea se adapte a nuestras preferencias y necesidades de forma casi transparente. Esta nueva era de la computación es lo que Mark Weiser describió en su artículo "The Computer for the 21st Century" en 1991 [1] como computación ubicua.

En estos entornos existe una mayor diversidad de dispositivos con diferentes limitaciones hardware para ejecutar aplicaciones. Además, gracias a los protocolos de comunicación inalámbricos, se forman redes en las que los dispositivos entran y salen de forma espontánea. A este tipo de redes se les denomina redes ad-hoc. Tanto por las limitaciones de los dispositivos, como por las características de las redes ad-hoc, parece claro que no es eficente migrar soluciones tradicionales a estos nuevos sistemas [2]. Además, la mayoría de estos nuevos dispositivos no son multipropósito, sino que proporcionan una serie de servicios concretos, pero cuando forman una red ad-hoc, es posible que se compongan estos servicios de forma inteligente, para ofrecer un nuevo servicio que satisfaga las expectativas del usuario de manera transparente al mismo, como quería Weiser.

En este escenario, las aplicaciones software de estos dispositivos limitados deben adaptarse a las restricciones de memoria y procesamiento que proporciona el dispositivo en el que se ejecutan, a una comunicación intermitente y de calidad cambiante, necesitan autonomía para poder alcanzar los objetivos que quiere el usuario, pero sin necesidad de interaccionar continuamente con él, y deben ser capaces de moverse por otros sistemas para poder obtener información o ejecutar tareas que las limitaciones del dispositivo no les permiten realizar de forma local, o la conectividad directa no les permite alcanzar. El paradigma de computación distribuida que se adapta a todas estas características es lo que se denomina Agente [3].

Los desarrollos llevados a cabo a nivel agentes se realizaron para redes como Internet, las plataformas se ejecutaban en PCs sin limitaciones en cuanto a su capacidad de proceso y comunicación, y aunque en su definición se consideraba que se adaptaban a entornos cambiantes, con conectividad intermitente y calidad cambiante, la realidad es que estos estados eran tratados más como excepciones a las que el sistema sabía adaptarse, que como las características habituales del entorno en el que se ejecutaban.

Nuestro trabajo actual se basa en emplear el paradigma de agentes móviles como tecnología middleware para el desarrollo de servicios en redes ad-hoc formadas por dispositivos limitados, que se comunican de forma directa sin necesidad de sistemas centrales. Para ello, tomando como pun-

to de partida el estándar FIPA hemos realizado un nuevo diseño de plataforma de agentes móviles adaptada a los nuevos requisitos, con el mínimo impacto en la especificación. Como tecnología para apoyar la viabilidad de implantación de nuestro diseño, utilizaremos la versión de Java para dispositivos limitados, J2ME.

En la sección 2, realizamos una breve revisión del estándar FIPA. Después en la sección 3, justificamos y proponemos simplificaciones a esta arquitectura para adaptarla tanto a las características de las redes ad-hoc, como a las restricciones de los sistemas limitados en los que se implantaría. En la sección 4, introducimos brevemente la plataforma software J2ME y analizamos la viabilidad de implementar la arquitectura diseñada sobre ella, describiendo parte del desarrollo que hemos realizado hasta la actualidad. Por último, finalizamos con las conclusiones y líneas futuras de nuestro trabajo, en la sección 5.

2. Estándar FIPA

Foundation for Intelligent Physical Agents[1] (FIPA) comenzó sus actividades en 1995 con el objetivo de estandizar aspectos relacionados con la tecnología de agentes y sistemas multiagente. En la actualidad se puede considerar el estándar más ampliamente reconocido y extendido internacionalmente, y se ha convertido en un referente a seguir a la hora de realizar desarrollos basados en agentes.

Las especificaciones FIPA se han agrupado en tres tipos: *component*, que se encargan de estandarizar todas las tecnologías básicas relacionadas con agentes, *informative* que describen posibles soluciones en aplicaciones realizadas con agentes en un determinado dominio y *profiles* que son conjuntos de especificaciones de tipo *component* que permiten validar cuándo una implementación es conforme al estándar.

En *FIPA Agent Management Specification* [4] se describe el modelo de referencia FIPA de plataforma de agentes y se describe la funcionalidad de cada uno de sus componentes, ver Figura 1. Estos componentes son:

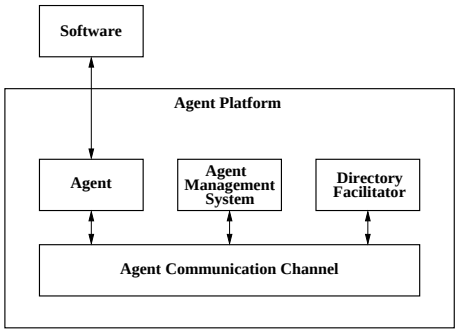


Figura 1: Arquitectura abstracta FIPA

- **Agent Management System** (AMS): que gestiona el ciclo de vida de los agentes, los recursos locales, y los canales de comunicación y proporciona un servicio de páginas blancas, que permite localizar agentes por su nombre.
- **Directory Facilitator** (DF) : que proporciona un servicio de páginas amarillas, que permite localizar agentes por sus capacidades y no por su nombre.
- **Agent Communication Channel** (ACC): que gestiona el envío de mensajes entre agentes de la misma plataforma o de plataformas distintas, y permite la migración de agentes.

3. Diseño de la plataforma

Como hemos visto en el apartado anterior FIPA define en su arquitectura tres elementos funcionales, que proporcionan una serie de servicios básicos para los agentes que residen en la plataforma. Para que una plataforma sea compatible con FIPA, debe tener obligatoriamente estos componentes, y proporcionar las interfaces y funcionalidad definida en sus estándares para cada uno de ellos.

A lo largo de este apartado analizamos cómo se puede simplificar esta funcionalidad para facilitar su implantación en dispositivos limitados, y para que el servicio que ofrecen se adapte a los requisitos impuestos por el entorno cambiante en el que se encuentran.

3.1. Agent Management System

El Agent Management System (AMS) gestiona el ciclo de vida de los agentes, incluidos los estados relacionados con movilidad, es decir, crea y borra agentes y gestiona su paso de una plataforma a otra. Además, el AMS da soporte a un servicio de búsqueda denominado de "páginas blancas", es decir, localizar agentes por su nombre.

Este elemento es fundamental en una plataforma de agentes, y en nuestro diseño actual lo mantenemos con la funcionalidad que se define en el estándar FIPA, pero reduciéndola a un ámbito local en la plataforma, es decir, eliminamos la posibilidad de que el AMS extienda sus búsquedas de agentes a otras plataformas, poniéndose en contacto con AMS remotos.

3.2. Directory Facilitator

La adaptación de la funcionalidad del DF para dispositivos limitados que operan en redes ad-hoc es uno de los temas más importantes de investigación que se está llevando a cabo en este terreno. El Working Group Ad-Hoc de FIPA centra su trabajo actual, exclusivamente, en la adaptación del DF. De hecho la propuesta que se describe en este

[1] http://www.fipa.org

apartado es una de las que están siendo consideradas en el grupo de trabajo [5].

El DF es el elemento de la plataforma que le permite a un agente descubrir los servicios que ofrecen otros agentes en el entorno que le rodea. En la definición del funcionamiento de un DF tradicional, la búsqueda de servicios remotos se realiza utilizando el concepto de federación de DFs. Un DF además de tener registrados unos servicios proporcionados por agentes locales (nativos o no), puede tener registrados otros DF, lo que le permite poder extender sus búsquedas a los servicios/agentes que tienen registrados esos otros DF. El número de saltos entre DF federados se limita mediante un parámetro de restricciones de búsqueda.

Analizamos a continuación los inconvenientes de la federación de DF para el descubrimiento de servicios remotos en redes ad-hoc:

- La búsqueda de un servicio remoto concreto siempre implica una comunicación con este sistema, porque a priori sólo conocemos que en esa plataforma existe un DF y no los servicios que están registrados en él. Lo que implica realizar transmisiones sin garantía de éxito.

- Si las condiciones de búsqueda permiten saltos de más de un nivel, es decir, la búsqueda en un DF remoto puede extenderse a otros DF remotos federados en él. Puede ocurrir, al contrario que en redes tradicionales, que los servicios que están registrados en esos DF no están accesibles al sistema inicial y por lo tanto, no sean resultados válidos.

3.3. Ad-hoc Discovery Agent

Para solventar estos dos problemas expuestos, y con el objetivo de mantener la mismas funciones que se especifican en FIPA00023 para el DF, de forma que los agentes no necesiten modificar la forma en la que interaccionan con él para descubrir servicios. Proponemos la introducción en la arquitectura FIPA para redes ad-hoc de un nuevo agente el **Ad-hoc Discovery Agent (ADA)** obligatorio en la plataforma, con la siguiente funcionalidad:

- Registrará y mantendrá actualizados, en el DF de la plataforma en la que se encuentre, los servicios remotos ofrecidos en la red ad-hoc. Es decir, el DF tendrá entradas a servicios remotos y no a un DF remoto, de esta manera, minimizaremos el número de transmisiones.

- Propagará búsquedas de servicios remotos solicitadas al DF, cuando éste se lo solicite, porque las condiciones de búsqueda así lo permiten.

- Anunciará a través del protocolo de descubrimiento asociado, lo servicios registrados en el DF, cuando esté permitido.

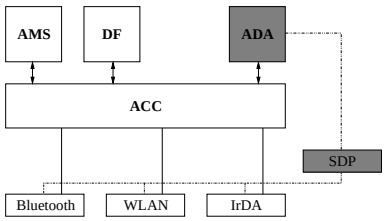


Figura 2: Introducción de ADA en FIPA

ADA utilizará un Service Discovery Protocol (SDP) para descubrir los servicios remotos. En este sentido deberá seleccionarse un protocolo que se adapte a la restricciones expuestas en el apartado anterior, para garantizar una solución eficiente en redes ad-hoc. En [6] se realiza un análisis de los posibles protocolos que se pueden emplear y se propone la utilización de Pervasive Discovery Protocol (PDP), que ha sido definido en nuestro grupo de trabajo. El Working Group FIPA AdHoc ha publicado un White Paper en el que se incluye parte de nuestro análisis y la descripción de nuestro protocolo [7].

3.3.1. Registro del ADA en el DF

El agente ADA proporciona un servicio que obligatoriamente tiene que estar registrado en el DF de la plataforma en la que reside, de esta forma en un DF que opere en una red ad-hoc siempre existirá una entrada correspondiente al ADA, que poseerá el siguiente AID reservado:

```
(agent-identifier
  :name ada@hap
  :addresses (sequence hap_transport_address)
)
```

Este agente se registra en el DF poniendo en el parámetro `:type` del `service-description` el valor `ada`.

El DF cuando procesa un `search` y en las restricciones de la búsqueda se le permite propagar la búsqueda a la red ad-hoc (se puede reutilizar la restricción `max-depth = 1`) el DF delega la búsqueda al ADA.

El ADA cada vez que tenga conocimiento a través del SDP de algún nuevo servicio remoto en la red ad-hoc, lo dará de alta en el DF mediante una petición de `register`. Como la red es cambiante, estos registros tendrán un `timeout` asociado que gestionará el ADA, de tal forma que cuando éste expire, realizará una petición de `deregister` al DF. El ADA sólo debe almacenar el `service-identifier` correspondiente y su `timeout` asociado.

El ADA a su vez puede proporcionar al SDP si éste lo solicita los servicios locales registrados en el DF, para ello realizará una petición de `search` sobre el DF.

3.4. Agent Communication Channel

En una sociedad multiagente, los agentes deben cooperar para realizar sus tareas y alcanzar sus objetivos. Los mecanismos de comunicación tradicionales permiten que dos agentes puedan transferirse información, pero es insuficiente cuando el objetivo es el de conseguir un comportamiento social. Es decir, se requieren mecanismos que permitan dotar a los mensajes intercambiados de un contenido semántico.

FIPA ha desarrollado el lenguaje Agent Comunication Language (ACL), una evolución de KQML, tomado como estándar durante mucho tiempo para el intercambio de conocimiento entre los agentes. La definición semántica formal con lógica modal de los mensajes ACL, ha sido uno de los esfuerzos mayores dentro del estándar FIPA y otorga a este lenguaje de una gran aceptación como estándar dentro del mundo de los agentes. El elemento ACC de la arquitectura FIPA es el que da soporte a la comunicación entre agentes empleando ACL. Para su simplificación hemos realizado un análisis de ACL, que presentamos en este apartado.

ACL está basado en **actos comunicativos** que se encargan del paso de información, solicitud de la información, negociación, realización de acciones y manejo de errores.

Las conversaciones entre agentes en ocasiones siguen patrones determinados, que se repiten en muchos casos. Aprovechando estos patrones y su repetición, FIPA ha definido unos **protocolos**. Un protocolo es un patrón que se usa para llevar por unos cauces concretos una conversación. Son como conversaciones guiadas, en que cada agente sabe qué mensaje enviar y cuáles puede recibir.

3.4.1. Comunicacion entre agentes

La idea central de los actos comunicativos entre los agentes gira entorno a la solicitud de la realización de una acción a otro agente. Esta operación se puede realizar de diferentes formas dependiendo si la acción se solicita a un agente concreto o si no se sabe quién provee esa acción.

Cuando un agente A solicita a otro agente concreto B una acción, envía un mensaje `request`. El destinatario puede aceptar o rechazar la petición con un mensaje `accept` o `refuse` respectivamente. En caso de aceptarla, deberá realizar la acción, e indicárselo al agente A cuando finalice mediante un mensaje de tipo `agree`. Si se produce un fallo en la realización de la acción se le notifica al agente que inició la petición mediante el mensaje `failure`.

Si el agente A necesita que se realice una operación pero no conoce quién tiene esa habilidad o se sabe que hay varios agentes que la proveen, se establece una comunicación de negociación. En este caso, se inicia la comunicación con el mensaje tipo `cfp`, pidiendo propuestas a los distintos agentes. En la petición se especifica la acción a realizar y algunas precondiciones a la hora de hacer las propuestas. Los agentes consultados envían sus propuestas al agente mediante `propose`, también indicando las condiciones de la propuesta y si no la aceptan lo notifican con el envío de un mensaje `refuse`. Para evitar que el agente A espere mucho tiempo, se establece un tiempo límite para la respuestas descartando las que lleguen después de éste. El agente A estudia las propuestas y elige las que le convienen, notificándoselo a los agentes mediante `accept-proposal` y rechazando el resto con `reject-proposal`. Una vez aceptadas y rechazadas las propuestas, el agente que inició el proceso puede cancelarlo si se produce algún cambio sobre la situación inicial. Así mismo, una vez aceptada una propuesta, el agente B puede responder con un `inform` indicando que se ha realizado la acción, o indicando que ha habido algún fallo con un `failure`.

Los agentes también se comunican para intercambiar información. De modo que el agente inicia la solicitud de información con `query-if` o `query-ref`, tras la petición de información, el agente B puede responder con la propia información (`inform`), con un fallo (`failure`), con un `not-understood` o con el rechazo de la petición, teniendo que alegar el motivo.

Todo lo comentado anteriormente hace referencia a la comunicación entre los agentes, pero ACL también se emplea para solicitar operaciones a los elementos integrantes de la plataforma, como por ejemplo el AMS y el DF.

3.4.2. Estructura del mensaje

El mensaje ACL FIPA contiene uno o más elementos de mensaje necesarios para la comunicación eficiente entre los agentes. Se compone de un **identificador** del tipo del acto comunicativo que identifica el tipo de mensaje o acción que el agente emisor solicita. También incluye una secuencia de **parámetros** definidos como un conjunto de parejas clave-valor que permiten asociar a cada acto de comunicación concreto toda la información necesaria.

accept-proposal	agree	cancel
cfp(call for proposal)	confirm	disconfirm
failure	inform	inform-it
inform-ref	not-understood	propagate
propose	proxy	query-if
query-ref	request	request-when
request-whenever	subscribe	

Figura 3: Mensajes ACL

3.4.3. Conjunto minimal de mensajes ACL

Como se ha visto en el apartado anterior, ACL cuenta con gran variedad de mensajes, además cada uno de ellos tiene hasta 11 parámetros diferentes, lo que hace que sea un lenguaje pesado para ser implementado completamente en una plataforma de agentes móviles para dispositivos limitados.

Este requisito ha hecho que se seleccione un conjunto de mensajes mínimo que provea las necesidades de comunicación entre los agentes.

La necesidad de petición de operaciones se puede cubrir con los mensajes `request`, `inform` y `agree`. De igual manera que las peticiones y la solicitud de información a los elementos que componen la plataforma.

El caso de la negociación explicado con anterioridad requiere muchos mensajes, pero se puede simplificar de la siguiente manera. Al agente A envía un `request` al AMS o al DF solicitando los agentes que realizan la operación en concreto. Los elementos de la plataforma envían la información mediante un mensaje `inform`, y de esta forma se ha obtenido la funcionlidad de `cfp` y de `proposal`. El siguiente paso es decidir a quién se le requiere el desempeño de esa operación, esta tarea se realiza de forma interna en el propio agente. Una vez que se ha decidido a quién realizar la petición se envía el `request` y se procede igual que en las peticiones de operación.

En lo referente a la solicitud de información también se puede realizar mediante peticiones `request` y luego mediante los parámetros especificar si se requiere la ejecución de una operación o si simplemente se solicita una información en concreto permitiendo así sustituir el mensaje `query` por `request`.

Con lo que se concluye que el grupo mínimo de mensajes para comunicarse en una plataforma en dispositivos limitados es `request`, `inform`, `agree`. El mensaje `cancel` se puede sustituir por el establecimiento de un tiempo de respuesta.

3.4.4. Ejemplo de mensaje ACL

A continuación se presenta un mensaje que envía un agente al AMS para registrarse en su plataforma. En el mensaje se incluye el identificador del receptor mediante el parámetro `receiver` y adjunta la información relativa a sí mismo con el parámetro `sender`. Después se indica el lenguaje, el protocolo y la ontología, para continuar con el contenido del mensaje donde se especifica la operación que se está solicitando, en este caso consiste en registrarse y se indica mediante `register`.

```
(request
  :sender
    (agent-identifier
      :name dummy@foo.com
      :addresses (sequence iiop://foo.com/acc))
  :receiver (set
    (agent-identifier
      :name ams@foo.com
      :addresses (sequence iiop://foo.com/acc)))
  :language FIPA-SL0
  :protocol FIPA-Request
  :ontology FIPA-Agent-Management
  :content
    (action
      (agent-identifier
        :name ams@foo.com
        :addresses (sequence iiop://foo.com/acc))
      (register
        (ams-agent-description
          :name
            (agent-identifier
              :name dummy@foo.com
              :addresses (sequence iiop://foo.com/acc))
          :state active))))
```

4. Implementación

En paralelo al diseño de una plataforma compatible con FIPA, nuestro grupo de trabajo comenzó a evaluar la posibilidad de poder implementar esta plataforma en dispositivos reales.

Este estudio nos llevó en primer lugar, a la elección de un lenguaje de programación para realizar la implementación. A día de hoy podemos decir que aunque se han desarrollado lenguajes específicos, ha sido Java el lenguaje en el que más plataformas de agentes móviles se han desarrollado, debido a las siguientes ventajas: independencia de la plataforma, ejecución segura, carga dinámica de clases, serialización, reflexión,...

Esto nos llevó a seleccionar la versión de Java para dispositivos limitados J2ME, y en concreto el perfil Mobile Information Device Profile (MIDP), como base de nuestros desarrollos. Aunque J2ME mantiene alguna de las características de Java, parte de las que convertían a Java en un buen lenguaje de programación de plataformas de agentes móviles, han desaparecido, o se han visto limitadas por motivos de seguridad, en concreto, aquellas que nos permitían implementar movilidad de objetos (carga dinámica de clases, serialización y reflexión). Veremos a lo largo de esta sección cómo hemos solventado estos problemas.

4.1. Tecnología de agentes y J2ME

Existen varias propuestas en la literatura que pretenden involucrar a dispositivos limitados J2ME en plataformas de agentes móviles, en este apartado haremos referencia a aquellas más importantes.

4.1.1. LEAP

El proyecto LEAP [8] engloba un consorcio de compañías entre las que se encuentran, entre otras, Motorola, British Telecom y Siemens. El objetivo del proyecto es el desarrollo de una plataforma de agentes en Java conforme al estándar FIPA que pueda operar tanto en dispositivos limitados (teléfonos móviles, PDA, pagers) con J2ME, como en PCs con J2SE. Para ello, han diseñado una arquitectura modular, con una parte obligatoria, común a todos los tipos de dispositivos, y otra opcional; y mediante un instalador, se puede componer la plataforma según las limitaciones del dispositivo en el que se instale.

El proyecto LEAP fue pionero a la hora de demostrar la viabilidad de construir plataformas de agentes en dispositivos limitados, aunque su aplicabilidad está centrada en redes con infraestructura y por lo tanto, en los terminales móviles no existe una plataforma completa de agentes como tal y su operación depende siempre de un sistema intermedio al que esté conectado.

Aunque era uno de los objetivos marcados en el proyecto, la movilidad a nivel agente no está soportada.

4.1.2. Monash University

En Monash University [9] se ha diseñado e implementado una plataforma de agentes para dispositivos personales móviles basados en PalmOS.

La plataforma desarrollada se denomina MAE y en la actualidad tienen dos implementaciones utilizando dos versiones reducidas de Java: la configuración CLDC de J2ME, y SuperWaba, que es una versión limitada de Java con algunas funcionalidades no soportadas por la versión oficial de Sun, como Java Native Interface (JNI).

Esta plataforma soporta movilidad a nivel agente, para solventar el problema de la carga dinámica de clases se han empleando mecanismos específicos de los sistemas PalmOS, por lo que esta característica no es migrable a otro tipo de dispositivos J2ME.

4.2. TAgentsP: nuestro perfil sobre CLDC para agentes móviles

Las propuestas anteriores han realizado importantes contribuciones en cuanto a la adaptación de plataformas de agentes en dispositivos limitados. LEAP es una plataforma válida para una red con infraestructura, en la que no es necesario tener un plataforma autónoma en el dispositivo limitado, y por lo tanto se puede depender de un sistema no limitado, pero el escenario que nosotros queremos abordar, que es una red ad-hoc esto no es válido. MAE es una plataforma completa pero no es compatible con FIPA y funciona sólo sobre PalmOS.

Nuestra investigación y desarrollo inicial se centró en dotar a J2ME de aquellas características que tenía Java como lenguaje de desarrollo de plataformas de agentes y que J2ME no posee, en concreto las que nos permiten *weak mobility*: serialización de objetos y carga dinámica de clases.

Para ello, siguiendo la filosofía modular de J2ME [10] complementamos el perfil MIDP para construir un nuevo perfil sobre el CLDC que proporcione la funcionalidad básica de un plataforma de agentes móviles, a este perfil lo hemos llamado TAgentsP (Travel Agents Profile), por analogía con nuestra plataforma de agentes sobre J2SE, TAgents [11].

Una vez proporcionados los servicios básicos, y para conseguir nuestro objetivo de implementar una plataforma de agentes móviles autónoma, es decir, sin necesidad de interaccionar con dispositivos no limitados, hemos desarrollado un reducido servidor HTTP que nos permite por una parte la movilidad de agentes y por otra, la comunicación entre ellos de forma directa.

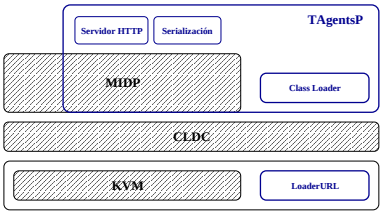


Figura 4: Arquitectura de TAgentsP

En la Figura 4 vemos la arquitectura de nuestra propuesta y en las siguientes secciones describiremos cada uno de los módulos desarrollados y probados con éxito en dispositivos limitados: el módulo de serialización y el del servidor HTTP. Todavía no hemos abordado la implementación de la carga dinámica aunque expondremos brevemente el porqué.

4.2.1. Implementación de TAgentsP

Una de las primeras carencias que se pretenden suplir en nuestro nuevo perfil TAgentsP, es la ausencia de mecanismos de carga dinámica de clases y de serialización de objetos, que nos permitan la movilidad de código entre dispositivos limitados.

La movilidad de un agente se consigue transformando un agente en un flujo de bytes que viaja a través de la red, para posteriormente ser reconstruido en el dispositivo destino. Este mecanismo se conoce como serialización. Pero además, a la hora de soportar la migración de agentes es necesario proporcionar carga dinámica de clases, ya que el sistema destino si no posee la clase a la que pertenece el agente serializado recibido, éste no podría ser reconstruido.

En J2ME no se dispone de un mecanismo de carga dinámica de clases, aunque es una de las funcionalidades que se pretenden suplir en próximas versiones. La implementación de mecanismos de carga dinámica de clases, implica modificar la implementación nativa de la máquina virtual, por lo que este mecanismo no es portable a distintos sistemas. Éste es el principal motivo que nos ha llevado a postergar esta implementación y suplirla introduciendo el mismo conjunto de clases en los sistemas limitados entre los que realizamos movilidad de código.

En las próximas secciones proporcionamos una descripción del mecanismo de serialización implementado y del servidor HTTP contruido.

Mecanismo de serialización en J2ME La *serialización* es un mecanismo mediante el que se puede convertir un objeto en un flujo de bytes que represente su estado, y consecuentemente ser transportado a través de la red o almacenado de manera persistente en un sistema de ficheros. Esta conversión no tendría sentido si no fuese a haber una posterior recuperación, mecanismo denominado *deserialización*.

La serialización es un requisito imprescindible para soportar la movilidad de agentes, ya que es

el mecanismo que permite convertir a un agente en algo transportable que conserve su estado. Además, también es un mecanismo necesario para la comunicación de mensajes entre agentes residentes en distintas plataformas. Java en su versión estándar proporciona serialización, pero en J2ME es una de las características que se han eliminado, aunque, si bien plantea serias dificultades de implementación, es abordable y además ofrece un mecanismo abierto y extensible, en cuanto a la movilidad de código se refiere.

Debido a la ausencia del mecanismo de reflexión en J2ME, por el cual las propias clases son capaces de inspeccionarse a sí mismas (averiguando sus métodos, atributos, parámetros, constructores), el mecanismo de serialización implementado va a tener ciertas limitaciones, no pudiéndose conseguir una serialización totalmente automatizada, en la que el programador no tenga que intervenir en este proceso.

Se puede decir que lo que se busca conseguir, es el soporte básico para que mediante éste y el conocimiento del programador sobre las clases que implementa, se disponga de un mecanismo genérico y extensible para transformar objetos en un flujo de bytes, transportarlos y posteriormente recuperar su estado en el dispositivo destino.

El principal problema a afrontar a la hora de desarrollar mecanismos de serialización, es la imposibilidad de conocer en tiempo de ejecución los métodos y atributos de las clases en J2ME, que se proporcionaba en las versiones estándar de Java.

La serialización llevada a cabo facilitará y se encargará de que el programador se despreocupe de cómo se lleva a cabo todo el proceso, del formato interno de los datos así como del proceso de recuperación de las clases serializadas, siendo únicamente responsable de la implementación del interfaz `Serializable` como se verá a continuación.

La solución planteada consiste en primer lugar en la creación de un interfaz denominado `Serializable` que contenga los dos métodos básicos que especifiquen las operaciones que toda clase serializable debe poder llevar a cabo: `writeObject` para serializarse y `readObject` para deserializarse. Este interfaz tiene el objetivo de poder "marcar" aquellas clases que sí saben cómo serializarse/deserializarse.

El siguiente paso es la creación de dos nuevas clases `ObjInputStream` y `ObjOutputStream`. Estas clases son las que van a controlar el proceso completo de serialización/deserialización realizando las llamadas necesarias a los métodos `writeObject` y `readObject` que todas las clases serializables deben implementar.

El método `writeObject` se va a encargar de escribir los valores de los atributos de un determinado objeto en un flujo de bytes, obteniendo de este modo un objeto serializado, que mediante el método `readObject` podrá ser interpretado permitiendo la construcción de un nuevo objeto a partir de la información extraida de este objeto serializado.

En el momento de serializar una determinada clase, por ejemplo `Agente_Serializable` se va a crear un objeto de tipo `ObjOutputStream`. Este objeto recibirá el objeto a serializar, creará un flujo de bytes en el que inicialmente escribirá el nombre de la clase del objeto que se está serializando[2] para finalmente llamar al método `writeObject` de la clase `Agente_Serializable` que se preocupará de escribir la información precisa (valor de atributos, tipos, ...) en un flujo de bytes de modo que posteriormente pueda recuperar su estado empleando para ello su propio método `readObject`.

Este mecanismo ofrece la ventaja de que aún careciendo de mecanismos de reflexión, si el programador se preocupa de implementar los métodos de serialización y deserialización (writeObject y readObject) toda clase en J2ME que implemente el interfaz serializable puede ser convertida en un único flujo de bytes que va a poder viajar de dispositivo en dispositivo, y en donde, con una simple llamada al método readObject, este flujo se convierta en un nuevo objeto con el mismo estado del objeto serializado.

Servidor HTTP en dispositivos limitados

El perfil TAgentsP desarrollado no tendría utilidad alguna si no existiese un mecanismo de comunicación directa entre los distintos dispositivos móviles. En nuestro desarrollo se ha optado por construir un reducido servidor HTTP 1.1 en J2ME, como mecanismo de comunicación directa entre dispositivos, ya que ofrece una solución abierta no orientada únicamente al intercambio de clases o agentes serializados.

A continuación se introduce la arquitectura básica del servidor. Esta arquitectura está compuesta por tres módulos principalmente. El primero de ellos es el de *configuración*; mediante éste se definen las opciones básicas de funcionamiento (número de peticiones a aceptar, puerto, nombre del servidor, archivos MIME soportados ...).

A través del *sistema de archivos* (segundo módulo) se van a poder gestionar los recursos a albergar por el servidor así como resolver las peticiones de los usuarios. Por último es mediante el *servicio de tratamiento de peticiones* que las diversas solicitudes HTTP (GET, POST, HEAD) van a poder ser tratadas.

Uno de los problemas surgidos en el desarrollo de este servidor, ha sido la ausencia de un sistema de archivos accesible en J2ME para poder proporcionar los recursos solicitados por los clientes HTTP. La solución ha consistido en implementar un pseudo sistema de archivos sobre RMS [3] de

[2] llamando a Object.getClass().getName() soportado en J2ME

[3] API para el almacenamiento persistente en J2ME, que permite la creación y almacenamiento de registros binarios de datos.

modo que puedan crearse directorios y archivos así como poder gestionarlos (recuperar, eliminar, listar ...).

Este servidor podrá utilizarse para diversas tareas como pueden ser: intercambio de clases entre diferentes dispositivos que permite el soporte a la movilidad de agentes y la comunicación directa entre dispositivos sobre HTTP, por ejemplo como mecanismo de transporte de mensajes ACL.

5. Conclusiones y trabajos futuros

En este artículo proponemos la utilización de la tecnología de agentes móviles como middleware para el desarrollo de aplicaciones en dispositivos limitados que se comunican mediante protocolos inalámbricos, formando lo que se denomina una red ad-hoc. Siendo ésta una contribución a la computación ubicua, ya que permite integrar a los dispositivos limitados y embebidos en el mundo físico en la computación distribuida.

Tomando como punto de partida el estándar FIPA, hemos realizado un análisis de los diferentes elementos funcionales que se definen y hemos propuesto una simplificación para que puedan implementarse en un dispositivo limitado. En el caso del DF, no sólo es necesario su simplificación sino su adaptación a las restricciones de las redes ad-hoc. En este sentido, los autores forman parte activa del Working Group FIPA Ad-Hoc, en el que en la actualidad se están discutiendo las propuestas relacionadas con el DF, entre ellas, la aquí propuesta.

En la actualidad, seguimos trabajando en el diseño de la plataforma definiendo la interfaz entre el agente ADA y el protocolo de descubrimiento de servicios subyacente y el propio DF. Y seguimos analizando el impacto que la simplificación de ACL puede tener en la comunicación entre agentes. También dentro de nuestro grupo estamos definiendo un modelo de seguridad aplicable a sistemas multiagente en redes ad-hoc.

Como hemos visto, para implementar el diseño realizado hemos seleccionado la tecnología J2ME y hemos definido e implementado un nuevo perfil J2ME, TAgentsP, que nos dé el soporte para movilidad de código y comunicación directa entre dispositivos, que J2ME no proporcionaba. Este perfil ha sido probado en dispositivos reales con resultados satisfactorios. Nuestro trabajo actual en esta línea se centra en implementar la plataforma de agentes, integrando nuestros desarrollos en la plataforma LEAP.

Agradecimientos

Los autores agradecen a David Camacho, Carlos García Rubio y Andrés Marín las aportaciones realizadas.

Celeste Campo da las gracias también a Florina Almenares y a los miembros de FIPA TC/WG Adhoc, especialmente a Michael Berger.

Referencias

[1] Mark Weiser. The Computer for the 21st Century. *Scientific American*, September 1991.

[2] Guruduth Banavar, James Beck, Eugene Gluzberg, Jonathan Munson, Jeremy Sussman, and Deborra Zukowshi. Challenges: An Application Model for Pervasive Computing. In *Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom 2000)*, 2000.

[3] H. S. Nwana. Software Agents: an overview. *Knowledge Engineering Review*, 1(3):205–244, 1996.

[4] FIPA. *FIPA Agent Management Specification*, March 2002.

[5] C. Campo. Directory Facilitator and Service Discovery Agent. Technical report, FIPA Adhoc Technical Committee, 2002.

[6] C. Campo. Service Discovery in Pervasive Multi-Agent Systems. In Tim Finin and Zakaria Maamar, editors, *AAMAS Workshop on Ubiquitous Agents on embedded, wearable, and mobile agents*, Bologna, Italy, July 2002.

[7] WG-AdHoc. Agents in Ad Hoc Environments. A Whitepaper, 2002.

[8] Lightweight extensible agent platform. http://leap.crm-paris.com/.

[9] Patrik Mihailescu and Elizabeth A. Kendall. MAE: A Mobile Agent Platform for Building Wireless M-Commerce Applications. In *8th ECOOP Workshop on Mobile Object Systems: Agent Applications and New Frontiers*, Spain, June 2002.

[10] Enrique C. Ortiz. A Survey of J2ME Today. Technical report, Wireless Java, 2002.

[11] Thomas Letsch. Redesign and implementation of a mobile agent system compliant with the maffinder part of masif stanadard. Master's thesis, Technische Universitat Munchen. Institut fur Informatik, 2000.

Arquitectura de Red Programable y sus Aplicaciones en Sistemas Móviles Inalámbricos[1]

D. Larrabeiti, M.F. Sedano, M. Calderón, B. Alarcos, M. Urueña, R. Romeral, M. Bagnulo, E. de la Hoz

Departamento de Ingeniería Telemática. Universidad Carlos III de Madrid.
Av. Universidad 30 - 28911 Leganés (Madrid)
E-mail: {dlarra, maria, muruenya, rromeral, marcelo}@it.uc3m.es

Departamento de Ingeniería de Sistemas Telemáticos. Universidad Politécnica de Madrid.
Ciudad Universitaria S/N - 28040 Madrid
E-mail: marifeli@gsi.dit.upm.es

Área de Ingeniería Telemática. Dpto. de Automática. Universidad de Alcalá.
Crta Madrid-Barcelona, Km 33,600 - 28871. Alcalá de Henares (Madrid)
E-mail: {bernardo, enrique}@aut.uah.es

Abstract*. This paper identifies a set of scenarios where intelligent programmable devices can find a field of application in the development of wireless networks, and presents a novel architecture for programmable nodes especially designed to ease the deployment of network applications in wireless networks, including a security architecture that enables scalable secure on-demand deployment of code. Finally, we describe preliminary results of this work on a prototype of this system before starting the deployment phase in a UMTS testbed.*

1. Redes activas, redes programables y redes móviles inalámbricas

Cada vez es más frecuente en la periferia de las redes IP, el uso de dispositivos con capacidad para realizar procesamiento sobre los paquetes más allá del simple encaminamiento de paquetes. Ejemplos de este tipo de dispositivos son cachés web transparentes, cachés de contenidos multimedia y de optimización de la distribución de video, cortafuegos y sistemas de detección de intrusión, multiplexores de direcciones y puertos NAPT, adaptadores de formato, traductores IPv4-IPv6, acondicionadores y marcadores de tráfico, etc.

El paradigma de las *redes activas* [2], hoy considerada una línea de investigación madura, llevó el estudio de este tipo de dispositivos hasta sus últimas consecuencias, analizando la posibilidad de que los propios usuarios finales distribuyeran código en los routers activos para cambiar el comportamiento del router sobre el tráfico. Dicho código podía transportarse en cápsulas [3] (paquetes conteniendo el propio código que debe procesarlos), debía ejecutarse en entornos de ejecución suficientemente seguros [4], y debería compatibilizar el uso compartido de los recursos del router entre las distintas aplicaciones activas [5]; todo esto constituye un reto de difícil implementación y aplicación industrial.

Más pragmáticamente, las *redes programables* [6], admiten un aceptable grado de programabilidad, ofreciendo un mayor control sobre las entidades capaces de desplegar dinámicamente los "programas de red". Estos programas, suficientemente probados y garantizando la compatibilidad entre ellos, son descargados desde almacenes de código seguros (nota: en ocasiones en la literatura los términos *redes activas* y *redes programables* aparecen usados indistintamente).

Entre las iniciativas de organizaciones de normalización que fomentan la programabilidad de la red se encuentran OPES [7], revelando la preocupación de IETF en el control de la alteración de contenidos por parte de dispositivos intermediarios, IEEE P1520 [8] y FORCES [9], donde se promueve el desarrollo de una arquitectura de router abierta con la definición de un interfaz entre elementos de reenvío y elementos de control. Asimismo, la industria ha respondido a esta necesidad con una amplia oferta de procesadores de red [10] como alternativa flexible pero de alta velocidad a la implementación de procesadores de interfaz mediante ASICs, tecnología esta última con una capacidad de reprogramación más limitada.

[1] Este trabajo está financiado por el Ministerio de Ciencia y Tecnología a través del proyecto TIC2001-1650-C02-01/02 AURAS (**A**rquitectura integrada **U**MTS-**R**edes **A**ctivas para la implantación rápida de **S**ervicios).

En todo este rango de posibilidades de programación del comportamiento de nodos de red se ha realizado una intensa investigación, tanto en la definición de arquitecturas [6] como en las posibles aplicaciones de las mismas [4]. La aplicación de estas tecnologías al campo de las redes móviles inalámbricas no ha sido una excepción. Así existen trabajos que intentan optimizar las prestaciones de sistemas inalámbricos, clásicamente rompiendo el principio de transporte extremo a extremo. Tal es el caso de TCP spoofing y relaying en PEP (Performance Enhancing Proxies) [11, 12] (especialmente útiles en escenarios con segmentos inalámbricos con alta tasa de errores y con protocolos de enlace sin capacidad de retransmisión), el de dispositivos que pretenden optimizar el transporte de flujos multimedia [13] (optimizaciones basadas en el conocimiento y caracterización de las áreas de cobertura y de la posición y la trayectoria del móvil), e incluso en la aplicación de técnicas de redes activas al cómputo de rutas en redes ad-hoc [14]. Por otra parte, algunos fabricantes de procesadores de red, como IBM [15], prevén una importante área de aplicación de estos dispositivos en el diseño de equipos de sistemas móviles 3G, aduciendo que su flexibilidad inherente para cambiar sus pilas de protocolos permitirá amortizar la inversión en estos equipos durante la transición a la futura arquitectura todo-IP [16].

En este contexto tecnológico, se sitúa el trabajo presentado en este artículo, que está siendo realizado en el marco del proyecto MCYT AURAS [1], y cuyo principal objetivo es el desarrollo de arquitecturas de nodos programables adecuadas para redes móviles y el estudio de sus aplicaciones en UMTS. En este artículo se propone una arquitectura de nodo programable que intenta dar soporte a aplicaciones de red para terminales móviles. En primer lugar se justifican los principios de diseño de esta arquitectura; a continuación, se describe brevemente la arquitectura de seguridad ideada para el despliegue seguro y escalable de código en la red; después se definen escenarios de aplicación de este sistema y, finalmente, se muestran resultados preliminares obtenidos sobre un prototipo que implementa la arquitectura propuesta.

2. Arquitectura de Red Programable SARA

2.1 Arquitectura de nodo programable

SARA (Simple Active Router Assistant) [17] es una arquitectura de nodo programable que tiene por objetivo principal la implementación y explotación de dispositivos de red reprogramables dinámicamente. Su programabilidad se basa en algunos conceptos desarrollados en el campo de las redes activas; es decir, propone la existencia de un entorno de ejecución dinámico en el nodo donde se ejecutan las aplicaciones de red; pero a diferencia de un nodo activo, las posibles aplicaciones a lanzar en este entorno se hallan controladas por el administrador de la red, y el usuario final simplemente queda habilitado para activarlas y cambiar su estado mediante señalización específica. Este procedimiento de distribución de código en la red y de activación del mismo mediante paquetes de señalización, se realiza de manera segura mediante la arquitectura de seguridad presentada en la sección siguiente. Asimismo se ha procurado dar soporte a aplicaciones para redes móviles, como se analiza en este trabajo.

Los principios de diseño de SARA son los siguientes:

I. Los routers convencionales deben delegar el procesamiento específico en asistentes. Partiendo de la existencia de routers convencionales de gama media con *fast path* hardware y *slow path* software, es evidente que la introducción de servicios de procesamiento específico en su CPU, mermaría rápidamente las prestaciones de los routers convirtiendo su capacidad de procesamiento en un cuello de botella. De hecho la carga de CPU es un factor que hoy en día es tenido muy en cuenta a la hora de introducir cortafuegos, túneles cifrados y conmutadores de nivel 4 en una red. Por consiguiente todo procesamiento complejo sobre los paquetes debería realizarse sobre elementos especializados, ajenos al propio router, y de la manera más desacoplada posible de su función principal de reenvío de paquetes. En definitiva, la arquitectura SARA propone que todo router que requiera ampliar su programabilidad con nuevas aplicaciones de red debe reprogramarse para delegar su procesamiento en procesadores co-ubicados (denominados asistentes) que pueden consistir en ordenadores conectados al router con una LAN de alta velocidad y conteniendo un entorno de ejecución apropiado para el tratamiento eficiente de paquetes. La penalización causada al router por esta nueva función queda determinada por el coste de identificar los paquetes que deben ser desviados, entre los paquetes que entraron en el router por sus interfaces convencionales, y desviarlos al asistente que debe procesarlos. El asistente puede realizar procesamiento transparente sobre los paquetes en capa 3 y superiores antes de reenviarlos al router para su encaminamiento normal, y debe disponer del mayor control posible sobre los recursos del router.

II. Cooperación router-asistente. La delegación de funciones al asistente/s precisa cooperación entre el router y el asistente. Los mecanismos establecidos para ello son:

A) Desvío de paquetes de señalización de aplicaciones de red. Esto permite lanzar, cambiar el estado o retirar aplicaciones del entorno de ejecución en los asistentes.

B) Vista del estado del router. Ciertas aplicaciones requieren consultar la tabla de rutas, capacidades y nivel de ocupación de los enlaces del router, longitud media de las colas, etc. El entorno de ejecución pone a disposición de las

aplicaciones esta información, mediante una vista seleccionada del estado del router – típicamente obtenida por SNMP – cacheada por motivos de eficiencia.

C) Protocolo Router-Asistente (RAP). Si se requiere disponer de capacidad de proceso de cualquier flujo convencional que atraviese el router, es necesario habilitar mecanismos de control más complejos y versátiles que los anteriores. RAP permite tareas tales como la programación de desvíos de flujos a asistentes, reparto de flujos entre varios asistentes o la salida de un paquete por un interfaz determinado del router (útil en retransmisiones sobre subárboles en aplicaciones de multicast fiable).

La figura 1 muestra la arquitectura software empleada en el asistente (derecha) y la interacción con el router preexistente. Lógicamente, existe una relación de compromiso entre funcionalidad disponible a las aplicaciones de red y sobrecarga en el router convencional. Por este motivo, la arquitectura ofrece dos niveles de cooperación: cooperación *ligera*, que requiere los mecanismos A) y B); y cooperación *completa*, basada en C). Nótese que la cooperación ligera es suficiente para muchas aplicaciones y sólo precisa pequeños cambios en los routers preexistentes.

III. Despliegue transparente de aplicaciones de red. Un objetivo de diseño de la arquitectura es facilitar la movilidad, ocultando la topología de la red a los terminales y la ubicación de los nodos programables. Las aplicaciones se lanzan en los nodos del trayecto automáticamente empleando paquetes de señalización direccionados al sistema final, no siendo necesario direccionar los nodos programables en el trayecto (o árbol) origen-destino(s). Para implementar esta funcionalidad de manera eficiente se emplea la opción *router alert* de IP (RFC2113 para IPv4 y RFC2711 para IPv6). La alternativa, la opción no transparente, muy frecuente en la mayoría de esquemas de redes activas donde se construye un overlay de nodos activos estableciendo vecindad explícita entre ellos, es también posible. En este caso pueden emplearse búsquedas multicast incrementales en anillo para localizar al nodo programable más cercano.

Los paquetes de señalización de aplicaciones de red llevan encapsulación ANEP, contienen referencias al entorno y a la aplicación que debe procesarlos. En caso de no hallarse residente la aplicación en el entorno, se procede a su descarga remota, a partir de su URI asociada, desde un almacén de código seguro (servidores de código). La aplicación tiene un tiempo de vida especificado en la propia señalización, que se refresca mediante mecanismos soft-state, con el fin de caducar aplicaciones que procesaban los paquetes de un móvil cuyas comunicaciones han dejado de atravesar el nodo en cuestión.

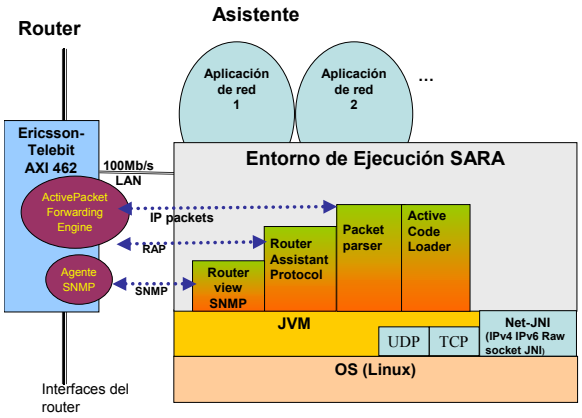


Figura 1. Arquitectura software SARA

2.2 Arquitectura de Seguridad

La introducción de programabilidad en la red introduce nuevos riesgos de seguridad que hay que abordar. Entre esos riegos cabe destacar la protección de la propia red programable frente a paquetes de señalización no auténticos que pueden provocar efectos negativos en la arquitectura del nodo programable. Uno de los principales efectos a evitar es el consumo de recursos por ejecución de aplicaciones no autorizadas. Para proteger a la red de este riesgo se ofrecen mecanismos de autenticación y autorización de los usuarios que solicitan un servicio y protección de los paquetes de señalización con autenticación e integridad. Con el objeto de cubrir estos riesgos se ha desarrollado una solución de seguridad para SARA denominada ROSA (Realistic Open Security Architecture for Active Networks) [18, 19].

El escenario real sobre el que se puede aplicar una red programable consta de usuarios que solicitan servicios, los cuales requieren un procesamiento por parte de los nodos programables. Para poder obtener este servicio, los usuarios establecerán sesiones que se identificarán de forma única mediante los siguientes parámetros, a los que denominaremos *parámetros de sesión*:

- Identificación del usuario, U, que solicita el servicio (es decir solicita la activación de una aplicación de red).
- Periodo de tiempo de validez de la sesión, marcado por el tiempo de inicio y fin de la sesión (SST, SET).
- Direcciones IP del origen y destino del flujo de paquetes de señalización correspondientes a la sesión, L.
- Identificador de código activo que deben ejecutar los nodos programables, Ci.

El planteamiento de seguridad se ha enfocado desde la perspectiva de protección de la red programable frente a usuarios maliciosos. Debido a esto, el principal riesgo es el uso no autorizado de una aplicación, es decir, el envío de paquetes de señalización no autorizados que provocan la

ejecución de dicho código. Para proteger la red de este riesgo hay que ofrecer mecanismos de autenticación y autorización de los usuarios que solicitan el servicio, y protección de los paquetes de señalización con autenticación e integridad. Por otro lado se ha buscado una solución que sea escalable en cuanto a sobrecarga de las cabeceras y sobrecarga de procesamiento de seguridad de los paquetes de señalización. Debido a esto se ha desestimado el uso de mecanismos basados en clave asimétrica. Se ha optado por proteger los paquetes de señalización con un código de autenticación de mensajes (generado con hmac) basado en clave simétrica, que tiene unos tiempos de ejecución rápidos y no sobrecarga en exceso las cabeceras con respecto a otros mecanismos como firma digital. El uso de clave simétrica sin embargo, plantea un problema de distribución de clave entre los distintos componentes de la red programable y el usuario.

ROSA implementa un mecanismo distribuido de generación de clave, en el que parte de la información que llevan los paquetes de señalización es usada como credencial de autorización. La red programable con seguridad estará formada (Fig. 2) por un *Servidor de Autorización* (AS), *Servidores de Código* (CS) y *Nodos Programables* (PN compuesto por un router y su asistente) que compartirán un valor secreto (Kci) asociado a cada aplicación (Ci). Los Kci son generados por el AS periódicamente y enviados a los CS por un canal confidencial. Cuando los PN descargan un código (aplicación) desde los CS también descargan el Kci asociado a dicho código. Podemos distinguir tres fases en el proceso de seguridad:

Solicitud de servicio: un usuario solicita el servicio a un AS, cuando éste comprueba que el usuario está autorizado, genera una clave de sesión (K) y se la da al usuario. Esta clave depende de los *parámetros de sesión* (U, SET, SST, L, Ci) y del *valor secreto* (Kci) asociado al código. Para generar K se utiliza una función de derivación de claves. La comunicación entre el usuario y el AS se hace con mecanismos de autenticación mutua y confidencialidad.

Generación de señalización protegida: el usuario envía paquetes de señalización desde el origen al destino. Los paquetes van protegidos con un código generado con hmac y la clave de sesión (K). Los paquetes de señalización llevan además los *parámetros de sesión* que identifican la sesión y con los cuales se ha calculado K.

Procesamiento en los nodos programables: cuando un paquete de señalización de una sesión llega a un PN por primera vez, este descarga el valor secreto (Kci) y la aplicación desde un CS. Con los parámetros de sesión que van en el paquete y Kci, el nodo genera la clave de sesión K. Con K y hmac verifica que el paquete es auténtico e integro. Con los parámetros de sesión, comprueba que el paquete está

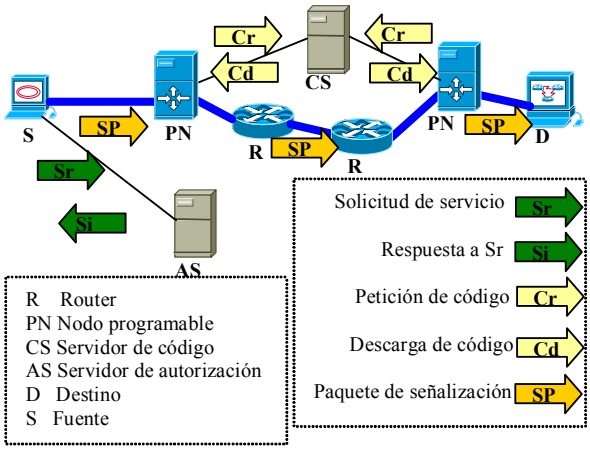


Figura. 2. Arquitectura ROSA

dentro de la ventana de tiempo de la sesión. El nodo comprueba de esta forma, que el paquete ha sido generado por un usuario que conoce K. Esto implica que el usuario ha sido autorizado previamente por el AS a enviar paquetes en el periodo de tiempo especificado, desde un origen a un destino y para ejecutar una aplicación de red concreta. Como vemos, los parámetros de sesión que van en el paquete actúan como credencial dado que permiten comprobar que el paquete está autorizado para ser procesado. Si la aplicación modifica el contenido del paquete de señalización, lo vuelve a proteger utilizando hmac y la clave K. Para el resto de paquetes de señalización de la sesión no es necesario descargar el código y el *valor secreto*, puesto que ya están en el nodo.

La solución de seguridad planteada soporta movilidad de los terminales dado que si un usuario se mueve por la red y este movimiento implica un cambio del nodo programable, el nuevo nodo será capaz de general la clave de sesión y comprobar la integridad y autenticad de los paquetes sin ningún tipo de procesamiento adicional.

La solución propuesta se ha implementado y se ha evaluado la influencia de la arquitectura de seguridad en el retardo extremo a extremo. Los resultados obtenidos muestran que ROSA introduce un pequeño incremento (7,6%) sobre el retardo extremo a extremo sin seguridad, cuando la descarga del *valor secreto* y de la aplicación desde el CS no es necesaria, es decir, para la mayoría de los paquetes procesados por los nodos programables. Sólo el primer paquete de señalización de la sesión experimenta un mayor retardo debido a la descarga segura del código y del *valor secreto*.

3. Aplicaciones en redes móviles inalámbricas

Como se ha descrito en la introducción, existen diversos escenarios en los que programas ejecutados dentro de la red pueden apoyar la comunicación en

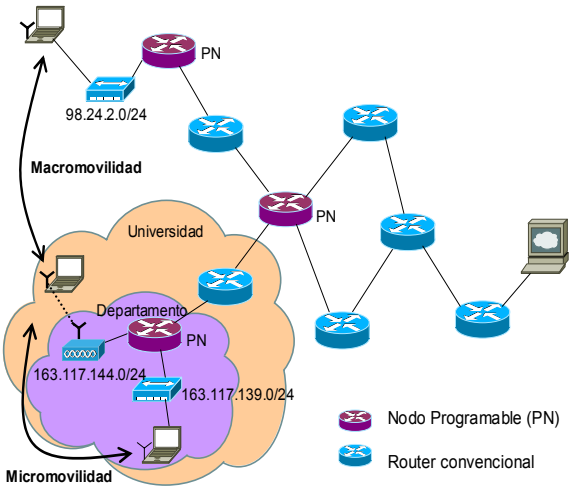


Figura 3. Micro y macromovilidad

entornos inalámbricos, entre ellas han sido muy estudiadas las aplicaciones encargadas de optimizar las prestaciones, como los PEP (Performance Enhancing Proxies) [12] y relays de aplicación con conocimiento exacto de cobertura y contexto del terminal móvil. Un sistema como SARA añade a este tipo de aplicaciones la posibilidad de ubicar automáticamente dichos proxies en los nodos óptimos del trayecto (en enlaces con tasas altas de errores o alto retardo). La información necesaria para caracterizar con precisión los enlaces de dicho trayecto puede obtenerse mediante paquetes de señalización que inspeccionan las vistas de estado de los routers. De este modo se da un cierto soporte a macromovilidad (Fig.. 3), en el sentido de que las aplicaciones de red necesarias vuelven a restaurarse al cambiar el trayecto de los paquetes debido al movimiento del terminal o cambios de enrutamiento.

Nótese que no se pretende dar una alternativa a la implementación de movilidad de terminal, ya que para ello existen protocolos específicos. Tampoco a la movilidad de aplicaciones, de la que se han ocupado ampliamente los estudios sobre movilidad de agentes. sino un soporte bastante más simple independiente de si el móvil ha precisado cambiar de dirección IP en su cambio de subred.

En un escenario de micromovilidad (Fig. 3) la programabilidad de un solo router en la red de una organización, sí nos permitiría una implementación rápida de mecanismos simples de movilidad. Por ejemplo, activando de manera controlada funciones de proxy-ARP. De esta manera un usuario puede desplazarse a otro departamento de la organización, y cambiar de subred física conservando su dirección IP y, con ella, sus permisos de acceso a sus recursos de intranet, etc.

UMTS

En UMTS se proponen distintos escenarios de aplicación de dispositivos programables de procesamiento de paquetes. En primer lugar, en la implementación completamente actualizable de los

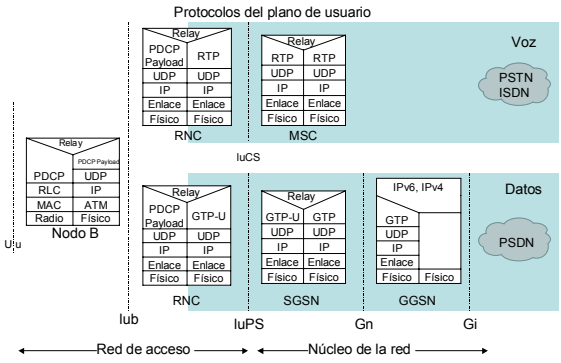


Figura 5. Plano de usuario UMTS all-IP

elementos del núcleo de la red. Así, algunos fabricantes de Network Processors [15] han añadido a sus diseños iniciales -sólo IP/IEEE802-, soporte de conmutación ATM con AAL2 y AAL5, segmentación y reensamblado (SAR). De esta manera pretenden que estos procesadores sean directamente aplicables a la implementación del plano de datos de BSC/RNC, SGSN, GGSN y MSC[2] desde la actual release 99 basada en ATM/IP (Fig. 4) a la futura arquitectura all IP (Fig. 5), prometiendo velocidades hasta OC-48 y haciendo innecesaria la actualización de los componentes físicos de los equipos.

Otras aplicaciones de red programables distintas de la mera implementación flexible de los protocolos UMTS que proponemos incluyen la optimización de prestaciones en aplicaciones TCP/IP en los terminales, de manera análoga a las propuestas existentes en la literatura para WiFi o satélite [12].

Las limitaciones en ancho de banda, de procesamiento y memoria de los terminales inalámbricos pueden mejorarse substancialmente con dispositivos activos dentro de la red (mediante retransmisiones, caching, control de flujo en función de la capacidad disponible en el segmento radio, etc) especialmente si éstos se ubican cerca de los terminales. Sin embargo, esto no es sencillo en UMTS debido a la movilidad y a los mecanismos de gestión de la misma. Por consiguiente, tanto en los nodos B como en la RNC sólo sería posible lanzar aplicaciones que no precisen traspasar su estado entre nodos B y RNCs. Si el traspaso del estado de la aplicación de red entre estos elementos fuera necesario, sería asimismo imprescindible implicar el traspaso de dicho estado en la señalización UMTS de traspaso gestionada por el SGSN. Esto no es imposible de realizar, pero no es general deseable, ya que impide mantener aislados los procesos de optimización de las aplicaciones de usuario de los

[2] Por limitaciones de espacio no se incluye en este artículo la descripción de los elementos, protocolos y acrónimos empleados en UMTS, remitiendo al lector a la norma para una documentación completa de los mismos.

procesos de señalización estándar. En definitiva, un funcionamiento compatible con la señalización actual de procesos que precisen traspaso de estado requiere que los procesos de usuario deban alojarse en equipos más allá del SGSN, donde no existe información específica de las condiciones del segmento radio móvil-nodo B. En este escenario la característica de SARA para el despliegue transparente de código a lo largo del trayecto de los paquetes daría soporte a un traspaso de GGSN, que no puede darse en la práctica.

Por último, la disponibilidad de interfaces abiertos de acceso a los recursos red como OSA/Parlay [21], abre en UMTS nuevas posibilidades para la explotación de las aplicaciones de red. La razón es que gracias al uso controlado de IP que se realiza en 3G se cubre una carencia específica de las aplicaciones de red en el contexto de IP: la imposibilidad práctica de aplicar un modelo de negocio en el uso de estas aplicaciones. Efectivamente, en 3G los usuarios están plenamente identificados y se dispone de mecanismos para facturar por el consumo que realizan de los recursos. Este procedimiento puede aplicarse al uso de las aplicaciones de red, agregándolas al conjunto de recursos contabilizados. La viabilidad de esta nueva perspectiva de uso del interfaz OSA/Parlay la estamos validando en la plataforma de pruebas del proyecto IST Opium [20]. En este caso, la aplicación de red es una caché transparente con pre-carga inteligente de objetos web ubicada tras el GGSN, y asociada al router de acceso a la internet pública. La figura 6 muestra la ubicación del nodo programable (etiquetado IWB en la figura) en la plataforma de pruebas.

4. Un prototipo de SARA

SARA [17] es un prototipo de nodo programable desarrollado en Java (y parcialmente C) para estudiar el paradigma *router*-asistente. Tal como se ha explicado anteriormente, el sistema es capaz de procesar transparentemente paquetes IP de señalización de las aplicaciones de red y cualquier flujo de paquetes de usuario que pasan por el router. Los primeros disparan la carga o refrescan el estado

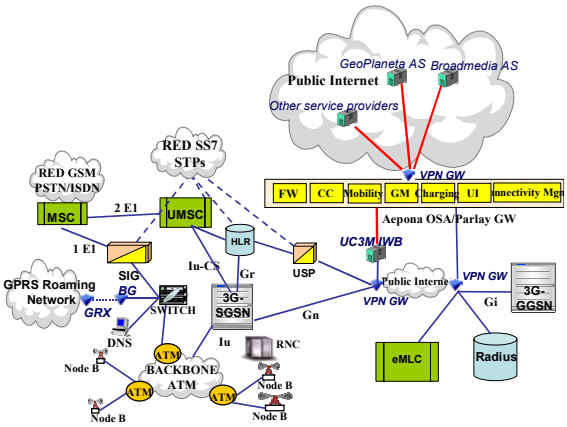


Figura 6. Escenario de uso de una pasarela OSA/Parlay para tarificación de un servicio de precarga web

de las aplicaciones, independientemente de su dirección destino. El router desvía los paquetes de señalización al asistente donde son analizados y procesados por las aplicaciones. Cada aplicación se ejecuta como un *thread* (nativo del S.O.) y puede hacer uso de las bibliotecas de análisis y modificación de sus cabeceras, extensión de JAVA (JNI/C para Linux) para dar servicio de *raw sockets*, vista del estado del router obtenida por SNMP, etc disponibles en el entorno.

Hoy en día hay dos posibles configuraciones. Ambas soportan IPv4 e IPv6, permitiendo a las aplicaciones activas un control total sobre los paquetes desviados. La primera está totalmente basada en Linux (cubriendo los dos papeles, como router y asistente) y la segunda es una plataforma híbrida donde se emplea un router Ericsson-Telebit AXI462 con un kernel modificado para interoperar con un PC asistente.

La experiencia preliminar con este prototipo indica que el techo de prestaciones de una aplicación es severo y viene impuesto por los cambios de contexto entre el kernel y el entorno de ejecución JAVA. Así una aplicación ejecutándose en la plataforma descrita en la figura 7, es capaz de analizar y procesar 2000 paquetes por segundo (Fig. 8), con tasas hasta 30 Mb/s, y con una penalización en retardo máximo de 2 ms por el desvío al asistente.

5. Conclusiones

La introducción de distintos grados de programabilidad en las redes de conmutación de paquetes está siendo objeto de investigación por parte de numerosos grupos de investigación, ya sea en forma de red activa, red programable o de procesadores de red. Como hemos presentado, uno de los más importantes campos de aplicación de esta tecnología puede ser la optimización de las comunicaciones en sistemas móviles. En este trabajo hemos presentado una arquitectura de red programable que incluye elementos que identificamos como esenciales para una aplicación práctica de esta tecnología en redes móviles: es una propuesta que permite una evolución progresiva a

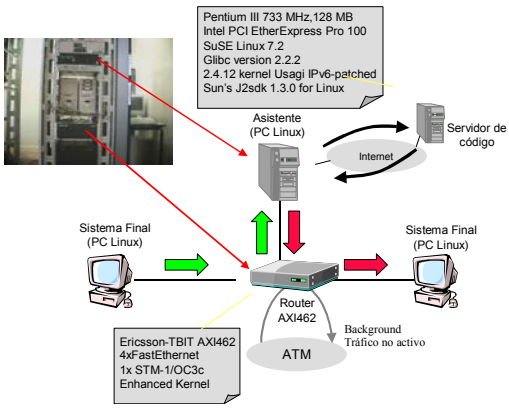


Figura 7. Banco de pruebas

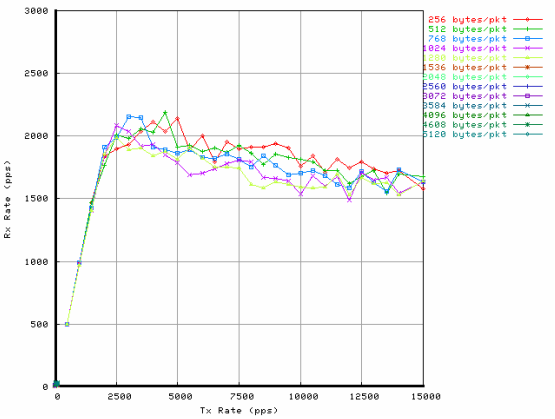


Figura 8. Tasa cursada vs tasa ofrecida (paquetes/s)

partir de routers convencionales, facilita la movilidad con su mecanismo de carga de código en nodos intermedios sin precisar su direccionamiento explícito, y presenta un compromiso razonable entre capacidad de control de los recursos del nodo y aislamiento de los procesos de tratamiento personalizado de paquetes. Además, la viabilidad de este enfoque se ha fundamentado con una arquitectura de seguridad que garantiza un uso controlado de las aplicaciones de red. Finalmente se han identificado escenarios de aplicación a implementar y se han descrito experiencias iniciales sobre una primera versión del prototipo. Queda por demostrar próximamente la viabilidad de algunas de las aplicaciones descritas sobre un escenario con tecnología UMTS real.

Referencias

[1] Proyecto MCYT AURAS. http://matrix.it.uc3m.es/~auras/.

[2] D. Wetherall, U. Legedza and J. Guttag, *Introducing new Internet services: Why and How*, IEEE Network Magazine, 1998.

[3] D. J. Wetherall, J. Guttag, and D. L. Tennenhouse, *ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols, IEEE OPENARCH'98*, San Francisco, CA, April 1998.

[4] Konstantinos Psounis. *Active networks: Applications, security, safety, and architectures*. IEEE Communications Surveys, 2(1), Q1 1999.

[5] L. Peterson, Y. Gottlieb, M. Hibler, P. Tullmann, J. Lepreau, S. Schwab, H. Dandekar, A. Purtell and J. Hartman. *An OS Interface for Active Routers*. In IEEE Journal on Selected Areas in Communications, 2001.

[6] Andrew T. Campbell, Herman G. De Meer, Michael E. Kounavis, Kazuho Miki, John B. Vincente, and Daniel Villela. *A survey of programmable networks.* Computer Communication Review, 29(2):7-23, April 1999.

[7] S. Floyd and L. Daigle, "IAB Architectural and Policy Considerations for Open Pluggable Edge Services," *RFC 3238*, January 2002.

[8] IEEE P1520: Proposed IEEE Standard for Application Programming Interfaces for Networks. http://www.ieee-pin.org/

[9] IETF Forwarding and Control Element Separation http://www.ietf.org/html.charters/forces-charter.html.

[10] P. Cowley, M. Fiuczynski, J-L. Baer, and Bershad. *Characterizing processor architectures for programmable network interfaces.* In Proc. International Conference on Supercomputing, Santa Fe, 2000.

[11] A. Calveras, X. de Porrata. Utilización de Proxies para mejorar el rendimiento de comunicaciones móviles en Internet. X Jornadas Telecom I+D. Barcelona-Madrid. Noviembre 2000. .

[12] J. Border, M. Kojo, J. Griner, G. Montenegro, Z. Shelby, Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations, *RFC 3135*. June 2001.

[13] V. Sunderam, J. Pascoe and G. Tonev. *Reconciling the Characteristics of Wired and Wireless Networks; The Janus Approach*. IEEE 4th Int. Workshop on Active Middleware Services. IEEE Computer Society, pp. 91-98, Edinburgh, Scotland. July 2002.

[14] Tschudin, C.; Lundgren, H.; Gulbrandsen, H. *Active routing for ad hoc networks* IEEE Communications Magazine, Volume: 38 Issue: 4 , pp: 122 -127Apr 2000.

[15] A. Millard. Tech.ical Report. *2.5G/3G wireless networks and the application of network processors.* http://www-3.ibm.com /chips/techlib/techlib.nsf/pages/main. August, 2002.

[16] Y-B Lin, A-C Pang, and Y-R. Haung and I. Chlamtac. *An All-IP approach for UMTS Third-Generation Mobile Networks*. IEEE Network, September/October 2002.

[17] D. Larrabeiti, M. Calderón, A. Azcorra and M. Urueña. *A practical approach to Network-based processing*. IEEE 4th International Workshop on Active Middleware Services. IEEE Computer Society, pp. 3-10. Edinburgh. July 2002.

[18] Marcelo Bagnulo, Bernardo Alarcos, María Calderón, Marifeli Sedano. "ROSA: Realistic Open Security Architecture for Active Networks". IWAN 2002, LNCS 2546, pp. 204-215. Zurich, Switzerland, December, 4-6 2002.

[19] Marcelo Bagnulo, Bernardo Alarcos, María Calderón, Marifeli Sedano. "Providing Authentication & Authorization Mechanisms for Active Service Charging". QofIS/ICQT 2002, LNCS 2511, pp. 337-456. Zurich, Switzerland, October 16-18, 2002.

[20] http://www.ist-opium.org/

[21] http://www.parlay.org/specs/

Gestión Semántica: Aplicando las Ontologías a la Gestión de Red

Jorge E. López de Vergara, Víctor A. Villagrá, Julio Berrocal.
Departamento de Ingeniería de Sistemas Telemáticos. Universidad Politécnica de Madrid.
E.T.S.I. de Telecomunicación. Av. Complutense, s/n. 28040 Madrid.
Teléfono: 91 549 57 00. Fax: 91 336 73 33.
E-mail: {jlopez, villagra, berrocal}@dit.upm.es

Abstract. The multiplicity of Network Management models (SNMP, CMIP, DMI, WBEM...) has raised the need of defining multiple mechanisms to allow the interoperability among all involved management domains. One basic component of such interoperability is the mapping between the information models that each domain specifies. These mappings, usually carried out with syntactical translations, can reach the semantic level by using ontologies. This article shows the advantages of using formal ontology techniques to improve the integration of current network management models. Applying this representation method, network managers can work and reason with an abstract view of the management information, independent of the specific management model used to interoperate with the managed resources, which can also include basic behaviour constraints.

1 Introducción

En la actualidad existen varios modelos de gestión de red integrada que usan distintas tecnologías. Entre ellos está SNMP (*Simple Network Management Protocol*, Protocolo Simple de Gestión de Red), CMIP (*Common Management Information Protocol*, Protocolo Común de Gestión de Información), DMI (*Desktop Management Interface*, Interfaz de Gestión de Ordenadores de Sobremesa) y WBEM (*Web Based Enterprise Management*, Gestión Basada en Web de Empresa). También se han aplicado tecnologías de procesamiento distribuido como CORBA (*Common Object Request Broker Architecture*, Arquitectura Común de Intermediarios de Peticiones de Objetos) [1]. Cada uno de estos modelos ha identificado la necesidad de definir información para describir los recursos a gestionar, permitiendo de esta forma que exista un conocimiento común entre gestores y agentes. Por ello, cada modelo de gestión posee un lenguaje de definición propio con el que se han definido bases de información de gestión (MIBs, *Management Information Bases*) también específicas de cada dominio de gestión.

Esta multiplicidad de modelos supone un problema cuando hay que utilizar distintas de estas tecnologías para acceder a los distintos recursos que componen un sistema. En estos casos es necesario establecer mecanismos que permitan la interoperabilidad entre los distintos modelos de gestión implicados. Sin embargo, las propuestas existentes hasta la fecha, tales como IIMC (*ISO-Internet Management Coexistence*, Coexistencia de Gestión ISO e Internet), JIDM (*Joint Inter-Domain Management*, Gestión Inter-Dominio Unificada), o incluso CIM (*Common Information Model*, Modelo de Información Común) sólo han planteado traducciones sintácticas que únicamente reescriben los modelos de información [2]. Esto quiere decir que si un mismo recurso está descrito en dos modelos de información, se puede realizar una traducción directa entre las estructuras que componen la especificación, pero no entre el significado que contienen, lo que supone un problema para alcanzar una gestión integrada. Para solventarlo, es necesario aprovechar la semántica contenida en la información.

Al mismo tiempo, las ontologías se han utilizado con éxito para resolver problemas similares en otros dominios como el de la Web Semántica [3], en el que esta técnica de representación del conocimiento proporciona semántica a las páginas y servicios *web*. Este artículo estudia cómo las ontologías también pueden ser útiles para la gestión de red, permitiendo unificar desde un punto de vista semántico las actuales definiciones heterogéneas de información. Para ello, antes que nada se presentan las ontologías y se comparan con los actuales modelos de información de gestión. Tras esto se proponen distintos pasos para obtener modelos con una mayor capacidad semántica, e integrables en una ontología de gestión. Finalmente se aportan las conclusiones obtenidas en este estudio.

2 Ontologías

Las Ontologías son una de las principales aproximaciones utilizadas en el ámbito de la Gestión del Conocimiento y la Inteligencia Artificial para resolver cuestiones relativas a la semántica. Para comprender cómo se pueden aplicar a los modelos de información de gestión, conviene explicar en qué consisten. Una Ontología

se puede definir como “una especificación explícita y formal de una conceptualización compartida” [4]:

- Es explícita porque define los conceptos, propiedades, relaciones, funciones, axiomas y restricciones que la componen.
- Es formal porque es legible e interpretable por máquinas.
- Es una conceptualización porque es un modelo abstracto y una vista simplificada de las entidades que representa.
- Finalmente, es compartida porque ha habido un consenso previo sobre la información, que ha sido acordado por un grupo de expertos.

En breve se puede decir que una ontología es la definición de un conjunto de conceptos, su taxonomía, interrelación y las reglas que gobiernan dichos conceptos.

Las ontologías se pueden clasificar en dos grandes grupos: las ligeras y las pesadas. Las primeras incluyen a aquellas que modelan la información referida a un dominio pero sin usar axiomas o restricciones, por lo que es difícil razonar con ellas. Las segundas sí que incluyen todos los elementos que les permiten inferir conocimiento a partir de la información definida.

Según esto, los modelos de información de gestión existentes se podrían entender como ontologías ligeras. Las MIBs de Internet o los esquemas CIM definen la información del dominio de la gestión de manera parcialmente formal y han sido acordadas en grupos de trabajo. Sin embargo, su semántica está limitada, al no poder especificar restricciones sobre la información [5].

Otra cuestión que diferencia a las ontologías de los modelos de información de gestión es la forma en que se aborda el problema de la interoperabilidad. Las soluciones que se aplican para integrar distintas ontologías no tratan sólo la traducción sintáctica de distintos lenguajes, sino otras cuestiones para permitir la interoperabilidad a un nivel semántico. Las propuestas por parte de los distintos grupos de investigación que trabajan en este ámbito incluyen la fusión de modelos para obtener un modelo común, o bien el establecimiento de correspondencias entre los modelos.

3 Aplicando las ontologías a la gestión de red

La presente sección contempla el problema de integración de los modelos de gestión desde la perspectiva de las ontologías y plantea una propuesta para mejorar la interoperabilidad semántica de las distintas especificaciones de información. Para ello se han estudiado tres pasos, contenidos en los tres apartados siguientes, ilustrados en la Figura 1:

1. Primero se estudian las posibilidades de emplear un lenguaje de definición de ontologías para especificar información de gestión, adaptando dicho lenguaje para que pueda expresar algunas construcciones típicas de los lenguajes de información de gestión.

2. Partiendo de los modelos de información, expresados con lenguajes de ontologías, se podrán aplicar de manera conjunta las técnicas de fusión y correspondencia definidas en este campo. Para ello se define un método que combina ambas técnicas y las particulariza para el caso concreto de la información de gestión. Con esto se podrá obtener un nuevo modelo que realmente integre los ya existentes teniendo en cuenta la semántica contenida en los mismos y declarando al mismo tiempo las distintas reglas de correspondencia con los modelos iniciales. De esta manera se podrá solventar el problema de interoperabilidad identificado anteriormente.

3. Finalmente, se analiza la forma de incluir comportamiento a las especificaciones de información de gestión aprovechando las características de formalización que proporcionan los lenguajes de ontologías, pudiendo integrar dichas especificaciones en el modelo común generado anteriormente para poder tener modelos de información completos como ocurre con las ontologías pesadas.

3.1 Escribiendo información de gestión con lenguajes de ontologías

Para especificar la información de gestión teniendo en cuenta el significado de la misma, sería necesario un lenguaje de definición cuya capacidad expresiva le permita expresar una semántica precisa, incluyendo restricciones sobre la

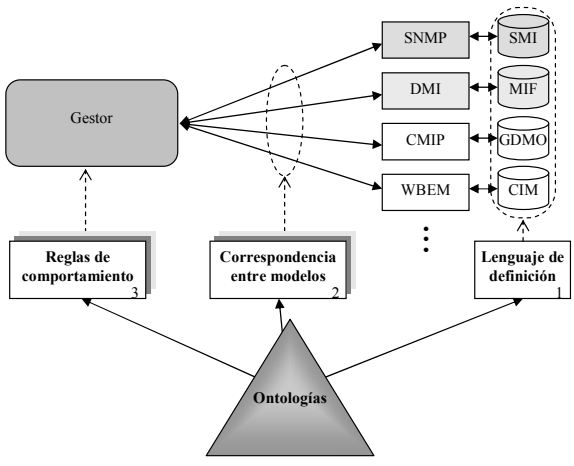


Figura 1. Aplicaciones de las ontologías.

información. Este apartado estudia las posibilidades de emplear un lenguaje de definición de ontologías para especificar información de gestión. Esto supone tener que adaptar dicho lenguaje para que también pueda expresar algunas construcciones típicas de los lenguajes de información de gestión.

Existen otras propuestas en este sentido. En [6] se propone una traducción entre SMIng (*Structure of Management Information, next generation*, Siguiente Generación de la Estructura de la Información de Gestión) y RDF (*Resource Description Framework*, Marco de Descripción de Recursos) para que lo puedan usar agentes inteligentes que se comunican mediante un vocabulario común basado en ontologías. Aunque en este caso se definen términos referidos a SMIng en RDF, no se hace para el caso general de cualquier lenguaje de gestión. Por otro lado, en [7] se trata de integrar el metamodelo de CIM dentro de una herramienta de ontologías. La justificación es poder utilizar la información definida en CIM para el intercambio de información entre agentes inteligentes usando OKBC (*Open Knowledge Base Connectivity*, Conectividad Abierta de Bases de Conocimiento) [8]. Sin embargo, no llega a utilizar directamente ningún lenguaje de ontologías para la definición de información de gestión.

Los lenguajes de ontologías actualmente en mayor uso y con mayor número de herramientas disponibles son aquellos relacionados con la Web Semántica. Entre estos destaca DAML+OIL (*DARPA Agent Markup Language + Ontology Inference Layer*, Lenguaje de Marcas de Agentes de DARPA + Capa de Inferencias de Ontologías) [9]. Otra ventaja de este lenguaje frente a los de definición de información de gestión es que está formalizado en KIF (*Knowledge Interchange Format*, Formato de Intercambio de Conocimiento) [10], con lo que su semántica es unívoca y puede ser usada por sistemas inteligentes. Por el contrario, DAML+OIL no es un lenguaje específico de gestión, y no posee construcciones para definir todas las facetas típicas de gestión.

Aunque el estudio que se muestra en este apartado se refiere a DAML+OIL, se puede generalizar para otros lenguajes de definición de ontologías con características similares. Todos aquellos lenguajes que permitan definir clases y propiedades pueden valer para definir información de gestión, si bien es posible que se pierda claridad si estos lenguajes no poseen las facetas adecuadas, ni un mecanismo para definir otras nuevas.

DAML+OIL es un lenguaje de ontologías bastante completo, pues permite definir clases y propiedades, que pueden estar en el dominio de una clase. A su vez, las propiedades poseen distintas facetas, como la restricción de tipo o de cardinalidad, así como la documentación. También permite definir especializaciones de clases, con herencia múltiple, así como otro tipo de relaciones con restricción de rango. Finalmente, también se pueden expresar ejemplares de estas clases. Como muestra de ello, a continuación se define en DAML+OIL la clase del esquema nuclear de CIM `CIM_ManagedSystemElement` y sus propiedades asociadas. Los tipos de datos utilizados son los definidos en XSD (*XML Schema Data types*, Tipos de Datos de Esquemas XML) [11].

```
<daml:Class rdf:ID=
   "CIM_ManagedSystemElement">
      <rdfs:comment>
CIM_ManagedSystemElement is the base class for
the System Element hierarchy. […]
      </rdfs:comment>
      <rdfs:subClassOf rdf:resource=
         "#CIM_ManagedElement"/>
</daml:Class>

<daml:DatatypeProperty rdf:ID=
   "InstallDate">
   <rdfs:comment>
A datetime value indicating when the object was
installed. A lack of a value does not indicate that
the object is not installed.
   </rdfs:comment>
   <daml:domain rdf:resource=
      "#CIM_ManagedSystemElement"/>
   <daml:range rdf:resource=
      "http://www.w3.org/2000/10/XMLSchema
#dateTime"/>
</daml:DatatypeProperty>

<daml:DatatypeProperty rdf:ID="Name">
   <rdfs:comment>
The Name property defines the label by which the
object is known. When subclassed, the Name
property can be overridden to be a Key property.
   </rdfs:comment>
   <daml:domain rdf:resource=
      "#CIM_ManagedSystemElement"/>
   <daml:range rdf:resource=
      "http://www.w3.org/2000/10/XMLSchema
#string"/>
</daml:DatatypeProperty>

<daml:DatatypeProperty rdf:ID="Status">
   <rdfs:comment>
A string indicating the current status of the object.
Various operational and non-operational statuses
are defined. […]
   </rdfs:comment>
   <daml:domain rdf:resource=
      "#CIM_ManagedSystemElement"/>
   <daml:range rdf:resource=
      "http://www.w3.org/2000/10/XMLSchema
#string"/>
</daml:DatatypeProperty>
```

No obstante, DAML+OIL no incluye facetas típicas de los lenguajes de gestión como el valor por defecto o el tipo de acceso, entre otras. Para que este lenguaje contenga dicha información habría que definir en RDF cada una de estas construcciones. A continuación se presenta un par

de ejemplos que especifican algunas facetas de gestión:

```
<!-- VALOR POR DEFECTO -->
<rdf:Property rdf:ID="defaultValue">
   <rdfs:label>defaultValue</rdfs:label>
   <rdfs:comment>
It defines the default value of a property.
   </rdfs:comment>
   <rdfs:domain rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-
syntax-ns#Property"/>
</rdf:Property>
```

```
<!-- ACCESO -->
<rdf:Property rdf:ID="access">
   <rdfs:label>access</rdfs:label>
   <rdfs:comment>
It defines the access of a property, which can be
readable and/or writeable.
   </rdfs:comment>
   <rdfs:domain rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-
syntax-ns#Property"/>
   <rdfs:range rdf:resource=
      "#accessString"/>
</rdf:Property>

<xsd:simpleType name="accessString">
   <xsd:restriction base="string">
      <xsd:enumeration value=
         "read-only"/>
      <xsd:enumeration value=
         "read-write"/>
      <xsd:enumeration value=
         "read-create"/>
   </xsd:restriction>
</xsd:simpleType>
```

Con todo, no es posible expresar métodos u operaciones en este lenguaje. Sin embargo, en general las especificaciones de información de gestión no suelen incluir muchas de estas construcciones, por lo que éste no es un gran problema.

Entonces, a partir de las estructuras básicas de DAML+OIL junto con las específicas de gestión, la información de gestión actual es fácilmente traducible a un lenguaje de ontologías. Únicamente habría que comparar las construcciones que posee cada lenguaje como se hace en [12] para llevar a cabo esta traducción.

3.2 Fusión y Correspondencia (Método M&M)

Aunque todos los modelos de información de gestión se expresen en un mismo lenguaje, esto no supone integrarlos semánticamente, a pesar de que el lenguaje sea de ontologías. Es necesario llevar a cabo otros procedimientos que identifiquen el significado contenido en estos modelos. Esto es posible haciendo uso de las técnicas de fusión y correspondencia empleadas en las ontologías. Para ello este apartado expone un método que no se basa en realizar reescrituras de la información, sino que combina distintas definiciones en un modelo común, declarando las distintas reglas de correspondencia con los modelos iniciales, como se ilustra en la Figura 2.

Para asistir en este proceso se ha definido el método denominado M&M (*Merge and Map*, Fusión y Correspondencia). En él se propone un conjunto de pasos para ayudar a obtener simultáneamente el modelo común y las reglas de correspondencia. Está basado en el método de fusión de [13] adaptándolo al caso particular de la gestión de red, y añadiendo todo lo que se refiere a la definición de reglas de correspondencia, con lo que por cada elemento que se fusione se añade una regla que relaciona los elementos fusionados en la ontología de correspondencia.

Este método no genera un resultado de manera automática. Se trata de un método que ayuda a la persona que deba realizar esta labor en el proceso de fusión y correspondencia. Para esto se proponen los siguientes heurísticos, que permitirán identificar candidatos a fusionar con gran probabilidad:

- De correspondencia por similitud en cadenas de caracteres, analizando y comparando aquellas que estén incluidas en identificadores de clases o propiedades, o en su descripción, y seleccionando aquellas que sean similares.

- De correspondencia por similares jerarquías de herencia, que permitirán fusionar clases si sus clases madres son semejantes, dado que las clases hijas de una dada suelen ser similares a las clases hijas de la clase fusionada con ésta.

- De correspondencia por el dominio al que pertenecen las propiedades, pues si una clase se corresponde con otra, será normal que las propiedades de dicha clase también se correspondan con las de la otra. Al mismo tiempo, si es posible identificar dos propiedades semejantes, también se puede suponer que las clases a las que pertenecen también se corresponden.

Para facilitar la fusión no se ha pensado en definir un nuevo modelo común, sino partir de uno de los

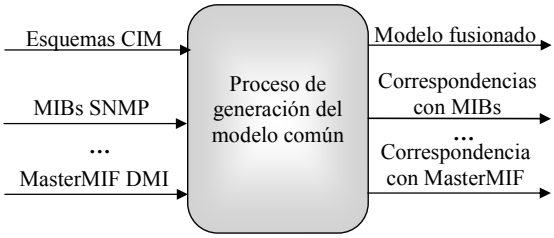


Figura 2. Proceso de fusión y correspondencia de información de gestión.

ya existentes [12]. En este caso se utiliza el esquema CIM como base para fusionar otros modelos ya existentes, como la MIB-II, la HOST-RESOURCES-MIB o la definida en la recomendación M.3100.

También se ha identificado que las correspondencias típicas entre los elementos de los modelos de información de gestión son las siguientes o una combinación de las mismas:

- Directa, si es una relación 1:1 en la que no hay que hacer ninguna transformación. En este caso, un elemento de un modelo es exactamente igual al de otro modelo.

- De valores, si es una relación 1:1 en la que cada elemento toma valores según una enumeración, que es distinta para cada caso.

- De tipos de datos, si es una relación 1:1 en la que cada elemento posee un tipo de datos distintos, teniéndose que adaptar para que cada dominio de gestión reciba el que tiene definido.

- De operación aritmética sobre un elemento, si es una relación 1:1 en la que un elemento se obtiene realizando una operación aritmética con una constante sobre el otro elemento. Este caso se da, por ejemplo, cuando hay que cambiar las unidades con que se mide ese parámetro.

- De operación aritmética sobre varios elementos, si es una relación 1:n en la que un elemento se obtiene por combinación aritmética del resto.

- De cadenas de caracteres, si es una relación 1:n en la que un elemento se compone de la concatenación de distintas cadenas de caracteres.

Para describir las reglas de correspondencia que traduzcan los ejemplares de un modelo concreto al modelo común se ha definido una ontología muy simple, representada en la Figura 3.

Su estructura es la siguiente: cada posible elemento (`Element`) que compone una ontología (clases, propiedades, etc.) posee una fórmula (`Formula`) de traducción. Además, cada elemento contiene

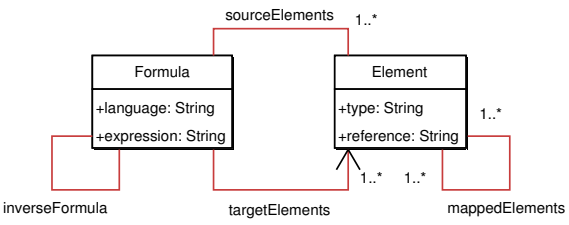


Figura 3. Ontología de correspondencia.

propiedades tales como el tipo (`type`) o la referencia a su definición (`reference`). Por su parte, las fórmulas se relacionan con el conjunto de elementos origen (`sourceElements`) y destino (`targetElements`), así como una expresión (`expression`) usada para traducir entre ambos en un lenguaje concreto (`language`). También se incluyen relaciones con los elementos de la otra ontología con que se corresponde (`mappedElements`) y la fórmula inversa (`inverseFormula`). Con esta ontología es posible representar cualquier correspondencia de las definidas anteriormente. Además, no existen las limitaciones del calificador `MappingStrings` de CIM, que sólo permite representar correspondencias directas.

La Figura 4 muestra el diagrama de actividad que describe el método M&M. Las actividades con fondo gris son aquellas que realizaría el usuario, mientras que las de fondo blanco serían efectuadas por el sistema. En resumen, el método M&M consiste en identificar a través de los heurísticos comentados clases similares, y tras esto fusionar los distintos atributos que hay en ellas. Al mismo tiempo, se define de manera automática en la ontología de correspondencia cada elemento a fusionar, y luego se van definiendo las fórmulas asociadas y elementos que se corresponden con éste. Durante este proceso, la persona que lo realiza debe ir validando cada una las operaciones propuestas por el método, pudiendo también definir otras diferentes. El resultado final es un modelo común y un conjunto de ejemplares de la ontología de correspondencia que representen las reglas de traducción de los modelos fusionados.

Un gestor basado en el modelo común y esta ontología de correspondencia funcionaría de la siguiente manera. Si necesita obtener todos los ejemplares de un elemento concreto del modelo común, lo buscará en la ontología de correspondencia, encontrando a la vez la fórmula y elementos correspondientes de los modelos fusionados. Se accederá a cada dominio y se obtendrán dichos elementos. Al aplicar la expresión contenida en la fórmula se traduciría el valor de los ejemplares obtenidos en cada dominio de gestión al modelo que maneja el gestor.

3.3 Definiciones de comportamiento

Un paso más allá de la integración de información es el que se consigue añadiendo un conjunto de restricciones al modelo común de gestión que se obtenga. Esto permite describir el comportamiento relativo a la información contenida en dicho modelo, que podrá ser comprobado en el gestor. Las actuales definiciones de información de gestión incluyen algunas reglas acerca del comportamiento de la información de gestión, pero están escritas en los campos `BEHAVIOUR` o `DESCRIPTION` en lenguaje natural que no es legible por una máquina.

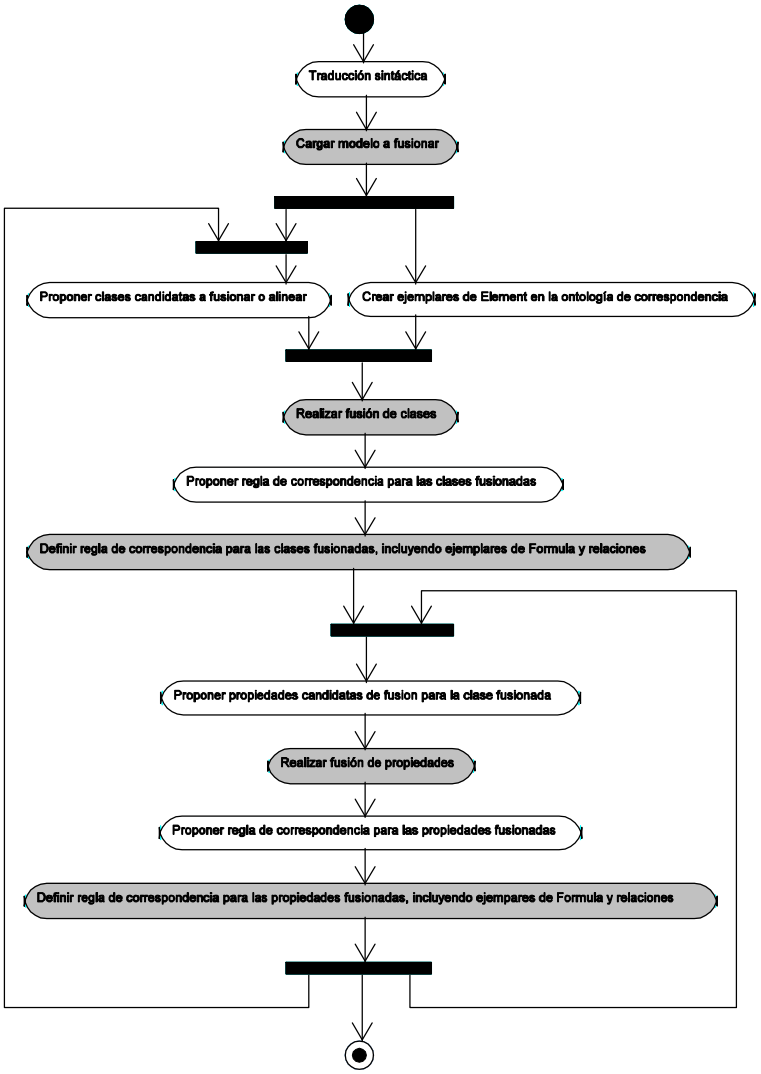


Figura 4. Diagrama de actividad del método M&M.

También aquí se pueden aplicar las ontologías, pues suelen incluir axiomas y restricciones para especificar su comportamiento, que al definirse formalmente sí pueden interpretarse de manera automática.

Las restricciones a incluir en la información pueden ser de dos tipos. El primero se ha denominado restricción implícita y se refiere a la información que en un estado de operación normal siempre se deben cumplir, porque así es como se ha definido la información. El segundo tipo se ha denominado restricción explícita y se refiere a que un gestor concreto pueda definir nuevas restricciones sobre información ya definida para particularizar de esta manera el comportamiento de los sistemas gestionados.

Las restricciones implícitas se encuentran habitualmente definidas en lenguaje natural en las descripciones de las clases y atributos de información de gestión. Un ejemplo de ello podrían ser las restricciones contenidas dentro de la clase `CIM_Printer` de los esquemas CIM, como la que dice "*a language that is used as a default by the Printer should also be listed in LanguagesSupported*", que restringe el lenguaje que debe manejar una impresora a los contenidos en la lista de lenguajes soportados.

Las restricciones explícitas siguen una política concreta. Por ejemplo, se puede definir una política que haga cumplir que el espacio disponible en los sistemas de archivos de un equipo nunca deberá ser inferior del 10% de la capacidad total.

Respecto a la definición de estas restricciones, DAML+OIL permite definirlas con lógica de primer orden respecto a propiedades algebraicas de de relaciones (simetría, transitividad, unicidad). Además, se pueden definir restricciones de universalidad y existencia para las clases y propiedades. Sin embargo, DAML+OIL no es un lenguaje totalmente completo a la hora de especificar restricciones, pues éstas se refieren a propiedades que referencian objetos, no siendo igual de expresivo con propiedades que contienen tipos de datos. Por este motivo, otros lenguajes de ontologías como KIF son mejores para definir este tipo de información.

El ejemplo propuesto para la restricción implícita podría definirse de la siguiente manera en DAML+OIL:

```
<daml:Class rdf:about"#CIM_Printer">
   <daml:Restriction>
      <daml:onProperty rdf:resource=
         "#LangaugesSupported"/>
      <daml:hasClass rdf:resource=
         "#DefaultLanguage"/>
   </daml:Restriction>
</daml:Class>
```

Asimismo, utilizando KIF, se expresaría de la siguiente manera:

```
(defrange ?printer :FRAME CIM_Printer)
(forall ?printer
   (element-of
      (DefaultLanguage ?printer)
      (LanguagesSupported ?printer)))
```

Si se emplea nuevamente KIF para el segundo ejemplo, esta sería:

```
(defrange ?fs :FRAME CIM_FileSystem)
(forall ?fs
   (> (AvailableSpace ?fs)
      * 0.10 (FileSystemSize ?fs)))
```

Para el caso de DAML+OIL se aprecia lo que se comentaba anteriormente. Es posible definir una restricción de valores aprovechando las construcciones de XSD, pero no se puede definir un valor máximo con una operación. Por ejemplo, si el espacio disponible se midiera en porcentaje y no en valores absolutos sí se podría definir la siguiente restricción:

```
<xsd:simpleType name="over10">
   <xsd:restriction base=
      "xsd:positiveInteger">
      <xsd:minInclusive value="10"/>
<!-- Como es un porcentaje se podría
incluir una restricción de máximo -->
      <xsd:maxInclusive value="100"/>
   </xsd:restriction>
</xsd:simpleType>

<daml:Class rdf:about="#CIM_FileSystem">
   <daml:Restriction>
      <daml:onProperty rdf:resource=
         "#AvailableSpace"/>
      <daml:toClass rdf:resource=
         "#over10"/>
   </daml:Restriction>
</daml:Class>
```

El comportamiento que se defina mediante estas restricciones se incluirá en la ontología que modele la información de gestión, con lo que puede ser más tarde automáticamente procesado por sistemas de gestión inteligentes que razonen con la información basada en las ontologías que se les proporcione.

4 Conclusiones

En este artículo se ha presentado cómo se puede aplicar la técnica de representación formal conocida como ontología para mejorar la definición e integración de la información de gestión de red.

Escribir la información de gestión con un lenguaje de ontologías como DAML+OIL mejora la expresividad de dicha información, existiendo asimismo múltiples herramientas desarrolladas para su uso y validación. Sin embargo, también es necesario ampliar este tipo de lenguajes para que puedan expresar toda la información contenida habitualmente en los modelos de gestión.

Los trabajos existentes hasta la fecha respecto de la integración de información de gestión trataban ésta de forma muy limitada, basada sobre todo en traducciones sintácticas. La aplicación del método M&M permite realizar esta tarea a partir del significado contenido en esta información, lo que permite que un gestor pueda manejar un único modelo, con total transparencia de los dominios de gestión subyacentes. Esto redundará en una mejora de las aplicaciones de gestión, que pueden relacionar datos que hasta la fecha no tenían una asociación directa al pertenecer a distintos dominios de gestión. Durante la experimentación se ha constatado que su aplicación a modelos de información de gran tamaño puede llevar bastante tiempo debido a la necesidad de intervención humana para comprobar la validez de las reglas propuestas. Sin embargo, este tiempo será en general menor que si se realiza esta tarea a mano.

Al aprovechar las características de los lenguajes de ontologías, la posibilidad de definir de restricciones se obtiene por añadidura. Estas reglas, a diferencia de los modelos habituales, pueden ser interpretadas. Con esto, se puede obtener una especificación de información de un modelo común e independiente del dominio de gestión, cuyos conceptos además posean restricciones acerca de comportamiento.

Basándose en estas ideas, se puede desarrollar un sistema de gestión como el que se muestra en la Figura 5, que aproveche esta aproximación basada en ontologías, integrando todos los modelos de gestión de manera inteligente, teniendo en cuenta la semántica de la información definida. Al mismo tiempo, se pueden construir pasarelas genéricas que utilicen las ontologías de correspondencia obtenidas al aplicar el método M&M para traducir la información a cada dominio de gestión.

Los trabajos actuales incluyen traductores automáticos de CIM y SMI a DAML+OIL. Al mismo tiempo, se está adaptando el método M&M a una herramienta de ontologías existente, para automatizar el proceso de fusión y correspondencia.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Ciencia y Tecnología a través del proyecto GESEMAN (TIC2002-00934).

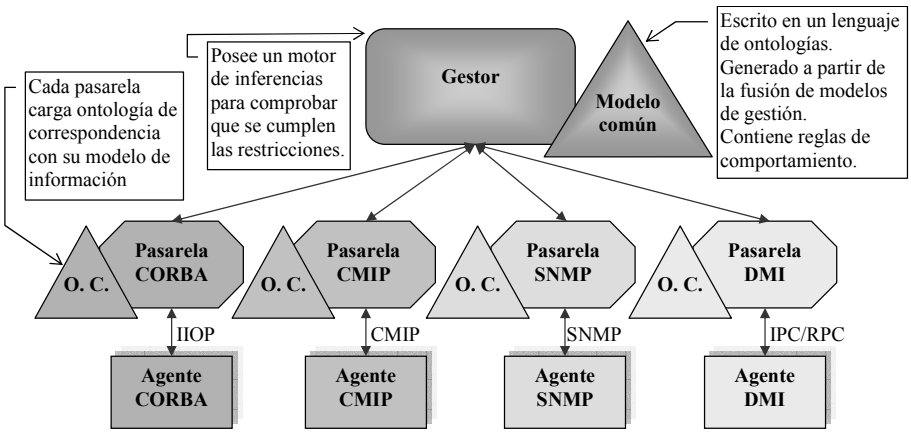


Figura 5. Arquitectura de un gestor que maneje ontologías

Referencias

[1] Heinz-Gerd Hegering, Sebastian Abeck, Bernhard Neumair, "Integrated Management of Networked Systems". Morgan Kaufmann, 1999.

[2] Jorge E. López de Vergara, Víctor A. Villagrá, Julio Berrocal, "Semantic Management: advantages of using an ontology-based management information meta-model", Proceedings of the HP Openview University Association Ninth Plenary Workshop (HP-OVUA'2002), Böblingen, Alemania, junio de 2002.

[3] Tim Berners-Lee, James Hendler, Ora Lassila, "The Semantic Web", Scientific American, mayo de 2001.

[4] R. Studer, V.R. Benjamins, D. Fensel, "Knowledge Engineering: Principles and Methods", Data & Knowledge Engineering. 25: 161-197, 1998.

[5] Jorge E. López de Vergara, Víctor A. Villagrá, Julio Berrocal, Juan I. Asensio, Roney Pignaton, "Semantic Management: Application of Ontologies for the Integration of Management Information Models", Proceedings of the Eighth IFIP/IEEE International Symposium on Integrated Network Management (IM'2003), Colorado Springs, Colorado, EE. UU. A., marzo de 2003.

[6] Jun Shen, Yun Yang, "RDF-Based Knowledge Models for Network Management", Proceedings of the Eighth IFIP/IEEE International Symposium on Integrated Network Management (IM'2003), Colorado Springs, Colorado, EE. UU. A., marzo de 2003.

[7] Emmanuel Lavinal, Thierry Desprats, Yves Raynaud, "A Conceptual Framework for Building CIM-Based Ontologies", Proceedings of the Eighth IFIP/IEEE International Symposium on Integrated Network Management (IM'2003), Colorado Springs, Colorado, EE. UU. A., marzo de 2003.

[8] Vinay K. Chaudhri, Adam Farquhar, Richard Fikes, Peter D. Karp, James P. Rice, "OKBC: A Programmatic Foundation for Knowledge Base Interoperability", Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI'98), Madison, Wisconsin, EE. UU. A., julio de 1998.

[9] Dan Connolly, Frank van Harmelen, Ian Horrocks, Deborah L. McGuiness, Peter F. Patel-Schneider, Lynn Andrea Stein, "DAML+OIL (March 2001) Reference Description", W3C Notes, 18 de diciembre de 2001.

[10] Richard Fikes, Deborah McGuinness, "An Axiomatic Semantics for RDF, RDF-S, and DAML+OIL (March 2001)", W3C Note, 18 de diciembre de 2001.

[11] Paul V. Biron, Ashok Malhtra, "XML Schema Part 2: Datatypes", W3C Recommendation, 2 de mayo de 2001.

[12] Jorge E. López de Vergara, Víctor A. Villagrá, Juan I. Asensio, Julio Berrocal, "Ontologies: Giving Semantics to Network Management Models", IEEE Network, special issue on Network Management, Volume 17, Number 3, mayo/junio de 2003.

[13] Natalya Fridman Noy, Mark A. Musen, "An Algorithm for Merging and Aligning Ontologies: Automation and Tool Support", Proceedings of the Workshop on Ontology Management, Sixteenth National Conference on Artificial Intelligence (AAAI-99), Orlando, Florida, EE. UU. A., julio de 1999.

Agentes móviles para composición de servicios web

Sergio F. Castillo y Juan R. Velasco
Universidad Industrial de Santander, Ciudad Universitaria
Bucaramanga (Colombia)
scastillo@uis.edu.co
Universidad de Alcalá, Escuela Politécnica, Departamento de Automática
Crtra. N–II, Km 31,600, 28871 Alcalá de Henares, Madrid (Spain)
juanra@aut.uah.es

Abstract *Web services are supposed to be designed as software components for an intercommunicated business world. In this way, they are published to be used by any other software process or person that needs them. The exchange information format is also defined. In addition to this, one of the added value capabilities is the possibility of web service integration, federation or composition. The idea of service composition is not new, as it has been studied in the telecommunication world. In this field, technologies are needed to be as independent as possible. Mobile agents, as proposed in this paper, may be one of these technologies.*

1. Introducción

La composición de servicios web puede definirse de manera sencilla como la creación de un nuevo servicio a partir de la combinación de otros ya existentes. Esta composición ofrece un valor añadido a los servicios originales y se trata, sin duda, de una actividad fundamental de cualquier proceso de negocio ofrecido a través de un sitio web. De hecho, no resulta extraño encontrar un sitio web que ofrezca información o productos a sus visitantes generando éstos a partir de diferentes fuentes combinadas. De esta forma, sus visitantes obtienen un valor añadido, al no tener que navegar por la red en busca de esa información, con el consiguiemnte procesamiento posterior.

la idea que se encuentra detrás de los servicios web es precísamente ésa. Una compañía ofrece información o productos, junto a un mecanismo para poder acceder a ellos. Esta forma de acceso se publica en algún lugar conocido, de manera que cualquier usuario que desee utilizarlos u ofrecerlos, pueda hacerlo (en algunos casos tras la firma de un contrato con el propietario). Uno de los problemas claves que podemos encontrar en este proceso consiste en asegurar que todos los usuarios o servicios que acceden a la información son capaces de comunicarse con el servidos de manera correcta. Para resolver el problema es necesario hacer uso de una ontología [4] compartida. Este artículo presenta dos maneras distintas de implementar la composición de servicios haciendo uso de agentes móviles: por un lado, los agentes móviles pueden representar al usuario o al servicio principal y navegar por la red en busca de la información o los datos, contactando para ello con diferentes servicios web. Por otro lado, el problema ontológico puede ser resuelto parcialmente por medio de un agente al que llamamos *Ontos* [3], capaz de traducir datos y mensajes entre ontologías diferentes (aunque, como veremos, forzosamente relacionadas).

La siguiente sección presenta los diferentes elementos que vamos a necesitar en nuestro trabajo: la composición de servicios, una breve descripción de los servicios web y su funcionamiento, y una subsección sobre agentes y agentes móviles.

La sección 3 presenta cómo los agentes móviles pueden ser utilizados para la composición de servicios web en los dos sentidos a los que hemos hecho referencia: interacción entre servicios y traducción de ontologías.

Finalmente se presentan las conclusiones y las líneas de trabajo futuras.

2. El punto de partida

2.1. Composición de servicios

A pesar de su importancia, la composición de servicios ha recibido poca atención. Uno de los usos principales puede ser encontrado en el mundo de las telecomunicaciones, y más concretamente en la definición que TINA (telecommunications information networking architecture) hace de federación y composición de servicios [14]. Aunque el consorcio TINA ha pasado definitivamente a la historia, eso no quiere decir que no podamos tomar de su trabajo algunas ideas que son, sin duda, relevantes. De este modo, la composición de servi-

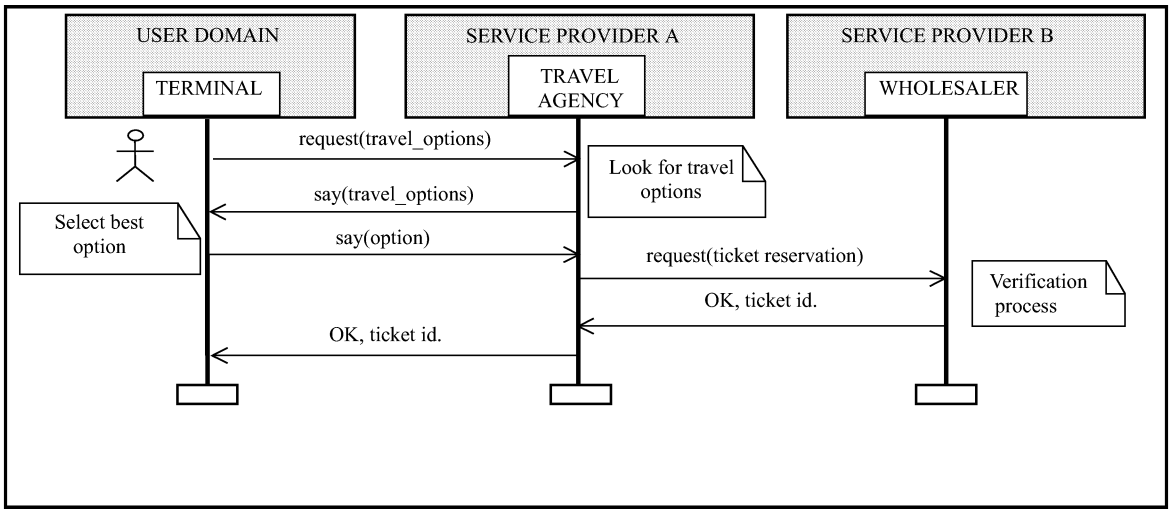


Figura 1: Composición de servicios

cios en TINA se basa en cuatro principios básicos [13]:

1. Identificación y localización de servicios en diferentes dominios.
2. Separación entre acceso al servicio y su uso posterior (esta es una característica básica de la arquitectura TINA.)
3. Gestión coherente de servicios compuestos.
4. Soporte para sesión e interacción.

Por su parte, los principios básicos de la federación son: acuerdo mutuo entre las partes, descentralización y autonomía de cada una de las partes.

Los autores de este artículo han trabajado en el proyecto SCARAB (Smart Card and Agent enabled Reliable Access, AC339)[10] y [1], financiado por la Unión Europea dentro del programa ACTS. En este proyecto, una de las mayores contribuciones técnicas consitió en la estructuración de la composición de servicios dentro de un entorno distribuido. Para ello se desarrolló una arquitectura basada en agentes móviles capaz de implementar esta composición de servicios dentro de una arquitectura TINA convencional [2]. Este proyecto fue el punto de partida para el trabajo desarrollado por los autores en composición de servicios web.

En primer lugar, es conveniente aclarar en qué consiste la composición de servicios. Para ello haremos uso de un ejemplo que nos acompañará a lo largo de todo el artículo. Consideremos un serviciode agencia de viajes online (al que llamaremos OTA, Online Travel Agency). OTA puede ofrecer sus propios productos, pero lo más habitual es que se abastezca de aquellos que las agencias mayoristas le ofrezcan. Supongamos, dentro de nuestro ejemplo, que OTA hace uso de diferentes servicios para ofrecer el producto final a sus clientes: uno de billete electrónico de avión, otro de reservas de hotel, un tercero de seguros de asistencia médica en el extranjero y un cuarto de entradas para espectáculos. Cuando un cliente solicita un viaje a OTA, con unas determinadas características, OTA puede componer los diferentes servicios que conoce y entregar a su cliente un producto completo y adaptado a sus desos. La figura 1 muestra, a través de un diagrama de secuencia UML, a un usuario que solicita un servicio a SP_A *Service Provider A* es la agencia de viajes), y SP_A solicita la composició de su servicio con el de un mayorista (en esta caso SP_B, *Service Provider B*, uno de sus proveedores).

2.2. Servicios web

Un servicio web es "un componente software, independiente de la plataforma y la implementación, que puede ser: descrito utilizando un lenguaje de descripción de servicios, publicado en un registro de servcios, descubierto por medio de mecanismos conocidos, invocado mediante una API declarada y compuesto con otros servicios" [5]. Los servicios se desciben por medio de WSDL (Web Service Description Language) [18] , se invocan haciendo uso de SOAP (Simple Object Application Protocol) [11], y pueden ser localizados por medio de un sistema basado en UDDI (Universal Description, Discovery and Integration) [15].

Los tres papeles que aparecen en el funcionamiento del modelo en el que se basan lo servicios web son:

- El proveedor del servicio (service provider): La entidad que alberga uno o más servicios web. El proveedor del servicio debe publicar sus servicios en el registro de servicios.

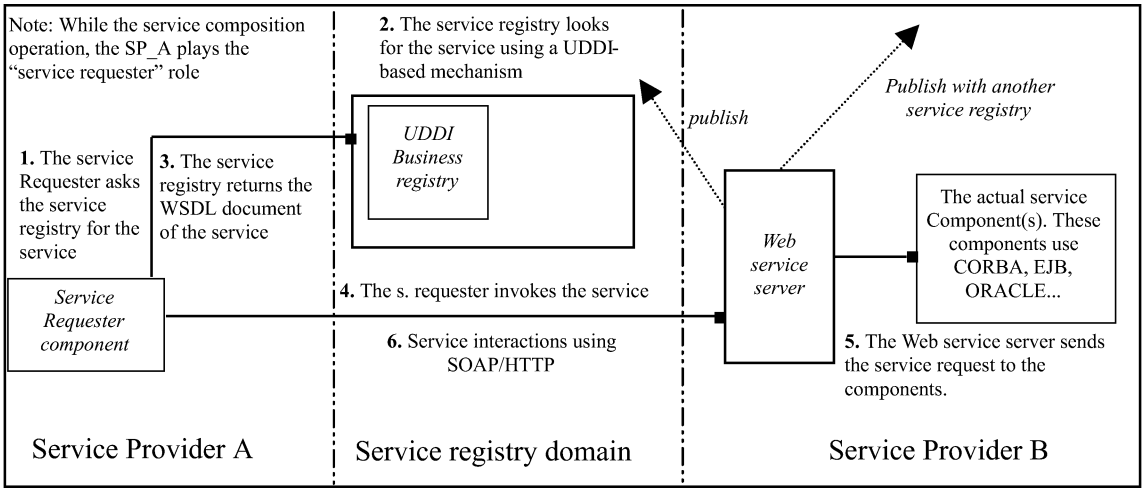


Figura 2: Elementos de los servicios web

- El solicitante del servicio (service requester): La entidad que invoca un servicio ofrecido por medio de SOAP. Los solicitantes de los servicios pueden localizarlos por medio del registro de servicios.

- El registro de servicios (service registry): La entidad donde se almacena la información sobre los servicios y sus proveedores. Esta entidad provee algún mecanismo de búsqueda para que los posibles ususarios de los servicios puedan localizarlos.

La figura 2 muestra la interacción entre estos tres papeles dentro del modelo de servicios web. Puesto que el objetivo de este trabajo se centra en la composición de servicios, la figura muestra cómo el proveedor de servicios A localiza un nuevo servicio por medio de UDDI, e inicia una interacción con el proveedor B, que lo ofrece.

Un aspecto fundamental para que se pueda establecer esta comunicación entre servicios web es que todos ellos deben compartir la misma ontología. Podemos definir una ontología como *una especificación formal y explícita de una conceptualización compartida* [12]. Para los lectores menos introducidos en estos temas, puede verse como una descripción de datos común, por lo que todos los usuarios de la misma hacen uso de una sintaxis y semántica conocidas por todas las partes. En la figura 3, un servicio web principal, al que accede un usuario, busca un servicio web que pueda ofrecerle alguno de los elementos que necesita (información, datos, productos, etc.). Todos los servicios contactados hacen uso de una ontología compartida, por lo que el servicio web principal no debe tener problemas a la hora de contactar con ellos. En el momento en que recibe las respuestas de todos los servicios contactados, selecciona cuál de ellos es interesante para su usuario. Haciendo uso nuevamente del ejemplo de la agencia de viajes, el usuario puede contactar con OTA para solicitar billetes de avión para un determinado lugar y fechas. Si OTA no dispone de asientos libres reservados para ese vuelo (esto puede ocurrir en algunos casos), localiza a sus mayoristas y comienza una negociación para localizar los billetes que se ajusten a las necesidades y deseos de su cliente. Una vez que los mayoristas devuelven toda la información, OTA puede utilizar a uno o varios de ellos para proporcionar el billete final al usuario.

Esta solución, haciendo uso de una ontología compartida ha sido adaptada a partir del mecanismo inicialmente propuesto por W3C para la composición de servicios web, tal y como puede ser encontrado en [17]. En todo caso, hay dos inconvenientes no despreciables en este proceso:

1. La asunción de que una ontología sea compartida por todos los servcios web no es realista. Incluso en el caso de que todas las entidades manejen un conjunto de conceptos y relaciones sobre su dominio de trabajo idénticas, cada una de las entidades la representará, probablemente, de una manera diferente. Por ejemplo, un proveedor de servicio puede hablar de "viaje exótico", mientras que otros pueden hacer uso de "viaje de aventuras" o "viaje de alto riesgo" para expresar la misma idea[1].

2. El propio mecanismo carace de funcionalidades importantes. Tal y como se detalla en [6], las entidades que colaboran en el modelo de servicios web no saben nada acerca de

[1] Estos nombres diferentes para ideas similares pueden ser debidos a motivos culturales, de marketing, etc.

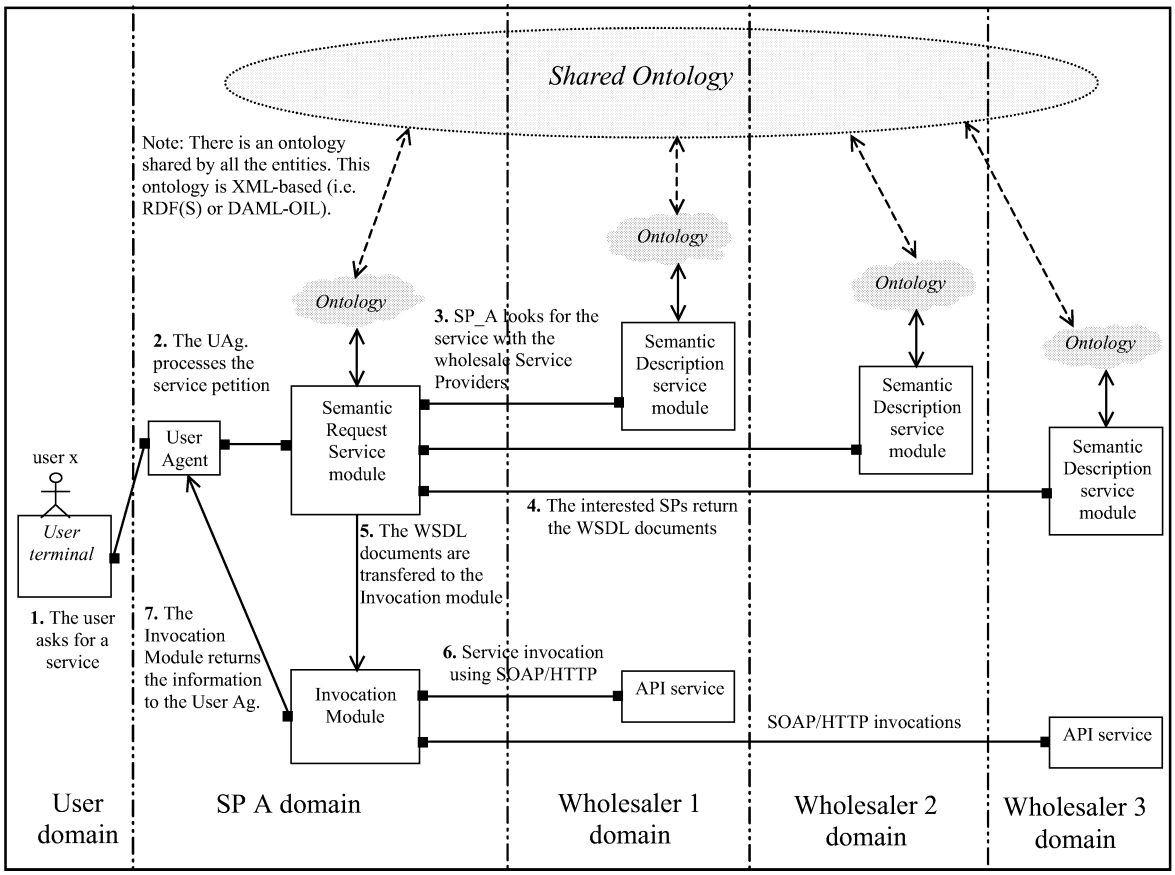


Figura 3: Diferentes servicios web con ontología compartida

los deseos o gustos del usuario, por lo que no pueden actuar de una manera proactiva.

2.3. Agentes y agentes móviles

Los sistemas basados en agentes [16] facilitan la creación de una arquitectura software enormemente distribuida, con altas capacidades para comunicación y negociación entre todos los componentes. Comenzando por una descripción informal de lo que es un agente, podemos decir que se trata de un programa que recibe información de su entorno, que es capaz de tomar decisiones, de actuar de manera proactiva, y que es capaz de comunicarse con otros agentes, bien para solicitarles la realización de alguna tarea que él no es capaz de llevar a acabo, bien para hacer él mismo ese trabajo para otro u otros agentes. Para poder ser considerados como tales, los agentes deben cumplir una serie de características básicas [7]:

- Autonomía: Los agentes deben ser capaces de resolver tareas sin necesidad de interacción con otros agentes o con humanos, y deben tener cierto control sobre sus propias acciones y sobre su estado.
- Sociabilidad: deben ser capaces de interactuar, cuando lo consideren necesario, con otros agentes o con humanos, para completar sus objetivos o ayudar a otros agentes en los suyos.
- Reactividad: deben ser capaces de percibir su entorno y responder en tiempo adecuado a los cambios que en él se produzcan.
- Proactividad: los agentes no deben actuar simplemente en respuesta a cambios en el entorno; deben ser capaces de mostrar un comportamiento dirigido por sus objetivos, y tomar la iniciativa cuando sea necesario.

Los agentes móviles son un tipo especial de agentes. El modelo de agentes móviles ha sido el resultado de la convergencia del campo del procesamiento remoto (RPC, migración de procesos, objetos móviles) y del campo de los sistemas multiagentes (módulos independientes, fuentes de conocimiento, actores, agentes). Continuando con las definiciones informales, podemos decir que un agente móvil es una entidad software que puede desplazarse a otra máquina y continuar allí su ejecución, y que actúa en representación del usuario o de otra entidad.

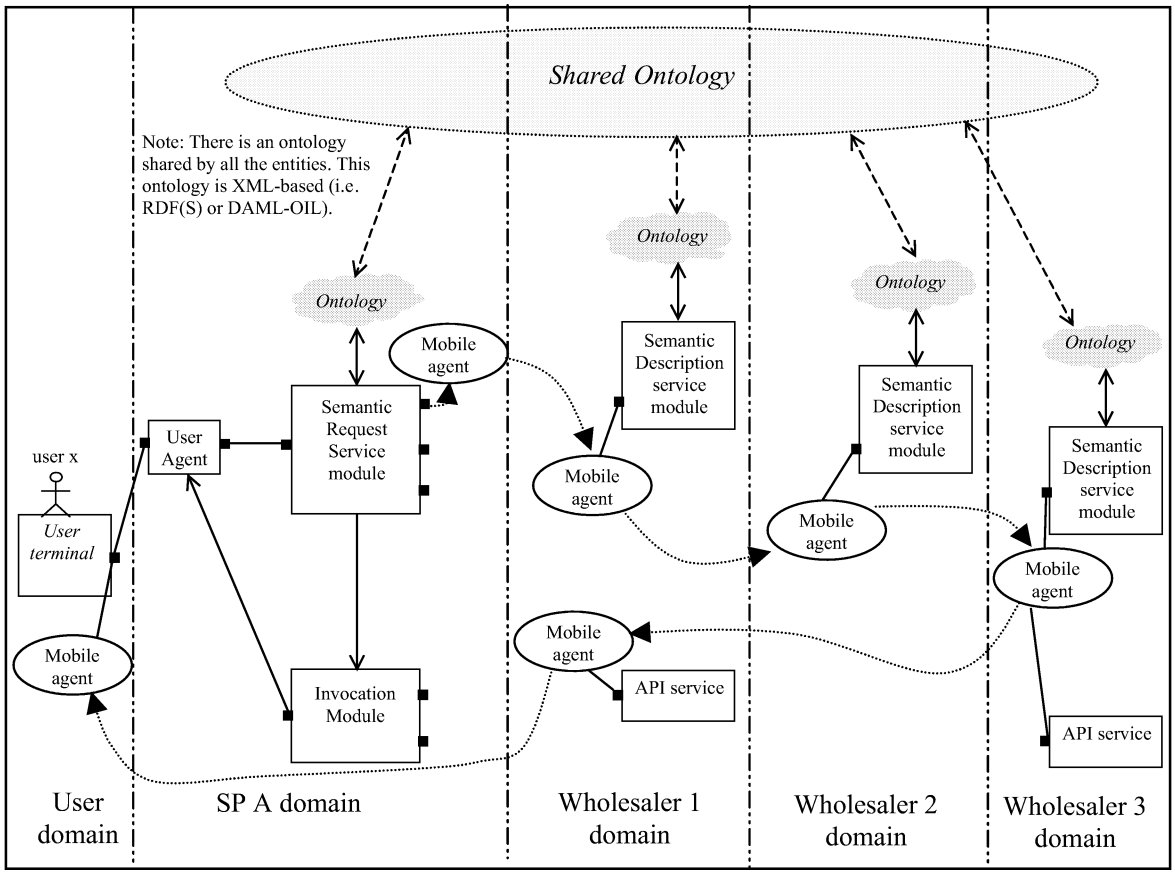


Figura 4: Agentes móviles contactando diferentes servicios web

Entre las principales ventajas del modelo de los agentes móviles destacan las siguientes:

- Uso más eficiente del ancho de banda disponible.
- Soporte para dispositivos móviles (con recursos limitados de computación y de comunicación).
- Facilidad para distribuir servicios a múltiples clientes o máquinas (aspecto que permite a los agentes móviles ofrecer servicios tales como los necesitados por las redes inteligentes).
- Escalabilidad.
- Interacción remota robusta.

3. Servicios web y agentes móviles

Tal y como ya ha sido indicado, los agentes móviles pueden jugar dos papeles relevantes en el proceso de composición de servicios web. Por un lado, gracias a su movilidad, los agentes pueden navegar por la red contactando con los diferentes servicios con los que se pueden componer, actuando como un representante del proceso o servicio principal. Esta situación se muestra en la figura 4. En este caso, el servicio web principal (en nuestro ejemplo, OTA), lanza un agente móvil a la red para negociar con otros servicios web. Una vez que el agente contacta con todos los servicios que pueden ser útiles a su usuario (el servicio principal o el propio usuario de ese servicio), comienza una fase de negociación con aquellos que sean realmente interesantes, moviéndose finalmente al terminal del usuario con la información o el producto. En el ejemplo de la figura 4, haciendo uso de su capacidad para tomar decisiones de manera autónoma, una vez que ha hablado con todos los servicios web, decide negociar sólo con el tercero (justo donde está en ese momento) y con el primero, moviéndose finalmente hasta el usuario con la información del viaje solicitado.

Esta sitiución es diferente de la que se muestra en la figura 3. En aquel caso, el servicio web principal era el encargado de realizar todas las comunicaciones tomar la decisión de qué servicios son intersantes, volver a contactar con ellos y, finalmente, enviar la información al usuario. Todo

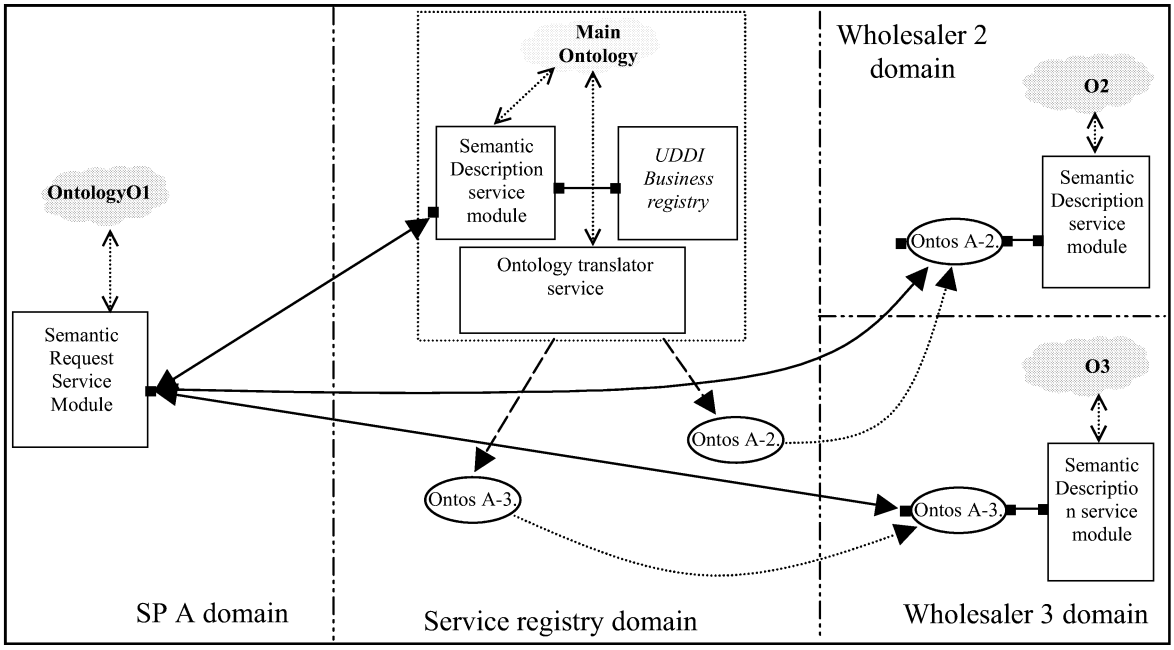


Figura 5: Agente *Ontos*

ello supone una sobrecarga para el SP_A (OTA) que ahora asume el agente en cada uno de los servidores

Con la propuesta de uso de agentes móviles, los beneficios que se obtienen por la composición de servicios se mantienen, y se añaden algunas ventajas que merece la pena destacar:

1. SP_A, (el proveedor de servicios A), una vez que comienza la negociación, tras crear el agente móvil y enviarlo a los diferentes mayoristas, queda liberado para atender nuevas peticiones de usuarios, redirigiendo la carga de la negociación a los servidores de los servicios
2. Todas las interacciones se realizan en modo local, por lo que datos que puedan ser importantes no viajan por la red de manera incontrolada. Si algunos datos no van a ser utilizados por el servicio principal (por ejemplo porque se trate de una oferta más cara que las ya conocidas), incluso puede ser desestimada directamente y no almacenada en el agente
3. Es incluso posible que el agente sea creado por el terminal de usuario (en la figura lo crea el proveedor de servicios, pero no se trata de la única opción). En ese caso, las preferencias del usuario pueden viajar junto con el agente, y posibilitar que la negociación sea más adaptada a los deseos reales del usuario.

Por supuesto, hay algunas desventajas en todo este proceso. Sin duda, la seguridad es la más importante [8]. Cuando utilizamos agentes móviles hay dos problemas diferentes relacionados con la seguridad. Primero, un problema ya conocido, derivado del hecho de que un determinado código, posiblemente no controlado, va a ejecutarse en una plataforma remota: la plataforma debe asegurarse de que ese código no le va a causar problemas, ni va a tratar de atacar a su sistema en modo alguno. Este problema se puede resolver de la manera más convencional (similar a la descarga de software que podemos hacer a través de la red), por medio de firmas, certificados, ejecución controlada en un entorno acotado (sandboxes), etc.

La segunda situación sí es novedosa, y consiste en prevenir a los propios agentes de la actuación de posibles plataformas maliciosas. En otras palabras: el sistema que se implemente (realizado por diferentes fabricantes o desarrolladores), tiene que garantizar que las plataformas de agentes no van a actuar sobre el código o los datos de los agentes, corrompiendo, falseando o alterándolos de manera que ese servidor pueda obtener algún beneficio frente a otros (o simplemente por el *placer* de atacar un software ajeno). Esta es una línea de investigación en la que se avanza a buen ritmo, pero no está dicha la última palabra (como ejemplo puede verse el trabajo desarrollado en [9]). Las ideas más utilizadas se centran en el cifrado/firma de todos los datos que se van almacenando en el agente, tranto por el propio agente como por la plataforma que los ofrece. De esta manera, cuando el agente móvil llega a un nuevo servidor, salvaguarda la información previa de un mal uso (no

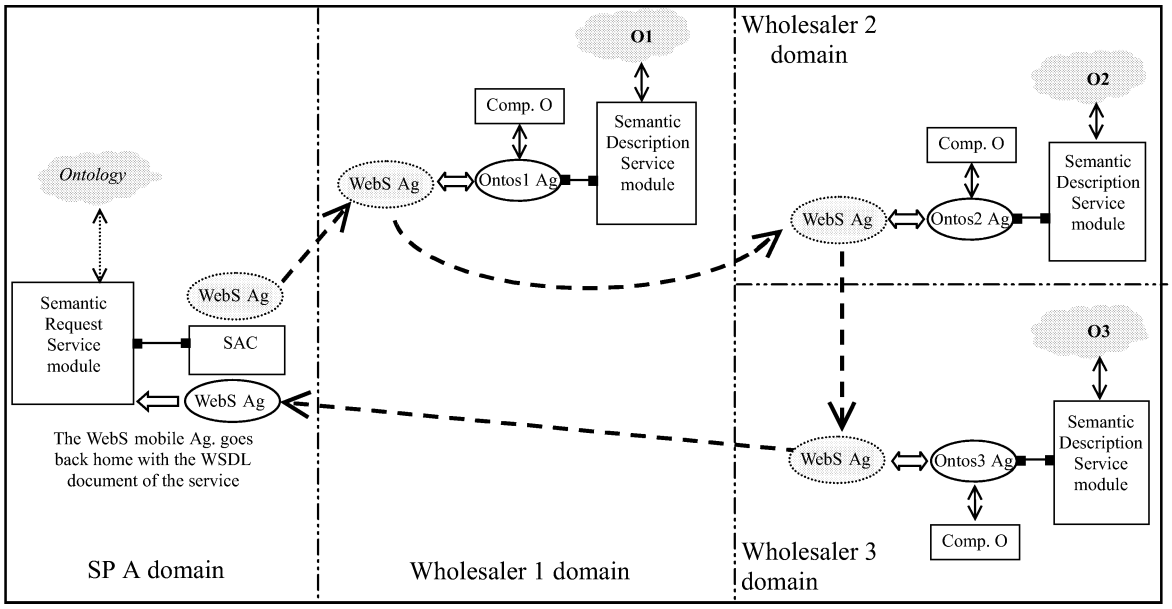


Figura 6: Agente *Ontos* y un agente móvil buscando información o productos

puede ser leida ni alterada). En el caso de que el servidor al que llega el agente lo destruya, reemplazándolo por uno nuevo con la información que la plataforma quiera hacer llegar al servicio principal o al usuario (por ejemplo, sus propias tarifas como las únicas y mejores encontradas), cuando el agente movil llega a su destino, puede ser descubierto, puesto que los datos originales no han sido firmados por el creador del agente móvil.

El segundo uso principal de los agentes móviles (compatible con el primero que acabamos de ver), es la posibilidad de actuar como traductores de ontologías. Este caso se muestra en la figura 5, en la que se representan diferentes sistemas, cada uno de los cuales hace uso de una ontología diferente, relacionándose.

Haciendo uso del ejemplo que estamos manejando, SP_A puede hablar de "viajes exóticos" mientras que el mayorista 2 utiliza "viajes de aventura" y el mayorista 3 "viajes de alto riesgo". En este caso, tras un primer contacto entre SP_A y el registro de servicios, se crean dos agentes *Ontos* nuevos, cada uno de los cuales realizará la traducción entre dos ontologías (A-2 y A-3). Estos agentes se mueven a los servidores donde se albergan los servicios web y esperan hasta que SP_A realize la comunicación. La ventaja principal de esta idea es que cada uno de los servicios web puede manejar su propia ontología (relacionada con una ontología principal, pero adaptada a sus necesidadaes en cualquier caso), y no es necesario que todos los servicios web conozcan cómo traducir entre todas estas ontologías. Este servicio de traducción se realiza de manera distribuida a medida que va siendo necesario.

Cuando los dos usos de los agentes móviles se combinan, el resultado se muestra en la figura 6. En este caso, el agente móvil viaja desde SP_A a los diferentes mayoristas, siendo capaz de comunicarse con ellos gracias al agente *Ontos* que se encuentra allí (activado cuando se han localizado los servicios), que se encarga de traducir los conceptos necesarios para que ambos sistemas puedan entenderse corectamente.

Las características principales del uso de agentes móviles para la composición de servicios que se propone en este trabajo son:

- Cada entidad puede hacer uso de una ontología diferente (O1, O2, O3 en la figura 6).
- Para poder realizar una comunicación eficaz entre servicios que manejan diferentes ontologías, se utiliza un agente mediador *Ontos*.
- Dado que el propio *Ontos* es un agente móvil, aparece en la plataforma del servicio web cuando es necesario, extendiendo entonces los servicios web por medio de un mayor nivel de interoperabilidad y cooperación.
- El solicitante del servicio compuesto (en nuestro ejemplo, el proveedor de servcios A, o OTA) utiliza un agente movil (que aparece en las figuras como *WebS*. Este agente se encarga de llevar a cabo la composición de los servicios interaccionando a través del agente *Ontos* con el servicio web. Esta interacción, realizada en el dominio del servicio web, descarga de trabajo al solicitante del servicio, permitiéndole atender a otros usuarios o realizar otras actividades.

4. Conclusiones y líneas de trabajo

Los agentes móviles pueden ser utilizados en el dominio de los servicios web para proporcionar un buen mecanismo para componer servicios. Por otro lado, el servicio principal puede lanzar los agentes móviles a la red en busca de servicios con los que cooperar, tratando de localizar mejor información o productos para su usuario. Al mismo tiempo, no es necesario que todos los servicios web hayan sido diseñados haciendo uso de la misma ontología exactamente, sino que pueden implementarse bajo ontologías relacionadas entre sí pero que soporten diferencias culturales o de criterios de mercado.

Por supuesto, queda mucho trabajo por hacer en este campo. En primer lugar la seguridad es un aspecto fundamental, y como hemos visto no todo está dicho en este tema. Es necesario establecer mecanismos fiables y ligeros que permitan garantizar que las plataformas a las que llegan los agentes móviles no van a alterar ni su código ni su contenido. Un segundo aspecto de trabajo en el que es necesario ahondar son las preferencias del usuario. Una buena representación de estas preferencias, pueden permitir que sea el propio terminal del usuario el que genere los agentes y éstos puedan realizar una negociación mucho más adaptada a los deseos reales del usuario, manejando aspectos que no sean exclusivos del ámbito del servicio.

Referencias

[1] Bos, L., Ciminiera, L., De Blieck, E., Sisto, R., Exploiting smart cards and mobile agents for personalised service provisioning: A case study en Proc. CLIMATE Workshop on Advanced Services in Fixed and Mobile Telecommunications Networks, Singapore, September 1999.

[2] Castillo S., Velasco J.R.: "Service Composition in m-commerce using the Mobile Agents approach", ICECR-5 5th International Conference on Electronic Commerce Research, Montreal, Canada, October 23-27, 2002.

[3] Castillo S.: "Composición de servicios mediante el modelo de los agentes móviles" (Tesis Doctoral), E.T.S.I.Telecomunicación, Universidad Politécnica de Madrid, 2002.

[4] Gomez-Pérez, A. y Corcho, O.: "Onlology Languages for the Semantic Web", IEEE Intelligent Systems Jan-Feb, 2002, pp.54-60.

[5] Graham S. et. al.: "Building Web Services with Java: Making Sense of XML, SOAP, WSDL and UDDI", Sams Publishing, USA, 2001.

[6] Huhns M.N.: "Agents as Web Services", IEEE Internet Computing, july/ag. 2002.

[7] Jennnings, N.R. and Wooldridge, M., "Software Agents", en IEE Review, January 1996, pp 17-20.

[8] Man M.C., Wei V.K.: "A Taxonomy for Attacks on Mobile Agent", IEEE EUROCON'2001 Trends in Communications, International Conference on, p. 385-388, 2001.

[9] C. Ruland, O. Weissmann, A. Friesen and N. Oikonomidis. "A distributed Certificate and Profile Management System for use in Agent Systems". 2nd international ACTS workshop: "Advanced Services in Fixed and Mobile Telecommunikations Networks" (Singapur, September, 1999)

[10] "SCARAB consortium: The SCARAB Software Architecture". Documento semipúblico disponible parcialmente a través de los autores (1999).

[11] SOAP (Simple Object Access Protocol): http://www.w3.org/TR/SOAP

[12] Rudi Studer, V. Richard Benjamins, Dieter Fensel. "Knowledge Engineering: Principles and Methods". In Data and Knowledge Engineering, 25, pp. 161-197, 1998.

[13] TINA consortium: "Overall Concepts and Principles of TINA", version 1.0, feb. 1995.

[14] TINA consortium: Telecommunications Information Networking Architecture- Service Architecture, version 5.0 (1996).

[15] UDDI (Universal Description, Discovery and Integration): http://www.uddi.org

[16] Wooldridge, M. and Jennnings, N.R. (eds). Proceedings of the 1994 workshop on Agent Theories, Architectures and Languages (ATAL 94), The Netherlands, August, 1994.

[17] World Wide Web Consortium: "Web service use case: Travel reservation - Use case 5 May 2002", (http://www.w3.org/2002/06/ws-example), 2002.

[18] WSDL (Web Services Description Languages): http://www.w3.org/TR/wsdl

Sesión 4A

Gestión de la calidad de servicio

Control de acceso al medio con objetivos de calidad de servicio para esquemas de transmisión con modulación adaptativa

G. Gómez, M. C. Aguayo-Torres, J. T. Entrambasaguas
Departamento de Ingeniería de Comunicaciones
Universidad de Málaga
Campus de Teatinos s/n, Málaga (España)
E-mail: {gerardo, aguayo, jtem}@ic.uma.es

Abstract- *In this paper, the performance of an Adaptive Quadrature Amplitude Modulation (AQAM) system in a fast fading downlink channel using Time Division Multiple Access (TDMA) is studied. The results of different scheduling algorithms over shared channels are analysed, where in particular, delay, packet loss, system efficiency and throughput performance are studied. Results confirm how the knowledge of the channel state information (CSI) as well as other system state information by the resource manager improves the system efficiency and the Quality of Service (QoS) experienced by the different flows.*

1 Introducción

En los últimos años se ha incrementado de forma notable la demanda de intercambio de información mediante tecnologías inalámbricas. Esto, a su vez, ha empujado el progreso de los sistemas de comunicaciones sin hilos a un paso sin precedentes. Puesto que los sistemas inalámbricos tienen un ancho de banda disponible escaso, los limitados recursos deben usarse de forma eficiente para proporcionar servicios satisfactorios a un número creciente de usuarios.

El esfuerzo en la mejora de las prestaciones de los sistemas de comunicaciones móviles resulta especialmente arduo pues deben enfrentarse a un ambiente particularmente hostil [1]: el movimiento tanto de los extremos de la comunicación como de su entorno lleva a que, incluso a lo largo de una comunicación, las condiciones del canal puedan cambiar de forma apreciable. El resultado es que la capacidad del canal para transmitir información es variable. A pesar de ello, la mayor parte de los sistemas de comunicaciones móviles actuales sigue transmitiendo con una tasa binaria constante y consiguen la recuperación de los errores producidos en los desvanecimientos mediante codificación de canal o retransmisiones.

Una interesante posibilidad para mejorar el rendimiento de la transmisión consiste en permitir al receptor vigilar las condiciones del canal y solicitar, durante la comunicación, cambios en la señal transmitida adaptados a las condiciones instantáneas del enlace [2]. Estas tecnologías, que han venido a denominarse modulación adaptativa, incluyen cualquier técnica que vaya modificando algún parámetro de la señal transmitida (potencia, período de símbolo, esquema de modulación utilizado, razón de codificación o una combinación de esos parámetros) en función de las características instantáneas del canal. La idea subyacente es modificar la señal transmitida de manera que la razón entre la energía de símbolo y la potencia de ruido (e interferencia) sea fija, transmitiendo más deprisa cuando las condiciones del canal lo permiten, y reduciendo la velocidad de los datos a medida que el canal los degrada. E l resultado es un sistema de transmisión binaria variable con una capacidad media mejor que los sistemas fijos de igual tasa de error (*Bit Error Rate*, BER).

Por otra parte, los futuros sistemas de comunicaciones móviles ofrecerán a los usuarios la posibilidad de intercambiar información de naturaleza muy distinta con requerimientos muy diferentes de calidad de servicio (*Quality of Service*, QoS). El logro de los objetivos de la QoS y el aprovechamiento de los recursos radio requieren mecanismos eficientes de acceso al medio (*Medium Access Control*, MAC).

Es posible mejorar el rendimiento del MAC en sistemas inalámbricos de velocidad binaria constante mediante algoritmos adaptativos que reciben información adicional para poder adecuarse dinámicamente a las condiciones del sistema y del canal [3]. Esta información adicional puede ser la relación señal a ruido instantánea de los enlaces con los usuarios o el estado de ocupación de las colas.

Estas mismas ideas sobre el acceso al medio pueden aplicarse al caso del esquema de velocidad variable con modulación adaptativa. Algunos estudios presentados en la bibliografía introducen una capa de acceso al medio para gestionar los recursos radio de acuerdo a algunas necesidades de QoS [4]-[7]. Así, en [8][9] se combinan técnicas de solicitud automática de repetición con un algoritmo de planificación que tiene en cuenta el estado del canal. Sin embargo, estos estudios se centran en el comportamiento del sistema desde el punto de vista de tasa de errores y eficiencia, sin tener en cuenta otros requisitos, como los de retardo, que según el caso pueden llegar a inutilizar el servicio prestado.

En esta comunicación se estudia el comportamiento de algunos algoritmos de acceso al medio con objetivos de calidad de servicio en un sistema que emplea un esquema de modulación adaptativa para la

transmisión. El escenario considerado está compuesto por varios equipos móviles que comparten los recursos servidos por una estación base, como presenta la Fig. 1. El sistema trabaja a una única frecuencia y ofrece canales multiplexados mediante división en el tiempo (*Time Division Multiple Access*, TDMA) sobre cada uno de los cuales, a su vez, pueden multiplexarse flujos de paquetes de diferentes usuarios. En la transmisión hacia cada usuario, la estación base emplea un esquema de modulación adaptado a las condiciones instantáneas del enlace que la une con el terminal móvil. El uso compartido del conjunto de canales por todos los usuarios se controla con un planificador de paquetes (*scheduler*).

El resto de esta comunicación se organiza como sigue. En primer lugar, el sistema en estudio es descrito en la sección 2. Posteriormente se describe la capa de acceso al medio y se presentan los algoritmos de planificación estudiados. Los resultados de las simulaciones se muestran en la sección 4. Finalmente, las conclusiones de la sección 5 resumen las principales ideas presentadas aquí.

2 Descripción del sistema

El modelo del sistema se muestra en la Fig. 2. En primer término, el sistema presenta N colas (una por usuario) donde se almacenan N flujos de paquetes de posiblemente diferente naturaleza (en términos de velocidad binaria, BER objetivo, prioridad, etc.) mientras esperan a ser reexpedidos hacia el enlace radio. Aunque la llegada de bits a las colas se considera por paquetes (de tamaño medio definido por los modelos de tráfico), son extraídos en grupos de cualquier número entero de elementos.

El planificador se encarga de asignar los turnos de transmisión a los distintos flujos a frecuencia de ranura temporal (*time-slot*, TSL). El comportamiento del planificador puede variar dependiendo de la información que se tenga en cuenta en la asignación. En la sección siguiente se describirán tanto los parámetros de entrada al planificador como los algoritmos propuestos.

En cada ranura temporal se transmiten M símbolos dirigidos al usuario k al que el planificador asignó el turno. El transmisor emplea un esquema de modulación adecuado a las condiciones instantáneas del canal radio k. Los N canales sufren desvanecimientos planos independientes entre sí. En

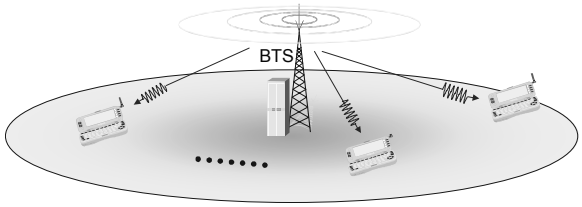


Figura 1: Escenario considerado en este trabajo

este estudio se ha considerado que siguen una distribución de tipo Rayleigh.

El esquema de modulación adaptativa empleado modifica el número de bits de información por símbolo (es decir, el nivel de modulación) en función de la relación señal a ruido (*Signal to Noise Ratio*, SNR) instantánea medida por el receptor. Un ejemplo de cómo funciona este algoritmo de modulación QAM adaptativa (*Adaptive QAM*, AQAM) se presenta en la Fig. 3, donde se presenta el nivel de modulación asociado a la SNR de dos usuarios en movimiento. Se usa una constelación más o menos robusta dependiendo de la SNR instantánea para mantener la tasa de error por debajo de una definida como objetivo, BER_T. Valores más estrictos de tasa de error objetivo obligan a superar umbrales más altos de SNR para aumentar el tamaño de la constelación.

Es interesante señalar que este esquema AQAM produce una velocidad binaria variable ya que, aunque el período de símbolo en la transmisión es constante, el número de bits entrantes al transmisor para transmitir un símbolo es variable.

3 Control de acceso al medio

El planificador está encargado de decidir qué flujo debe transmitir en cada turno (*slot*) al mismo tiempo que persigue cumplir los requerimientos de QoS para cada uno de ellos. Se supone aquí que en el proceso de iniciación de cada flujo se entrega al sistema la información necesaria de QoS que permite tratar a cada uno de forma diferenciada.

En este estudio se han empleado dos parámetros de calidad: la tasa de error objetivo y la prioridad de los flujos. El primero de ellos permite determinar los umbrales para seleccionar la constelación empleada en función de la relación señal a ruido instantánea en

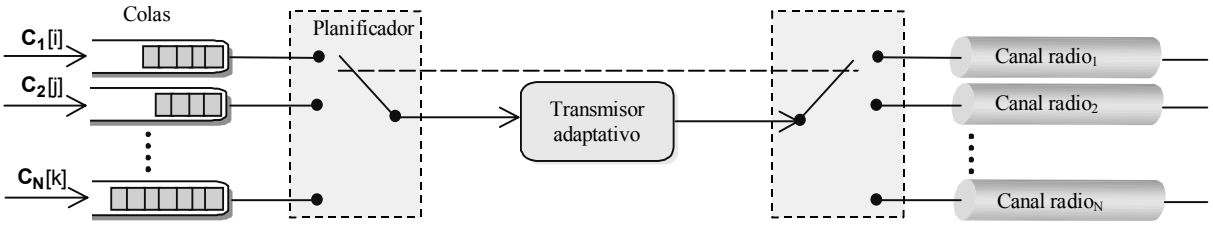


Figura 2: Modelo del sistema propuesto

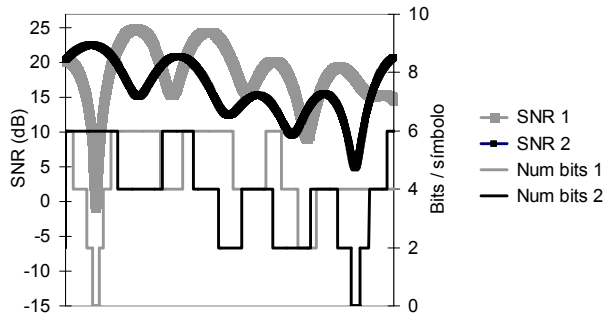


Figura 3 Ejemplo de procedimiento de modulación adaptativa AQAM

la forma que se comentó en la sección anterior.

La prioridad de los flujos determina el orden en que se sirven los distintos flujos: paquetes con mayor prioridad son servidos antes que los de prioridad menor. La consecuencia inmediata es una diferenciación del retardo sufrido por los flujos. Aunque sería posible un control más estricto del retardo, el uso de un parámetro de prioridad permite la escalabilidad del sistema.

Se han simulado cuatro algoritmos de planificación que podrían ser clasificados en dos grupos: algoritmos fijos (con parámetros predefinidos) y algoritmos adaptativos (basados en información del canal y del sistema).

3.1 Algoritmos de planificación fijos

Los algoritmos de planificación fijos quedan definidos por medio de parámetros determinados al comienzo de la comunicación a los que se atienen de forma estricta, sin tener conocimiento de las condiciones del canal o del estado del sistema. En cualquier caso se supondrá que existe una cola diferente para cada flujo.

En este estudio se han implementado dos algoritmos fijos, para los que la Fig. 4 muestra el intercambio de información entre bloques del sistema:

- Algoritmo cíclico sin prioridades (*Round Robin*). Este algoritmo rota los turnos de transmisión entre las colas no vacías sin ninguna información adicional. Como consecuencia, distribuye el ancho de banda por igual entre los distintos flujos sin tomar en consideración sus necesidades o las condiciones del canal radio. Su simplicidad es esperable que se pague con una eficiencia baja en términos de velocidad neta de transmisión o diferenciación de QoS.
- Algoritmo con prioridades fijas. En este caso, los recursos se asignan a aquellos flujos con prioridad mayor y colas no vacías. Entre flujos de igual prioridad, los recursos se distribuyen de forma cíclica (como en el algoritmo anterior). Este algoritmo tiene un impacto directo en el retraso en las colas en el sentido de que flujos con mayor prioridad sufrirán menores retardos.

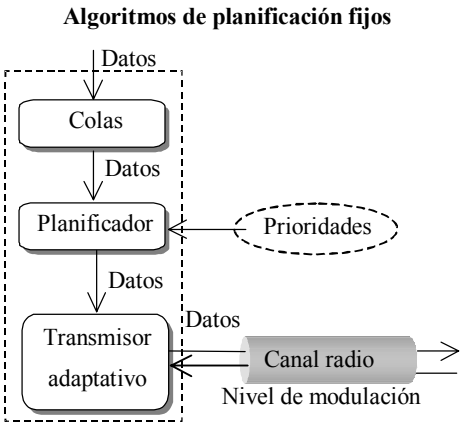


Figura 4: Algoritmos de planificación fijos

Sin embargo, tampoco se espera que sea un método eficiente ya que no toma en cuenta las condiciones del enlace radio.

3.2 Algoritmos adaptativos de planificación

Los algoritmos adaptativos reciben información adicional que permite al sistema adecuarse dinámicamente a las condiciones variables tanto del propio sistema como del canal. Tal información podría ser, por ejemplo, la relación señal a ruido instantánea de los N enlaces o la ocupación de la cola a lo largo del tiempo.

El intercambio de información entre los bloques del sistema para los algoritmos adaptativos se presentan en la Fig. 5. De este grupo de métodos de planificación, se han implementado dos algoritmos.

- Algoritmo basado en el nivel de modulación. Puede considerarse que este esquema emplea para asignar los turnos una discretización de la relación señal a ruido instantánea. Esta estrategia elige el flujo al que asignar el turno entre aquellos a los que el algoritmo de modulación adaptativa les ha asignado mayor número de bits por símbolo (mayor nivel de modulación). De esta forma maximiza el uso del ancho de banda

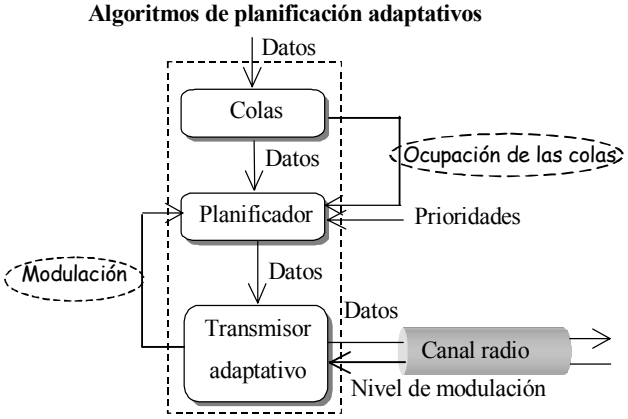


Figura 5: Algoritmos de planificación adaptativos

y, adicionalmente, reduce el desbordamiento de las colas comparado con los algoritmos fijos. Las prioridades son también consideradas en la asignación del turno para permitir elegir entre flujos que usan el mismo nivel de modulación.

- Algoritmo basado en la modulación con control de pérdida de paquetes. Los algoritmos mencionados anteriormente no toman en cuenta la ocupación de las colas para la asignación de los turnos. El último esquema de planificación presentado se basa en el anterior pero define un "umbral de ocupación": cuando la cola de un flujo supera este umbral (por ejemplo, el 80%), precede al resto de flujos de igual nivel de modulación para la asignación del turno, independientemente de su prioridad.

4 Resultados de las simulaciones

Esta sección proporciona los parámetros seleccionados y los resultados de las simulaciones que permiten comparar los diferentes algoritmos de planificación en términos de eficiencia del sistema, retardo, pérdida de paquetes y velocidad de transmisión binaria neta.

Se han considerado tres tipos de tráfico, correspondientes a voz, vídeo y tráfico de datos. La Tabla 1 muestra los parámetros más relevantes de los modelos usados en las simulaciones y la Tabla 2 sus requerimientos de QoS, definidos como tasa de error objetivo (BER_T) y prioridad. Se han realizado simulaciones con 15 usuarios simultáneos (5 usuarios de cada tipo de tráfico).

Se supone que los usuarios se mueven a 50 Km/h alrededor de la estación base, a una distancia tal que produce una SNR media del enlace en un rango de ±2dB alrededor de 20dB. Los símbolos se transmiten a una frecuencia de 2Mbaudios sobre una portadora a 2 GHz.

La constelación transmitida se escoge como la más densa que mantiene la tasa de error por debajo de la objetivo (que depende del tipo de tráfico). Existen cinco posibles regiones de modulación: no-transmisión (NTX), QPSK 16QAM, 64QAM y 256QAM.

Una vez asignado el turno a un flujo, éste transmite 512 símbolos consecutivos con la constelación elegida como la más adecuada dependiendo de la BER objetivo del flujo y la relación señal a ruido instantánea del enlace sobre el que se transmite. El tamaño de las colas es 10 paquetes para todos los tipos de tráfico.

Tabla 1: Modelos de tráfico

Tipo de tráfico	Tamaño de los paquetes	Tiempo entre paquetes	Velocidad
Voz	268 bytes	Modelo ON-OFF Exponencial ON=OFF=33.5 ms	64 kbps (ON)
Vídeo	Distribución lognormal Media: 1000 bytes Desviación 400 bytes	15.625 ms	512 kbps (media)
Datos	1250 bytes	Distribución exponencial Media: 10 ms	1 Mbps (media)

Tabla 2: Requerimientos de QoS

Tipo de tráfico	BER objetivo	Prioridad
Voz	1e-2	1
Vídeo	1e-3	1
Datos	1e-4	2

4.1 Retardo medio en la cola

Esta sección proporciona los resultados de retardo (considerado desde el instante de llegada de un paquete a la cola hasta que es reexpedido hacia el enlace radio) para diferentes algoritmos de planificación. Los resultados de retardo están principalmente influenciados por las prioridades y la velocidad binaria de la fuente. La Fig. 6 muestra los resultados para cada usuario, donde los usuarios 1-5 usan un servicio de voz, 6-10 corresponden a vídeo y 11-15 a tráfico de datos.

Puede observarse que los retardos experimentados por los diferentes flujos son suficientemente bajos para sus correspondientes servicios.

El algoritmo cíclico es el más igualitario y proporciona retardos similares a todos los flujos, aunque aquellos con mayor velocidad binaria (usuarios 6-15) normalmente tengan más bits esperando en la cola, lo que aumenta el retardo.

El algoritmo basado en prioridades maximiza la diferenciación de retardo, es decir, obtiene retardos mínimos para usuarios de alta prioridad (voz y vídeo) y retardos máximos para tráfico de datos, lo que puede conducir a grandes pérdidas de paquetes en las colas de prioridad menor (si no son correctamente dimensionadas).

Los algoritmos adaptativos minimizan el retardo en media para todos los usuarios (asumiendo similar SNR) debido a que el planificador proporciona prioridad a aquellos flujos con mayor nivel de modulación y, por consiguiente, la eficiencia del sistema es mayor. No hay diferencia significativa en términos de retardo entre los dos algoritmos adaptativos.

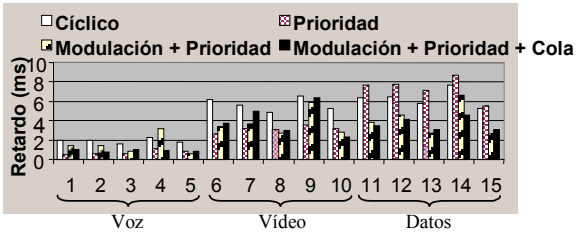


Figura 6. Retardo medio en la cola

4.2 Paquetes perdidos

La Fig. 7 muestra la tasa de paquetes perdidos que se produce debido al desbordamiento de las colas.

Únicamente el tráfico de datos (usuarios 11-15) pierde algunos paquetes debido a su menor prioridad y su mayor velocidad binaria media. Aunque el algoritmo cíclico también produce pérdidas de paquetes, el algoritmo de peor comportamiento es el de prioridades estrictas, para el que las pérdidas de paquetes para el tráfico de datos están por encima del 60%. Esto se debe a que los recursos nunca son asignados a este tipo de tráfico a no ser que no exista tráfico de vídeo o voz esperando para ser enviado, incluso si la cola está llena para el tráfico de datos. Dado que las condiciones del canal no se consideran en la decisión, los recursos podrían ser asignados a un flujo sin posibilidad de transmisión (*outage*).

Ya que los algoritmos adaptativos toman en cuenta la SNR asociada a cada usuario, los resultados de pérdidas de paquetes disminuyen considerablemente debido a la alta eficiencia del sistema, es decir, el transmisor usa en media un número mayor de bits por símbolo.

El algoritmo basado en la modulación con control de pérdida de paquetes elimina las pérdidas ya que da preferencia a aquellas colas cuyo nivel de ocupación está por encima de cierto umbral para evitar los desbordamientos. Para apreciar la mejora de este algoritmo, la Fig. 8 muestra los resultados de los algoritmos adaptativos para un sistema más cargado (tráfico de voz de 640kbps y datos a 1.2Mbps). Este ejemplo ratifica la mejora significativa en términos de pérdida de paquetes cuando la ocupación de la cola es considerada en la asignación de recursos. En la mayoría de los casos los resultados se reducen a la mitad.

En todos los casos sólo se produce pérdida de paquetes apreciable en el tráfico de datos, que requerirá un protocolo de nivel superior con retransmisiones como TCP (*Transmission Control Protocol*). Otra opción sería un dimensionamiento adecuado de las colas, aunque este aspecto tiene un impacto directo en el retardo máximo. Adicionalmente, también se asume que el sistema necesita un control de admisión responsable de la admisión de aquellos flujos que el sistema es capaz de soportar con cierta QoS.

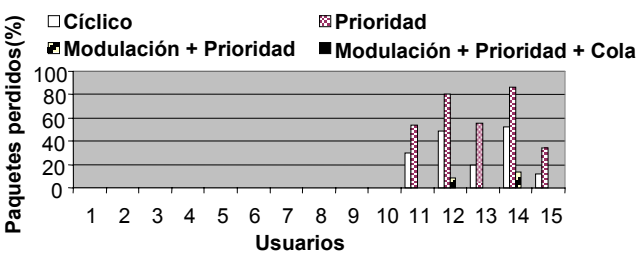


Figura 7: Paquetes perdidos para las condiciones de la Tabla 1

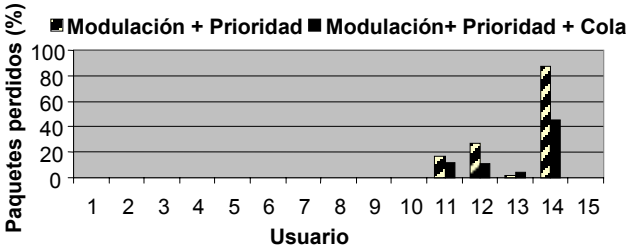


Figura 8: Paquetes perdidos para vídeo a 640kbps y datos a 1.2Mbps

4.3 Eficiencia del sistema

La eficiencia del sistema da una idea de la utilización del canal, es decir, del número medio de bits por símbolo transmitidos por cada usuario considerando sólo sus correspondientes turnos (mayores valores de eficiencia media implican mejor uso del ancho de banda).

Los resultados obtenidos en las simulaciones se presentan en la Fig. 9. Como se esperaba, la eficiencia obtenida con algoritmos adaptativos es siempre mayor que los obtenidos con algoritmos fijos. Además, ambos algoritmos obtienen resultados similares (incluso aunque uno de ellos tenga control de pérdida de paquetes) ya que ambos asignan los recursos a aquellos flujos con mayor nivel de modulación.

La Fig. 10 presenta el promedio de la eficiencia de todos los usuarios. Puede obtenerse una estimación de la capacidad máxima a partir de estos resultados y la frecuencia de símbolo: para algoritmos fijos la capacidad máxima es de unos 7.2Mbps mientras que

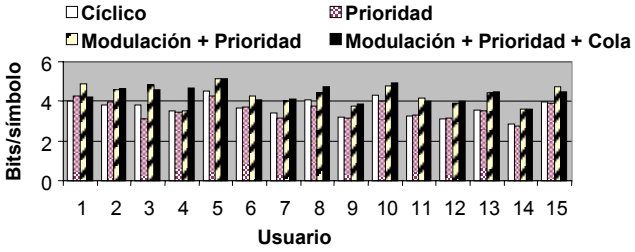


Figura 9: Eficiencia del sistema

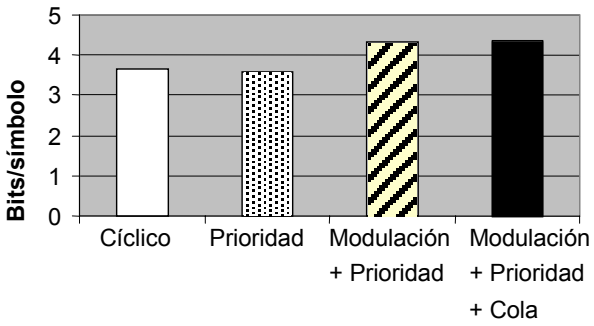


Figura 10. Eficiencia media del sistema

los algoritmos adaptativos obtienen 8.6Mbps de velocidad binaria.

4.4 Velocidad neta

Esta sección muestra los resultados de la velocidad de transmisión neta total transmitida hacia los receptores, considerando las pérdidas en las colas. Los resultados se presentan en la Fig. 11.

La velocidad neta obtenida para tipos similares de tráfico puede variar ya que los modelos de tráfico ofrecen una velocidad binaria variable (*variable BitRate*, VBR) para vídeo y datos. El tráfico de voz puede también variar debido al modelo exponencial usado en la conmutación ON-OFF.

La velocidad binaria experimentada por los usuarios de voz (1-5) y vídeo (6-10) es aproximadamente la velocidad media definida en los modelos de tráfico ya que no hay pérdidas de paquetes para estos flujos. En el caso de tráfico de datos (usuarios 11-15) la velocidad experimentada para los algoritmos fijos se ve muy afectada por las pérdidas de paquetes. Este problema se resuelve con los algoritmos adaptativos, especialmente con el que tiene control de pérdida de paquetes.

5 Conclusiones

Esta comunicación ha abordado el estudio del comportamiento de un sistema basado en modulación adaptativa y una capa de control de acceso al medio responsable de la asignación de recursos para maximizar la eficiencia del sistema mientras pretende cumplir los requerimientos de QoS de los flujos.

El sistema de modulación adaptativa permite cumplir los requerimientos de tasa máxima de error de los flujos mientras que el planificador es el encargado de cubrir las necesidades de retardo e intentar maximizar la utilización del sistema.

Se han simulado varios algoritmos de planificación que se basan la asignación de los turnos de transmisión a partir de varios elementos de información: prioridades de tráfico, nivel de modulación en la transmisión y ocupación de las colas.

La diferenciación de los retardos sufridos por los paquetes es posible dando prioridad a aquellos flujos con necesidades estrictas de retardo (como voz o vídeo) en la asignación de recursos.

El uso de los recursos es más eficiente (aproximadamente un 15%) y, por tanto, el sistema presenta una capacidad mayor, si los algoritmos de acceso al medio son adaptativos y dan precedencia a aquellos flujos cuya velocidad binaria instantánea es mayor. Si, además, el algoritmo adaptativo considera el factor de ocupación de las colas, minimiza las pérdidas de paquetes debido al desbordamiento de las colas mientras además mantiene la ganancia en eficiencia respecto a los algoritmos fijos.

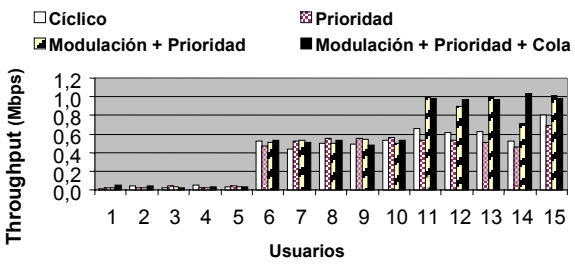


Figura 11. Velocidad binaria neta

Referencias

[1] D. Parsons, "The Mobile Radio Propagation Channel," Petench Press, 1992

[2] S.T. Chung, A.J. Goldsmith "Degrees of freedom in adaptive modulation: A unified view", S.T. Chung, A.J. Goldsmith, vol. 49, nº 9, septiembre 2001, págs. 1561-1571

[3] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, "Providing Quality of Service over a Shared Wireless Link", Communications Magazine, febrero/marzo, 2001

[4] N. Morinaga, M. Nakagawa, R. Kohno, "New Concepts and Technologies for Achieving Highly Reliable and High Capacity Multimedia Wireless Communications Systems", IEEE Communications Magazine, 1997

[5] N. Casimiro, "Scheduling and Adaptive Transmission for the Downlink in 4G Systems", Future Telecommunication Conference Beijing, China, noviembre 2001

[6] V. K. N. Lau, Y. K. Kwok, "Joint design of adaptive channel coding and cell scheduling for wireless ATM", Electronics Letters , marzo 2000

[7] N. Casimiro, "Adaptive Modulation and Scheduling for Fading Channels", Global Telecom. Conference, GLOBECOM '99

[8] S. Falahati, A. Svensson, "Hybrid type-II ARQ/AMS and Scheduling using Channel Prediction for Downlink Packet Transmission on Fading Channels", Proceedings PCC Workshop, Nynäshamn, Suecia, abril 2001

[9] T. Ikeda, S. Sampei, N. Morinaga, "The performance of adaptive modulation with dynamic channel assignment in multimedia traffic", IEEE Universal Personal Communications, 1998

QoS en redes Wireless LAN IEEE-802.11

Anna Calveras, Alex Berdonces
Departamento de Ingeniería Telemática. Universidat Politécnica de Cataluña.
Campus Norte, Modulo C3
08034 Barcelona
Teléfono 93 401 60 13 Fax 93 401 59 81
E-mail: anna.calveras@entel.upc.es

Abstract *– Mobile communications are growing rapidly, joint to Internet and other new technologies, that provide network quality of service (QoS). Therefore, the QoS concept is also going to be considered in mobile communications. In this paper we present how to achieve QoS on 802.11 networks. Having QoS at the IP level is not effective without the modification of the 802.11 MAC level. So, we describe different proposals related to that. Finally, we present one proposal to have better performance on presentaremos una propuesta para mejorar las comunicaciones TCP/IP en este tipo de entornos.*

1 Introducción

Actualmente el mundo de las comunicaciones móviles y el de Internet están sufriendo un crecimiento espectacular. Por tanto no es de extrañar que se estén integrando los dos para ofrecer al usuario final las ventajas de ambos mundos. La tecnología "wireless" está teniendo mucho empuje en el mercado. Cada vez es más común ver todo tipo de terminales que acceden a servicios de Internet de forma inalámbrica tales como *smart phones* y terminales GPRS (General Packet Radio Service) u ordenadores portátiles y PDA's mediante WLAN (Wireless Local Area Netwok) o Bluetooth.

Por otro lado el mundo de Internet ya no se centra únicamente en la World Wide Web, sino que continuamente aparecen nuevas aplicaciones como la telefonía IP (Internet Protocol), videoconferencias, vídeo bajo demanda, etc. Por eso está en auge el mundo de la calidad de servicio en las redes IP, ya que las nuevas aplicaciones tienen restricciones estrictas en cuanto a retardos y ancho de banda. Es por este motivo que han surgido métodos para ofrecer QoS (Quality of Service) en IP como IntServ [11] y DiffServ [12].

Es lógico pensar pues, que haya un creciente interés en reunir estas tecnologías dentro de los estándares para redes inalámbricas. Si nos centramos en el IEEE 802.11 [2], que es el más comercializado, nos damos cuenta que ofrecer QoS a nivel IP [1] (o superior) no es posible sin hacer modificaciones en las técnicas de acceso al medio. Se ha comprobado que, en las redes inalámbricas, los servicios diferenciados de IP son subóptimos sin el soporte de las capas inferiores [8]. Por tanto el Task Group E (TGe) del IEEE 802.11 centra sus esfuerzos en modificaciones de la capa MAC (Medium Acces Control) que permitan proporcionar un cierto soporte para QoS a los niveles superiores, pero que a la vez no creen incompatibilidades con lo desarrollado hasta el momento. En este artículo se prsentan las diferentes soluciones para ofrecer QoS en el nivel MAC 802.11 así como un nuevo esquema para redes inalámbricas. La idea es dotar a los reconocimientos del protocolo de transporte de prioridad sobre el resto de tráfico, como ocurre con los de nivel MAC.

2 Wireless LAN IEEE 802.11

En 1990 se formó el grupo de trabajo del IEEE 802.11 para la estandarización de las redes inalámbricas de área local. El 802.11 llevó a cabo un estándar global que inicialmente tenia tres especificaciones de nivel físico: una de infrarrojos y otras dos RF (Radio Frecuencia) que trabajan en la banda de los 2,4 GHz, no requieren licencias, y funcionan a una velocidad de 1 o 2 Mbps. Actualmente tenemos velocidades mayores: con el 802.11b se llega a los 11 Mbps y con el 802.11a se consiguen velocidades de transmisión de 36 Mbps.

2.1 Capa MAC del 802.11

La capa MAC [2] la encontramos sobre el nivel físico de las WLANs. Proporciona dos tipos básicos de funcionamiento: DCF (Distributed Coordination Function) y PCF (Point Coordination Function).

El DCF es de implementación obligada y es el que se utiliza en redes *adhoc*. Sobre éste se implementa el PCF que permite el acceso mediante infraestructura y proporciona un intervalo libre de contenciones. El PCF es opcional.

DCF (*Distributed Coordination Function*): DCF es el mecanismo básico de acceso al medio del 802.11. En este modo de funcionamiento cada estación puede conectarse con otras que estén visibles y en su proximidad sin que haya un control centralizado (redes *adhoc*). El CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) en el que se basa consiste en escuchar el canal antes de la transmisión. Si el canal está libre durante un tiempo DIFS (DCF InterFrame Spacing) entonces se procede a transmitir los datos. Si el canal está ocupado se espera hasta que

quede libre y después de DIFS se inicia un mecanismo de backoff aleatorio. El temporizador de backoff va descontándose mientras el canal esté vacío y se detiene cuando haya otra estación transmitiendo. En este caso, cuando el canal vuelva a estar libre, después de esperar DIFS de nuevo continuará la cuenta atrás donde se había quedado. Cuando el contador expira se pueden enviar los datos. El valor del tiempo de espera de backoff se escoge dentro de la ventana de contención (Contention Window). Este valor estará entre [0,CW]. Inicialmente CW es 31 por defecto en el 802.11b. Cada trama tiene que ser validada mediante la transmisión de un ACK (Aknowledgement) por parte del receptor. Los paquetes de ACK esperan un tiempo SIFS (Short InterFrame Spacing) para acceder al canal. SIFS siempre es menor que DIFS, de esta manera los ACK tienen prioridad frente a nuevas transmisiones. Como el canal wireless es *half duplex* (debido a la diferencia entre las potencias de la señal recibida y la transmitida) no podemos detectar colisiones. Por tanto consideramos que se ha producido una colisión cuando no se recibe el ACK del receptor. En este caso se dobla el valor de CW y se vuelve a intentar la retransmisión de forma que el nuevo backoff que se escoge tiene más posibilidades de ser diferente al de la otra estación con la que se ha colisionado. Con las sucesivas colisiones se va aumentando CW y con las retransmisiones correctas se disminuye. Tenemos pues un backoff aleatorio exponencial gracias al cual se minimizan el número de colisiones entre estaciones que previamente hayan colisionado.

Detección de portadora virtual: En sistemas de acceso por radio puede darse el caso de tener estaciones que no se escuchen entre ellas e interpreten que el canal está libre originando una colisión en una tercera que sí pueda escucharlas a ambas. Para solucionarlo se utiliza un esquema adicional de RTS/CTS [2] (Request To Send / Clear To Send) que es opcional y no tiene por qué estar implementado en todas las WLAN. Los paquetes de RTS compiten por el canal esperando un tiempo DIFS como los paquetes normales. La estación emisora envía un RTS antes de transmitir para realizar la reserva y el receptor contesta con un CTS, quedando reservado el canal. El receptor envía su CTS después de esperar un tiempo SIFS a que el canal esté libre para evitar que otra transmisión pueda acceder al canal antes que éste. El resto de estaciones que hayan escuchado el RTS, el CTS o los datos, actualizan su NAV (Network Allocation Vector) con el tiempo de reserva y no compiten por el canal hasta que éste haya transcurrido. El NAV se utiliza como mecanismo de detección virtual de portadora. Este esquema también se utiliza cuando se envían tramas muy largas ya que si no, se reduce la eficiencia de la transmisión debido a errores introducidos por el canal que hacen que tenga que retransmitirse la trama.

PCF (Point Coordination Function): PCF funciona sobre DCF en redes con infraestructura en las que un punto de coordinación (PC) centraliza el control de acceso al canal mediante el *polling* de las estaciones. Cuando se utiliza este método se divide el tiempo en supertramas con un periodo libre de contención para PCF y otro con contención en el que se usa DCF. Una supertrama empieza con una trama de *beacon* enviada por una estación utilizando DCF, esté activo el modo PCF o no [3]. Esta trama se utiliza para sincronizar a las estaciones y como trama de gestión. El PC las genera a intervalos regulares conocidos como *Target Beacon Transiton Time* (TBTT), que se anuncian en la propia trama de *beacon*. Para entrar en modo PCF el AP (que suele ser el PC) debe competir por el canal esperando un tiempo PIFS (PCF InterFrame Spacing) para acceder al medio. Se cumple SIFS<PIFS<DIFS. Durante el periodo libre de contención, el AP mantiene una lista de todas las estaciones sondeables y las interroga de forma predeterminada aunque no disponga de datos para ellas. Utiliza un algoritmo Round Robin para hacerlo y por eso tenemos garantizado un tiempo de latencia máximo, pero no podemos dar prioridad a las diferentes estaciones. Todas las estaciones contestan al *poll* y envían sus datos si los tienen. Para aumentar la eficiencia se puede realizar en la misma trama el envío de datos, *el polling* y el ACK. Es lo que se conoce como *piggybacking*. Cuando el AP gana el canal todas las estaciones actualizan su NAV al valor $CFP_{maxduration}$ (Contention Free Period) que contiene un valor máximo. No se entrará en el modo DCF hasta que expire el NAV o hasta que el AP envíe un paquete de final del periodo libre de contención (CF-End). El PCF inicialmente era el método pensado para ofrecer unas ciertas garantías a los servicios con requisitos temporales, pero se ha visto que son del todo insuficientes [13]. Los principales problemas que presenta son el retardo impredecible de los *beacons* y el hecho de no conocer la duración de las tramas que enviarán las estaciones sondeadas. Además este modo de funcionamiento no es escalable, ya que el AP debe sondear todas las estaciones que entren en el PCF, y puede ser muy ineficiente para redes grandes con muchos terminales.

3 QoS en WLAN 802.11

Para poder ofrecer QoS a los niveles superiores se hace necesaria una modificación de la capa MAC. Si no se asegura un acceso al medio condicionado por las prioridades, los esfuerzos que se realicen a nivel superior serán infructuosos. Como resultado del trabajo realizado por el TGe y del análisis de las diferentes propuestas que se están haciendo, han surgido el EDCF (*Enhanced Distributed Coordination Function*) y el HCF (*Hybrid Coordination Function*), que son técnicas orientadas a DiffServ [12].

Existen otros métodos para proporcionar QoS en el 802.11 a nivel MAC [4] a parte de los del 802.11e. Propuestas como la limitación de la longitud de la trama máxima, el escalado de la ventana de contención o la diferenciación mediante IFS se basan

en modificaciones de parámetros del DCF y pueden considerarse sustituidas por el EDCF. Funcionando sobre DCF también, tenemos el esquema de DENG, DFS y Blackburst. Sobre PCF encontramos el *Distributed TDM* (Time Division Multiplex). Finalizaremos con AEDCF que es una mejora para que EDCF sea adaptativo.

A continuación veremos la idea principal de cada uno de ellos junto a las propuestas del TGe.

3.1 Limitar la longitud de trama máxima

Uno de los métodos para conseguir esta QoS consiste en limitar la longitud máxima de las tramas transmitidas por cada estación [8]. Es decir: permitiremos enviar tramas mayores a las estaciones de más prioridad. En este aspecto distinguimos dos posibilidades. Descartar paquetes que excedan la longitud máxima permitida a cada estación (o configurarlo para que esto nunca pase); o permitir fragmentar los paquetes que superen la longitud de trama máxima a las estaciones de mayor prioridad.

Como se ha visto anteriormente, entre los paquetes que se fragmentan no hay interrupción y por tanto no hace falta competir de nuevo por el canal, de forma que lo tenemos reservado hasta que se produzca un error debido al ruido o se acabe la transmisión. Es lo que se conoce por *packet bursting*. La transmisión se acabará cuando no queden paquetes o cuando se haya llegado al tiempo máximo permitido a cada estación. Este tiempo máximo debe limitarse ya que puede afectar negativamente en el *jitter*.

3.2 Escalado de la ventana de contención

Otro de los métodos propuesto en [8] se basa en el tamaño de la ventana de contención para proporcionar una cierta diferenciación en los mecanismos de acceso al medio. Asignando una ventana de contención menor a las estaciones de mayor prioridad conseguiremos una cierta calidad estadística. En la mayoría de los casos las estaciones de alta prioridad transmitirán antes que las de prioridad baja ya que escogen el tiempo de *backoff* en una ventana más pequeña. La fórmula que utiliza es:

$$Nueva\_CW = 2^{i+Pj} * Antigua\_CW$$

Donde P_j es el factor que controla la diferenciación e i es el número de colisiones. Cuanto mayor sea P_j, la ventana será más grande y la estación tendrá menor prioridad.

3.3 Diferenciación basada en el IFS

Una nueva forma de dar prioridades a la capa MAC mediante la modificación del tiempo de espera a que el canal quede libre (el IFS) se expone también en [8]. Se consigue prioridad asignándoles un tiempo de espera $DIFS_j$ menor a las estaciones más prioritarias (siendo j la prioridad de cada estación). Se persigue la misma idea del mecanismo original por el que los paquetes ACK tenían prioridad respecto a los paquetes normales de datos. Cada prioridad *j* tiene asociado un tiempo de espera $DIFS_j \geq DIFS$. Esto significa que será DIFS más un número determinado de slots (pudiendo ser 0). Tiene que cumplirse $DIFS_{j+1} < DIFS_j$ para que se produzca la diferenciación. Además se fija el intervalo máximo que puede añadirse a la ventana de contención de cada estación, denominado *Random Range* (RR_j). Si hacemos que $RR_j = DIFS_{j-1} - DIFS_j$ entonces tenemos una diferenciación total sin solapamientos. Así conseguimos que ninguna estación de prioridad *j+1* empiece a transmitir hasta que se hayan enviado todos los paquetes de la estación con mayor prioridad *j*. Pero de esta forma los tráficos de baja prioridad sufren desigualdades muy severas en el acceso al canal. Si queremos que la diferenciación sea menos estricta tan solo tenemos que hacer que $RR_j > DIFS_{j-1} - DIFS_j$.

3.4 Esquema de Deng

Deng y Chang propusieron en [6] un esquema basado en DCF que utiliza tanto IFS como el intervalo de *backoff*. Mediante IFS asigna a las estaciones de mayor prioridad un tiempo de espera PIFS y DIFS a las de menor. Vemos que con este esquema no podremos trabajar de forma correcta con PCF ya que no se garantizaría que el AP pueda tomar el canal cuando sea necesario. Por tanto en redes con infraestructura tendremos que optar por otra técnica. Por otro lado en el algoritmo de backoff la ventana de contención queda dividida en dos partes diferentes no solapadas $[0,CW_a]$ y $[CW_a,CW_b]$. Combinando ambas técnicas obtenemos un esquema de 4 prioridades de acceso al canal.

3.5 DFS (Distributed Fair Scheduling)

DFS [7] utiliza el esquema de backoff del 802.11 para determinar qué estación transmite primero, pero le añade ideas de *fair queueing*. Antes de transmitir se inicia siempre un backoff, que será directamente proporcional a la longitud del paquete que desee transmitirse y a la vez inversamente proporcional a la prioridad que se haya asignado al flujo. Las estaciones de mayor prioridad generarán backoffs más pequeños y por tanto podrán transmitir antes. Para evitar situaciones injustas en las que solamente se envíe tráfico de alta prioridad se ha incluido el tamaño del paquete en el cálculo del backoff. De esta manera los paquetes pequeños se envían más a menudo. Si se produce una colisión se calcula el nuevo intervalo de backoff utilizando el algoritmo estándar del 802.11.

3.6 Blackburst

Otra solución pasa por utilizar la técnica de *blackburs*t [5] que minimiza el retardo del tráfico en tiempo real. Para ello se envía un paquete que bloquea el canal, denominado paquete *blackburst*. La longitud de este paquete es proporcional al tiempo que una estación haya estado esperando por el canal

con el método DCF normal. Esta técnica exige unos requisitos que deben cumplir las estaciones de *mayor prioridad*. Tienen que intentar acceder al medio a intervalos constantes de tiempo t_{sch}, y tienen que poder bloquear el canal durante un cierto periodo de tiempo. Cuando estas estaciones tienen una trama lista para ser transmitida escuchan el canal y, si está libre, la envían tras esperar un tiempo PIFS. Si está ocupado esperan a que quede libre, esperan PIFS y después entran en un periodo de contención por *blackburst*. En este periodo envían el paquete de *blackburst* para bloquear el canal y escuchan para ver que no haya otra estación que hubiera estado esperando más tiempo y por tanto haya enviado un *blackburst* más largo. Si no es así envían su trama y planifican su próxima transmisión para t_{sch} segundos después. Imponiendo un valor mínimo de trama se asegura que cada intervalo de contención mediante *blackburst* tenga un único ganador. Así que una vez sincronizados los tráficos de tiempo real, tenemos como un acceso TDM que no requerirá de más periodos de contención, a no ser que alguna trama de baja prioridad consiga hacerse con el medio.

3.7 TDM distribuido

En [4] se habla acerca de la propuesta de TDM distribuido. A diferencia de los anteriores, este método se propone dentro del modo de funcionamiento PCF. Mediante esta técnica se realiza el sondeo como en PCF normal, pero este nuevo mecanismo nos permite configurar periodos de acceso como si tuviéramos TDM y especificar qué "*slot*" corresponde a cada estación. Una vez que cada estación sepa en qué "*slot*" tiene que transmitir el AP, a diferencia de en PCF, prácticamente no tiene que intervenir más en la comunicación.

3.8 Mecanismos propuestos por el 802.11e

A continuación veremos los mecanismos que está incluyendo actualmente el TGe en el estándar [9]. Las estaciones que trabajan bajo el 802.11e se llaman *enhanced stations* y el punto coordinador, que suele estar en el AP, *Hybrid Coordinator* (HC). Continúan existiendo dos periodos diferenciados en la supertrama, uno con contención y otro sin ella controlado por el HC. EDCF se usa solo en el CP y HCF se usa tanto en CP como en CFP, por eso se le denomina híbrido.

EDCF (Enhanced DCF): EDCF es la base para HCF e introduce el soporte de QoS, mediante las denominadas Categorías de Tráfico o *Traffic Categories* (TCs), al esquema utilizado en el funcionamiento normal. Se puede ofrecer servicios de diferenciación de hasta ocho TC diferentes. Una de las nuevas características más importantes del nivel MAC del 802.11e es la de las TXOP (Transmission Opportunities). Para disminuir el retardo, el *jitter* y conseguir una mayor utilización del medio, se permite el *packet bursting* mediante estas oportunidades. Una TXOP es un intervalo de tiempo durante el cual se permite iniciar la transmisión a una estación en particular. Está definida por un tiempo de inicio y una duración máxima (TXOPLimit). Las TXOP se pueden conseguir mediante contención, las denominadas *EDCF-TXOP*, o a través de HCF, denominadas entonces *polled-TXOP*, y pueden ser para una única vez o persistentes. El límite para las EDCF-TXOP se indica en las tramas de *beacon*, mientras que para las *polled-TXOP* se hace en las tramas de *poll*. Aunque estas tramas son nuevas del 802.11e, las estaciones "antiguas" también ajustan sus NAV. Para lograr QoS, cada TC inicia un backoff aleatorio propio y compite por las TXOPs de forma independiente al resto. Para ello espera a que el canal esté libre un tiempo AIFS (*Arbitration InterFrame Space*) diferente para cada TC. El backoff se elige asignando a cada prioridad una determinada CWMin. La diferenciación se consigue asignando valores menores de CWMin para las prioridades altas. Varios DCF's corren en paralelo con su contador de backoff independiente de forma que las colas con menor CWMin ganen el acceso al canal más fácilmente. Es posible que se produzcan colisiones virtuales (en una misma estación) cuando dos de los contadores lleguen a cero, pero en este caso el scheduler le garantiza la TXOP a la TC de mayor prioridad. El hecho de que el backoff se escoja de forma aleatoria facilita que las colas de menor prioridad también puedan transmitir algún paquete pese a no estar vacías las de mayor prioridad. Por otro lado, también se limita el valor máximo al que puede llegar la ventana de contención (CWMax) de cada TC. Así las TCs de mayor prioridad tendrán valores de CWMax más pequeños. Después de una colisión se calcula un nuevo valor para CW con la ayuda del factor de persistencia PF. Este PF puede ser diferente para cada TC en función de la prioridad que quiera otorgarse. Para el caso de DCF normal, PF era 2.

$$Nueva\_CW[TC] \geq ((Antigua\_CW+1)\cdot PF[TC])-1$$

Mediante AIFS=DIFS y CWMin[TC]<31 (valor por defecto en 802.11b) se consigue prioridad sobre las estaciones que funcionan con el DCF normal. El problema es que AIFS tiene que ser mayor o igual que DIFS, y esto complica la interoperabilidad con estaciones antiguas.

EDCF proporciona una mejora en el servicio a las clases de mayor prioridad, mientras que ofrece un servicio mínimo a las de baja prioridad. El problema es que los parámetros no se adaptan a las condiciones de la red. Por tanto, como cada TC está implementada como una estación virtual, la tasa de colisiones crece cuando las contenciones para acceder al medio son elevadas y se pierde así parte de la mejora inicial. Se puede observar que el tráfico con comportamiento a ráfagas se ubica mejor en el EDCF, mientras que el que tenga altos requerimientos temporales o de jitter tendrá mejor trato bajo el HCF que se explica a continuación.

HCF (Hybrid Coordination Function): HCF dispone de nuevos servicios y formatos de trama para

ofrecer QoS extremo a extremo. HCF provee mecanismos para transferencias libres de contención y con contención controlada en cualquier instante (CFP o CP) permitiendo que el HC genere ráfagas de CFPs en lugar de un único intervalo seguido (como ocurría con PCF). Por tanto se pueden crear un conjunto de "mini-CFPs" durante el CP para ajustarse a las expectativas del tráfico. Esto beneficia a las TCs que necesiten intervalos periódicos de transmisión, ya que podrán extenderse durante el CP. Siempre que el HC lo requiera puede conceder TXOP después de detectar el canal libre durante PIFS (esto le da prioridad a la hora de acceder al canal). Durante el CP cada TXOP empieza cuando el medio está disponible bajo los criterios de EDCF, o cuando una estación recibe una trama especial de *poll* (*QoS CF-Poll*) del HC. Durante el CFP solamente el HC puede asignar esas TXOP entre las estaciones utilizando las mismas tramas *QoS CF-Poll*. Como parte del 802.11e se ha definido un protocolo adicional de acceso aleatorio que permite la resolución rápida de colisiones: la contención controlada. El HC va sondeando a las estaciones que tengan MSDUs disponibles para saber qué estaciones necesitan ser sondeadas, en qué instante y durante cuánto tiempo. Este mecanismo permite a las estaciones reservar las TXOPs enviando peticiones de recursos sin tener que competir por el canal con el resto del tráfico. Esto ocurre durante el intervalo de contención controlada que inicia el HC con una trama especial de control, forzando a las estaciones normales a actualizar su NAV para que no participen durante este tiempo. Esta trama define un número de oportunidades de contención controlada (intervalos cortos separados por SIFS) y un filtro de máscara indicando las TCs permitidas. Las estaciones con datos que pertenezcan a esas TCs enviaran sus tramas de petición de recursos indicando su TC específica y la duración de la TXOP, o el tamaño de la cola de la TC requerida. Para resolver las colisiones más rápidamente, el HC confirma la recepción de la petición generando una trama de control con un campo de *feedback* para que las estaciones que están solicitando sus TXOP puedan detectar colisiones durante el periodo de contención controlada. Por otra parte las estaciones sin QoS pueden competir por el canal en los CP y pueden ser interrogadas por el HC en los CFP, del modo que lo hacían hasta ahora.

3.9 AEDCF (Adaptative EDCF)

Recientemente también han aparecido propuestas que se basan en el EDCF del 802.11e, pero que intentan que la elección de los parámetros que condicionan el grado de calidad de servicio ofrecido pueda ser fijado de forma adaptativa [10]. En concreto se adapta el parámetro del tamaño de la ventana de contención CWMin. De este modo si tenemos un conocimiento general del estado de la red podemos reaccionar de forma adecuada frente a diferentes condiciones de carga. Los estudios realizados mediante simulaciones en el mismo artículo [10] muestran su buen comportamiento.

4 Evaluación del mecanismo EDCF

Llegados a este punto, tenemos los conocimientos suficientes para entender cómo funcionan las redes IEEE 802.11 a nivel MAC y qué modificaciones se han realizado para ofrecer QoS. Por tanto es momento de realizar un análisis mediante simulación con el Netwok Simulator 2 (NS-2) [14] de la problemática que presentan las comunicaciones en entornos inalámbricos con QoS, con el método EDCF [15]. Para ello vamos a ver el comportamiento tanto de las conexiones TCP (Transmission Control Protocol) como UDP (User Datagram Protocol) en los entornos inalámbricos bajo diferentes condiciones. Además aprovecharemos el hecho de tener un medio priorizado para probar un nuevo esquema para redes inalámbricas. La idea detrás de este esquema es que a nivel de transporte los ACK del TCP tengan prioridad sobre el resto de tráfico, como ocurre con los ACK a nivel MAC.

4.1 Comportamiento en DCF

En primer lugar vamos a comprobar cómo se comporta el tráfico TCP y UDP en modo *AdHoc* bajo DCF, por tanto no tenemos prioridades (lo que equivale a usar CWMin=15, CWMax=1023 y difs=0). En las siguientes figuras vemos que el comportamiento para tráficos UDP y TCP es diferente tanto en ancho de banda como en retardo.

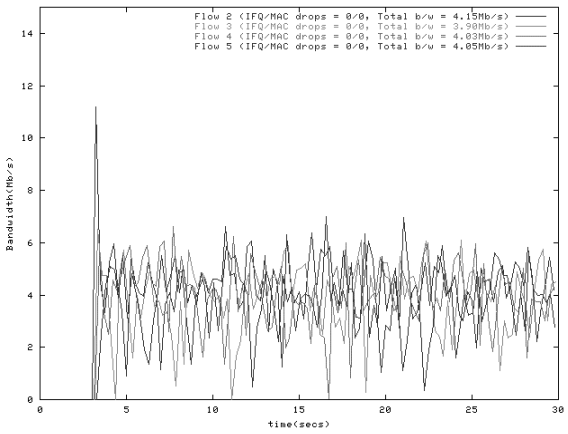


Fig 1 - Ancho de banda para TCP en modo DCF

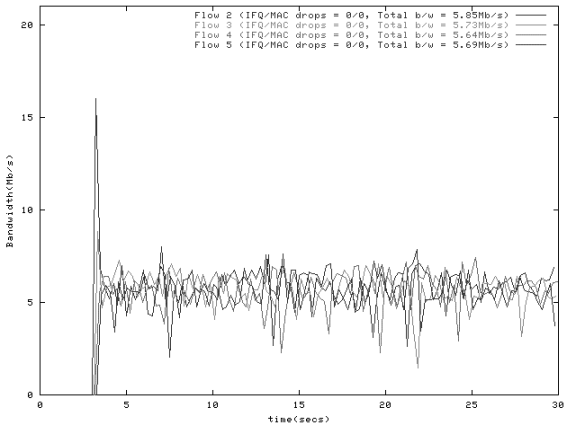


Fig 2 - Ancho de banda para UDP en modo DCF

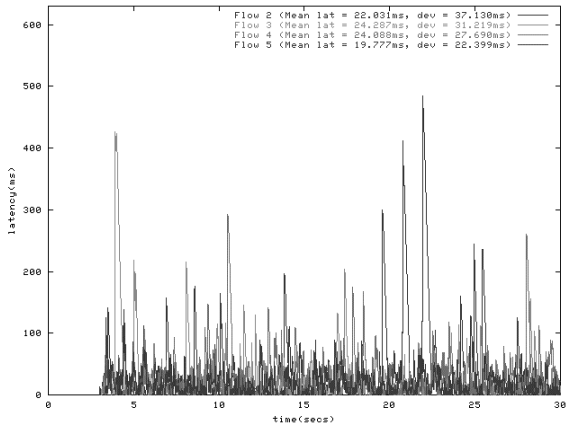


Fig 3 - Retardo instantáneo para TCP en modo DCF

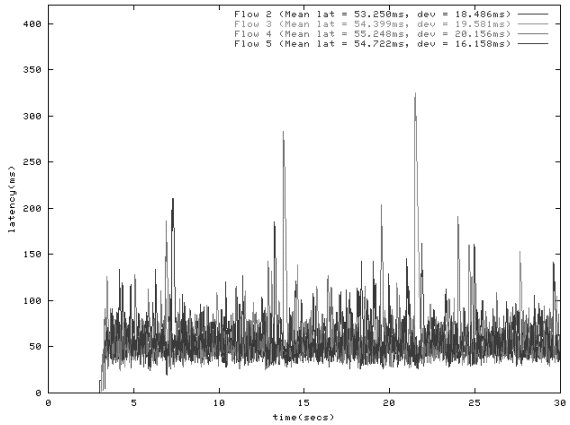


Fig 4 - Retardo instantáneo para UDP en modo DCF

Viendo las gráficas podemos constatar que el comportamiento de UDP es mejor en cuanto a ancho de banda y a retardo. Esto es así porque TCP está sincronizado con la llegada de ACKs. Como el canal es compartido y *half-dúplex* los paquetes de reconocimiento de TCP compiten con el resto de datos con la misma prioridad. La fuente no puede enviar nuevos datos fuera de la ventana hasta que no reciba los ACK que hagan que la ventana vaya desplazándose. Entonces, mientras que UDP aprovecha todo el caudal posible, TCP por cada paquete que quiera transmitir introduce en el canal (y compitiendo por él) un paquete adicional de ACK que no tenemos en UDP. Esto nos lleva a un mayor número de paquetes intentando acceder al canal y por tanto nos aumenta el número total de colisiones. En concreto en TCP tenemos 10223 colisiones mientras que en UDP tan solo encontramos 4778. Esto explica la diferencia en la utilización final del enlace. UDP obtiene un *goodput* de 23,6 Mbps mientras que TCP se queda con 16,4 Mbps (estamos trabajando con un máximo de 36 Mbps).

No es de extrañar tampoco que veamos caídas más bruscas de ancho de banda en TCP por esta misma razón ya que, si ha colisionado uno de los ACK, el tráfico entrante nuevo tiene más opciones de acceder al canal antes que el que ha colisionado (esto es así en entornos CSMA/CA). Por tanto una de las fuentes puede estar esperando un ACK durante un periodo relativamente largo de tiempo. Aún así este tiempo no excede nunca el contador de retransmisión del TCP y el ACK acaba llegando a la fuente. En ningún momento tenemos pérdidas a nivel TCP. Todo esto no ocurre con UDP. Por otro lado los retardos son mayores en UDP que en TCP, pero en cambio su varianza es algo menor. El hecho de tener el medio compartido es el que agrava nuevamente la diferencia. TCP debido a su ventana deslizante de control de flujo, sincroniza las fuentes con la llegada de los ACKs, por tanto es fácil que en determinados momentos una fuente esté enviando varios paquetes seguidos mientras otras esperan confirmación. Esto contribuye a disminuir el retardo en media. En cambio, como tiene que esperar a los ACK, la varianza de este retardo es mayor por el mismo motivo. En resumen, vemos que no hay ningún reparto de ancho de banda ni ningún tipo de garantías en cuanto a retardo. En estas circunstancias no podríamos ofrecer ningún tipo de calidad de servicio a los niveles superiores. Incluso utilizando UDP, que es el que usan las aplicaciones a tiempo real, tenemos un comportamiento aceptable pero con un *jitter* que todavía podemos considerar excesivo.

4.2 Priorización de los ACK de TCP

Una vez vista la necesidad de tener un medio con prioridades analicemos diferentes posibilidades para optimizar las transmisiones en las redes 802.11. Por lo visto en el apartado anterior, lo primero que se nos podría ocurrir es mejorar el comportamiento del TCP. Una primera idea sería intentar reproducir el funcionamiento que tenemos a nivel MAC en el nivel de transporte. Es decir: dotar a los ACK de TCP de prioridad respecto al resto de tráfico. Lo conseguiremos haciendo que los nodos destino envíen su tráfico (ACKs) con prioridad respecto al resto de paquetes. Por tanto asignaremos a los flujos de los nodos destino un AIFS (difs) y una CWmin inferior al resto de paquetes de datos.

Tabla 1 - Parámetros para prioridad en ACK

Tipo de paquete	Prioridad	CWMin	CWMax	difs
ACK	0	15	1023	0
Datos	1	31	1023	1

Vemos que, respecto a la Fig 1, en cuanto a ancho de banda no tenemos una mejora cuantitativa importante hablando en términos de valores medios. Seguimos teniendo el pico inicial debido a que van añadiéndose flujos a la simulación y continuamos moviéndonos alrededor de los 4 Mbps. Pero sin embargo, observamos una mejora cualitativa ya que no tenemos tanta fluctuación y la tasa individual de cada flujo no llega a tener intervalos de transmisión nula. De hecho, en cuanto a comportamiento podemos ver que es mucho más parecido al caso UDP de la Fig 2.

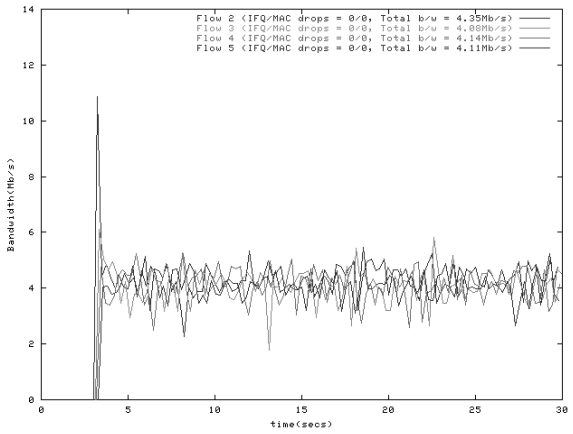


Fig 5 - Ancho de banda TCP con prioridad en los ACK. DCF

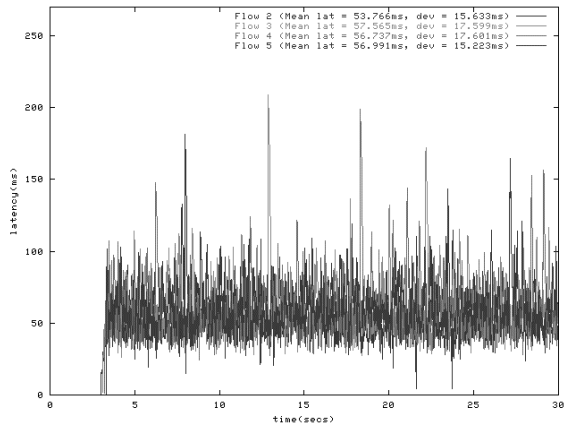


Fig 6 - Retardo instantáneo TCP con prioridad en los ACK. DCF

Seguimos teniendo una utilización del canal similar con un *goodput* de 17 Mbps, pero el número de colisiones se ha reducido casi al mismo nivel que en el escenario con UDP: 5249. En cuanto al retardo, comparando con la Fig 3, comprobamos que hemos logrado reducir su varianza (*jitter*) a cambio de aumentar el retardo en media de la transmisión. Aproximadamente doblamos el retardo pero reducimos el *jitter* a la mitad. De nuevo asimilamos el comportamiento al de UDP.

Podemos decir que a rasgos generales hemos conseguido tener un comportamiento similar al obtenido en UDP en condiciones normales, con la ventaja de estar trabajando con un protocolo fiable y orientado a conexión. Incluso se consigue un *jitter* menor al de UDP con este nuevo esquema. Por tanto sería aplicable en situaciones en las que no nos importe el tanto el retardo como su varianza, como podría ser aplicaciones de video *streaming* en las que se dispone de un buffer de almacenamiento. Tenemos que notar que aún mejorando esta situación, hemos estado hablando de valores medios pero seguimos teniendo picos máximos que podrían provocar que los buffers de recepción llegasen a vaciarse. De todas formas estos picos rondan los 200 ms, muy por debajo de los casi 500 ms a los que podíamos llegar con el esquema tradicional de TCP. El nuevo esquema propuesto reúne lo mejor de los dos protocolos. Pese a todo, son necesarias más pruebas en diferentes entornos antes de poder asegurar que su comportamiento pueda ser siempre mejor al tradicional.

5 Conclusiones

Hemos visto que la actual demanda de determinadas aplicaciones (como voz sobre IP, video,...) con requerimientos de ancho de banda garantizado, retardo controlado,... hace necesario ofrecer una calidad de servicio. Para aplicar estos requisitos en entornos de redes 802.11 necesitamos ciertas modificaciones en la capa MAC.

En un inicio se pensó en proporcionarla con PCF pero no garantiza ningún tipo de reserva y las garantías temporales que ofrece son del todo insuficientes debido a factores como el retardo impredecible de los *beacons* y el desconocimiento de la duración de las tramas que enviarán las estaciones sondeadas. Por eso es un campo de gran investigación en los últimos años y en el que han surgido diferentes propuestas.

Hemos visto que el TGe del 802.11 está trabajando actualmente en ello pero todavía no hay un estándar cerrado. Sus propuestas son el EDCF y HCF que proporcionan una diferenciación en base a las TC's. EDCF tiene un buen comportamiento con el tráfico a ráfagas y es la base para el HCF. El tráfico que tenga requisitos en cuanto a retardos o jitter tendrá mejor trato bajo el HCF ya que puede garantizar las TXOP tanto durante el periodo libre de contención, como durante el periodo en que se trabaja bajo EDCF. Propuestas como las de DENG, escalado de la ventana de contención, la diferenciación mediante IFS y el *packet bursting* quedan englobadas y redefinidas dentro del EDCF. Blackburst requiere que las estaciones de alta prioridad puedan transmitir a intervalos regulares, si no, se degrada mucho su funcionamiento. Distributed TDM está indicado para voz pero no tiene un buen comportamiento para datos. DFS soluciona problemas de *fairness* en el reparto del throughput entre estaciones, pero no queda claro lo que pasa con el retardo.

Gracias a los dos nuevos métodos de acceso al medio del TGe podremos ofrecer QoS y seguir permitiendo tráfico Best Effort en las WLAN. Pero uno de los problemas no resueltos todavía es qué hacer con las estaciones que no implementen estos métodos, ya que tendrán prioridad en EDCF debido a los parámetros que utilizan por defecto. La propuesta es encaminar todo su tráfico al AP y que éste se encargue de gestionarlo como Best Effort

A rasgos generales también se ha visto cuál es el comportamiento de TCP y UDP en entornos inalámbricos del IEEE 802.11a. Por lo general UDP consigue un mayor ancho de banda que TCP dado a que no tiene que sincronizarse con los ACK y no utiliza mecanismos de ventana deslizante para el control de flujo. Por este mismo motivo, UDP tiene mayor retardo que TCP pero tiene menor *jitter*. Como

TCP va esperando a que lleguen los ACK que deben competir por el medio con el resto del tráfico, es más fácil que haya una mayor varianza con respecto al retardo. En cuanto al retardo medio, el hecho de que haya un control de flujo facilita que no tengamos tantos paquetes compitiendo a la vez por el canal y que puedan enviarse paquetes más rápidamente. Por otro lado, aprovechando la posibilidad de introducir prioridades en el medio, se ha analizado un esquema en el que los ACK a nivel de transporte tuviesen prioridad sobre el resto del tráfico. Esta idea persigue clonar el comportamiento que encontramos en las redes a nivel MAC por el cual los ACK se retransmiten con prioridad respecto al resto. Se ha analizado bajo DCF y EDCF. Este esquema lo que consigue es que el comportamiento de TCP se asemeje al de UDP, pero con la ventaja de estar trabajando con un protocolo fiable orientado a conexión. Gracias a esta nueva propuesta tenemos tráfico con un *jitter* mucho menor que para el caso en que trabajábamos con TCP normal. Por el contrario el retardo aumenta, con lo que no irá bien en entornos interactivos como telefonía IP. Para el caso en que introducimos la prioridad en los ACK trabajando con EDCF, el esquema sigue comportándose correctamente haciendo que el tráfico TCP se parezca más al UDP. Seguimos pudiendo tener como mínimo una TC con el retardo y el *jitter* controlados. En escenarios con TCP y UDP juntos vemos que el introducir prioridad en los ACK hace que no sólo el comportamiento de TCP sea parecido al de UDP sino que incluso sea mejor en cuanto a ancho de banda utilizado, retardo y *jitter*. Por tanto este nuevo esquema podría utilizarse para las TC de mayor prioridad y que tuviesen requisitos más estrictos en cuanto a QoS.

Aún así es necesaria una mayor profundización y un mayor análisis de la propuesta para poder evaluarla totalmente. En los escenarios que hemos propuesto se ha demostrado su eficacia, pero son muchos los casos diferentes que podrían evaluarse. Pese a ello, se puede decir que la priorización de los ACK del TCP es efectiva cuando trabajamos con medios compartidos *half-dúplex,* al menos bajo ciertas circunstancias.

Agradecimientos

Parte de este trabajo ha estado financiado por el proyecto TIC 2000-1041-C03-01.

Referencias

[1] G. Armitage. “Quality of Service in IP Networks: Foundations for a Multi-Service Internet”. MTP, Indianapolis, IN., Abril 2000

[2] IEEE 802.11, Wireless LAN medium access control (MAC) and physical layer (PHY) specifications.

[3] S. Mangold et al.. “IEEE 80.11e Wireless LAN for Quality of Service”, European Wireless 2002 Proceedings, 26-28 de Febrero 2002.

[4] Y. Drabu. “A survey of QoS techniques in 802.11”, Febrero 2002: http://trident.mcs.kent.edu/~ydrabu/research/wmac/mac.pdf

[5] J.L. Sobrinho et al.. “Real-time traffic over the IEEE 802.11 medium access control layer”. Bell Labs Technical Journal, pages 172-187, Otoño 1996.

[6] J. Deng et al. “A priority Scheme for IEEE 802.11 DCF Access Method”. IEICE Trans. Comm., vol: E82-B, NO.1 Enero 1999.

[7] N. H. Vaiday et al. “Distributed Fair Scheduling in a Wireless LAN”. Sixth Annual International Conference on Mobile Commuting and Networking, Boston, Agosto 2000.

[8] I. Aad et al., “Differentiation mechanism for IEEE 802.11”, IEEE Infocom 2001, Anchorage – Alaska, 22-26 de Abril, 2001.

[9] IEEE 802.11 WG, Draft Supplement to International Standard for Information Technology-Telecommunications and Information Exchange between systems LAN/MAN Specific Requirements. IEEE 802.11e/D2.0, Noviembre 2001

[10] L. Romdhani et al. “AEDCF: enhanced service differentiation for IEEE 802.11 wireless ad-hoc networks”. INRIA Research Report No. 4544, Septiembre 2002.

[11] R. Braden et al. “Integrated Services in the Internet Architecture: an Overview”. RFC-1633, Junio 1994.

[12] S. Blake et al., “An Architecture for Differentiated Service”. RFC-2475, Diciembre 1998.

[13] A. Lindgren et al., “Evaluation of Quality of Service Schemes for IEEE 802.11 Wireless LANs”. Proceedings of the 26th Annual IEEE Conference on Local Computer Networks (LCN 2001), 15-16 Nov. 2001, Florida, USA.

[14] The Network Simulator (NS-2): http://www.isi.edu/nsnam/ns.

[15] Módulo EDCF para ns-2: ftp://ftp-sop.inria.fr/rodeo/qni/ns-edcf.tar.gz

Modelo basado en la percepción de los usuarios para la gestión de la calidad de servicio en redes de datos

Armando Ferro[3], Eva Ibarrola[3], Fidel Liberal[3], Alberto García[1] y Joaquín Salvachúa[2]
[1]Ingeniería Telemática. Universidad Carlos III
[2]Departamento Ingeniería Telemática. Universidad Politécnica de Madrid
[3]Departamento de Electrónica y Telecomunicaciones. Universidad del País Vasco
ESI de Bilbao, Alameda de Urquijo s/n 48013 Bilbao
Teléfono: 94 601 42 09 Fax: 94 601 42 59
E-mail: {jtpfevaa|jtpibara|jtplimaf}@bi.ehu.es, alberto@it.uc3m.es, jsr@dit.upm.es

Abstract. *During the 90's the exponential growth of the number of Internet users led to the birth of many ISPs and data operators. Nowadays, after the technological bubble, only some of them survive. Since prices have reached the lowest limit, some kind of service quality differentiation between competitors is a must. Unfortunately, QoS measuring methods don't always agree with users' expectations. Moreover, many times data operators don't know how to improve the quality that users perceive. Any investment on infrastructures seems to result on an improvement, but there are no guarantees. This article describes the effort of a group of Spanish Universities towards the definition of a method to set up and measure the relationships between users' satisfaction and network parameters. The model developed would allow both ISPs and legislators to control the degree of users' satisfaction, as well as how any change in network parameters results on a variation of their perceptions.*

1 Introducción

En la década pasada asistimos al desarrollo de Internet en el entorno comercial y, junto con ello, a un crecimiento exponencial el número de usuarios de la red. Esto supuso a muchas empresas privadas y operadores públicos una oportunidad de negocio, para proporcionar a sus clientes acceso al mundo de Internet como ISPs. Sin embargo, después del auge de tantos proveedores de servicio, son pocos los que todavía permanecen en condiciones de competitividad adecuada. Hoy en día el negocio de acceso a Internet se puede considerar que ya está repartido. Las empresas proveedoras difícilmente serán capaces de aumentar su facturación por estos servicios. Es más, sus ingresos en general tenderán a disminuir dado que la competencia en precio es enorme y es difícil que un cliente esté dispuesto a pagar más.

Esto obliga a los proveedores a intentar diferenciarse de su competencia ofreciendo servicios más adecuados al cliente. Aquí es donde surge la importancia de medir la satisfacción que el cliente pueda tener del servicio recibido por su proveedor. La gestión adecuada de la calidad de servicio es fundamental para asegurarse que los usuarios están recibiendo lo que esperan.

En algunos países son las propias Administraciones Públicas las que han puesto en marcha iniciativas que pretenden velar por la calidad de servicio que ofrecen sus ISPs. El problema que surge es cómo hacerlo. Las Administraciones e incluso los operadores de telecomunicaciones están acostumbrados a compromisos de calidad por parte de las empresas prestatarias muy directamente relacionados con parámetros estables de la infraestructura. Sin embargo, en las redes de datos, como Internet, es muy difícil establecer un compromiso de calidad a partir de un parámetro concreto de la red, pues en general, la tecnología de estas redes se rigen en base a un comportamiento estadístico.

Hasta ahora, para los proveedores de conectividad predominantes, la gestión de la calidad se reducía a proporcionar mayor capacidad en su red. Con más capacidad se consigue mejor servicio y los usuarios pueden ver una mejora indirecta de la calidad. Sin embargo, esto no es del todo cierto, y en cualquier caso, basar la calidad en la contratación de más recursos supone repercutir al cliente unos costes que quizás se podrían evitar con una buena gestión. Aparece, además, el problema de evaluar en qué medida esa inversión en infraestructuras ha redundado en una mejora de la percepción del servicio por parte de los usuarios.

Desgraciadamente, a la hora de analizar la calidad de servicio basada en la percepción del usuario, son muchos y muy diversos los factores a tener en cuenta. Por otra parte, nos encontramos con que no existen modelos adecuados para realizar este tipo de análisis. En este artículo se presenta parte del trabajo realizado por varias universidades españolas involucradas en el desarrollo de un nuevo modelo para la gestión de la calidad de servicio en las redes de datos. Este modelo se fundamenta en el análisis de la percepción del usuario.

2 Experiencias previas

Son muchas las organizaciones, tanto públicas como privadas, que han mostrado interés en la definición de los parámetros de calidad que pudieran servir para

medir la Calidad de Servicio (QoS) percibida. Nuestro trabajo se ha basado en la experiencia adquirida por esas instituciones intentando hacer un enfoque particular al entorno de nuestro mercado. Las iniciativas que consideramos más importantes son:

- La Comisión para el Seguimiento de la Calidad en la prestación de los servicios de telecomunicaciones creada el 21 de Diciembre de 1999 como órgano asesor del Ministerio de Fomento (ahora Ministerio de Ciencia y Tecnología). Dentro de esta comisión se crea un grupo de trabajo específico para el seguimiento de la calidad de servicio en los servicios relacionados con Internet. De sus trabajos cabe destacar la elaboración de una propuesta de modelo de regulación sobre la calidad en Internet [1], y una propuesta de un sistema de evaluación y seguimiento de la calidad [2].

- El Instituto de Estándares de Telecomunicaciones Europeo (ETSI) que, siendo consciente de las carencias actuales en cuestiones de normalización elabora una serie de normas y guías que intentan orientar en cuanto a la metodología para la identificación de parámetros relevantes en servicios de telecomunicaciones [3], los parámetros relacionados con el usuario sobre la prestación de un servicio específico [4] [5] y los aspectos a tener en cuenta en el establecimiento de acuerdos con los usuarios sobre niveles de servicio (SLAs) [6].

- El Foro de Estandarización de Internet (IETF) ha creado un grupo de trabajo denominado IPPM (IP Performance Metrics). Este foro ha estudiado el establecimiento de parámetros de medida sobre la calidad, rendimiento y fiabilidad de los servicios de distribución de datos en Internet. Estas medidas pueden ser utilizadas por operadores, usuarios finales o grupos independientes de medida. Este grupo tiene identificados parámetros y procedimientos para realizar esas medidas de una forma adecuada [7].

- La recomendación I.380 del ITU-T [8] define los parámetros que pueden ser usados a la hora de especificar determinados aspectos de calidad de servicio sobre Internet. Considera servicios extremo a extremo de la red, entre dos puntos cualesquiera y en determinadas partes de la red. Estos parámetros son útiles en la planificación y oferta de servicios de tráfico IP a nivel internacional.

- La Dirección General de la Sociedad de la Información de la Comisión Europea, ha encargado un estudio [9] sobre los parámetros de QoS en la provisión de servicios en Internet. El informe final recoge bastante adecuadamente la dificultad a la hora de analizar la QoS y hace un esfuerzo por definir una serie de parámetros significativos que pudieran ser objetivos a la hora de medir la calidad percibida por los usuarios.

En general, prácticamente todos los trabajos concluyen con una lista similar de parámetros genéricos que sirven para medir la QoS pero no atienden a criterios de usuario, sino más bien a parámetros medibles por los proveedores de conectividad y/o proveedores de servicios.

3 Necesidad de un nuevo modelo

A pesar de que la calidad de servicio ofrecida por la red se puede establecer en torno a unos parámetros que pueden ser fáciles de medir de forma totalmente objetiva, en el caso de la calidad percibida por el usuario esto no resulta tan sencillo. La percepción de la calidad depende de las necesidades, de las diferencias culturales, de las expectativas de los usuarios o de su aplicación concreta. Es evidente que la máxima calidad satisfará a todos los usuarios, pero esto no es razonable. Es por ello necesario proponer una metodología que permita realizar una correlación entre los parámetros subjetivos que percibe el usuario y aquellos que son evaluables a nivel de red. Es necesario destacar que la percepción de calidad es holística: se percibe el resultado global, mientras que las medidas son reduccionistas: se evalúan como parámetros individuales. Por ello se identificarán diversas combinaciones y se valorará la aportación de cada elemento de forma individual.

Dada la tipología de servicios en Internet es difícil encontrar una colección de parámetros universales para todo tipo de servicios. Conviene analizar cuales de esos parámetros pueden ser realmente relevantes, a la hora de estimar la percepción de los usuarios, para determinados servicios.

Se han realizado múltiples modelados para intentar medir la calidad de servicios ofrecidos, no solo telemáticos, sino también los servicios de una empresa. Al ser un servicio un bien intangible no se puede medir mediante métricas estándar. Para intentar solucionar el problema se han desarrollado varios modelos donde destaca entre ellos el modelo SERVQUAL [10]. SERVQUAL define la Calidad del Servicio como la medida en que éste cubre las expectativas creadas en el usuario. Esto no establece la calidad objetiva del servicio, sino en qué medida se cumplen las expectativas del usuario. Un servicio mediocre puede satisfacer muy bien las expectativas de algunos usuarios.

Muchas empresas utilizan este modelo para fidelizar a sus clientes, buscando que su satisfacción al recibir el servicio sea suficientemente aceptable como para no pasarse a la competencia.

En otros casos son los Agentes Prestatarios los que establecen ciertas expectativas para el servicio que ofrecen. En esos casos se establecen unas discontinuidades (gaps) entre la calidad ofertada por el Agente Prestatario y la percibida finalmente por el usuario. Estas discontinuidades se sitúan a varios niveles que son competencia de diversos departamentos de las compañías.

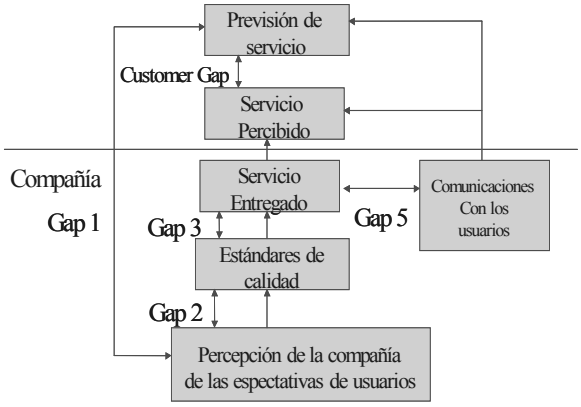


Fig 1 Modelo SERVQUAL

El Agente Prestatario del servicio deberá optimizar los "gaps" internos mediante una adecuada comunicación con sus usuarios.

El punto principal es cómo se relaciona este modelo con las métricas y cómo podemos cuantificar, monitorizar y analizar para poder caracterizar, evaluar y planificar los servicios ofrecidos. En este caso nos encontraremos en el denominado "Customer gap" dentro del cual el usuario percibe un servicio y lo compara con las expectativas que tiene por sus experiencias anteriores, por las acciones realizadas por la empresa, etc. De hecho algunas empresas utilizan la metodología ISO 9001 2000 junto con el modelo SERVQUAL para conseguir auditar la calidad de sus servicios prestados.

El modelo SERVQUAL nos permite analizar las carencias que el servicio puede tener frente a las expectativas que se planteen pero no ofrece una metodología ni elementos útiles que nos permitan evaluar la participación de los diferentes agentes en la prestación final de servicio y, por lo tanto, en la percepción del usuario. Es necesario algún modelo que pueda orientar en la forma de relacionar los parámetros de calidad que cada agente participante en la provisión de servicios controla con la percepción final que puede recibir de ese servicio un usuario. De esta forma podríamos detectar las causas posibles de pérdida de calidad y los agentes podrían conocer sobre qué parámetros de su servicio actuar para mejorar esa percepción final del usuario.

4 Entorno de aplicación del modelo

El modelo que se presenta contempla los planteamientos de diferentes autores y grupos de trabajo pero pretende poner orden en lo que sea posible a la problemática de análisis de la calidad de servicio en redes de datos proporcionando una visión de gestión de la misma orientada a medir la percepción de los usuarios.

Este modelo debemos situarlo en un marco concreto de operación y por ello conviene considerar los aspectos de regulación de nuestro entorno y el modelo de red sobre el que se puede aplicar.

4.1 Marco regulatorio

La Orden Ministerial publicada el 14 de octubre de 1999, es el documento básico en términos de calidad. En ella se regulan los servicios de red fija (telefonía y alquiler de redes), algo sobre Internet y el compromiso de regular la calidad del servicio móvil en un plazo corto. Además en esta orden se acuerda el establecimiento de una "Comisión de Seguimiento de la Calidad en la Prestación de los Servicios de Telecomunicaciones", con el fin de asesorar al Gobierno en la implantación del sistema que se propugna.

Esta Comisión se constituyó en diciembre de 1999, y en su seno se han formado tres Grupos de Trabajo: Telefonía fija/Líneas alquiladas, Móviles e Internet, con unos mandatos concretos.

En el caso concreto del Grupo de Trabajo de Servicios de Acceso a Internet (GT-3) se le otorgó el mandato de analizar, desarrollar y proponer un sistema objetivo de evaluación y de seguimiento de los niveles de calidad de servicio que facilitan los distintos proveedores de servicios de acceso a Internet. Este mandato se ha materializado en tres actividades: tipificar los servicios y proveedores de servicios, definir el parámetro o el conjunto mínimo de parámetros para la medida y evaluación de la calidad de los servicios y establecer los sistemas y procedimientos de medida para los parámetros anteriores (garantizando la necesaria objetividad, precisión y fiabilidad de los resultados).

4.2 Modelo de red

El modelo pretende servir de ayuda para la identificación de los distintos agentes que toman parte en el proceso por el que un cliente accede a Internet, así como, de las relaciones entre dichos agentes.

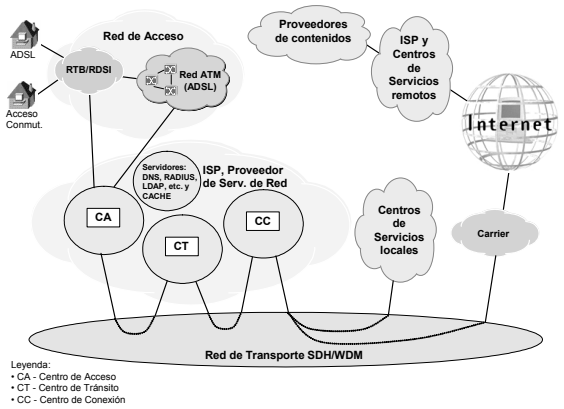


Fig 2 Modelo de Red

Como se puede ver en la figura 2, en una primera aproximación, se pueden considerar los siguientes agentes:

- *Plataforma de usuario:* puede tener fuertes implicaciones en la percepción de la calidad. Presenta la dificultad añadida de no ser responsabilidad de los operadores.
- *Red de Acceso:* conecta los clientes con el ISP a través de las redes de acceso disponibles bien sean RTB/RDSI, ADSL, Cable, etc.
- *Internet Service Provider* (ISP): proporciona los servicios básicos que permiten el acceso a Internet de los clientes.
- *Proveedor de Servicios de Red*: proporciona los servicios básicos que facilitan el uso de los recursos de Internet (DNS, autenticación, autorización, caché, etc.). A veces coincide con el ISP.
- *Carrier*: proporciona conectividad entre distintos ISP y con el backbone Internet.
- *Red de Transporte*: conecta los distintos centros del resto de agentes. Normalmente su influencia es baja siempre que las funciones de conectividad se hagan con una muy alta fiabilidad (establecimiento de redundancias en equipos, líneas, etc).
- *Centro de Servicio*: proporciona el hospedaje de los servicios finales.
- *Proveedor de Contenidos*: es el responsable de generar los contenidos a los que los clientes accederán.

5 Modelo de gestión de la QoS

El modelo que se presenta tiene una estructura matricial tal y como se ve en la figura 3.

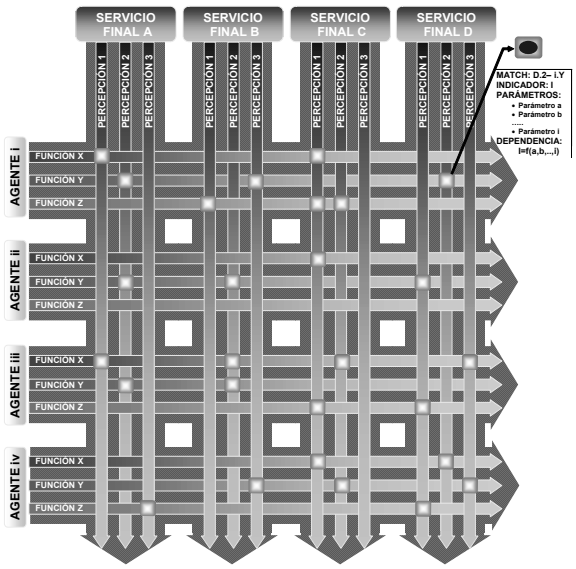


Fig 3 Modelo de gestión de la QoS

Esta estructura matricial permite representar las complicadas relaciones que existen entre los Agentes participantes en la prestación de servicios con las percepciones de los usuarios para diferentes servicios.

5.1 Agentes y sus funciones

El concepto de agente se ha de entender como cualquiera de las partes de un sistema que pudiera tener entidad propia a la hora de ofrecer una serie de funciones determinadas al usuario, en su proceso de acceso a un servicio. Por ello se puede considerar como agente desde la propia plataforma del usuario, que le resuelve una serie de funciones básicas como el procesado o presentación, hasta los servicios que pudiera contemplar un proveedor ISP que proporciona unas funciones más amplias como conectividad, transferencia, etc.

El concepto de función se ha de entender como los servicios básicos que un agente puede proporcionar y que es necesario resolver para facilitar el proceso de acceso de un usuario a un servicio.

Estos elementos forman parte de la vista horizontal del modelo. Para ello habrá que determinar qué agentes participan en el modelo y con qué funciones. Los agentes constituyen, por tanto, primer nivel de abstracción. Dentro de ellos hay que considerar qué funciones básicas resuelven, intentando identificar funciones suficientemente genéricas para evitar crear una colección inmanejable. El último nivel de abstracción en esta vista horizontal serían los parámetros. Cada función, analizada desde el punto de vista de la calidad de servicio que pudiera proporcionar, tendrá una dependencia de determinados parámetros que son susceptibles de medida.

El modelo de red puede servir para identificar alguno de estos agentes. Realmente, el modelo de gestión propuesto permitirá identificar como agente cualquier cosa que se pueda entender que puede afectar en la percepción final de servicio que tenga el usuario. Por ejemplo, la propia plataforma del usuario se puede entender que es un agente importante, pues de sus características pueden derivarse percepciones de calidad diferentes para cada servicio.

5.2 Indicadores

Quizás quede un poco confusa la relación entre estas funciones que realizan los agentes con los parámetros que se pueden gestionar o monitorizar internamente en él. Digamos que la calidad de servicio de una función determinada de un agente puede estar afectada por una colección de parámetros que internamente conoce el agente pero que quizás sea difícil de evaluar externamente. Por ello en el modelo aparece el concepto de indicador que permite, en aquellas funciones que sea necesario, establecer qué parámetros son de interés a la hora de medir una percepción determinada de QoS por parte de los usuarios.

5.3 Servicios y percepciones

Si analizamos la perspectiva vertical del modelo nos encontramos que debemos identificar en una primera fase los servicios finales que recibe el usuario. Posteriormente se identificarán para cada uno de los servicios, aquellas percepciones que se entienda que son de relevancia para estimar la QoS percibida por los usuarios.

El usuario final no tiene una percepción directa de los parámetros de calidad de servicio de una red. No experimenta, por ejemplo, el retardo de la respuesta de un servidor de DNS. Sin embargo, sí que observa que al introducir una URL en un navegador experimenta un retraso hasta que la página aparece. La experiencia de uso es global, involucrando todos los parámetros de red, de la configuración de su propio ordenador, así como de la velocidad de respuesta del servidor. La percepción de calidad será satisfactoria o no en función de múltiples parámetros subjetivos como son las expectativas, experiencias anteriores o percepciones del usuario.

Se entiende por percepción aquellos aspectos referentes a la calidad de servicio que un usuario puede percibir en el proceso de acceso a un determinado servicio. Un error bastante frecuente en muchos estudios en el análisis de la calidad desde el punto de vista del usuario, es el de buscar parámetros excesivamente técnicos para intentar establecer el grado de calidad percibida por el usuario en un servicio. Creemos que el modelo ayuda a establecer abstracciones más adecuadas a la hora de identificar las percepciones que un usuario puede recibir de un servicio. Se puede así escapar de un enfoque excesivamente técnico y tener una aproximación más adecuada a las estimaciones que un usuario pudiera realizar de un servicio. Posteriormente será necesario relacionar esas percepciones con parámetros concretos, pero ello se podrá hacer en una etapa posterior a partir de los indicadores.

Las percepciones de servicio dentro del modelo han de considerarse desde la perspectiva de un usuario concreto, generalmente el usuario final. Según el tipo de usuario que se considere podemos hablar de diferentes percepciones. No es igual la percepción sobre el servicio Web de un usuario que quiere albergar un sitio web en un servidor, de la del usuario final que le interesa acceder a esa información.

5.4 Puntos de cruce (matches).

Una vez identificados los agentes y sus funciones por un lado y los servicios y sus percepciones por otro, en este apartado se desarrolla un análisis más detallado de los cruces (matches) que hay entre una determinada percepción de un servicio y las funciones que proporcionan los diferentes agentes. De esta forma quedará más claro cuales son los puntos más sensibles en el modelo para una determinada percepción. Para cada uno de estos cruces será necesario definir algún indicador que pueda ayudar para estimar el peso y la relación que determinados parámetros dentro de una función tienen con la percepción considerada. No se pretende profundizar en el análisis detallado de los parámetros internos del agente, ni tampoco definir ninguna estrategia de control ni medida, pero sí que es necesario en el modelo definir las relaciones entre ellos. Para ello, tendrán mucha importancia los métodos de medida que se definan.

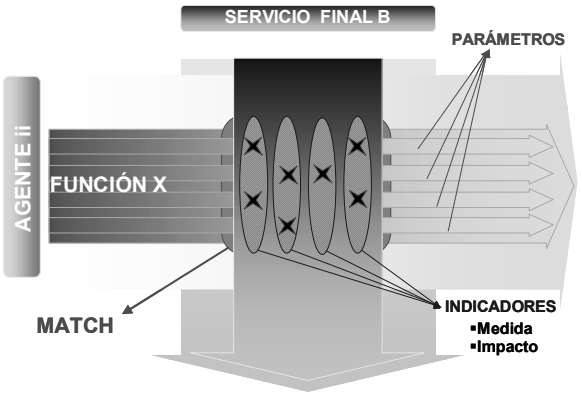


Fig 4 Indicadores, parámetros y puntos de cruce

5.5 Métodos de medida

Otra cuestión que ha surgido a lo largo del trabajo es cómo resolver en el modelo la relación que ha de existir entre los indicadores y los parámetros que controla un agente. Al final, interesa que los agentes puedan llegar a conocer cómo han de optimizar sus parámetros para influir en las percepciones de los usuarios. En la estimación del grado de cumplimiento de un indicador pueden estar afectados muchos parámetros. Intentar controlarlos todos puede resultar inviable. Una labor importante a realizar es la definición de métodos de medida que sirvan para estimar el grado de cumplimiento de un indicador. Estos métodos ayudarán a conocer cómo realizar esas medidas sobre un escenario determinado pero no serán muy útiles a la hora de intentar identificar cuales son los parámetros óptimos de control. Para esto último será necesario desarrollar herramientas de simulación o definir modelos teóricos que permitan jugar con diferentes configuraciones para observar la evolución de un indicador.

6 Análisis del modelo

En este apartado intentaremos dar una explicación más detallada de la necesidad de estos elementos y cómo se pueden combinar para resolver muchas de las necesidades de gestión de la calidad.

6.1 Lógica general

Evidentemente, el modelo debe hacer referencia a los dos tipos de entidades generales que intervienen, los usuarios finales y los agentes. Estos últimos son los que intervienen en la prestación de los servicios.

Pretender cuantificar la calidad global que obtiene un usuario no parece razonable, sino que es preciso referirse de forma separada a cada uno de los servicios finales que se ponen a su disposición. Por tanto, otro de los elementos que interviene en el modelo son los servicios.

La primera pregunta que guía el desarrollo del modelo es bien simple ¿qué hace pensar a un usuario que un determinado servicio es de mayor o menor calidad?. En última instancia, el objetivo es desarrollar un modelo que permita verificar que el usuario está obteniendo lo que desea. Surge la pregunta, ¿qué es lo que el usuario desea de un servicio?. Cualquier usuario común de Internet puede responder a esa pregunta inmediatamente, y es que, desde su punto de vista, la calidad depende de sus propias percepciones del servicio.

Entre las respuestas se podría encontrar: "que los ficheros se bajen rápido", "que no se corte la conexión", "que las radios on-line no se entrecorten", etc. Se trata, por tanto, de valoraciones que para el usuario son más o menos subjetivas pero que un técnico sería capaz muchas veces de medir (extremo a extremo) o asignar un valor o porcentaje. Esas "valoraciones" se realizan sobre las percepciones del usuario y, por tanto, esas percepciones han de ser otro elemento del modelo, puesto que son determinantes para que reflejen claramente las preferencias de los usuarios.

En este punto disponemos de los elementos verticales del modelo: los usuarios evalúan la calidad global que tiene para ellos Internet, a través de los diferentes servicios finales (DNS, WWW, FTP, mail, video, voz…) por medio de percepciones tales como disponibilidad del servicio requerido, tiempo de respuesta, etc...

En el otro eje tenemos a los agentes, que contribuyen a que los usuarios acaben obteniendo el servicio deseado y con la calidad deseada. Evidentemente, cada uno de los agentes que intervienen en la cadena ejerce un determinado rol, que puede afectar a ciertos servicios y a otros no, con un determinado peso, en función de sus características. Cada agente, por tanto, tiene una determinada responsabilidad con respecto a un servicio, en cuanto a que tiene que cumplir su función, además, con unas garantías de calidad mínimas. Es decir, el modelo debe permitir identificar de forma sencilla cuáles son las funciones (y por tanto las responsabilidades) de cada uno de los agentes que intervienen en la prestación de los servicios. Para ello, se representan cada una de las funciones genéricas que proporciona un agente.

Se completa por tanto la parte horizontal del modelo, considerando los diferentes agentes o entidades que intervienen en algún punto del servicio final a usuario con sus respectivas funciones (conectividad, autenticación, prestación de contenidos, etc…).

6.2 Análisis de los puntos de cruce

Queda pendiente una cuestión inicial que se plantea dentro del modelo: identificar cómo depende la sensación de calidad que acaba obteniendo el usuario de un determinado servicio de las funcionalidades de que es responsable cada uno de los agentes que intervienen en ese servicio.

El modelo nos proporciona la solución en los puntos de cruce o "matches" de los elementos verticales. Cada uno de esos "matches" indica que esa percepción del usuario (el elemento vertical) se ve afectada por la función concreta del agente. Evidentemente, es preciso obtener más información que la simple influencia o no. Por ejemplo (ver figura 5), se debe categorizar cómo afecta a la percepción "*D.2*" la función "*i"*. Uno o varios indicadores asociados a cada "match" pueden permitir determinar la cuestión de "cómo afecta". Con ello se da respuesta a la primera de las preguntas planteadas. En general esos indicadores podrían estar cuantificados por el agente correspondiente mediante medidas.

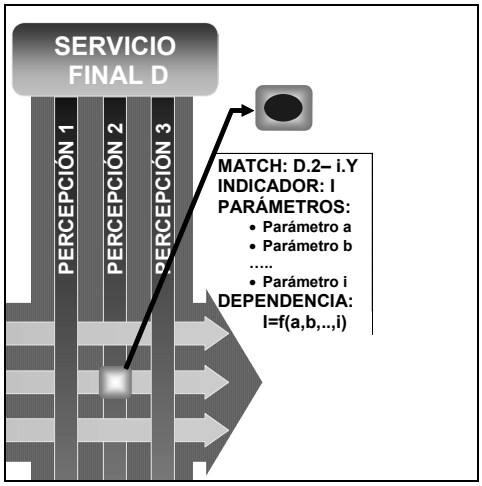


Fig 5 Ejemplo de identificación de un indicador

Sin embargo, existe una cuestión adicional relativa a la posibilidad de establecer la correspondencia entre los parámetros internos de los agentes prestatarios y la percepción de la calidad de los usuarios finales. De esta manera los agentes podrían variar dichos parámetros para mejorar ésta. Debe quedar claro que, aun cuando los indicadores indican una dependencia con esa percepción, son en general magnitudes medibles pero *no necesariamente parámetros internos modificables por los agentes*, dado que, muchas veces, o dependerán de la conjunción de varios de ellos, o ni siquiera existirá una correspondencia inmediata. Para ilustrar esa diferencia de matiz, asociado a cada identificador de match se debe representar en el modelo una serie de parámetros internos del agente asociados, debiendo definirse la función que relaciona el indicador con los parámetros.

6.3 Esquema de valoraciones

En este apartado se documenta un esquema de valoraciones inicial que se puede utilizar dentro del modelo para analizar el grado de cumplimiento de una percepción a partir de las estimaciones que se realicen de los indicadores y de los factores de impacto.

En el modelo se estudian las relaciones que tiene cada función de cada agente con las percepciones definidas para cada servicio. A la hora de valorar la influencia que tiene cada punto de cruce en la percepción final de un usuario son dos las magnitudes que conviene tener en cuenta:

- *El valor del indicador* de referencia. Esta magnitud permite establecer el grado de cumplimiento de un indicador por parte de un agente prestatario. A mayor grado de cumplimiento el impacto sobre la percepción final ha de ser mayor.

- *El factor de impacto* del indicador en la percepción. Es claro que no todos los indicadores que participan en una percepción tienen por qué tener el mismo grado de importancia sobre una percepción. Por ello esta magnitud permite establecer la relevancia que tiene un indicador para una determinada percepción.

Estos dos elementos combinados correctamente pueden servir para establecer un esquema de valoraciones adecuado de una percepción. Convendría aclarar que, en función de cómo se definan los indicadores y los factores de impacto, puede haber problemas en el esquema de valoración. Algunos problemas que surgen a la hora de elegir estos indicadores y factores de impacto y que conviene tener en cuenta son:

- *Independencia de funciones*. Es necesario garantizar en lo posible la independencia entre las funciones de los diferentes agentes que se considere. Si se definen funciones entre agentes prestatarios que son dependientes puede ocurrir que se estén imputando factores de impacto redundantes en el esquema de valoración.

- *Relatividad de los factores de impacto*. Los factores de impacto tienen un peso relativo en el esquema de valoraciones y conviene definir un sistema de asignación de pesos que pueda ser escalable. Si aparecen nuevos factores de impacto es necesario corregir el impacto de los anteriores.

- *Estabilidad de los factores de impacto*. Puede ocurrir en el modelo que una determinada percepción pueda depender de un factor de impacto que no mantenga un valor estable en el tiempo pues quizás dependa de variables dinámicas. En estos casos resultará difícil la estimación de estos factores de impacto. Una solución válida puede ser considerar esos elementos como nuevos indicadores.

- *Necesidad de indicadores complejos*. En muchos casos será necesario utilizar indicadores indirectos que pueden a su vez estar estimados por la medida de otros indicadores. Esto puede realizarse con la idea de simplificar la dependencia de una percepción de múltiples factores. Una forma de realizar esto es utilizando como indicadores percepciones de otros servicios. En el modelo se puede hacer este análisis utilizando algunos servicios que sirvan de referencia para otros.

Una vez identificadas las magnitudes a medir, tanto de indicadores como de factores de impacto, la valoración del grado de cumplimiento de una percepción se puede realizar de múltiples formas. Algunos autores proponen la utilización de lógica borrosa para ofrecer un resultado. Otra propuesta podría ser utilizar una valoración global por puntuación que se podría traducir a un éxito porcentual a través de una transformación por aproximación exponencial. Una forma sencilla de hacerlo es utilizar una valoración por aproximación lineal tal y como se sugiere en la siguiente función:

$$P_{B2} = \alpha_{iY} \cdot I_{iY} + \alpha_{iiZ} \cdot I_{iiZ} + \alpha_{ivX} \cdot I_{ivX}$$

En principio esta formulación permite observar directamente cuáles son los factores de impacto de cada indicador, el peso relativo de los mismos y el nivel de cumplimiento de cada criterio para cada indicador. Aunque se pueden estudiar formulaciones que puedan llevar a resultados con un comportamiento más estable, desde el punto de vista de la percepción, es una aproximación válida que si es necesario puede corregirse fácilmente.

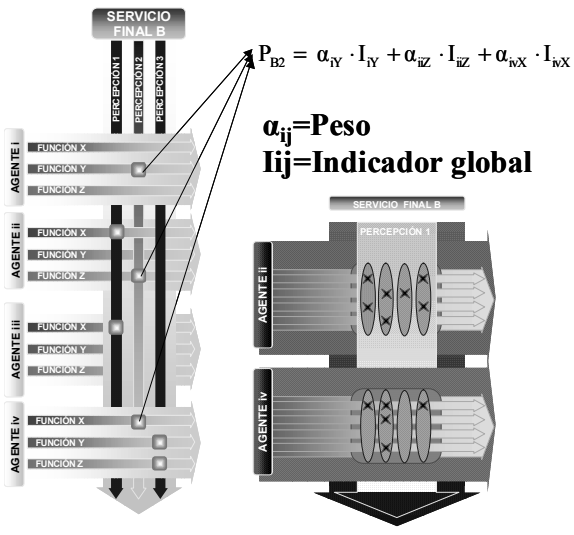


Fig 6 Esquema de valoraciones

Los términos en α representan los factores de impacto que de alguna forma permiten ponderar la importancia de un determinado indicador en la evaluación global de la percepción, y los términos en *I* representan el valor de un indicador en el punto de cruce entre percepciones y funciones. Esos indicadores estarán relacionados con los parámetros a través del método de medida que se establezca.

7 Aplicación del modelo

En este artículo se han presentado las ideas principales del modelo de gestión de calidad que se ha desarrollado. Se está estudiando la viabilidad técnica de la implantación de este modelo en un entorno real. Las primeras impresiones de los usuarios fueron que la aplicación de este modelo resultaba bastante compleja. Ello no se debe al modelo en sí mismo sino más bien al enfoque que se pueda dar a la implantación concreta del modelo.

El problema práctico que aparece consiste en que, si se intentan considerar todos los aspectos que puedan influir en una determinada percepción, el análisis de la misma puede resultar excesívamente complejo. Es necesario conseguir abstracciones adecuadas para facilitar luego la metodología de análisis y evaluación. El modelo ofrece elementos que pueden ayudar en conseguir esas abstracciones y ofrecer una vista más simplificada en su implantación.

Actualmente se está trabajando en esa línea, buscando abstracciones en el modelo que permitan una visión más simple del mismo y se pueda así facilitar su uso práctico. Las ideas sobre las que se trabaja a grandes rasgos consisten en: buscar una definición adecuada de indicadores que se aproximen a las percepciones de los usuarios, definir métodos de medida de esos indicadores que puedan abstraernos suficientemente de los parámetros que gestionan los operadores, definir una estructuración en capas de los servicios que permita simplificar el análisis de las percepciones y analizar el impacto relativo de los diferentes puntos de cruce para poder eliminar elementos poco significativos en el análisis.

8 Conclusiones

Hemos presentado un modelo de gestión de la calidad que permite analizar de forma adecuada las implicaciones de los diferentes agentes en la percepción que un usuario puede llegar a tener de un servicio determinado. El modelo facilita la identificación de las responsabilidades de los agentes en cada uno de los servicios que se le proporcionan al usuario y ayuda a identificar los factores que un agente debe tener en cuenta a la hora de gestionar de forma adecuada sus recursos.

El modelo contempla un esquema de valoraciones lo que permite poder llegar a estimar el grado de cumplimiento de una determinada percepción a partir del análisis de los diferentes indicadores de referencia. En ese análisis van a tener mucha importancia los métodos de medida que se definan. Con ello es posible desarrollar un sistema real de gestión de la calidad de servicio que, basándose en medidas reales, pueda estimar el grado de satisfacción de los usuarios para diferentes servicios.

Referencias

[1] Comisión para el Seguimiento de la Calidad en la Prestación de los Servicios de Telecomunicaciones. Propuesta de bases para la elaboración de un modelo de regulación sobre calidad de servicio en la prestación de servicios relacionados con Internet. 21 Junio 2002.

[2] Grupo de Trabajo de Servicios de Acceso a Internet. Comunicación del Grupo de Trabajo de Servicios de Acceso a Internet a la Comisión para el Seguimiento de la Calidad en la Prestación de los Servicios de Telecomunicaciones.

[3] Quality of telecom services. Part 1: Methodology for identification of parameters relevant to the Users basis (ETSI EG 202 009-1 V1.1.1). 2002-02.

[4] Quality of telecom services. Part 2: User related paraters on a service specific basis (ETSI EG 202 009-2 V1.1.1). 2002-02.

[5] User related QoS parameter definitions and measurements. (ETSI EG 202 057 V1.1.1 -1/ -2). 2002-09.

[6] Quality of telecom services. Part 3: Template for Service Level Agreements (SLA) (ETSI EG 202 009-3 V1.1.1). 2002-02.

[7] IP Performance Metrics
RFC 2678: IPPM Metrics for Measuring Connectivity. Septiembre 1999.
RFC 2330: Framework for IP Performance Metrics. Mayo 1998.
RFC 2679: A One-way Delay Metric for IPPM. Septiembre 1999.
RFC 2680: A One-way Packet Loss Metric for IPPM. Septiembre 1999.
RFC 2681: A Round-trip Delay Metric for IPPM. Septiembre 1999.
http://www.advanced.org/IPPM/docs.html

[8] ITU-T draft recommendation I.380 on IP performance.
http://www.advanced.org/IPPM/T1A1/I.380-8dec98.pdf

[9] Quality of Service Parameters for Internet Service Provisions. Informe para la DG para la Sociedad de la Información. Agosto 2000.
http://europa.eu.int/ISPO/eif/InternetPoliciesSite/QoS/FinalReport.pdf

[10] Grönross, C, (1994) "From marketing mix to relationship marketing: towards a paradigm shift in marketing", Management Decision, Vol.32, No.2, pp.4-20

QoS en redes móviles de cuarta generación

Carlos García, Pedro Antonio Vico, Antonio Cuevas, Ignacio Soto, José Ignacio Moreno

Departamento de Ingeniería Telemática
Universidad Carlos III de Madrid
Avenida de la Universidad, 30
28911 Leganés (MADRID)

E-mail: {cgarcia, pvico, acuevas, isoto, jmoreno}@it.uc3m.es

***Abstract**. This paper describes an architecture for the provisioning of Quality of Service over a fourth generation network. Due to the special requirements of these networks, other services, such as mobility and AAA (Authentication, Authorization and Accounting) must be taking into account. Internal network procedures, allowing the interaction between these components, are described and detailed. Finally, some QoS measurements have been done, over the proposed architecture, in order to demonstrate the capabilities of the proposed solution.*

1. Introducción

En este artículo se presenta una arquitectura de los sistemas móviles de 4G basada en la utilización del protocolo IPv6 tanto en los accesos como en el núcleo de red. Esta nueva arquitectura basada en conmutación de paquetes, requiere la incorporación de técnicas que soporten mecanismos de calidad de servicio (QoS), movilidad, seguridad y contabilidad basados en IP. En este sentido el artículo se centra en la descripción de soluciones basadas en QoS y en su interacción con los módulos de movilidad y AAA (Autorización, Autenticación y Contabilidad). Estas propuestas constituyen parte del trabajo desarrollado en el marco del proyecto europeo Moby Dick [1].

A continuación describiremos el esquema del resto del artículo. En el capítulo 2 se analizará el estado del arte respecto a las redes de cuarta generación, haciendo especial énfasis en la provisión de calidad de servicio. En el capítulo 3 se presenta la solución propuesta dentro del proyecto Moby Dick para QoS, centrándose en la entidad "QoSManager". Y a continuación, en el capítulo 4, se presenta la implementación de dicho elemento. El capítulo 5 presenta una serie de tests y medidas realizadas sobre la arquitectura propuesta. Finalmente se resumirán los resultados del artículo en el capítulo 6.

2. Descripción de la Arquitectura de Red

La principal característica de las propuestas de redes móviles 4G es la utilización de tecnologías IP en el núcleo y en las redes de acceso, para soportar todos los servicios [2] [3]. Mientras en redes 3G coexistirá un núcleo IP para la red de datos con otro núcleo basado en conmutación de circuitos para la prestación de servicios de voz, en las redes 4G sólo existirá un núcleo IP sobre el que se transportará todo el tráfico.

Una imposición para el núcleo de las redes de cuarta generación será el soporte del protocolo IP en su versión 6, IPv6, con lo que quedarían resueltos problemas como el espacio de direcciones, vital para el despliegue de una nueva red dónde sería deseable el uso de direcciones públicas. Concretamente el escenario implementado dentro del proyecto Mobydick es IPv6 nativo.

Existen diferentes tecnologías de acceso que aparecerán en un escenario 4G. No se trata de tecnologías complementarias, de manera que todas podrán coexistir, y en función de las necesidades del cliente podrá optar por alguna de las siguientes: WCDMA (UMTS), Wireless LAN 802.11, Ethernet.

En la figura 1 se representan los elementos funcionales de los que se compone una red de cuarta generación.

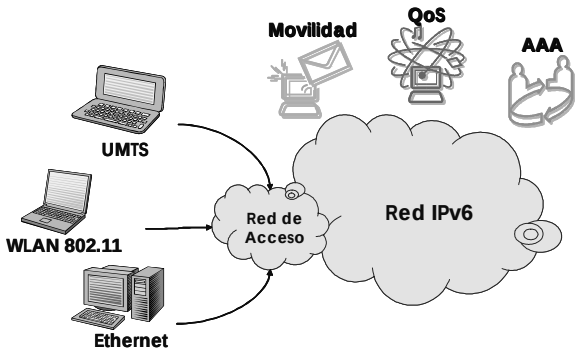


Figura 1. Arquitectura de red de 4ª generación

Los elementos más representativos de esta arquitectura son:

- **QoS**. La tecnología IP tal como se concibió originalmente, no ofrece ningún tipo de garantías

de Calidad de Servicio. Sin embargo, existen servicios, entre ellos el telefónico, con rigurosos requisitos de retardo y variación del retardo (jitter), lo que hace necesario añadir funcionalidad a IP para que las redes basadas en este protocolo sean capaces de soportar este tipo de servicios.

- **AAA**. Los sistemas tradicionales de contabilidad basados en la generación de CDR (Call Detail Record) deben ser modificados para soportar de forma eficiente movilidad de usuarios sobre una red basada en datagramas. Adicionalmente deben soportarse mecanismos de autenticación y autorización para ofrecer mecanismos seguros de identificación y acceso de usuarios. En este sentido el IETF ha definido los sistemas AAA [4], [5] encargados de comprobar la identidad de los usuarios, de controlar los servicios que usan y de tarificarles por ello. Estos sistemas utilizan las redes IP para transportar la información de señalización necesaria. El IETF propone el protocolo DIAMETER [6], [7], sustituto del tradicional RADIUS [8] y capaz de soportar movilidad Inter Dominio (roaming) de usuarios.
- **Movilidad**. Las redes de 4G deberán soportar mecanismos eficientes que permitan la movilidad de usuarios, que utilizando el mismo o distinto terminal se conecten a la red mediante distintas redes de acceso (WCDMA, WLAN, Ethernet, etc.) operadas por distintas entidades. Esto requiere mecanismos que soporten handover entre subredes bajo igual o distinta tecnología (handover horizontal y vertical) de forma eficiente, teniendo como elemento común el transporte IP. La base del soporte de movilidad en redes IP son los protocolos Mobile IP. La propuesta de Fast Handover [9] permitirá conseguir handovers sin interrupción apreciable de las comunicaciones. Esta movilidad requiere interaccionar con los procesos de soporte de QoS en el caso de traspasos entre áreas con distintos recursos de red disponibles y con los mecanismos de AAA para el caso de traspasos entre redes pertenecientes a distintos dominios administrativos.

3. Soporte de QoS en redes 4G

Existen diferentes iniciativas para proporcionar QoS en una red IP. El IETF divide sus esfuerzos en dos grupos Intserv [10] y Diffserv [11]. La implementación de la tecnología Intserv presenta problemas de escalabilidad. La tendencia es el uso de Diffserv en el núcleo combinado con Intserv como solución en la red de acceso.

Como los principales problemas de recursos aparecen normalmente en la red de acceso, y dado que sobredimensionar el núcleo es relativamente sencillo y barato, el uso combinado de Intserv y Diffserv en el acceso y núcleo respectivamente proporciona un buen compromiso entre coste y eficiencia.

Sin embargo esta solución como técnica de QoS presenta algunas limitaciones:

- En Diffserv, al no existir una reserva extremo a extremo, la QoS no está garantizada al 100%. Lo más que podremos alcanzar es una alta probabilidad de obtener el nivel de calidad de servicio deseado, si bien un buen dimensionado de la capa de transporte asegurará un buen servicio.
- Las reservas realizadas por el usuario se traducirán en un código (DSCP [12]) presente en los paquetes que éste envíe, que determinará el tratamiento de nuestro tráfico. El número de códigos es limitado y será el proveedor el encargado de definir éstos así como su implementación. Aparece entonces la posibilidad de que un mismo código DSCP no tenga el mismo significado para diferentes proveedores de servicio, de manera que la calidad de servicio final vendrá determinada por la relación entre los diferentes proveedores que se atraviesen.

El modelo se basa en el uso de un elemento encargado de la gestión de calidad de servicio, el **QoSBroker**. Este componente se encarga de administrar la reserva de recursos y gestionar los routers de la red de acceso y del núcleo. El QoSBroker se comunica con los routers usando el protocolo COPS para el intercambio de información relativa a gestión y administración de la red. COPS [13] define un modelo cliente (routers) servidor (QoSBroker). El QoSBroker, corazón del sistema de calidad de servicio, conocerá el estado de los enlaces hacia cada red de acceso, y podrá autorizar o denegar el acceso de un usuario a la red según la carga. Este elemento mantendrá una relación entre los códigos DSCP utilizados y el comportamiento (PHB) [14] [15] que debe ofrecerse al tráfico. Para ello se han definido una serie de servicios que podemos consultar en la tabla 1.

Tabla 1. Servicios ofrecidos al usuario

Service		Relative	Service parameters	Service
Name	Class	Priority		Description
S1	EF	1	Peak BW: 32 kbit/s	Real time services
SIG	AF41	2a	unspecified	Signalling
S2	AF21	2b	CIR: 256 kbit/s	Priority (urgent) data transfer
S3	AF1*	2c	Three drop precedences (kbps): AF11 – 64 AF12 – 128 AF13 – 256	Olympic service (better then BE: streaming, ftp, etc)
S4	BE	3	Peak bit rate: 32 kbit/s	Best effort
S5	BE	3	Peak bit rate: 64 kbit/s	Best effort
S6	BE	3	Peak bit rate: 256 kbit/s	Best effort

Las especiales características de la clase *Expedited Forwarding* la hacen idónea para servicios en tiempo

real como podrían ser conferencias de audio o video conferencias. Este tipo de tráfico no admite un retardo excesivo, ni la variación del mismo (jitter), además de requerir un ancho de banda bien determinado.

Las clases *Assured Forwarding* podrían utilizarse para diferentes tipos de tráfico. Por un lado el tráfico de señalización podría tratarse con una clase AF, resultando necesario realizar una previa caracterización del mismo para definir correctamente las técnicas de encolamiento requeridas. El tradicional sistema de servicios olímpicos, definiendo las subclases: oro, plata y bronce, según el orden de precedencia en el descarte de paquetes. Este sistema permite una gran flexibilidad para ofrecer una gran variedad de servicios al usuario. Finalmente podríamos destinar otra subclase AF, para algún tipo de tráfico de alta prioridad que no deseamos que compita por los recursos con el tráfico de servicios olímpicos.

Por último, resulta interesante definir el tradicional servicio *Best Effort* para el tráfico que no presenta ningún requisito de calidad de servicio. Debido a las especiales características de las redes de 4G dónde el acceso podría ser una red Ethernet con una capacidad de hasta 100 Mbits, resulta necesario imponer un límite al tráfico inyectado por el usuario para evitar el colapso de la red. Este límite se puede implementar a través de la definición de diferentes subclases de tráfico BE, con diferentes límites de ancho de banda, que se corresponderían con diferentes filtros en los routers de acceso.

Como hemos comentado la interacción entre el QoSBroker y los routers determinará la QoS obtenida. Para ello podemos distinguir entre routers frontera o de acceso (Access Router) y routers del núcleo (Core Routers).

Las funciones referentes a QoS que deberán implementar todos los routers serán: clasificación, acondicionamiento y encaminamiento de tráfico. Estas funciones son lo suficientemente sencillas para ser escalables a toda la red. De esta manera no aparecerá ningún problema de implementación en los **Core Routers,** evitando así el principal problema de escalabilidad del modelo Intserv. Por otra parte los **Access Routers** serán los encargados de controlar el acceso a la red, para ello deberán comunicarse con las entidades anteriormente comentadas: AAA Server y QoSBroker.

El módulo encargado de la gestión y provisión de QoS en los router de acceso es el **QoSManager**. Las principales funciones desarrolladas por este módulo son las siguientes:

- Aplicar algoritmos de gestión y planificación de colas de QoS bajo configuración del QoS Broker. Esta configuración podrá cambiar en tiempo de ejecución.
- Mantener una comunicación COPS con el QoS Broker actuando como cliente PEP.
- Traducción de los protocolos de autentificación CHAP-DIAMETER para permitir el intercambio de señalización entre el usuario y el AAAC que se encuentra dentro de la red DiffServ.
- Capturar flujos de tráfico dirigidos hacia el núcleo de red DiffServ. Esos tráficos deberán estar marcados con determinados DSCPs para ser susceptibles de aplicárseles QoS.
- Generar estadísticas de uso de sus colas e interfaces para su envío al QoSBroker.

4. Implementación del QoS-Manager

4.1. Implementación

Pasamos primero a presentar la arquitectura funcional de nuestro Router de acceso. En la figura 2 podemos ver los diferentes bloques de los que éste se compone. Durante la explicación de los procesos podremos evaluar como estos bloques interoperan entre si.

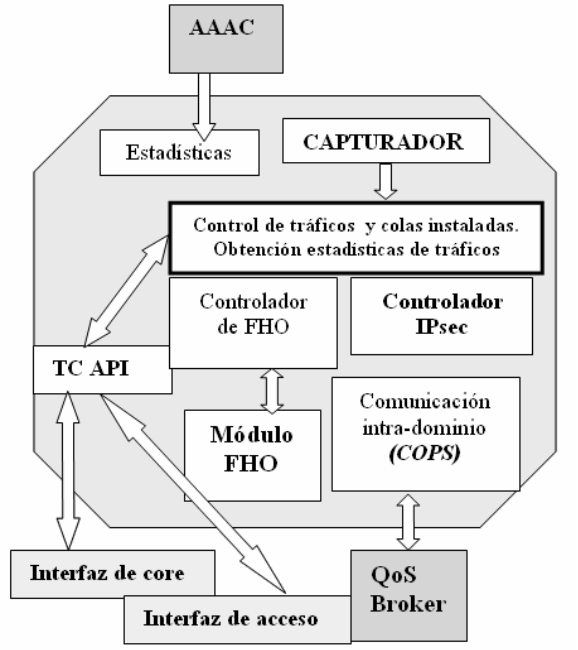


Figura 2. Diagrama de bloques del Router de acceso

Las principales contribuciones tecnológicas a la hora de implementar el QoSManager se corresponden a sus dos principales funcionalidades: calidad de servicio/conformado y políticas remotas. Para un mayor detalle del la implementación del QoSManager consulte [17].

Para lo primero se utilizan las librerías TC API (Traffic Control Application Program Interface) de IBM [16]. El TC API es una interfaz programable para manejar los mecanismos de QoS del núcleo (*kernel*) de red en Linux. La funcionalidad de QoS en Linux se limita a clasificar paquetes de red,

conformar y planificar. Para ello se desarrollaron una serie de funciones que permiten interactuar desde el espacio de usuario con el *kernel* de red. Esa interacción, transparente al usuario, se realiza mediante los sockets *netlink* y nos permite construir árboles de QoS y obtener estadísticas de ellos.

Respecto a la segunda característica utilizamos el protocolo COPS (Common Open Policy Service). Este protocolo, descrito en la RFC 2748 [13], define un modelo cliente/servidor sencillo para proporcionar control de políticas a protocolos de señalización de calidad de servicio. El protocolo COPS se basa en sencillos mensajes de petición y respuesta utilizados para intercambiar información acerca de políticas de tráfico entre un servidor de políticas (**PDP**, Policy Decision Point) y distintos tipos de clientes (**PEP**s, Policy Enforcement Points). Utiliza TCP como protocolo de transporte, es extensible en semántica y guarda el estado de todas las políticas.

Cada mensaje COPS consta de una cabecera COPS y un conjunto de objetos COPS ya definidos. En la figura 3 podemos ver un ejemplo:

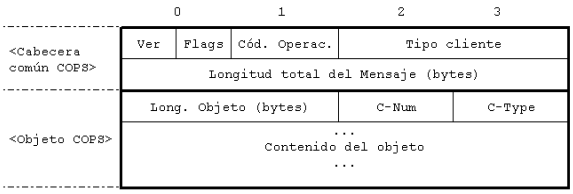


Figura 3. Ejemplo de mensaje COPS

4.2. Procesos

En este apartado vamos a describir los diferentes procesos en los que se ve involucrado nuestro router de acceso. De entre ellos cabe destacar el registro y configuración del router, la autorización de un usuario para el uso la red MobyDick y el proceso de movilidad.

A. REGISTRO DE UN USUARIO EN LA RED

Cuando un cliente desea registrarse en la red MobyDick debe contactar en primer lugar con el servidor de AAAC. Mediante conexiones CHAP (red de acceso) y DIAMETER (red MobyDick) se realiza la autenticación del usuario, por lo tanto, el Router debe encargarse de la conversión entre ambos tipos de protocolos.

Antes de que se le envíe al usuario la confirmación de registro, el AAAC debe instalar en el QoSBroker toda la información de conformado para todos los posibles tráficos del cliente. Este intercambio de información se realiza mediante el protocolo COPS. La información transferida, llamada '*Servicios de Red*', especifica el tiempo de vida del servicio, el ancho de banda, el tamaño de las ráfagas y la prioridad.

Finalmente el usuario, si la autorización ha sido exitosa, debe recibir la confirmación de su registro y una tabla con los DSCPs (Differentiated Services CodePoints) con los que le está permitido marcar sus tráficos de salida.

En la figura 4 podemos ver un esquema de este proceso:

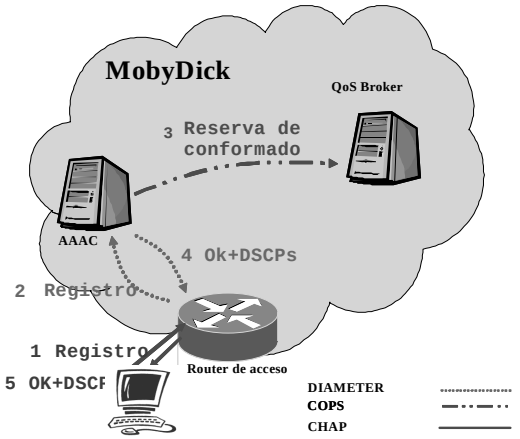


Figura 4. Fase de registro en la red

B. REGISTRO Y CONFIGURACIÓN DEL QOSMANAGER

El QoSManager, al arrancar su sesión COPS, debe registrase en su QoSBroker y acto seguido debe pedirle la lista de correspondencias entre DSCPs y parámetros de calidad de servicio (PHBs). Esa configuración la mantiene el QoSBroker en una tabla llamada: '*tabla de comportamiento*' y se usa en inicializaciones y reconfiguraciones de la interfaz de acceso de nuestro QoSManager.

La información es interpretada en el Router y, mediante las librerías TC API, se crea un árbol de disciplinas de colas en su interfaz de acceso. Los paquetes que viajen hacia la red de acceso recibirán diferentes calidades dependiendo de su DSCP. En la figura 5 podemos ver un esquema de este proceso:

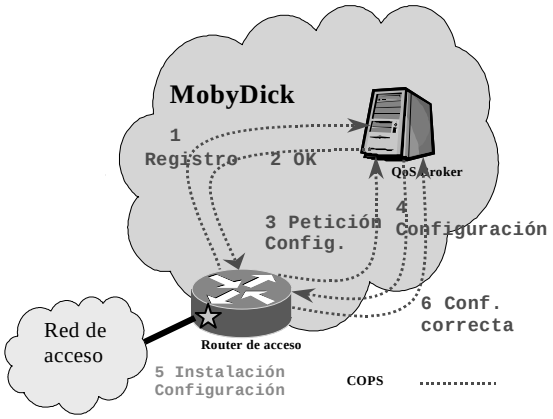


Figura 5. Registro y configuración del router

C. AUTORIZACIÓN Y ACCESO A LA RED MOBYDICK POR UN USUARIO

Una vez que el usuario está registrado en la red MobyDick, éste puede comenzar a cursar tráfico a ésta marcando los paquetes con los DSCPs recibidos. La elección de un DSCP determinado vendrá impuesta por el tipo de tráfico enviado.

Cuando el capturador del router de acceso detecta un nuevo tráfico desde la red hogar a la MobyDick, pregunta al QoSBroker si tiene que autorizar ese tráfico o no. En caso afirmativo, el QoSBroker debe enviar al router los parámetros de conformado para ese flujo. Toda la interacción se realiza a través del protocolo COPS.

El router, en la petición, envía al Broker la siguiente información de identificación del tráfico: direcciones IPv6 de origen y destino y el DSCP del tráfico. El QoSBroker comprueba entonces si la conexión estaba previamente instalada por el AAAC. Si la conexión está instalada, el QoSBroker envía al router un mensaje de autorización positiva junto con los parámetros de conformado (régimen binario y ráfaga). Si la conexión no está instalada la respuesta es negativa y el tráfico será bloqueado por el router.

El QoSManager, mientras todo esto ocurre y hasta que el Broker no le confirma la autorización, bloqueará el tráfico por seguridad.

Una vez que el router ha aplicado la orden del QoSBroker debe enviarle a éste un mensaje COPS informándole sobre el éxito o no de la decisión.

Tanto la autorización como el conformado tienen un tiempo de vida en el router. Cuando ese tiempo se agote, el router deberá realizar una nueva petición para saber si ese tráfico continúa estando autorizado para usar los recursos de la red MobyDick o deberá ser bloqueado. De esta manera las autorizaciones y el conformado de los tráficos pueden ser refrescados periódicamente. En la figura 6 podemos ver un esquema de este proceso en el caso de una autorización positiva.

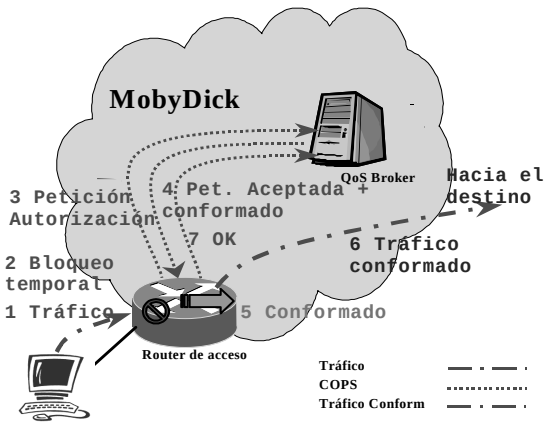


Figura 6. Acceso a la red de un tráfico

D. OBTENCIÓN DE ESTADÍSTICAS Y RECONFIGURACIÓN

Al mismo tiempo que las capturas y conformados de nuevos tráficos ocurren, el router realiza otras operaciones. En este apartado vamos a comentar dos de ellas: la obtención de estadísticas del uso de su interfaz de acceso y la comprobación de peticiones de reconfiguración provenientes del QoSBroker.

Según explicamos en el arranque de nuestro router, el QoSBroker configura las colas de calidad de servicio del interfaz de acceso de éste. Por estas colas pasan todos los tráficos que viajan de la red MobyDick a la red de acceso. Mediante unas funciones del TC API, el router puede obtener estadísticas de uso de las colas tales como: régimen binario, paquetes por segundo, descartes por segundo, etc. Estas estadísticas son enviadas al QoSBroker a través de la conexión COPS, por lo que este servidor puede hacer cálculos con ellas y estimar la carga del router. El incremento de tiempo que debe utilizarse entre dos informes de estadísticas consecutivas es también configurado por el Broker mediante un parámetro en el arranque.

Si el algoritmo de planificación del Broker estima necesario una modificación en los parámetros de las colas de acceso del router, puede cambiar la '*tabla de comportamiento*' y ordenar una reconfiguración. De este modo el router volvería a solicitar al QoSBroker la información sobre las colas de QoS y las instalaría de nuevo actualizadas en el interfaz de acceso.

En la figura 7 podemos ver un esquema de este proceso.

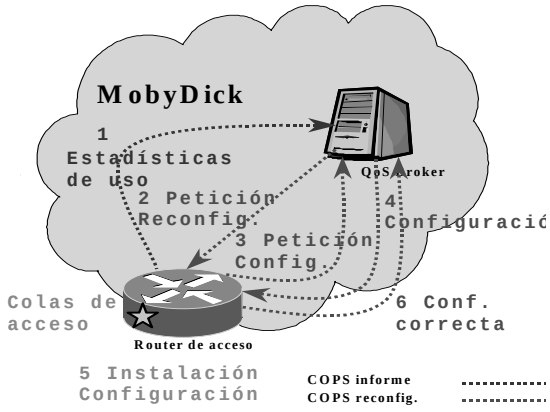


Figura 7. Transferencia de estadísticas y reconfiguración

E. MOVILIDAD

El QoSManager permite la interacción con el módulo de '*Fast Hand-Over*' de la red MobyDick para traspasos rápidos de la configuración de calidad de servicio de un router a otro.

Cuando un usuario se mueve de una red de acceso gobernada por un router a otra, se dice que se ha producido un '*Hand-Over*'. En nuestro caso interesa que esa operación sea lo más rápida posible para no interferir al tráfico.

En esta acción están implicadas al menos tres máquinas: el antiguo router de acceso, el nuevo y uno o más QoSBrokers.

El router de acceso antiguo debe informar a su QoSBroker sobre el cliente que va a moverse y hacia qué nuevo router de acceso. La información que le pasa el módulo de *Hand-Over* al router y este a su vez le comunica al Broker es la siguiente: CoA antigua, CoA nueva y dirección del nuevo router.

Dado que la transmisión ha de ser inmediata, el módulo de *Fast Hand-Over* debe interrumpir la ejecución del viejo router para que mande el mensaje de traspaso al Broker.

Cuando el QoSBroker recibe el mensaje comprueba si el router destino está dentro de su red. En caso afirmativo, debe buscar todos los tráficos instalados para ese usuario y mandárselos inmediatamente al nuevo router para que este instale las autorizaciones y los filtros de conformado correspondientes.

Si el router nuevo no está en la red que gobierna el primer QoSBroker, éste debe mandar la información de los tráficos del cliente al QoSBroker que administra al nuevo router de acceso. El nuevo Broker enviará la información al router final y ambos instalarán la parte de configuración que les corresponde.

La información que se transfiere al router de acceso final es el conjunto de todos los tráficos que tenía instalados en la red hogar. Los parámetros son: direcciones de origen y destino, DSCPs e informaciones de conformado.

Una vez que el nuevo router ha recibido el mensaje del Broker, instala todos los tráfico requeridos e informa al servidor de la correcta (o no) instalación de estos.

En la figura 8 podemos ver un esquema de este proceso en el caso de que el mismo QoSBroker gobierne a los dos routers de acceso involucrados.

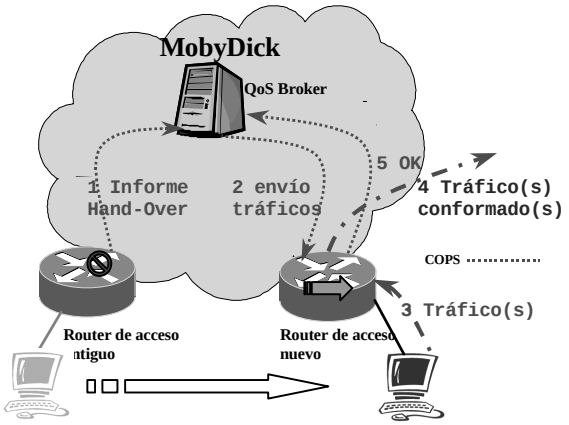


Figura 8. Fast Hand-Over

F. BORRADO DE UNA CONEXION CADUCADA

Cuando el QoSBroker ha autorizado una conexión en el router y ésta deja de transmitir paquetes, el programa capturador de tráfico asociado no detecta nada. Por lo tanto, cuando el QoSManager procede a actualizar esa conexión y se percata de que no está en la lista de conexiones capturadas, espera algunos segundos más por tráfico antes de borrarla. La cantidad de segundos que espera es configurable en el programa. Si el capturador sigue sin detectar tráfico por esa conexión pasado el tiempo de guarda, el router la desautorizará e informará al QoSBroker de ello mediante un mensaje COPS.

5. Medidas de QoS sobre el QoSManager

En este apartado presentaremos una serie de pruebas que realizamos con nuestro router en las que se podrán observar QoS, reconfiguraciones y prioridades de tráficos. Se pueden encontrar pruebas más detalladas en [17].

A. RESERVA DE ANCHO DE BANDA

En esta prueba se pretenderá comprobar como afecta a una cola de la interfaz de acceso –que tendrá activado el '*borrow_flag*'– la ocupación de otra(s). Este interesante parámetro permite, a la cola que lo tenga activado, tomar para ella el ancho de banda que no usan las otras colas o que sobra en la interfaz.

Para realizar esta prueba se midió el ancho de banda que conseguía cursar la cola '*Best-effort*' (creada con el *borrow_flag*) frente a diversos valores de regímenes binarios en la cola EF (DSCP = 0x2e)

Para el test se generó un tráfico '*best-effort*' que intentaba ocupar todo el ancho de banda de la interfaz (100Mbps). Al mismo tiempo, la cola EF que tenía reservados 30Mbps, comenzaba a cursar tráfico incrementalmente hasta 60Mbps.

En la gráfica 9 representamos los resultados obtenidos.

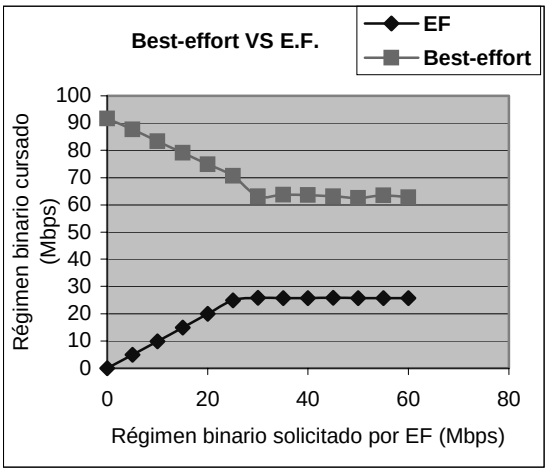


Figura 9. Reserva de ancho de banda

Como podemos observar, cuando por la clase EF no se cursa ningún tráfico, la clase '*Best-effort*' intenta apropiarse de todo el ancho de banda de la interfaz para ella. A medida que la clase EF comienza a requerir ancho de banda lo consigue ya que lo tiene reservado. No obstante cuando la clase EF requiere más ancho de banda del que tiene aprovisionado (30Mbps) no se la deja continuar y lo sobrante se le deja a la '*best-effort*'.

B. RECONFIGURACIÓN DINÁMICA

En este test vamos a probar el mecanismo de reconfiguración de las colas de calidad de servicio bajo petición de QoSBroker. Como el nombre del test indica, presentaremos la evolución temporal del tráfico cursado.

Los pasos a seguir son los siguientes: insertar en una clase cualquiera un tráfico que la sature. Cuando el QoSBroker reciba el informe de estadísticas a través del protocolo COPS llamará a la función que controla el ancho de banda. Esta función, según esté programada, procesará los datos recibidos y la '*tabla de comportamiento*' y, dependiendo de los resultados, solicitará al router una reconfiguración de las colas de su interfaz de acceso.

Nuestra función de planificación de las colas opera de la siguiente manera: si la clase tiene más de 20 descartes por segundo incrementará el ancho de banda de todas las colas que la forman en un factor 1,3. Acto seguido pedirá al router una reconfiguración. Es importante recalcar que las estadísticas se obtienen por clase DiffServ, no por cola.

Para este test se generó un tráfico unidireccional UDP de 800kbps. Se marcó en la fuente como clase AF1h (DSCP = 0x0e). Esta clase tenía inicialmente reservados 180kbps en tres colas de 70, 60 y 50 kbps. Por este motivo la clase se saturó y el resultado de un test de 200 segundos fue el siguiente.

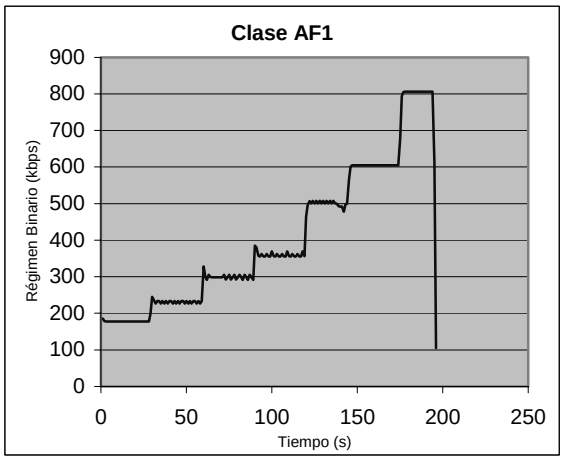


Figura 10. Reconfiguración dinámica

En la gráfica podemos apreciar claramente como, al principio, el tráfico fue clasificado a su cola AF1h de 180kbps. El Router, más tarde, tomó estadísticas de la cola y transfirió al QoSBroker el régimen cursado así como los paquetes, descartes y sobrecargas por segundo.

A la vista de los resultados (más de 20 descartes por segundo) el QoSBroker decidió que había que incrementar cada tráfico de esa clase por un factor 1,3. El Broker cambió entonces los valores de las colas virtuales de esa clase: la primera a 91kbps (70·1,3kbps), la segunda a 78kbps (60·1,3kbps) y la tercera a 65kbps (50·1,3kbps). Al final, la clase AF1 tuvo un ancho de banda de 234kbps (91+78+65kbps). De igual forma también activó el flag de reconfiguración. Cuando el Router necesitó actualizar este tráfico hizo una petición al BB y este le contestó solicitando una reconfiguración. El Router repitió la secuencia de configuración y aplicó las nuevas colas.

Obviamente como todavía se seguirían produciendo más de 20 descartes por segundo el proceso se repitió hasta alcanzar los 800kbps del tráfico. Los valores teóricos de los escalones según el factor que aplica el QoSBroker serán: 180, 234, 304, 395, 514, 668kbps y finalmente el máximo (800 kbps). El resultado es el que tenemos en la figura y es bastante fiel a los valores teóricos.

C. CONFORMADO EN LA INTERFAZ DEL NÚCLEO

En este test vamos a presentar los resultados de una batería completa de pruebas sobre la interfaz de core. Como se explicó en los procesos, en esta interfaz los tráficos son autorizados o rechazados bajo petición al QoSBroker. Si éste da permiso para instalar uno, en el mensaje de admisión viajarán los parámetros de conformado.

El QoSBroker guardaba la siguiente tabla de conformado según los DSCPs de tráficos instalados:

0x22 = 90kbps, 0x24 = 80kbps, 0x26 = 70kbps, 0x1a = 60kbps, 0x1c = 50kbps, 0x1e = 40kbps, 0x12 = 30kbps, 0x14 = 20kbps, 0x16 = 10kbps, 0x0a = 9kbps, 0x0c = 8kbps, 0x0e = 7kbps. Todos con una ráfaga de 1514 bytes.

Se instalaron tráficos para todas esos tipos y se introdujeron flujos que saturaban el conformador. Los resultados de regímenes de salida fueron los expuestos en la tabla2.

Tabla 2. Resultados de las pruebas de conformado

Regímenes binarios obtenidos	Clase de DSCP: 001b	Clase de DSCP: 010b	Clase de DSCP: 011b	Clase de DSCP: 100b
Bits de descarte: 010b	9.2 kbps	29.2 kbps	58.6 kbps	80.5 kbps
Bits de descarte: 100b	8.3 kbps	19.9 kbps	49 kbps	77.9 kbps
Bits de descarte: 110b	7.3 kbps	10.2 kbps	39.3 kbps	68.3 kbps

Observando los resultados y teniendo en cuenta que los valores que en ella se especifican son a nivel de datos UDP sobre IPv6 y Ethernet con un tamaño de datos de 1000Bytes, los resultados están bastante conformes a las especificaciones de la tabla del QoSBroker.

5. Conclusiones

Este artículo presenta una propuesta de arquitectura para las futuras redes de cuarta generación, ofreciendo servicios esenciales en estas redes tales como calidad de servicio, movilidad y sistema AAA. Se ofrece una descripción detallada del sistema de provisión de calidad de servicio basado en la solución Diffserv del IETF.

Igualmente se realiza un estudio detallado de los elementos QoSBroker y QoSManager como pilares de la red para la consecución de una gestión adecuada de la calidad de servicio. Y se analiza con mayor profusión la implementación del QoSManager dentro de los router de acceso.

Finalmente se ofrecen una serie de resultados obtenidos sobre una plataforma de pruebas con respecto a la reserva de ancho de banda y a la reconfiguración dinámica de QoS, demostrando de esta forma la viabilidad del proyecto.

Agradecimientos

Este trabajo ha sido financiado parcialmente por la comisión Europea a través del proyecto "Moby Dick - Mobility and Differenciated Services in a Future IP Network".

Referencias

[1] Proyecto Moby Dick: “Moby Dick - Mobility and Differentiated Services in a Future IP Network” (IST-2000-25394). http://www.ist-mobydick.org

[2] V. Marques et al., “An Architecture Supporting End-to-End QoS with User Mobility for Systems Beyond 3rd Generation”, IST Mobile Summit 2002

[3] V. Marques et al., “An IP-based QoS Architecture for 4G operator scenarios”, IEEE Wireless Communication, June 2003

[4] C. de Laat et al.: “Generic AAA Architecture” IETF, Experimental RFC 2903, August 2000

[5] IETF, AAA Working Group - http://www.ietf.org/html.charters/aaa-charter.html

[6] Pat R. Calhoun, "Diameter Base Protocol", <draft-ietf-aaa-diameter-10.txt>, April 2002

[7] Stefano M. Faccin, "Diameter Mobile IPv6 Application",<draft-le-aaa-diameter-mobileipv6-00.txt>, 2001

[8] Rigney, C. et al, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

[9] G. Dommety, ed. "Fast Handovers in Mobile IPv6", Internet Draft, work in progress, <draft-ietf-mobileip-fast-mipv6-3.txt>, July 2001

[10] IETF, Intserv Working Group - http://www.ietf.org/html.charters/intserv-charter.html

[11] IETF, Diffserv Working Group - http://www.ietf.org/html.charters/diffserv-charter.html

[12] K. Nichols, S. Blake, F. Baker, D. Black. “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers”, RFC 2474, December 1998.

[13] D. Durman et al. “The COPS (Common Open Policy Service) Protocol” RFC 2748, January 2000

[14] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski. “Assured Forwarding PHB Group”, RFC 2597, June 1999

[15] V. Jacobson, K. Nichols, K. Poduri. “An expedited forwarding PHB”, RFC 2598, June 1999

[16] “*IBM TC API Project*” http://oss.software.ibm.com/developerworks/projects/tcapi/

[17] P. A. Vico, J. I. Moreno, A. Cuevas. “Implementation of QoSManager in AR”. Departamento de Ingeniería Telemática, Universidad Carlos III Madrid, Abril 2003.

Propuesta de arquitectura multiprotocolo para la implantación incremental de un modelo de servicio con garantías QoS sobre redes IP

Alfonso Gazo Cervero, José Luis González-Sánchez
Área de Ingeniería Telemática. Departamento de Informática. Universidad de Extremadura.
Av/ Universidad s/n. 10071 Cáceres
Teléfono 927 257 253 Fax: 927 257 202
Email:{agazo,jlgs}@unex.es

Abstract *Internet currently offers a very simple delivery service, based in the best-effort model. Using this model, the higher guarantee that the network provides is reliable data delivery. Nevertheless, new kinds of applications have been recently developed, like videoconference, VoIP or VPN, which are sensitive to the QoS that they obtain from the network. In particular, traditional data delivery, where reliability can be obtained in change of not-controlled delays, becomes inacceptable.*

Before these new kinds of applications can be widely used, the Internet infraestructure must be modified to offer QoS at real-time and end to end controlled delays. Nevertheless, it seems reasonable to consider this requirement high enough to, in case of taking it on, it could only be satisfied at long time and big effort.

In this work we propose an architecture that enables an incremental implantation over Internet of a model of service in a way that the user QoS requirements could be satisfied at the places where certain mechanisms would be present.

1. Introducción

La simplicidad del modelo de servicio basado en *best-effort* es una de las razones esenciales del éxito de Internet. Esta simplicidad ha permitido que IP sea implementado sobre cualquier nivel de enlace concebible y también ha provocado que la gestión de red y la interconexión de proveedores sean tareas asumibles por organizaciones de muy diferente tamaño. Junto con las ventajas inherentes de la tecnología no orientada a conexión y el principio de diseño extremo a extremo, el modelo de servicio *best-effort* ha permitido una Internet rápida, sencilla, barata y altamente escalable.

Sin embargo, ahora el reto consiste en conseguir diferenciar esquemas de tráfico como un servicio de valor añadido a la red, sin añadir una complejidad adicional significativa y sin poner en peligro los principios de diseño que han hecho de Internet la red de mayor éxito en la actualidad.

La implantación de tecnologías como VPN sobre Internet, VoIP, aplicaciones de telemedicina, televigilancia o domótica requiere esta diferenciación con objeto de ofrecer garantías con respecto a la calidad del servicio (QoS, *Quality of Service*) [1] ofertada. Un escenario especialmente atractivo para la implementación de QoS sobre IP es la posibilidad de proporcionar un entorno multiservicio sobre el *núcleo* de una red de un proveedor de servicios.

Si bien en el contexto de un único proveedor de servicios se pueden adoptar, e incluso desarrollar, soluciones propietarias para la provisión de garantías QoS, la dificultad surge cuando se trata de aplicar un modelo *outsourcing* donde, para la provisión de un servicio, se requiera la intervención coordinada de más de un proveedor. Por ello se hace necesaria la implantación de un modelo de servicio que, si no es único para todos los proveedores de servicio presentes en Internet, sí que debería permitir al menos la interoperabilidad entre modelos diferentes.

Este trabajo comienza analizando el modelo de servicio utilizado actualmente en Internet, para a continuación describir las propuestas más relevantes para la provisión de garantías QoS. Analizando algunos parámetros organizativos de Internet en la actualidad, se presenta una propuesta cuya característica principal es la de no requerir la migración completa de los elementos de red en Internet para lograr la provisión de garantías QoS. Por último, se presentan los trabajos futuros y las conclusiones extraídas hasta el momento.

2. Trabajos relacionados

Posiblemente el modelo de servicio más popular es el *best-effort*, en el que el tráfico se procesa lo más rápidamente posible, pero no existen garantías ni restricciones temporales ni siquiera de entrega real. Éste es el único modelo de servicio soportado hasta ahora por Internet. Otro modelo de servicio ampliamente extendido es el ofrecido por la red telefónica, donde se asigna un circuito fijo para cada llamada.

Últimamente se han desarrollado dos modelos para proporcionar QoS en Internet: el modelo de *Servicios Integrados* (IntServ) [2] y el modelo de *Servicios Diferenciados* (DiffServ) [3]. La base de IntServ es reservar recursos (ancho de banda y espacio en búfer) para cada flujo de forma que la QoS pueda ser garantizada en caso de que sea necesario. La base de DiffServ es dividir el tráfico en clases diferentes y dar a cada una de estas clases un tratamiento diferente. Ambos modelos necesitan utilizar mecanismos ubicados en los niveles de red y de transporte para poder proporcionar esta QoS.

El enfoque DiffServ está considerado actualmente por muchos expertos como el modelo a implantar para proporcionar QoS en Internet. Desde esta perspectiva, técnicas como la ingeniería de tráfico, MPLS [4], encaminamiento basado en restricciones, reencaminamiento rápido, redirección de tráfico o equilibrio de carga son independientes. Sin embargo, en [5] se argumenta que la utilización de los modelos IntServ/DiffServ no resuelve todos los problemas relacionados con la obtención de QoS en Internet y se expone una arquitectura que incluye el modelo DiffServ, pero también otras técnicas en los niveles de aplicación y de red (tabla 1).

DiffServ proporciona una degradación diferenciada (o no degradación) en el rendimiento para clases de tráfico diferentes allí donde hay congestión en la red. Si puede evitarse la congestión, el rendimiento de todo el tráfico será bueno incluso sin DiffServ. La ingeniería del tráfico puede evitar la congestión causada por una distribución de tráfico no equitativa a lo largo de la red. Es por esto por lo que la ingeniería del tráfico resulta útil para proporcionar QoS.

De forma adicional, DiffServ no proporciona mecanismos para atender fallos en enlaces o en *router*. En estas situaciones en las que el tráfico enviado a lo largo de la ruta se pierde, el tiempo de recuperación es crítico. De ahí la necesidad de implantar mecanismos de reencaminamiento rápido para la provisión de QoS.

La redirección de tráfico resulta útil para evitar congestionar ciertas zonas de la red, de modo que existan varias fuentes dispersas geográficamente que sean elegidas por los clientes de forma dinámica. El equilibro de carga permite la utilización de varios servidores, en este caso geográficamente cercanos, entre los que se distribuyen las peticiones de los clientes. El objetivo de la aplicación de mecanismos de equilibro de carga es el incremento de la disponibilidad del servicio y la reducción de la carga por servidor.

En la figura 1 se muestra una evolución de algunas de las tecnologías implicadas en la provisión de garantías QoS en Internet.

3. Arquitectura propuesta para la implantación incremental de garantías QoS

3.1. Motivación

Desde la especificación de IntServ se ha intentado incorporar una arquitectura que permitiera ofrecer QoS sobre Internet, ya que es deseable que Internet se utilice como una infraestructura común para soportar tanto la comunicación en tiempo real como la que no es en tiempo real. Se podría construir una infraestructura totalmente nueva para los servicios en tiempo real, de forma paralela, dejando Internet tal y como está actualmente. Sin embargo, se perderían las ventajas significativas de la compartición estadística entre los tráficos en tiempo real y no tiempo real, además de convertirse en algo más complejo de construir y administrar que una infraestructura común.

El tamaño actual de Internet provoca que la modificación de su arquitectura para poder ofrecer QoS esté lejos de ser una tarea sencilla. Cada dominio de encaminamiento (conocido como *Autonomous System* o *AS*) tiene su propia configuración hardware y software sobre la que no siempre se puede actuar para incorporar extensiones a los protocolos soportados y/o configurados. Por diversos motivos, existirán organizaciones que, siendo responsables de uno o varios AS, decidan migrar su arquitectura hacia una arquitectura que pueda proporcionar QoS, mientras otras postpongan esta decisión.

El hecho de que no se realice una implantación completa de la nueva arquitectura en Internet provoca que no se puedan ofrecer garantías estrictas relativas a la obtención de QoS, aunque sí se proporcionen garantías estadísticas bajo ciertas restricciones.

Bajo el prisma de DiffServ, la situación no es necesariamente. Aunque los requerimientos en los elementos de la red para la implantación de una arquitectura DiffServ en un AS parecen inferiores a los de la implantación de una arquitectura IntServ, no dejan de existir unos requisitos mínimos que muchos AS actualmente no alcanzan.

Nuestra premisa de partida consiste en que el requisito de migrar completamente toda la arquitectura de Internet para poder ofrecer garantías de QoS a los usuarios resulta inaceptable. Las propuestas relacionadas con la arquitectura Internet2 QBone [6] todavía están lejos de llegar a los usuarios finales, que actualmente demandan de la red la obtención de QoS. Las aplicaciones se encuentran en este sentido por delante de los servicios que la red puede ofrecer.

Este trabajo está dirigido a la propuesta de una especificación que permita realizar una migración de la actual arquitectura de Internet hacia una arquitectura capaz de proveer QoS. La característica fundamental de esta arquitectura es la posibilidad de realizar una implantación *incremental*, de forma que puedan realizarse garantías QoS incluso si existieran nodos en la red que tan sólo proporcionaran un modelo de servicio

Tabla 1: Algunos mecanismos que garantizan o ayudan a la provisión de QoS en Internet

Nivel	Esquema de QoS	Mecanismos	Objetivo del esquema de QoS
Aplicación	Redirección de tráfico y equilibrio de carga	Redirección mediante URL, equilibro de carga	Redirigir el tráfico lejos de una parte congestionada de la red o un servidor
Transporte / Red	Diffserv	Clasificación, contador, marcador, adaptador, descartador, RED	Proporcionar servicios diferenciados para diferentes clases de tráfico, especialmente durante congestión en la red
Red	Ingeniería de tráfico	MPLS, encaminamiento basado en restricciones, señalización de rutas LSP y protocolos IGP de estado de enlace mejorados	Evitar congestión en la red
	Reencaminamiento rápido	Recuperación local	Evitar la pérdida de paquetes debido a fallo de enlaces o router.

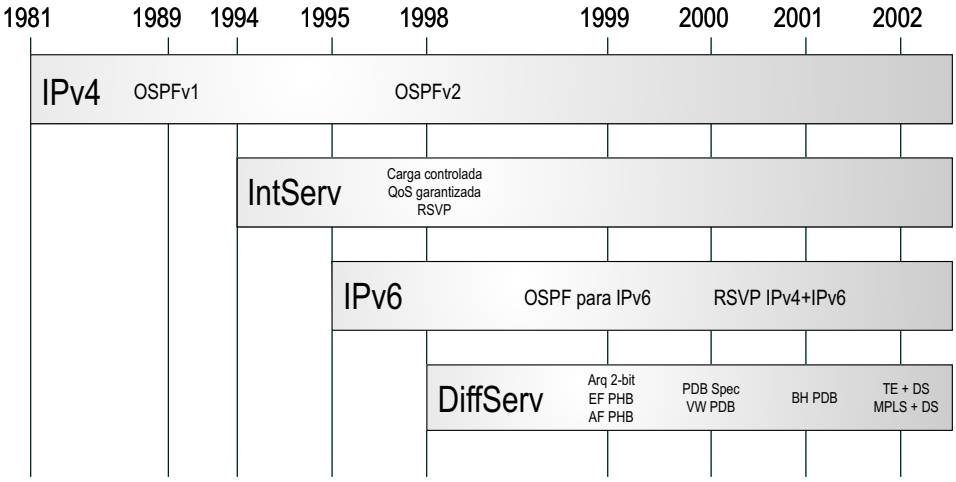


Figura 1: Evolución de algunas tecnologías implicadas en la provisión de QoS en Internet

best-effort. Esto da lugar a una evolución de la actual arquitectura de Internet, no una modificación completa (figura 2).

A continuación se exponen cuáles son los objetivos de este trabajo.

3.2. Alcance y objetivos

La arquitectura actual de Internet no ofrece ninguna garantía con respecto a los parámetros de QoS, salvo la entrega fiable. Partimos entonces de que cualquier modificación de la arquitectura que permita a Internet ofrecer una garantía mayor será considerado un avance, pero sólo si no repercute negativamente sobre el rendimiento actual de la red.

Por ello, en este trabajo se pretende:

- Incorporar mecanismos que permitan negociar la QoS obtenida desde la red. Si la red no puede ofrecer los requerimientos de QoS mínimos aceptables por el cliente, debe indicárselo rechazando la negociación [7].
- Que la incorporación de estos mecanismos tenga un impacto negativo mínimo sobre el rendimiento de la arquitectura actual. Los tráficos *best-effort* y QoS deben coexistir, no estar enfrentados.
- Que la incorporación de estos mecanismos tenga un impacto mínimo sobre cambios de configuración de la equipación actual, proporcionando mecanismos de configuración automática allí donde puedan estar disponibles.
- Que las pilas de protocolos *best-effort* no se vean modificadas en ninguno de los elementos de la red.

El fin último es proporcionar QoS a los clientes que lo soliciten y contraten en caso de que pueda encontrarse algún modo de proporcionar esta QoS. Si los parámetros de QoS son inaceptables, el cliente no se encontrará con un caso diferente al actual funcionamiento de Internet, que no realiza ninguna garantía con respecto a estos parámetros. Si es imposible garantizar los parámetros de QoS incluso con estos nuevos mecanismos,

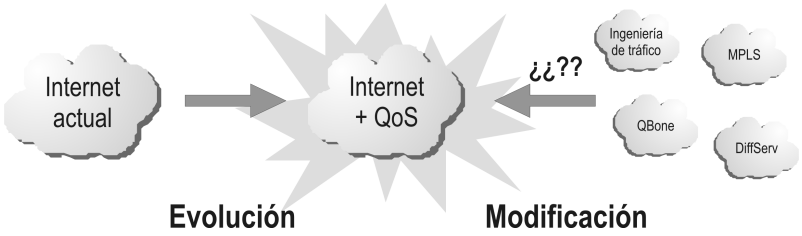


Figura 2: Evolución hacia una arquitectura que permita la provisión de QoS sobre Internet

tampoco se habrá llegado a una situación peor a la actual.

De este modo se aportarían ventajas a los siguientes agentes:

- Los usuarios se verán beneficiados de las ventajas que supone la posibilidad de contratar una conexión QoS.
- El ISP tiene garantizada una ventaja competitiva con respecto al resto de ISP que no ofrezcan un servicio QoS.
- Los *carrier* que oferten un servicio QoS acabarán manejando más volumen de tráfico, ya que a través de ellos seguirá circulando tráfico best-effort y adicionalmente, todo el tráfico de las conexiones QoS que los *carrier* de la competencia pierden por no ofertar QoS. Este incremento en el volumen de tráfico supondrá, de forma directa o indirecta [8], un aumento en sus beneficios.

Uno de los escenarios de clara aplicación de la propuesta puede encontrarse en los AS Multihoming, tanto en el caso de que el AS pertenezca a un cliente o a un proveedor. En este caso, el AS dispone de conectividad mediante varios proveedores. Si suponemos que alguno de estos proveedores oferta un servicio QoS, los equipos del AS Multihoming podrán negociar automáticamente el establecimiento de rutas QoS a través de los proveedores que dispongan de una arquitectura capaz de garantizar conexiones QoS. Para ello bastará con haber negociado un acuerdo de nivel de servicio (*Service Level Agreement o SLA*) entre el AS Multihoming y el AS proveedor que oferta QoS. Con respecto a las conexiones no-QoS, podrán encaminarse bien por los proveedores que ofertan QoS (ya que además mantienen el servicio *best-effort*) o bien por los que no (figura 3). De este modo, si cada AS que atraviesa un flujo de aplicación dispone de la posibilidad de encontrar una ruta QoS a través de alguno de sus proveedores, se podrá garantizar un servicio QoS incluso si no todos los nodos en la red ofertan un servicio QoS.

3.3. Arquitectura

3.3.1. Bandwidth Brokers (BB)

Todavía existen decisiones de diseño abiertas relacionadas con los BB, aunque podemos encontrar varios trabajos en curso [9, 10, 11, 12, 13, 14, 15] en los que se fijan algunas de ellas, llegando en algunos casos a desarrollos funcionales.

La figura de un agente BB como parte de un dominio DiffServ (*DiffServ Domain* o *DSD*) se introduce en [16], aunque para nuestra propuesta se requiere de la presencia de un agente en el contexto de un AS que realice tareas complementarias a las del BB. Decidimos incorporar esta nueva funcionalidad como parte del BB del AS, siendo la siguiente:

- **Control de los *router* del DSD mediante diferentes protocolos de señalización**, e incluso de mecanismos como Telnet/SSH allí donde los *router* no soportaran ningún protocolo de señalización específico (como COPS o RSVP). Este control está relacionado con el establecimiento de los parámetros requeridos por las diferentes clases de servicio. Equipos de encaminamiento diferentes pueden soportar protocolos de señalización diferentes para el establecimiento de estos parámetros y el BB debe poder controlar todos los dispositivos de encaminamiento del DSD.
- **Soporte de señalización directa RSVP para reserva de recursos**. Con objeto de permitir la interoperabilidad entre DSD y redes IntServ, los *ingress router* presentes en un DSD y conectados a redes IntServ señalizarán directamente a su BB las peticiones RSVP que reciben desde los *egress router* de la red IntServ (figura 4).
- **Monitorización de la información de encaminamiento realizado en el DSD**. Para ello, el BB debe comportarse como un elemento que recibe la información del protocolo de encaminamiento como si fuera un *router* más, pero sin embargo no propaga la información de encaminamiento computada al resto de los *router*.
- **Actuación directa sobre las restricciones QoSR**. Si, utilizando un algoritmo de encami-

[1]Los router son los responsables de calcular esta ruta, pero el BB también puede calcularla, ya que dispone de la información de encaminamiento que los router del AS le proporcionan. De este modo, el BB puede determinar si sobre esta ruta se satisfacen los parámetros de QoS solicitados.

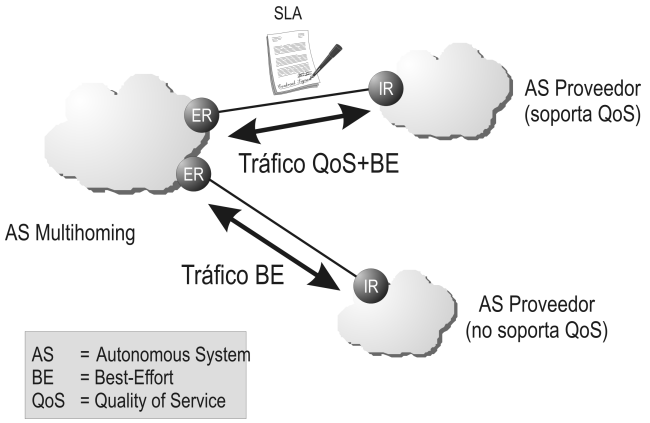


Figura 3: Un AS Multihoming puede enviar tráfico QoS a través del proveedor que lo soporta, y tráfico *best-effort* a través de cualquiera

namiento no-QoSR, la ruta calculada[1] para establecer un flujo *best-effort* no satisficiera los requerimientos para establecer un flujo QoS, el BB deberá actuar para que los *router* implicados utilicen una ruta diferente. Si se utiliza un algoritmo de encaminamiento QoSR, esta funcionalidad no es necesaria y el BB se limitará a la monitorización de la provisión global de QoS. Esto puede permitir la implantación de provisión de QoS en un AS incluso sin que todos sus elementos estén preparados para proporcionar garantías QoS.

En la figura 4 se muestra una perspectiva general de la relación de los BB con el resto de elementos de su entorno.

Los BB deben conocer la topología del AS en el que se encuentran, para lo que disponen de una base de datos de interfaces, además de la información de encaminamiento obtenida mediante el protocolo de encaminamiento. El conocer la topología permite a los BB, entre otras tareas, comunicarse directamente con los *router* del AS para controlarlos. Este control puede realizarse mediante diferentes mecanismos de señalización y control:

- RSVP [17], donde no se utilizan todos los mecanismos descritos en su especificación, sino tan sólo aquellos de señalización de reservas mediante mensajes RESV. A través de estos mensajes, el BB indica directamente al *router* correspondiente las características de las reservas para cada uno de los niveles de servicio en el contexto del propio *router*.
- COPS [18], un protocolo basado en TCP que utiliza un modelo cliente/servidor basado en los siguientes elementos:
  - El PEP (*Policy Enforcement Point*) es una entidad donde se aplican las políticas establecidas. Generalmente el PEP será un *router* o un *gateway* ubicado en una frontera del AS.
  - El PDP (*Policy Decision Point*) es el responsable de la obtención de las reglas a aplicar y de la generación de decisiones de acuerdo con las peticiones recibidas desde el PEP. El PDP actúa por tanto como servidor para el PEP. Por sus características, podría integrarse con el BB del AS.
- SNMP [19], incorporado en la mayoría de los *router*, puede permitir el control del *router* por parte del BB para establecer los parámetros de clasificación y planificación de paquetes en caso de que el *router* no soporte otros protocolos como RSVP o COPS.
- Telnet/SSH. Si el *router* no dispusiera de capacidad de ser controlado mediante COPS o RSVP y su agente SNMP no implementara el MIB correspondiente para el establecimiento de los parámetros de clasificación y planificación de paquetes, el BB todavía podría controlar el *router* mediante mecanismos Telnet/SSH.

En caso de utilizar un control basado en COPS, los *router* además deben conocer la localización del BB en el AS. Esta localización del BB desde los *router* puede estar preconfigurada (mediante SNMP o Telnet/SSH) o bien puede realizarse automáticamente mediante SLP[20].

Además, los BB deben localizar los BB de los DSD adyacentes. Como decisión de diseño abierta se proponen varias opciones:

- Incorporar la información de localización en la información distribuida mediante el protocolo BGP.
- Incorporar señalización entre *router ingress/egress* para que indiquen directamente la localización de los BB de sus respectivos AS.
- Utilizar el protocolo SLP, aunque para ello deben encontrarse disponibles capacidades de encaminamiento multicast en los AS adyacentes.

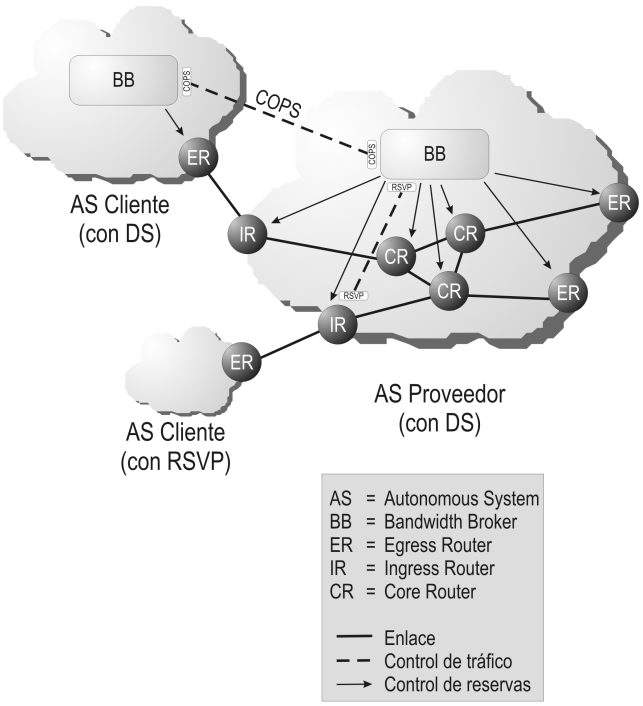


Figura 4: Relación entre BB de AS adyacentes

3.3.2. Arquitectura intra-DSD

De forma generalizada se acepta que el protocolo OSPF no es lo suficientemente flexible como para proporcionar un buen equilibro de carga. Precisamente esta es una de las razones para introducir tecnologías más flexibles, como MPLS, siendo esta una línea de trabajo abierta en la actualidad cuando se relaciona con la provisión de QoS en Internet.

Para el caso concreto de la utilización de OSPF, se proponen varias alternativas para los aspectos relacionados con el encaminamiento QoS en el contexto de un AS:

- Incorporación de extensiones QoSR al protocolo OSPF, mediante la propuesta de LSA opacos [21]. En este caso, los dispositivos de encaminamiento implicados en la provisión de QoS deben soportar las extensiones propuestas.
- Utilización de la capacidad de OSPF para la construcción de bases de datos de estado de enlace diferentes en función del campo TOS de IPv4. De este modo se sigue utilizando el encaminamiento *best-effort* sin necesidad de modificar el protocolo, y se pueden añadir métricas diferentes a los enlaces en función de parámetros QoS que se utilizarán para procesar los flujos QoS. En cualquier caso, debe realizarse un mapeo entre la clase de servicio asignada a un flujo y el campo TOS sintético aplicado para determinar qué base de datos de estado de enlace debe utilizarse.
- Actuación sobre las métricas de encaminamiento OSPF, mediante un agente externo que monitorice la correcta provisión de QoS conforme a las métricas establecidas en el contexto del DSD y actúe cuando estas métricas no permitan el establecimiento de una nueva conexión QoS. De este modo, en ciertas ocasiones, se provocaría la actualización de las bases de datos de estado de enlace cuando se requiera una nueva conexión. Para evitar un impacto excesivo sobre la red, se podrían utilizar mecanismos de histéresis que limiten estas actualizaciones.

3.3.3. Arquitectura inter-DSD

La arquitectura inter-DSD todavía no ha sido estudiada en profundidad, pero se parte de dos alternativas:

- Modificación de BGP con extensiones QoSR. Para determinar qué *router* disponen de capacidades QoS, se podría utilizar la propuesta [22].
- Señalización interDSD a través de los BB de los DSD adyacentes. De este modo son los propios BB los que acaban determinando los AS que cruzará el flujo QoS que está tratando de establecerse. Una vez que los BB han determinado los AS que atravesará el flujo, tan sólo tienen que indicar a los *router* de sus DSD respectivos cuál es el *egress router* que debe utilizarse.

4. Conclusiones y trabajos futuros

En este trabajo presentamos una propuesta de arquitectura para una implantación incremental de ga-

rantías de QoS sobre redes IP. Su diseño está orientado a permitir el establecimiento de garantías QoS para los flujos de aplicación incluso en el caso de que no todos los elementos de la red permitan la asignación de restricciones QoS.

Por un lado, en el intra-dominio, para el establecimiento de garantías QoS a los clientes basta con que exista al menos una ruta desde el *ingress router* hasta el *egress router* correspondiente de modo que los elementos de red atravesados permitan el establecimiento de mecanismos de clasificación y planificación de paquetes. El BB del dominio es el responsable de las actuaciones orientadas a que los paquetes que conforman el tráfico QoS sigan las rutas con restricciones QoS establecidas, así como del establecimiento de estas restricciones.

Por otro lado, en el inter-dominio, se pueden establecer garantías QoS si existe al menos una ruta en la que todos los AS atravesados disponen de capacidad de provisión de QoS. Puesto que todo AS que permita la provisión de garantías QoS debe disponer de un BB, esto permite que los BB de los AS adyacentes puedan intercambiar información para el establecimiento de rutas QoS a través de ellos, ignorando los AS sin capacidad de provisión de QoS.

Este escenario resulta especialmente atractivo para el caso de los clientes que disponen de conectividad mediante más de un proveedor (formando un AS Multihoming), que corresponde con más del 61 % de los AS en Internet. Si alguno de los proveedores dispone de capacidad de provisión QoS, en un AS Multihoming el tráfico QoS se puede dirigir a través de estos proveedores, mientras que el tráfico *best-effort* se puede dirigir a través de cualquier proveedor, disponga de capacidad para la provisión de QoS o no.

Actualmente estamos trabajando en el desarrollo de un prototipo que permita demostrar la potencial viabilidad de la propuesta. En este sentido se está trabajando sobre el desarrollo de un agente Bandwidth Broker sobre el sistema operativo Linux para el control automático de los parámetros de control de tráfico, inicialmente sobre nodos Linux, pero con una arquitectura que permitirá el control por parte del Bandwidth Broker de otros elementos de red.

Referencias

[1] R. Steinmetz y L. Wolf. Quality of Service: Where are We? En *IWQOS'97*, páginas 211–221. 1997.

[2] R. Braden, D. Clark y S. Shenker. Integrated Services in the Internet Architecture: an Overview. IETF RFC 1663, Julio 1994.

[3] K. Nichols, S. Bloake, F. Baker y D. Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 headers. IETF RFC 2474, Diciembre 1998.

[4] E. Rosen, A. Viswanathan y R. Callon. Multiprotocol Label Switching Architecture. IETF RFC 3031, Enero 2001.

[5] Xipeng Xiao. *Providing Quality of Service in the Internet*. Tesis Doctoral, Department of Computer Science and Engineering, Michigan State University, 2000.

[6] Benjamin Teitelbaum y Rüdiger Geib. Internet 2 QBone: A Test Bed for Differentiated Services. En *INET'99*. Junio 1999.

[7] E. Crawley, R. Nair, B. Jajagopalan y H. Sandick. A Framework for QoS-based Routing in the Internet. IETF RFC 2386, Agosto 1998.

[8] Geoff Huston. Interconnection, Peering, and Settlements. En *INET'99*. Junio 1999.

[9] British Columbia Institute of Technology. CA*net II Differentiated services: bandwidth Broker High Level Design, Noviembre 1998.

[10] M. Günter y T. Braun. Evaluation of Bandwidth Broker Signalling. En *Proc. of IEEE 7th International Conference on Network Protocols*, páginas 145–152. Octubre 1999.

[11] Ibrahim Khalil y Torsten Braun. Implementation of a Bandwidth Broker for Dynamic End-to-End Capacity Reservation over Multiple Diffserv Domains. *Lecture Notes in Computer Science*, 2216:160–??, 2001.

[12] B. Stiller, T. Braun, M. Günter y B. Plattner. The CATI project: charging and accounting technology for the Internet. En *Proc. of 4th European Conference: Multimedia Applications, Services and Techniques*, páginas 281–296. Mayo 1999.

[13] R. Rajan, D. Verma, S. Kamat, E. Felstaine y S. Herzog. A policy framework for Integrated and Differentiated Services in the Internet. *IEEE Network Magazine*, 13(5):35–41, Sep / Oct 1999.

[14] Andreas Terzis, Jun Ogawa, Sonia Tsui, Lan Wang y Lixia Zhang. A Prototype Implementation of the Two-Tier Architecture for Differentiated Services. *5th IEEE Real-Time Technology and Applications Symposium (RTAS99)*, Junio 1999.

[15] Jussi Lemponen. *Implementation of Differentiated Services Policy Information Base on Linux*. Proyecto Fin de Carrera, Tampere University of Technology. Department of Information Technology, Abril 2001.

[16] K. Nichols, V. Jacobson y L. Zhang. A Two-bit Differentiated Services Architecture for the Internet. IETF RFC 2638, Julio 1999.

[17] R. Braden, L. Zhang, S. Berson, S. Herzog y S. Jamin. Resource ReSerVation Protocol (RSVP). IETF RFC 2205, Septiembre 1997.

[18] D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan y A. Sastry. The COPS (Common Open Policy Service) Protocol. IETF RFC 2748, Enero 2000.

[19] F. Baker, K. Chan y A. Smith. Management Information Base for the Differentiated Services Architecture. IETF RFC 3289, Mayo 2002.

[20] E. Guttman, C. Perkins, J. Veizades y M. Day. Service Location Protocol, Version 2. IETF RFC 2608, Junio 1999.

[21] R. Coltun. The OSPF Opaque LSA Option. IETF RFC 2370, Julio 1998.

[22] R. Chandra y J. Scudder. Capabilities Advertisement with BGP-4. IETF RFC 3392, Noviembre 2002.

Servicio de medida de la calidad de servicio en Internet para usuarios y proveedores: Velocimetro.org

Eva Ibarrola, José María Perera, Armando Ferro, Alex Muñoz, Cristina Perfecto
Grupo de Ingeniería Telemática
Escuela Superior de Ingenieros de Bilbao. Alameda Urquijo s/n. 48013 Bilbao
Teléfono: 94 6013900 – Fax: 94 6014259
jtpibara@bi.ehu.es, jtbperij@aintel.bi.ehu.es, {jtpfevaa, jtpmumaa, jtppeamc}@bi.ehu.es

Abstract. This paper presents velocimetro.org, the service developed by the Engineering Faculty of Bilbao to provide access speed measurements in Internet. The explosion of Internet Service Providers (ISP) and different types of technologies in Internet access, makes it difficult for Internet users to decide about the best or more acquaint access to cope with their needs. Velocimetro.org offers the possibility to obtain a neutral measurement of the quality of service (QoS) of the access to Internet. In this way, users can evaluate and decide if their ISP is covering their demands or, on the contrary, is not complying with the terms of the contract. From the point of view of an ISP, velocimetro.org allows to make comparisons of their service against their competitors.

1 Introducción

El crecimiento de Internet en los últimos años junto con la proliferación de nuevas tecnologías de acceso ha provocado que los usuarios de Internet se planteen, cada vez más a menudo, cuál puede ser el proveedor de servicios o la tecnología idónea de cara a cubrir sus necesidades en el acceso a la red. Así, el usuario final es cada vez más exigente con la calidad de servicio (QoS) que recibe y es consciente del lugar que ocupa la selección del proveedor más adecuado, junto con la tecnología que mejor se adapta a sus necesidades.

Conocer si el ISP contratado cumple de forma adecuada con un determinado nivel de QoS, en términos de tiempo y velocidad de acceso a los servicios finales adquiere cada vez más relevancia. En este sentido, es importante qué sea el propio usuario final quien pueda realizar la medida de su velocidad ya que, de este modo, puede obtener datos reales del estado de su conexión en los instantes en los que observe anomalías con respecto al servicio contratado, es decir, en tiempo real. También es igualmente interesante que el usuario pueda conocer, de forma objetiva, la calidad que obtiene con su proveedor frente a la calidad de otros usuarios con las mismas características a fin de disponer de una referencia frente a la que poder valorar el servicio que se le presta.

Por otro lado, los ISPs conocen la numerosa competencia a la que se enfrentan y saben que su cuota de mercado va a depender, en gran medida y cada vez más, de la calidad del servicio ofertado a sus clientes. De la misma forma saben que la publicidad proveniente del ISP, acerca del buen servicio ofertado, no es un factor objetivo para los usuarios. Ellos, al igual que sus clientes, se verían beneficiados de un servicio de medidas desarrollado por una entidad neutral. Esto permitiría a los usuarios evaluar, en cualquier momento, la velocidad de su acceso a Internet y a los proveedores el servicio que ofertan comparado con el de su competencia.

En este artículo se presenta la arquitectura del servicio de medidas de calidad denominado velocímetro.org desarrollado por el Grupo de Ingeniería Telemática de la Escuela de Ingenieros de Bilbao en colaboración con la Universidad Politécnica de Madrid. Se trata de un servicio público de medidas para los usuarios de Internet que puede realizar estadísticas de uso y proporcionar gráficas comparativas de calidad tanto a usuarios como a proveedores.

2 Necesidad de un servicio de medidas

Dentro del panorama actual, en el ámbito de los accesos a Internet que ofrecen los ISPs, nos encontramos en una época en la que predominan los fuertes cambios y las novedades, sobre todo respecto a los accesos tradicionales como la conexión telefónica básica o las líneas RDSI.

Entre estas novedades, se pueden destacar tecnologías de acceso como ADSL o el cable-módem. Estas nuevas tecnologías consiguen unas velocidades mucho mayores que las tradicionales y proporcionan una conectividad permanente que supera, en la mayoría de los casos, velocidades de 256 Kbps. El usuario ya no sólo busca el acceso económico sino aquél que le garantiza la calidad de servicio requerida.

En este sentido, la Comisión para el Seguimiento de la Calidad en la prestación de los servicios de telecomunicaciones, creada el 21 de Diciembre de 1999 como órgano asesor del Ministerio de Ciencia y Tecnología, constituyó un grupo de trabajo específico para el seguimiento de la QoS en los servicios relacionados con Internet. De sus trabajos cabe destacar la elaboración de una propuesta de modelo de regulación sobre la calidad en Internet[1], en la que se especifica que "*la estimación de la calidad de servicio ofrecida deberán llevarla acabo agentes externos al proveedor, realizando un muestreo indicativo de los niveles de QoS proporcionados por los diferentes proveedores".*

Por otra parte, según los datos del Ministerio de Ciencia y Tecnología[2] ,el porcentaje de hogares del estado que dispone de conexión a Internet es del 29,5% mientras que entre las empresas se alcanza un 82,6%. El 14% de los usuarios disponen de ADSL mientras sólo el 4% utilizan el cable. Estos datos son similares, aunque algo inferiores, a los de otros países de la Unión Europea.

Los proveedores de acceso a Internet que proporcionan estas tecnologías, se ven limitados en muchas ocasiones por cuestiones ajenas a su propia infraestructura de red. A veces se ven obligados a contratar servicios a las operadoras de redes que cuentan con infraestructuras de red de mayor nivel que a su vez, normalmente, también ofrecen servicios de acceso a Internet.

En el caso concreto de ADSL, se trata de una tecnología que aprovecha el bucle local o bucle de abonado que, en algunos casos como España, es todavía propiedad en su mayor parte de las empresas que han tenido el monopolio durante muchos años.

En la actualidad, el bucle local se ofrece en régimen de alquiler a otros operadores y aunque algunos de ellos están instalando los suyos propios, todavía pasará mucho tiempo hasta que puedan llegar a todos los usuarios. Por lo tanto, aunque un proveedor de servicios de Internet cuente con unos accesos óptimos, siempre puede estar condicionado por otros factores externos, como por ejemplo la red alquilada, y desconocer realmente qué calidad final están ofreciendo a sus clientes.

Por otra parte, si nos ponemos en el lugar del cliente que tiene contratado uno de estos accesos, para éste puede resultar decepcionante observar la lentitud con la que se realiza la descarga de grandes archivos o simplemente la navegación ordinaria por la Web en determinados instantes. Es obvio, que para él es de gran interés conocer cuál es su velocidad de conexión a la red en ese momento. También puede ser interesante conocer cómo evoluciona la calidad de su acceso en el tiempo.

3 ¿Qué es velocímetro.org?

Velocímetro es un servicio de medidas de la calidad de servicio en el acceso a Internet que permite a un usuario poder estimar su velocidad de transferencia de información a las zonas más importantes de la red. Existen muchos servicios de medida que ofrecen a los usuarios este tipo de estadísticas[3]. Sin embargo velocímetro pretende llegar más allá:

- Se ha definido un método de medida específico que pretende proporcionar una estimación lo más adecuada posible a la propia percepción de un usuario del servicio.

- Se almacenan estadísticas comparativas de todas las medidas para poder proporcionar a los usuarios una referencia de su calidad frente a la calidad que puedan percibir otros usuarios de similares características y para esa misma franja horaria.

- Se permite a los usuarios guardar un perfil histórico de las medidas de calidad que realizan. De esta forma se les proporciona información de uso sobre franjas horarias, disponibilidad de servicio de su operador y rendimiento de su enlace a lo largo del tiempo. Pueden así conocer si su acceso experimenta problemas puntuales o bien si la buena o mala calidad percibida son constantes.

- Aprovechando las estadísticas recogidas de los usuarios, velocímetro es capaz de ofrecer servicios de diagnóstico y consultoría a proveedores de servicios de Internet.

- Muchos de los servicios que está ofreciendo velocímetro se hacen por suscripción de los usuarios de tal forma que se pueden guardar perfiles de comportamiento de los mismos para poder estimar el grado de confianza que puede proporcionar cada usuario que haga medidas.

- Velocímetro proporciona herramientas automatizadas para facilitar a los usuarios el lanzamiento de las pruebas de medida de calidad siguiendo un plan predeterminado. De esta forma los usuarios pueden programar sus pruebas y finalmente velocímetro les ofrecerá los resultados de una forma mucho más completa y compacta.

- Dado que velocímetro recoge muchas pruebas de múltiples usuarios de diferentes proveedores desde diferentes localizaciones con métodos de acceso variados, es posible proporcionar una información muy completa de la estimación de calidad para perfiles de comportamiento muy dispares.

- Esto permite a velocímetro ofrecer un mapa de calidad geográfico donde se puede analizar para diferente tipo de usuarios la calidad percibida en sus conexiones a diferentes partes de Internet.

4 Funcionalidades del sistema

A continuación se describen, de manera más detallada algunos de los servicios ya mencionados que es capaz de ofrecer una herramienta de este tipo a usuarios y proveedores de servicios de Internet y, en particular, www.velocimetro.org.

4.1 Servicios a Particulares

Desde el punto de vista del cliente de un ISP, mediante el servicio básico de velocímetro se le va a permitir estimar su velocidad de conexión, tanto en la descarga como en el envío de información. Finalizada cada prueba el usuario tendrá a su disposición:

- Las velocidades medias de su conexión para la descarga y subida de archivos así como los valores medios obtenidos para ese mismo tipo de acceso y en la misma franja horaria. Adicionalmente, a fin de facilitar la asimilación de la información se representará en % la velocidad del usuario frente a la media.

- Las velocidades detalladas de subida y bajada obtenidas para cada uno de los tres destinos inicialmente considerados y las medias de subidas y bajadas para cada uno de ellos obtenidas en las mismas condiciones anteriormente enunciadas junto con los valores en % obtenidos frente a los valores medios.

- Una tabla resumen en la que se recoge toda la información mencionada para que ésta se pueda consultar fácilmente.

Del mismo modo, se le informará de ciertos aspectos relevantes para él y para el sistema en sí mismo, como son:

- Posibilidad de automatizar el proceso de medida.

- Estimación del tipo de acceso que tiene contratado, en base a la velocidad alcanzada en la muestra inicial que se realiza al comienzo de las pruebas.

- Fecha y hora de realización de la prueba, número de pruebas realizadas hasta el momento, etc.

- Resultados de las pruebas realizadas por el usuario clasificados en función del tipo de conexión empleado, el servidor, la franja horaria, el sistema operativo, el navegador, etc.

- Comparativas con las medias obtenidas por el resto de usuarios del sistema.

- Evolución de sus resultados a lo largo del tiempo.

- Servicio de consulta de informes de resultados propios para usuarios y proveedores de servicios de Internet .

4.2 Servicios a ISPs

Desde el punto de vista de los ISPs, está herramienta es muy útil para conocer, con un cierto margen de error, cuál es la velocidad final que alcanzan sus clientes, y compararse con las velocidades que consiguen el resto de los proveedores.

Así mismo, se pueden extraer todo tipo de estadísticas en base a los datos que almacene el test en sus sucesivas operaciones, como por ejemplo una curva de velocidad según el tramo horario de conexión, tipo de conexión más utilizada por los usuarios de Internet, regiones geográficas donde se obtienen peores resultados, etc. Esta previsto facilitar este tipo de información a ISPs a modo de informes y bajo suscripción.

4.3 Servicios a Terceros

Por otra parte, la nueva versión de velocímetro implementa como servicio adicional la medida de la velocidad de descarga de páginas Web, independientes del sistema. Estas páginas podrán pertenecer a portales que ofrecen servicios muy demandados por los usuarios, como por ejemplo correo electrónico, banca, tiendas, periódicos, organismos públicos, etc.

Velocímetro.org permitirá que sus usuarios registrados configuren aquellos destinos contra los que desean medirse de entre una lista de destinos disponibles y en base a la información recogida se prevé poder ofrecer estadísticas y datos valiosos de cara a los administradores de estos sitios.

5 Acceso al Servicio

Para que el test pueda arrojar datos coherentes y elaborar estadísticas clasificadas por zonas, la plataforma que realiza la prueba se alberga en lugares estratégicos separados geográficamente y con una calidad en la conectividad garantizada, como son Estados Unidos, España y un país europeo. Desde estos nodos se remitirá la información obtenida para cada prueba al nodo central de Madrid albergado en Espanix[4], punto neutro para el intercambio del tráfico de los distintos ISPs nacionales.En relación con el modo de operación, y ya de forma más concreta, en la fig. 1 se muestra la pantalla de presentación del servicio y de inicio del test.

Para acceder al mismo basta con conectarse mediante un navegador convencional a la URL velocimetro.org.

Desde esta página se ofrece la posibilidad de que los usuarios se registren, lo que les permitirá acceder de forma totalmente gratuita a los servicios de medidas de valor añadido. Los usuarios no registrados, en cualquier caso, podrán realizar pruebas y acceder a los servicios básicos del sistema.

Los dos tipos de test que se ofrecen actualmente consisten en: la estimación de la velocidad del acceso del usuario y la estimación de la velocidad de descarga de páginas Web. En ambos casos, el test puede realizarse on line, mediante la interfaz que se ha presentado en la fig. 1, o bien mediante una aplicación cliente que puede descargarse desde la propia página. Esta aplicación además permite automatizar ambos procesos .

5.1 Test de Estimación de Velocidad de Usuario

En caso de que se solicite realizar el test de la velocidad de conexión del usuario, de forma transparente se realiza una prueba inicial mediante la toma de una muestra de corta duración que permite estimar el tipo de acceso utilizado para la prueba. Los resultados obtenidos con la muestra si bien, como se ha indicado, no se le muestran al usuario, son utilizados internamente para evaluar, en una primera aproximación, el tipo de conexión utilizado en el test y de esta forma poder configurar la realización de la prueba sobre los diferentes destinos, con los parámetros más adecuados, en cuanto a tamaño del fichero empleado para la descarga y subida, para ese acceso.

Durante el tiempo en el que transcurre esta primera medida, el test muestra un formulario en el que se solicitan datos de interés general para el posterior estudio y tratamiento de los resultados. Si el usuario se encuentra registrado en el sistema este paso no será necesario pero, en cualquier caso, siempre se le ofrecerá la posibilidad de modificar la información previamente almacenada en el sistema.

La muestra inicial permite además realizar una comprobación adicional: Protege al sistema frente a la selección errónea o fraudulenta de tipos de acceso que no se correspondan con los resultados obtenidos para la muestra. Así, un usuario con un acceso lento, por ejemplo módem a 56K, no podrá seleccionar como tipo de acceso empleado uno rápido, como pudiera ser una línea dedicada ocasionando en consecuencia una modificación a la baja de los valores de velocidad medios registrados hasta el momento para este último tipo de conexión. Y viceversa, no se podrán alterar al alza las velocidades medias de accesos teóricamente lentos indicando que se dispone de una conexión de estas características si esta afirmación no se ajusta a la realidad.

Si el tipo de conexión que elige el usuario no es acorde con los resultados obtenidos durante la prueba inicial, se le mostrará un mensaje informándole del error e instándole a seleccionar su tipo de conexión entre un listado de accesos cuya velocidad sea congruente con la velocidad estimada.

Si el usuario desconoce las características o el nombre del producto contratado, se le ofrecerá también la posibilidad de consultar un listado que recoge todas las modalidades de contrato de los diferentes ISPs del Estado a fin de que localice el suyo.

Figura 1– Ventana de inicio de la herramienta

Este proceso puede automatizarse programando la herramienta para que durante un período de tiempo definido por el usuario el sistema realice tests contra los servidores según la secuencia preestablecida por el mismo, simplificando, de este modo, notablemente las operaciones a realizar.

Como resultado del test, básico o automatizado, se obtendrá una medida de la de velocidad de descarga (fig.2) junto con los resultados medios anteriormente indicados. En una pantalla posterior de información detallada, podrá solicitar los resultados en subida y bajada para los diferentes destinos.

Si el usuario está registrado, podrá consultar información adicional, sobre cualquier prueba realizada con anterioridad. De esta forma podrá realizar comparativas entre sus resultados obtenidos a lo largo del tiempo y en función de los parámetros que para el son más significativos, como pueden ser tipo de conexión utilizada, franja horaria, etc. En la se muestra la presentación de resultados para un usuario registrado.

El sistema de base de datos está pensado para poder generar informes de resultados más completos a disposición de otras entidades interesadas o usuarios particulares a los que puede interesar, por ejemplo, conocer la evolución de su velocidad de acceso a Internet en el tiempo.

5.2 Test de Descarga de Páginas Web

Si la prueba realizada es de descarga de páginas Web, se procederá de forma similar, salvo que en este caso el sentido en el que se desarrollarán los tests será únicamente el de descarga.

Los usuarios podrán configurar los destinos que les interesen como por ejemplo portales de correo electrónico, bancos, tiendas, periódicos, organismos públicos, etc. Estas pruebas también podrán automatizarse y una vez concluidas el usuario podrá acceder a toda la información obtenida, de forma similar a la comentada con anterioridad.

6 Arquitectura del sistema

En este apartado se describirá la arquitectura del test de velocidad de transferencia de información en Internet para el servicio Web (fig. 3). Cabe destacar que la división modular que se va a realizar a continuación se basa, exclusivamente, en criterios de funcionalidad y en ningún momento se atiende a criterios físicos, ya que varios módulos funcionales pueden ser implementados en una o varias máquinas.

El sistema está compuesto por varios servidores. Por un lado se dispone de un servidor central cuya función principal es la de ofrecer los servicios de medidas a los usuarios, realizar las operaciones de mantenimiento, llevar el control del proceso de medida y generar los resultados adecuados. Se trata del gestor de los agentes de medidas, los cuales se encuentran en el resto de servidores del sistema separados geográficamente.

Para conocer cuál es la velocidad de conexión de la que disfruta un usuario se mide el tiempo necesario para transferir una cantidad determinada de información entre el equipo del usuario y los distintos agentes de medidas. A continuación se describen los módulos de que consta el sistema.

6.1 Interfaz de usuario

Es el medio a través del cual los usuarios del sistema acceden a los servicios de medidas. Consta básicamente de una aplicación Web a la que se le añaden funcionalidades mediante el desarrollo de una aplicación cliente (applet) que permite a los usuarios acceder a determinados servicios de valor añadido e incorpora las funciones necesarias para establecer conexiones con los diferentes módulos que componen el sistema.

Por otro lado, proporciona los recursos y protocolos necesarios para que se puedan garantizar los aspectos básicos de seguridad de la información especialmente sensible. La implementación de este servicio de seguridad se basa en el protocolo SSL.

Figura 2– Ventana inicial de resultados

6.2 Módulo de medidas

Cada uno de los servicios de medida de los que consta el sistema se modela en un nivel inferior dentro de este módulo. De esta forma se tiene un submódulo de medidas básico accesible mediante la interfaz Web.

Por otra parte, se dispone de otro submódulo que se encarga de calcular la velocidad de descarga de páginas Web de una entidad independiente del sistema. Por último, existe un tercer submódulo cuya función es la de realizar estos mismos procesos pero e forma totalmente automatizada y que además ofrece resultados de los resultados obtenidos en cada una de las fases parciales de las que constan las pruebas automaciadas.

6.3 Módulo de planificación de pruebas

Las pruebas de velocidad pueden ser planificadas. El usuario a través de la interfaz adecuada seleccionará los períodos durante los cuales desea realizar pruebas automatizadas y el número y el orden de los servidores que intervendrán en dichas pruebas o, por el contrario, las páginas Web cuya velocidad de descarga quiere conocer.

6.4 Módulo de gestión remota

Este módulo permite al administrador controlar el estado en el que se encuentra el sistema y actuar en consecuencia. Detecta posibles situaciones de pérdida de rendimiento o fallos en las conexiones entre los servidores que componen el sistema. Permite la configuración del sistema, y la monitorización de los procesos que tienen lugar en cada momento.

Las funciones que lleva a cabo este módulo son:

- Configuración de parámetros.
- Control del estado de los servidores.
- Control del estado de las bases de datos.
- Registro de incidencias.
- Monitorización de pruebas.
- Gestión de proveedores.

6.5 Módulo de bases de datos

Almacén de la mayor parte de la información del sistema, tanto la relativa a usuarios como los datos de mediciones realizadas.

Este módulo está compuesto por el sistema gestor de la base de datos que manipula la información almacenada, y por las bases de datos cuya estructura se basa en un conjunto de tablas que siguen el modelo relacional.

Las bases de datos (BD) del sistema requieren un mantenimiento mínimo: La información almacenada en las BD debe ser filtrada previamente para eliminar muestras erróneas que puedan corromper otras muestras válidas.

6.6 Módulo de gestión de usuarios

Debido a que la información de velocidades obtenida corresponde a un usuario, el sistema debe encargarse de gestionar las cuentas de usuarios (procesos de alta, baja y modificación) así como controlar el acceso de los usuarios a los recursos y servicios autorizados. Algunos servicios serán prestados a usuarios no registrados, por lo cual, otra tarea de este módulo es la identificación de las pruebas realizadas por usuarios no registrados. El método empleado es asignar a cada usuario un identificador que se almacena en una cookie, siendo la gestión de éstas otra de las funciones del módulo.

Toda la información que se transfiera entre los extremos, referente a este tipo de operaciones, estará protegida convenientemente con el protocolo SSL.

6.7 Módulo de resultados

La función principal de este módulo es realizar los cálculos estadísticos necesarios para generar las tablas y gráficos que, de una forma sencilla, permitan presentar los resultados y datos más relevantes.

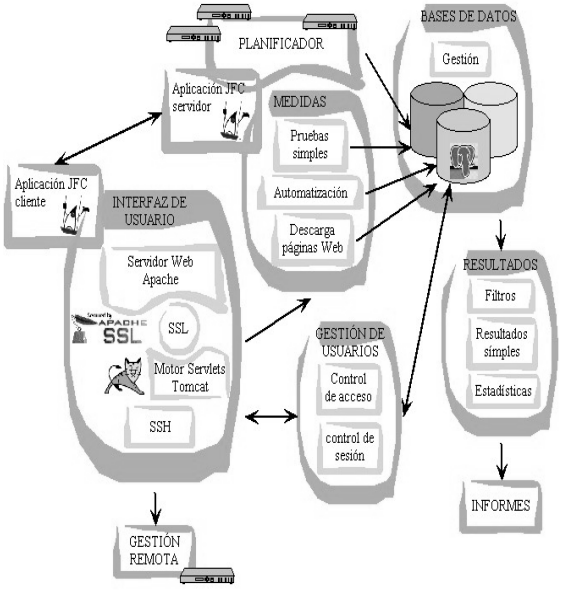


Figura 3- Arquitectura del sistema.

Adicionalmente, es el encargado de calcular la evolución temporal de las pruebas de los usuarios registrados así como de generar resúmenes de los resultados de ISPs y permitir la obtención de comparativas entre proveedores en base a múltiples parámetros.

6.8 Módulo de generación de informes

Este módulo se encarga de adecuar el formato de los resultados de las pruebas para su visualización, descarga, o publicación. Además, desarrolla las tareas requeridas para la generación de resúmenes de comparativas de proveedores y tipos de conexión. Por otra parte, ofrece la posibilidad de imprimir los resultados visualizados por el módulo de resultados, aplicándoles un formato adecuado para la impresión. Para llevar a cabo estas funciones este módulo obtiene la información necesaria del módulo de resultados. Sin embargo, estas tareas pueden consumir los recursos de los equipos en los que se encuentra instalado el sistema por lo cual se aplican distintas estrategias que aligeran estos procesos.

Para el desarrollo de los módulos funcionales descritos en este apartado se han utilizado los componentes SW que se describen a continuación:

En el servidor central del sistema se ha instalado el sistema operativo Solaris 8 de Sun Microsystems, el servidor Web seguro Apache 1.3.16 y el motor de Servlets y JSP Tomcat 4.1.12. Los elementos instalados en el resto de los servidores que componen el sistema son similares a los descritos anteriormente. Para el almacenamiento de la información recogida por el sistema se ha utilizado el sistema gestor de bases de datos relacionales PostgreSQL 7.2.3. Por último, para la elaboración de los informes de resultados se ha optado por emplear XML (eXtensible Markup Language) debido a que sus características se ajustan a las necesidades del sistema.

7 Resultados

Las pruebas realizadas durante el período comprendido entre abril de 2002 y diciembre de 2002 han permitido obtener resultados de diferentes tecnologías e ISPs que compiten en el mercado español de los accesos a Internet utilizando para ello la herramienta descrita.

En las fig. 4 y 5 se muestran los resultados medios obtenidos para dos de las tecnologías de mayor implantación como son ADSL y cable-módem. Se han omitido los nombres de los proveedores para mantener la confidencialidad de los resultados.

Los proveedores de ADSL de 256 Kbps en el enlace de bajada han alcanzado valores medios inferiores a 180 Kbps, mientras que en el enlace de subida los valores registrados presentan mayor disparidad y no superan los 90 Kbps, salvo alguna excepción. También se aprecia una disminución progresiva en los valores obtenidos para la descarga de datos, esto hace indicar que se está produciendo una sobrecarga de los recursos que estas empresas disponen para proporcionar este servicio. El número de pruebas válidas de este tipo contabilizadas ha sido de 107.768.

En relación a los proveedores de cable, se observa que una tecnología que en teoría se suponía simétrica resulta en la práctica asimétrica. Los valores obtenidos para la descarga de datos no superan la cifra de 190 Kbps y los valores obtenidos para el envío de datos apenas llegan a 90 Kbps. La excepción en este caso es el proveedor 3 cuyos datos reflejan que sí aplica la simetría en su servicio. Por otro lado, aquí también se observa, aunque en menor medida, una disminución paulatina de los resultados que podría suponer una saturación de los recursos de estos proveedores. En este caso el número de pruebas válidas contabilizadas ha sido de 78.732.

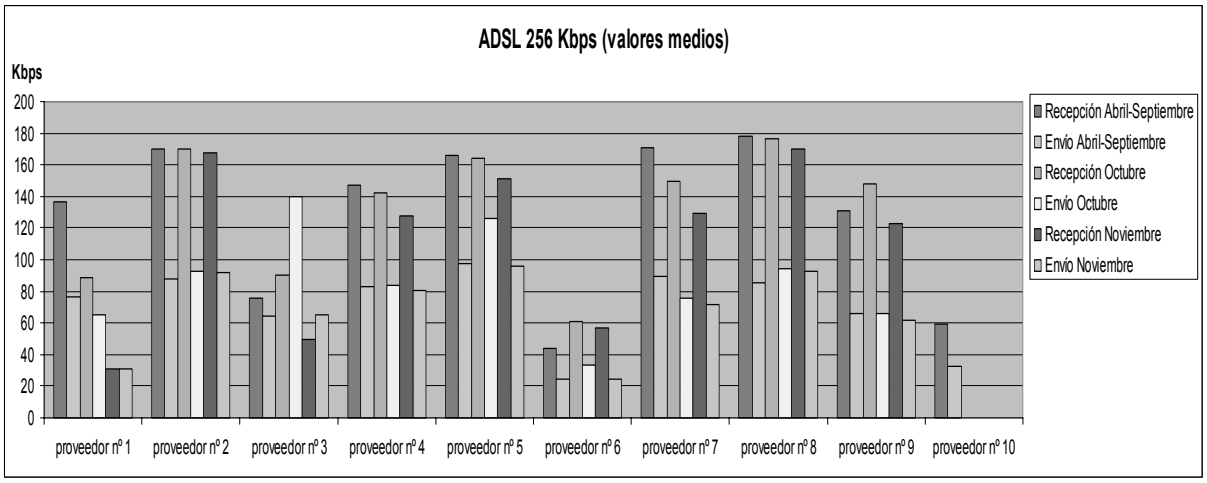


Figura 4– Resultados para ADSL 256 Kbps

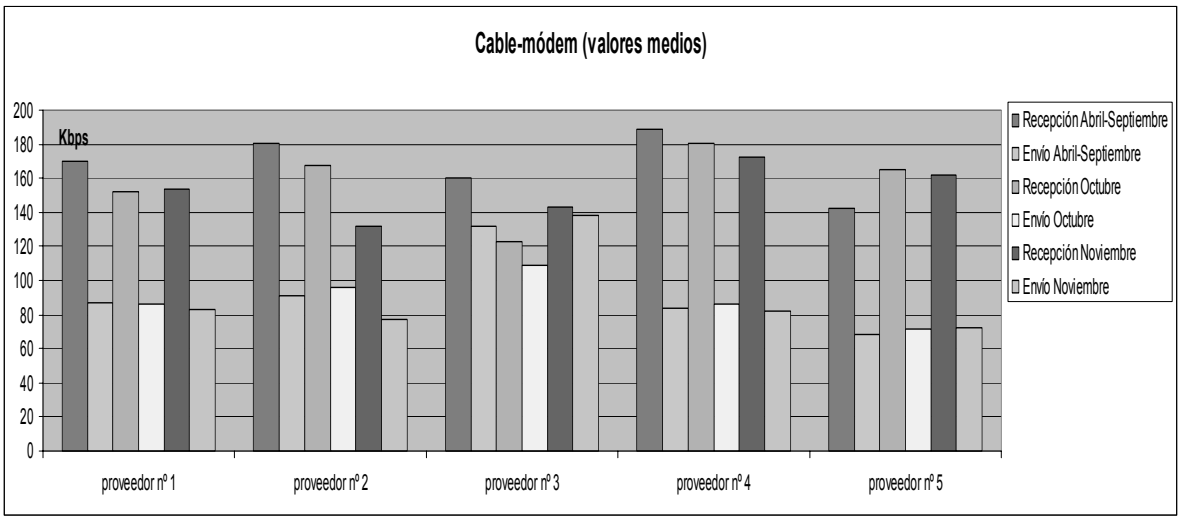


Figura5– Resultados para cable-módem

8 Conclusiones

El sistema de medidas implementado proporciona una serie de beneficios tanto para los usuarios finales de Internet, como para ISPs y empresas vinculadas con el servicio Web.

En el caso de los usuarios, este sistema les permitirá realizar, ellos mismos, y en el momento que consideren más oportuno, sus propias medidas para la estimación de su velocidad de acceso a Internet. Los resultados obtenidos junto con las comparativas que el sistema les proporciona, les llevará a una serie de conclusiones posibilitándoles tomar decisiones congruentes respecto a la QoS de su acceso y la idoneidad del mismo para los fines que persiguen. Teniendo en cuenta que los datos han sido obtenidos de una entidad neutral, podrán incluso exigir a su proveedor la mejora del servicio proporcionado en base a los resultados obtenidos con el sistema de medidas.

Por otro lado, los ISPs necesitan, cada vez más, datos comparativos frente a sus competidores ya que la QoS ha pasado a ser el factor diferenciador entre todos ellos. Además, saben que para que los datos tengan validez deben obtenerse utilizando criterios de análisis comunes y herramientas avaladas por entidades neutrales que garanticen que los resultados obtenidos son objetivos e independientes. El sistema de medidas aquí presentado les proporciona todo eso, permitiéndoles, además, detectar posibles problemas en el servicio de red o en su infraestructura.

Por último, las empresas que desarrollan parte de su actividad en entornos Web, como pueden ser empresas de compra online, podrán conocer mediante las pruebas específicas de portales, la calidad del acceso a su página Web, así como datos de utilización de la misma por parte de los usuarios.

Cabe destacar que, desde el punto de vista de investigación, los datos obtenidos por el servicio www.velocimetro.org, están sirviendo de base para el desarrollo y validación de varios proyectos relacionados con la QoS en redes de datos en los que se encuentra trabajando actualmente el Grupo de Ingeniería Telemática de la Escuela Superior de Ingenieros de Bilbao

Referencias

[1] *Comisión para el Seguimiento de la Calidad en la Prestación de los Servicios de Telecomunicaciones. Propuesta de bases para la elaboración de un modelo de regulación sobre calidad de servicio en la prestación de servicios relacionados con Internet. 21 Junio 2002*

[2] *Ministerio de Ciencia y Tecnología.* Indicadores de la Sociedad de la Información en España y varios países de la OCDE (1995-2002), http://www6.mcyt.es/indicadores/

[3] *Otros test de velocidad:*

http://www.bandwidthplace.com/speedtest
http://extranet.iies.es/CGI/speed.cgi
http://www.telefonica-data.es
http://www.gibroadband.com/pages/speedtest.asp
http://www.toast.net/perfomance/
http://www.cfl.rr.com/speedtest/
http://www.beelinesoftware.nl/bandwidth/
http://www.dslreports.com/stest?loc=1
http://homepage.tinet.ie/~leslie/testpage

[4] Punto Neutro español de Internet
http://www.espanix.net

Sesión 4B

Seguridad en redes de comunicación

Políticas de revocación para la protección de agentes móviles

Oscar Esparza Miguel Soriano
Universitat Politècnica de Catalunya
{oscar.esparza, soriano}@entel.upc.es

Abstract *Mobile agents are software entities that consist of code, data and state, and that can migrate autonomously from host to host executing their code. Code mobility is a powerful tool in distributed systems that can perform automatically functions related to data mining, electronic commerce or network management. Despite its benefits, security issues restrict the use of code mobility. The protection of mobile agents from the attacks of malicious hosts is considered by far the most difficult to solve security problem in mobile agent systems.*

The approach that is presented here aids to solve the problem of malicious hosts by using a Trusted Third Party, the Host Revocation Authority (HoRA). The HoRA was previously introduced in [4], and its main task is controlling which are the hosts that acted maliciously in the past, and as a result were revoked. This paper introduces two new protocols that can be used to revoke malicious host if some proofs of its malicious behaviour can be found.

1. Introducción

Un agente móvil es una entidad software compuesta básicamente por código, datos y estado, y con capacidad para migrar de host a host ejecutando dicho código mientras realiza ciertas acciones en nombre de un usuario. El ultimo host que ejecuta el agente manda los resultados (si los hay) al host origen. Si lo comparamos con los sistemas distribuidos habituales basados en traspaso de mensajes, el uso de agentes móviles ahorra ancho de banda y permite la ejecución autónoma de los mismos. Los agentes móviles son especialmente útiles para automatizar tareas en casi todos los servicios electrónicos, como son el comercio electrónico, la gestión de red o la búsqueda exhaustiva de contenidos por Internet. A pesar de los beneficios que comportan los agentes móviles, su uso masivo está restringido por razones de seguridad.

Consideraremos que en este escenario disponemos de dos entidades principales, el agente y el entorno de ejecución (o host). La protección de dichas entidades es necesaria cuando las relaciones de confianza entre ellas no pueden asegurarse. Estos son los casos que podemos encontrarnos:

- El agente ataca al host: es posible proteger a un host de los ataques de un agente malicioso mediante técnicas que limiten el entorno de ejecución de tipo sand-box y un control de acceso adecuado.
- Seguridad de las comunicaciones: es posible proteger un agente mientras migra de un host a otro utilizando técnicas criptográficas bien conocidas o protocolos como TLS.
- El host ataca al agente: no ha sido publicada ninguna propuesta que consiga dar una protección total a un agente móvil de los ataques del host que lo ejecuta. Este tipo de ataques son conocidos como el problema de los hosts maliciosos.

Este artículo introduce una nueva propuesta que ayuda a resolver el problema de los hosts maliciosos utilizando una autoridad de revocación de hosts o HoRA (Host Revocation Authority). La HoRA debe considerarse como una tercera parte de confianza (TTP) en un sistema de agentes móviles, de la misma manera que se considera una entidad de confianza a la autoridad de certificación en la infraestructura de clave pública (PKI). La HoRA fue introducida previamente en [4], y se dedica a almacenar los identificadores de aquellos hosts que se ha probado que actuaron de forma maliciosa y por esa razón han sido revocados. Antes de enviar un agente, cada host origen debe consultar la información de revocación (1) consultando directamente a la HoRA, o bien (2) consultando una lista de hosts revocados local que ha sido previamente descargada de la HoRA. Los hosts revocados serán eliminados del itinerario, de manera que ya no recibirán más agentes. Además, un host origen puede intentar revocar un host si demuestra que actuó maliciosamente. En ese sentido, son necesarias técnicas de detección y prueba de ataques [12, 3]. Este artículo presenta dos nuevos protocolos para revocar hosts maliciosos.

El resto del artículo está organizado de la siguiente forma: la Sección 2 presenta el problema de los hosts maliciosos; la Sección 3 describe las principales propuestas publicadas para la protección de agentes móviles; la Sección 4 detalla cuáles son las principales políticas de revocación de hosts, así como dos nuevos protocolos para revocar hosts maliciosos. Finalmente, las conclusiones y las lineas futuras se encuentran en la Sección 5.

2. Hosts Maliciosos

El problema de los hosts maliciosos es con diferencia el más difícil de resolver respecto a seguridad en entornos de agentes móviles. Los hosts maliciosos pueden intentar sacar provecho del agente modificando el código, los datos, las comunicaciones o incluso los resultados, ya que tienen control total sobre la ejecución del agente. El agente no puede transportar una clave de descifrado ya que los hosts podrían leerla. Además, nadie asegura que los hosts ejecuten el código de forma correcta o hasta la finalización, o simplemente no permitan la migración del agente hacia otros hosts.

Estos son los principales ataques que un host malicioso puede realizar:

- Negación de servicio: es imposible evitar los ataques de negación de servicio pues el host tiene control total sobre la ejecución. Solamente es posible detectar este tipo de ataques y castigar al host que los realiza. También se consideran ataques de negación de servicio a los cambios aleatorios en el código, los datos o los resultados que pueda realizar un host malicioso, ya que pueden llevar al agente a comportarse de forma inesperada.

- Repudio: un host realiza un ataque de repudio cuando niega una acción que sí tuvo lugar. Este tipo de ataques se resuelven mediante el uso de técnicas de firma digital en el intercambio de cualquier información entre entidades.

- Escuchas: los agentes normalmente se ejecutan en claro, con lo cual partes esenciales como el código, los datos o las comunicaciones deben estar disponibles para la plataforma de ejecución, y por tanto no se puede asegurar su confidencialidad. De hecho, cualquier dato no cifrado podrá ser leído por el host que ejecuta el agente.

- Manipulación: es posible asegurar la integridad y la autenticidad del código, datos o resultados que provienen de otros hosts usando técnicas de cifrado o firma digital, sin embargo es difícil detectar o prevenir ataques realizados por un host malicioso durante la ejecución, esto es, integridad de la ejecución.

- Confabulación: en un sistema abierto como Internet las relaciones de confianza son limitadas. Por ello se podría suponer que es difícil que un grupo de hosts confabulen contra un agente porque (1) deben estar en el itinerario del mismo agente y (2) deben confiar entre ellos. Sin embargo, la posibilidad de confabulación se dificulta pero no desaparece. Los ataques de este tipo son difíciles de detectar o prevenir, y esa es la razón por la cual la mayoría de propuestas publicadas son vulnerables a ellos [10, 9].

3. Propuestas Actuales

3.1. Propuestas de Detección de Ataques

En la opinión de los autores de este artículo, un esquema de protección de agentes basado sólo en métodos de detección de ataques es claramente insuficiente. Son necesarios mecanismos de castigo para aquellos hosts que actúan maliciosamente, y para ello es necesaria una TTP que imparta dichos castigos. En este sentido, la HoRA pretende resolver dicha carencia en los sistemas de agentes móviles. A continuación se enumeran las principales propuestas de detección de ataques.

En [7], Minsky et al. introducen la idea de replicación y voto. En cada etapa un grupo de hosts ejecutan el mismo agente de forma paralela y envían un conjunto de réplicas del mismo hacia la siguiente etapa. En algunas de estas etapas los hosts comparan los resultados, de manera que se toman como correctos los resultados obtenidos por la mayoría. Esta propuesta ofrece un mecanismo tolerante a errores, y además permite detectar actitudes maliciosas por parte de los hosts. La propuesta tiene sin embargo dos principales inconvenientes: (1) La replicación de agentes malgasta recursos; (2) Se asume que los resultados de todas las réplicas deben ser los mismos si todos los hosts actúan de forma honesta, con lo cual todos los hosts de una misma etapa deben tener los mismos recursos y datos. Esto sin embargo no concuerda con la propiedad de independencia de los hosts.

En [12], Vigna introduce la idea de las trazas criptográficas. Mientras se ejecuta, el agente toma trazas de las instrucciones que alteran el estado del agente debido a variables externas. Cada host que ejecuta el agente guarda las trazas durante un cierto tiempo y envía al host origen un hash de las mismas como prueba. Las trazas sólo se envían en caso de sospecha, ya que su tamaño a priori será demasiado grande. Si el host origen sospecha de un cierto host y quiere verificar su ejecución, pide que se le envíen las trazas. El host origen ejecuta el agente de nuevo y se asegura que la nueva ejecución concuerda con las trazas enviadas. Si no concuerdan es que ese host ha alterado el agente y por tanto es malicioso. Esta propuesta no sólo detecta los ataques, sino que también aporta pruebas del comportamiento malicioso de un cierto host. Sin embargo, tiene los siguientes inconvenientes: (1) La verificación se realiza en caso de sospecha, pero no se indica cómo un host llega a ser sospechoso; (2) Cada host debe almacenar las trazas durante un periodo de tiempo indefinido ya que el host origen puede solicitarlas.

En [3], los autores presentan un protocolo para detectar hosts sospechosos basado en limitar el tiempo de ejecución del agente. Cada host debe

guardar el tiempo de llegada y finalización de la ejecución del agente, de manera que cuando el agente vuelve al host origen se verifica si alguno de dichos hosts ha tardado más tiempo del esperado, y por tanto es sospechoso de ser malicioso.

3.2. Propuestas de Prevención de Ataques

Las técnicas de detección no son útiles en aquellos servicios donde los beneficios obtenidos manipulando un agente pueden ser mayores que el castigo que pueda imponerse. En dichos casos sólo son útiles las técnicas de prevención de ataques. Desafortunadamente no se ha publicado ninguna propuesta que consiga evitar los ataques de un host malicioso completamente.

En [13], Yee introduce la idea de un "Santuario" o subsistema cerrado donde se ejecutan de forma segura los agentes y al cual no tiene acceso ni el propio dueño de la plataforma. Esta solución obligaría a cada host con capacidad para ejecutar agentes a adquirir un equipo hardware. Además se debería considerar al suministrador del hardware como una entidad de confianza.

En [10], Roth presenta la idea de la protección mutua. En la opinión del autor, en un entorno hostil como Internet, en el cual las relaciones de confianza son limitadas, es posible suponer que es difícil que varios hosts confabulen para actuar en contra de un agente ya que difícilmente confiaran el uno en el otro. Por ello, el autor propone el uso de agentes cooperativos que comparten secretos y decisiones. El almacenamiento de resultados confidenciales y la toma de decisiones se realizan en el agente cooperativo, que además viaja por una ruta disjunta. Sin embargo esta propuesta tiene el inconveniente que la pérdida del agente cooperativo implica la pérdida de los resultados. Además, la posibilidad de confabulación entre hosts maliciosos no se elimina por completo.

En la propuesta de generación de claves de entorno (Environmental Key Generation) [9], el agente "despistado" (clueless) busca una clave para descifrar su código y para ello monitoriza el entorno. El código sólo se puede descifrar si se dan las condiciones de contorno adecuadas. Con esta medida evitamos que la plataforma pueda analizar el código antes de ejecutarlo. El problema del esquema estriba en que una vez descifrado el código, el host malicioso dispone de él para modificarlo a su conveniencia. Además obliga a monitorizar continuamente el entorno.

Una Blackbox es un entorno software del cual sólo podemos tener conocimiento de las entradas y salidas, y tanto el código como los datos internos no pueden ser leídos ni modificados en ningún momento. Desgraciadamente no se conoce hasta el momento ningún algoritmo o función que ofrezca una protección de este tipo. Sí es posible implementar una Blackbox limitada en tiempo [5]. Por tanto, se puede asumir que se protege al agente durante un cierto tiempo, después del cual ya no se asegura ni la confidencialidad ni la integridad de la ejecución. La propuesta está basada en la ofuscación del código como método para dificultar la lectura y por tanto la comprensión del código. Sin embargo, es difícil realizar una estimación del tiempo para el cual se asegura la confidencialidad del código, ya que depende de la capacidad que tenga el host malicioso de analizar el código ofuscado. Además, la longitud del código ofuscado es sustancialmente mayor que la del código normal.

El uso de programas cifrados [11] ha sido propuesto como el único modo de dar confidencialidad e integridad al código móvil. Los hosts ejecutan directamente el código cifrado, con lo cual no pueden extraer ninguna información de él, y por tanto no es posible modificarlo a su favor. Para recuperar los resultados es necesaria una función de descifrado. En [2] se mejora la propuesta haciendo que el agente pueda atravesar múltiples hosts. Más tarde en [1] el esquema permite la toma de decisiones mientras el agente viaja usando una TTP. El principal inconveniente de estas propuestas es la dificultad de encontrar funciones que puedan ejecutarse directamente cifradas

4. Políticas de Revocación de Hosts

En [4], se introdujo una nueva entidad en el sistema de agentes móviles, la autoridad de revocación de hosts o HoRA. La HoRA controla qué hosts han sido revocados en el sistema de agentes móviles, esto es, los hosts que se ha probado que actuaron maliciosamente. La HoRA debe considerarse una TTP en el sistema de agentes móviles, de la misma forma que la autoridad de certificación es considerada una TTP en la PKI. En este sentido, la HoRA no realiza tareas de autenticación sino de autorización, ya que los hosts sólo ejecutarán agentes si no han sido revocados. Otras propuestas han considerado necesario insertar una TTP en el sistema de agentes [1, 12, 13], pero en ninguna de ellas dicha TTP se utilizaba como revocador de hosts maliciosos.

La revocación de hosts [4] no puede considerarse ni una propuesta de detección ni de prevención de ataques, sino una mezcla de ambas. El primer ataque realizado por un host no puede ser evitado, sin embargo, si el host origen prueba que dicho host atacó al agente, este host puede ser revocado y por tanto sus ataques podrán evitarse, ya que ningún otro host le enviará agentes.

Si asumimos que la HoRA funciona de una manera similar a la autoridad de certificación respecto revocación de certificados, dos políticas de revocación de hosts pueden seguirse.

4.1. Política de Revocación Off-line

Está basada en la distribución de la información de revocación usando una lista de hosts revocados o HRL (Host Revocation List) firmada por la HoRA. Todos los hosts origen deben descargarse una copia de la HRL para consultarla antes de ejecutar el agente y eliminar los hosts revocados del itinerario. Los hosts origen también deben actualizar la lista periódicamente para tener en cuenta los nuevos hosts maliciosos. En ese sentido, la HRL funciona de una manera similar a la CRL [6] de la PKI.

4.2. Política de Revocación On-line

Antes de enviar el agente móvil, cada host consulta a la HoRA si hay hosts revocados en el itinerario del agente. La HoRA consulta internamente la lista de hosts revocados y envía una respuesta firmada al host origen indicando cuales son los revocados. Este mecanismo funciona de una manera similar al protocolo OCSP(Online Certificate Status Protocol) [8] usado en la PKI.

4.3. Funcionamiento de la HoRA

La HoRA realiza una serie de tareas de forma independiente a la política de revocación:

- Almacena y realiza el mantenimiento de la lista de hosts revocados.
- Revoca aquellos hosts que se demuestra que han actuado de forma maliciosa.

Si bien, también realiza tareas que sí dependen de la política de revocación utilizada:

- Política Off-line: genera la HRL y la distribuye para que pueda ser descargada.
- Política On-line: recibe peticiones de los hosts origen y manda respuestas firmadas a los mismos.

4.3.1. Mantenimiento de la Información de Revocación

El objetivo de la revocación de hosts es distinguir los hosts malicioso de los honestos. Desgraciadamente, no es posible adivinar si un host honesto se volverá malicioso en la transacción actual. Sin embargo, sí es posible conocer cuales son los hosts que actuaron de forma maliciosa en el pasado.

La HoRA sabe qué hosts fueron maliciosos porque guarda sus identificadores en una lista. Los identificadores de hosts deben ser únicos e inequívocos en el sistema de agentes móviles, por ejemplo, direcciones IP o nombres DNS. La HoRA almacenará una lista ordenada alfabéticamente en caso que los identificadores de hosts estén compuestos por caracteres, como un nombre DNS, o bien almacenará una lista ordenada numéricamente en caso de que los identificadores de hosts sean números, como direcciones IP. Esta forma de almacenar los identificadores de hosts revocados también es útil si se utiliza la política de revocación off-line, ya que la HRL no es más que una copia firmada de esta lista interna.

4.3.2. Revocación de Hosts Maliciosos

Antes que un host origen empiece el proceso de revocación de un host, debe haber usado uno de los mecanismos de detección y prueba de ataques. Como la propuesta de las trazas criptográficas [12] es la más conocida de este tipo, será la que utilicemos en nuestro esquema. La propuesta de Vigna tiene dos inconvenientes principales: (1) Cómo detectar a los hosts sospechosos; (2) Cada host origen debe almacenar las trazas por un periodo indefinido. Estos inconvenientes pueden solucionarse usando técnicas de detección de sospechosos [3] como se mencionó en la sección 3.1. Utilizando ambos mecanismos de forma conjunta es posible detectar hosts sospechosos cuando el agente vuelve al host origen, de manera que es posible pedirles que envíen sus trazas para verificar la integridad de la ejecución. En este artículo se presentan dos nuevos protocolos para revocar hosts maliciosos que utilizan de forma conjunta ambos mecanismos.

4.3.3. Generación de la HRL

Esta tarea sólo debe realizarse en caso que la política de revocación sea off-line. La HRL no es más que una lista ordenada de identificadores de hosts revocados firmada por la HoRA. La HRL debe estar disponible para que los hosts origen la descarguen y pueda consultarla. Como es una lista firmada, puede ser descargada desde repositorios que no sean de confianza.

4.3.4. Recepción y Respuesta de Peticiones

Estas tareas sólo deben realizarse en caso que la política de revocación sea on-line. Teniendo en cuenta que hay dos entidades involucradas:

- Hosts origen: antes de enviar un agente, cada host origen envía una petición a la HoRA para conocer el estado de los hosts del itinerario del agente.
- HoRA: recibe las peticiones de todos los hosts origen, y responde a cada una de ellas con un mensaje firmado indicando qué hosts están revocados.

El resto de la sección presenta dos nuevos protocolos que pueden utilizarse para revocar hosts maliciosos.

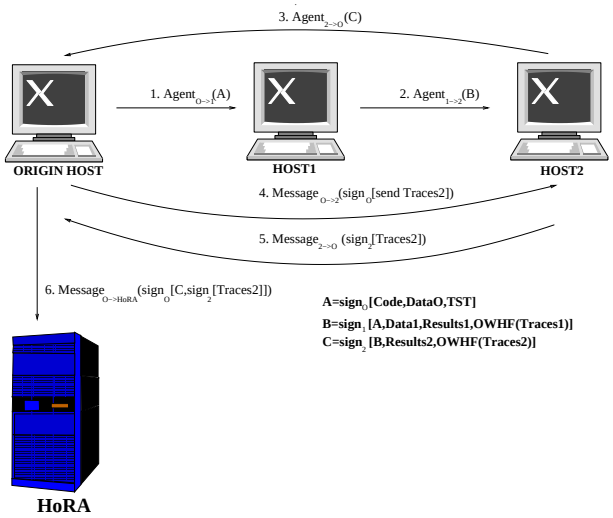


Figura 1: Protocolo de Revocación de Hosts

4.4. Protocolo de Revocación de Hosts

El protocolo para revocar un host malicioso que deben seguir las entidades que intervienen en la transacción se muestra en la figura 1. Por facilidad, presentaremos el protocolo mediante un ejemplo en el cual el itinerario es fijo con sólo 2 hosts. Asumimos que el host origen utiliza una política de revocación off-line, de manera que no realizará consultas a la HoRA, sino que consultara la HRL interna. Ninguno de los hosts del itinerario ha sido revocado, pero el segundo va a actuar maliciosamente en esta transacción. El ejemplo asume que no se requiere confidencialidad aunque, si fuera necesaria, podrían cifrarse aquellas partes consideradas confidenciales.

Utilizaremos la siguiente notación para el traspaso de mensajes o agentes en el protocolo:

- Denotamos un agente móvil que va desde el host x hacia el host y como $Agent_{x \to y}()$.
- Denotamos un mensaje que va desde el host x hacia el host y como $Message_{x \to y}()$.
- Denotamos la copia firmada del documento D como $sign_\alpha[D]$, donde α es el host firmante.
- Denotamos el valor de la función de hash del documento D como $OWHF(D)$.

A continuación se describen los pasos del protocolo para el ejemplo planteado:

1. El host origen (O) consulta internamente la copia local de la HRL. Como ninguno de los hosts ha sido revocado, el itinerario permanece igual y por tanto el agente se puede enviar al primer host. El agente transporta el código y algunos datos de entrada que se deben tener en cuenta en la ejecución del agente. Adicionalmente se incluye una caducidad de las trazas o TST (Traces Storage Timestamp), que indica a los hosts cuándo el host origen pierde el derecho de iniciar un proceso de revocación de hosts. Pasada dicha fecha de caducidad todas las pruebas pueden ser eliminadas porque dejan de tener validez. Todos los datos que transporta el agente deben estar convenientemente firmados para evitar ataques de repudio. Por tanto, el host origen envía al primer host del itinerario el siguiente agente:

$$Agent_{O \to 1}(A)$$

donde

$$A = sign_O[Code, DataO, TST].$$

2. El Host1 recibe el agente y lo ejecuta, de manera que con la ejecución del código se crean automáticamente las trazas [12]. Como las trazas a priori serán demasiado grandes, se envía como prueba al host origen el valor hash de las mismas. Las trazas completas sólo se enviarán en caso que el agente se considere sospechoso y se quiera revisar la integridad de la ejecución. El agente no sólo transportará el hash de las trazas, sino que también llevará los resultados y algunos datos de entrada para el siguiente host. La firma del Host1 certifica que existe una unión entre todos los parámetros de la ejecución, esto es, el código, los datos, las trazas y los resultados. Por tanto, el agente que se envía al siguiente host tiene el siguiente formato:

$$Agent_{1 \to 2}(B)$$

donde

$$B = sign_1[A, Data1, Results1, OWHF(Traces1)].$$

3. El Host2 recibe el agente y pretende ejecutarlo de forma maliciosa para obtener un cierto provecho. Para ello analiza y modifica el código y/o los datos. Como Host2 es el último host del itinerario no incluye datos de entrada para la ejecución del siguiente host (que correspondería al host origen). Por tanto, el agente que se envía al siguiente host tiene el siguiente formato:

$$Agent_{2\rightarrow O}(C)$$

donde

$$C = sign_2[B, Results2, OWHF(Traces2)].$$

4. Cuando el host origen recibe el agente, inicia el procedimiento de búsqueda de hosts sospechosos [3] para pedirles las trazas y así verificar la integridad de la ejecución. En el ejemplo, Host2 sería identificado como sospechoso y por tanto el host origen le enviaría un mensaje como el siguiente:

$$Message_{O\rightarrow 2}(sign_O[sendTraces2])$$

5. Host2 responde con un mensaje firmado que contiene las trazas completas. El formato del mensaje es el siguiente:

$$Message_{2\rightarrow O}(sign_2[Traces2])$$

6. Cuando recibe las trazas, el host origen realiza las siguientes verificaciones:

   - Verifica que las trazas $Traces2$ coincidan con el valor hash $OWHF(Traces2)$ que fue enviado en el paso 3. si hay alguna inconsistencia en el valor hash de las trazas, existe una prueba que certifica que Host2 no ejecutó el agente correctamente y por tanto puede ser revocado.
   - Ejecuta el agente de nuevo y verifica que la ejecución concuerda con las trazas $Traces2$. Si hay alguna inconsistencia en la ejecución, las trazas pueden utilizarse como prueba que certifica que Host2 no ejecutó el agente correctamente y por tanto puede ser revocado.

   Cuando el host origen tiene alguna prueba que certifique el carácter malicioso de algún host, puede iniciar el proceso de revocación. Básicamente, el proceso de revocación consiste en enviar a la HoRA las pruebas firmadas que certifican que el host ejecutó el agente maliciosamente:

   $$Message_{O\rightarrow HoRA}(sign_O[C, sign_2[Traces2]])$$

7. La HoRA recibe el mensaje que inicia el proceso de revocación de Host2, y como primer paso verifica que la caducidad de las trazas no haya pasado. Para ello verifica que el valor de TST es inferior a la fecha en curso, de manera que si es una fecha anterior el mensaje se descarta directamente. Una vez revisado esto, la HoRA realiza las mismas verificaciones que realizó el host origen en el paso 6. Obviamente, la HoRA debe disponer de un módulo con capacidad para ejecutar agentes para realizar esta tarea. Si finalmente alguna de las pruebas se considera válida, el identificador del Host2 se incluye en la lista que la HoRA tiene internamente, de manera que dicho host pasa a estado revocado. Host2 no puede realizar un ataque de repudio porque todas las pruebas están convenientemente firmadas por él.

4.5. Protocolo de Revocación Provisional

Puede darse el caso que un host quede fuera de servicio después de ejecutar un agente, con lo cual si algún host origen lo detecta como sospechoso y le pide las trazas no podrá mandárselas. En este caso, dicho host puede ser temporalmente revocado hasta que se pruebe que ejecutó el agente correctamente. El hecho de revocar un host de forma temporal no tiene consecuencias porque dicho host ha quedado fuera de servicio y por tanto no puede ejecutar ningún agente. El traspaso de mensajes se inicia en el paso 5, ya que el mensaje que mandaba el Host2 y que contenía las trazas no llega a mandarse porque se supone que dicho host está fuera de servicio. La Figura 2 muestras los nuevos mensajes:

5. Como el host origen no recibe las trazas de Host2, puede iniciar un proceso de revocación provisional. Para ello envía un mensaje a la HoRA informando que las trazas no han sido enviadas por el Host2. El mensaje además contiene las pruebas de la ejecución que dispone el host origen:

   $$Message_{O\rightarrow HoRA}(sign_O[C, Traces2\, not\, recieved])$$

6. La HoRA recibe el mensaje que demanda la revocación provisional de Host2. Como primer paso, la HoRA pide las trazas directamente a Host2 . El formato del mensaje que se envía es el siguiente:

   $$Message_{HoRA\rightarrow 2}(sign_{HoRA}[sendTraces2])$$

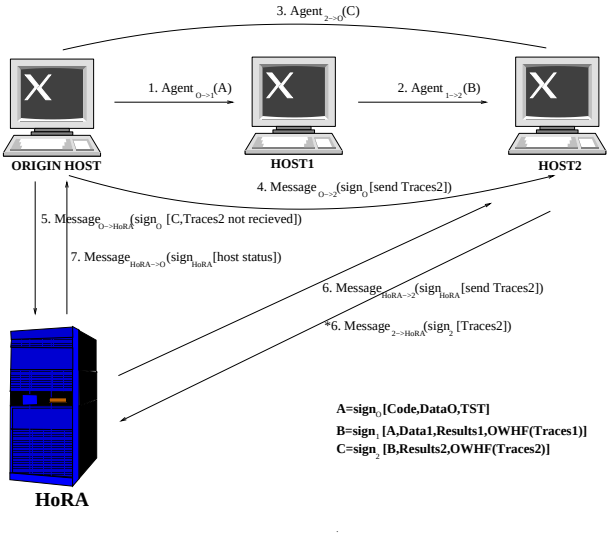


Figura 2: Protocolo de Revocación Provisional

- Si Host2 no responde al mensaje, su identificador se incluye en la lista que la HoRA tiene internamente, con lo cual se revoca provisionalmente. Durante un cierto periodo de tiempo, Host2 tiene la posibilidad de enviar las trazas a la HoRA para demostrar su inocencia, si bien después de dicho tiempo el estado del host pasa a permanentemente revocado. Sin embargo, para evitar que hosts honestos que han quedado fuera de servicio dejen de recibir agentes durante un periodo de tiempo largo, los hosts revocados provisionalmente no deben incluirse en la HRL hasta que no pasen a permanentemente revocados.

- Si Host2 responde al mensaje con las trazas, la HoRA dispone de toda la información necesaria y puede realiza las mismas verificaciones que se realizaban en el proceso de revocación de hosts no provisional:

$$* \; Message_{2 \rightarrow HoRA}(sign_2[Traces2])$$

7. En ambos casos, la HoRA debe informar al host origen del estado en que queda el host (revocado o no) para decidir qué hace con los resultados de la ejecución de dicho host:

$$Message_{HoRA \rightarrow O}(sign_{HoRA}[host\ status])$$

4.6. Ataques

Los ataques que pueden realizarse contra el sistema están focalizados en la ocultación de las pruebas:

- Un host malicioso puede modificar los datos de entrada para el siguiente host, los resultados, el hash de las trazas o incluso las mismas trazas. En el ejemplo anterior el Host2 puede modificar $Results2$, $OWHF(Traces2)$ o $Traces2$. Si finalmente el host es sospechoso y el host origen le pide las trazas, se encontrará una prueba que certifique que no ejecutó el agente de forma correcta y por tanto podrá ser revocado.

- El host malicioso puede no enviar las pruebas. Como sólo hay dos pruebas, hay dos ataques de este tipo:

  - El host malicioso no envía el valor hash de las trazas $OWHF(Traces2)$. Sin el hash de las trazas no existe una prueba que enlace los datos de entrada con los resultados y las trazas. Este ataque se podría considerar un ataque de negación de servicio ya que el agente está incompleto. En este caso el host puede revocarse directamente mandando a la HoRA la prueba que dicho host no mandó el hash de las trazas

    $$Message_{O \rightarrow HoRA}(sign_O[C, Incomplete\ Agent])$$

    donde $C = sign_2[B, Results2, -]$.
  - El host malicioso no envía las trazas $Traces2$. En este caso, el host pretende estar fuera de servicio, con lo cual se iniciará un proceso de revocación temporal. Si finalmente el host no envía las trazas, el estado de dicho host pasará a revocado permanentemente.

- Un host origen puede intentar involucrar un host honesto iniciando un proceso de revocación temporal:

  $$Message_{O \rightarrow HoRA}(sign_O[C, Traces1\, not\, recieved])$$

  Este tipo de ataque puede evitarse si el host honesto guarda las trazas $Traces1$ hasta que TST caduque.

4.7. Inconvenientes

Estos son los principales inconvenientes de la propuesta:

- Un error no intencionado de un host durante la ejecución de un agente puede provocar que dicho host sea revocado en caso que sea considerado sospechoso. Esto podría considerarse una medida desproporcionada, pero en la opinión de los autores, los hosts deben asegurar la correctitud de todas las transacciones.
- La lista que almacena internamente la HoRA crece indefinidamente. Este problema podría resolverse si se utilizara un certificado emitido por la HoRA que permitiera la ejecución de agentes con un cierto periodo de validez. En caso de comportamiento malicioso la HoRA no revocaría al host sino al certificado.
- De cara a liberar a la HoRA de alguna de sus tareas, podría pensarse en una topología alternativa basada en repositorios con una política de replicación adecuada.

5. Conclusiones

La propuesta que se presenta en este artículo ayuda a solucionar el problema de los hosts maliciosos con el uso de una TTP, la autoridad de revocación de hosts o HoRA [4]. La HoRA almacena una lista con los identificadores de aquellos hosts que existen pruebas que actuaron maliciosamente y por tanto fueron revocados. Antes del envío de un agente, los hosts origen deben consultar la información de revocación (1) consultando directamente a la HoRA (on-line), o bien (2) descargando localmente una lista de hosts revocados o HRL (off-line). Los hosts revocados serán eliminados del itinerario del agente, y por tanto no volverán a recibir agentes para su ejecución. Un host origen además puede revocar un host si demuestra que actuó maliciosamente. En este artículo se presentan dos nuevos protocolos que pueden ser usados para revocar hosts maliciosos y que utilizan técnicas de detección de ataques [12] y técnicas de detección de sospechosos[3].

Las lineas futuras son las siguientes:

- Evaluar cómo se pueden incluir los permisos de ejecución de agentes en un certificado emitido por la HoRA, o bien en otro tipo de certificados existentes como los certificados de identidad de la PKI o los certificados de atributo de la PMI.
- Construir una nueva topología basada en repositorios para hacer más accesible la información de revocación a los hosts.

Agradecimientos

Este trabajo ha sido desarrollado dentro del proyecto DISQET CICYT TIC2002-00818.

Referencias

[1] J. Algesheimer, C. Cachin, J. Camenisch, and G. Karjoth. Cryptographic security for mobile code. In *IEEE Symposium on Security and Privacy*, 2001.

[2] C. Cachin, J. Camenisch, J. Kilian, and Joy Müller. One-round secure computation and secure autonomous mobile agents. In *27th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 1853 of *LNCS*. Springer-Verlag, 2000.

[3] O. Esparza, M. Soriano, J.L. Muñoz, and J. Forné. A protocol for detecting malicious hosts based on limiting the execution time of mobile agents. In *IEEE Symposium on Computers and Communications - ISCC'2003*, 2003.

[4] O. Esparza, M. Soriano, J.L. Muñoz, and J. Forné. Host Revocation Authority: a Way of Protecting Mobile Agents from Malicious Hosts. In *International Conference on Web Engineering (ICWE 2003)*, LNCS. Springer-Verlag, 2003.

[5] F. Hohl. Time Limited Blackbox Security: Protecting Mobile Agents From Malicious Hosts. In *Mobile Agents and Security*, volume 1419 of *LNCS*. Springer-Verlag, 1998.

[6] ITU/ISO Recommendation. X.509 Information Technology Open Systems Interconnection - The Directory: Autentication Frameworks, 2000. Technical Corrigendum.

[7] Y. Minsky, R. van Renesse, F. Schneider, and S.D. Stoller. Cryptographic Support for Fault-Tolerant Distributed Computing. In *Seventh ACM SIGOPS European Workshop*, 1996.

[8] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP, 1999. RFC 2560.

[9] J. Riordan and B. Schneier. Environmental Key Generation Towards Clueless Agents. In *Mobile Agents and Security*, volume 1419 of *LNCS*. Springer-Verlag, 1998.

[10] V. Roth. Mutual protection of cooperating agents. In *Secure Internet Programming: Security Issues for Mobile and Distributed Objects*, volume 1906 of *LNCS*. Springer-Verlag, 1999.

[11] T. Sander and C.F. Tschudin. Protecting mobile agents against malicious hosts. In *Mobile Agents and Security*, volume 1419 of *LNCS*. Springer-Verlag, 1998.

[12] G. Vigna. Cryptographic traces for mobile agents. In *Mobile Agents and Security*, volume 1419 of *LNCS*. Springer-Verlag, 1998.

[13] B.S. Yee. A sanctuary for mobile agents. In *DARPA workshop on foundations for secure mobile code*, 1997.

Metodología para el Análisis Formal de los Protocolos de Seguridad

Javier López, Juan J. Ortega, José M. Troya
Dpto. Lenguajes y Ciencias de la Computación
Universidad de Málaga
e-mail: {jlm, juanjose, troya}@lcc.uma.es

Abstract. *Nowadays, it is widely accepted that critical systems have to be formally analysed in order to achieve well-known formal method benefits. In order to study the security of communication systems, we have developed a methodology for the application of the formal analysis techniques commonly used in communication protocols to the analysis of cryptographic ones. In particular, we have extended the design and analysis phases with security properties. Our proposal uses a specification notation based on Message Sequence Charts (MSC), which can be automatically translated into a generic Specification Description Language (SDL) specification. This SDL system can then be used for the analysis of the desired security properties, by using an observer process schema. Apart from our main goal of providing a notation for describing the formal specification of security systems, our proposal also brings additional benefits, such as the study of the possible attacks to the system, and the possibility of re-using the specifications produced to describe and analyse more complex systems.*

1 Introducción

El aumento en la utilización de las comunicaciones mediante las redes públicas y la continua evolución que han seguido las aplicaciones que hacen uso de ellas, como es el caso del comercio electrónico, ha obligado hacer una revisión en los sistemas sobre sus características de seguridad, centrándose mayormente en los protocolos de seguridad. La correcta definición de estos protocolos y una apropiada interacción con los restantes elementos del sistema, requiere de la formalización de las propiedades de seguridad en las etapas de diseño y especificación.

A partir de una especificación se puede diseñar, validar y documentar [7] el sistema con un menor riesgo de error. La semántica de los formalismos que proporciona las técnicas de descripción formal aporta reglas precisas de interpretación que evitan muchos problemas encontrados en una especificación en lenguaje natural. Es decir, un lenguaje con ricas facilidades de estructuración puede producir mejores especificaciones.

Además de producir una buena especificación, la aplicación de métodos formales proporciona la base para un procesamiento automático mediante el cual se da la posibilidad de generar una implementación del sistema a partir de su especificación.

En el desarrollo de protocolos de seguridad, además de encontrarnos con los problemas mencionados, existen otras consideraciones a tener en cuenta:

- Las propiedades que tienen que garantizar son extremadamente delicadas.

- Habitan en entornos complejos y hostiles. Aparece la figura del intruso que intentará "romper" la seguridad.

- Conocer las capacidades del intruso es muy complicado.

- Debido a su naturaleza, estos protocolos llevan asociados un alto grado de concurrencia.

Por otro lado, un protocolo de seguridad resulta ser un componente crítico en cualquier arquitectura de seguridad distribuida. Un mal diseño del mismo puede causar que presente vulnerabilidades ante ciertos ataques.

De este modo, por su naturaleza, los protocolos de seguridad resultan ser los candidatos ideales a los cuales aplicar técnicas rigurosas de especificación y diseño.

Durante muchos años se han estado utilizando este tipo de formalismos para propósitos generales pero sólo en la última década se han empezado a utilizar en los protocolos de seguridad.

Las técnicas de análisis de protocolos de seguridad [16] se pueden dividir en tres tipos. La primera utiliza lógica de primer orden diseñada específicamente para el análisis de protocolos de seguridad. La más significativa es la lógica BAN [3], aunque también se han definido extensiones como la lógica GYN [6]. Esta técnica está en desuso, sobre todo por su dificultad de aplicación. La segunda es la que utiliza los probadores de teoremas (*theorem-proving techniques*). Estos hacen uso de reglas de reescritura para el análisis de las propiedades de seguridad. Los más destacados son la traducción de CAPSL [4] a PVS y Maude, CASRUL [19,20]. El último tipo se basa en técnicas de exploración del espacio de estado, sobre todo *model-checking* [1,14]. Es la que mejores resultados han producido hasta ahora, ya que aunque no puede garantizar en la mayoría de los casos que no existe un fallo en el protocolo debido a la no completitud del problema [13,21] , si que la exploración es lo suficientemente amplia como para deducirlo. En este tipo se enmarcan el NRL Protocol Analyzer [15], Casper [22], LOTOS [10] y SDL [11,18] entre otros.

El artículo se divide en los siguientes apartados: la sección 2 explica las características principales de los protocolos de seguridad, así como los diferentes ataques a los que se enfrentan en sistemas abiertos. A continuación en la sección 3 se expone las propiedades de seguridad y los mecanismos de análisis de cada una de ellas. En la sección 4 se explica brevemente la metodología diseñada por los autores, donde se especifica y analiza las propiedades de seguridad como confidencialidad y autenticación. Por último, se expone las conclusiones y trabajos futuros.

2 Protocolos de seguridad

Como cualquier protocolo, un protocolo de seguridad [17] consiste en una secuencia de interacciones entre entidades para conseguir un resultado final. Pero, además, el objetivo fundamental que persigue este tipo de protocolos es proporcionar servicios de seguridad en un sistema distribuido. Entre estos servicios se encuentran la autenticación de los agentes que intervienen, el establecimiento de claves de sesión y la aportación de confidencialidad, integridad, anonimato y no-repudio. Para conseguir estos servicios se hace necesario el intercambio de mensajes entre los agentes implicados y, en algunos casos, la participación de una tercera parte confiable.

El servicio de confidencialidad, que ofrecen muchos protocolos de seguridad consiste en hacer que el intruso sea incapaz de deducir nada acerca de la actividad de los usuarios legítimos. Este tipo de servicio se requiere en aquellas situaciones en las que la información que se están intercambiando dos agentes es sensible, es decir, datos de relevancia.

En el caso de la autenticación lo que se pretende es que los agentes que intervienen en una comunicación estén seguros de la identidad de la persona con la que se están comunicando.

Un agente, aún estando seguro de la identidad del agente con el que está manteniendo un diálogo, puede necesitar tener evidencias de que se producen determinadas situaciones durante dicha comunicación. En este caso se necesita que el protocolo utilizado ofrezca el servicio de no-repudio. Este servicio se confunde a veces con el de autenticación pero su misión es otra completamente distinta. Lo que pretende es que un agente no pueda alegar que no ha recibido o enviado algo cuando sí que ha sucedido dicha situación.

El servicio o propiedad de anonimato es necesario en aquellas situaciones en las que se desea que no conozca la identidad del agente que ha llevado a cabo una determinada acción. Puede ser muy útil en compras a través de la red donde no se hace necesario el conocimiento del usuario que está comprando. Es lo contrario del no-repudio.

Normalmente un protocolo proporcionará algunos de estos servicios y no necesariamente todos los mencionados. Por ejemplo, en el caso de los protocolos dedicados al intercambio de claves lo que se pretende es establecer una clave que será utilizada por dos agentes que desean comunicarse de forma segura. Para ello será necesario que dicha clave se envíe a la persona apropiada además de hacerlo de forma confidencial.

En cambio, en los protocolos de autenticación lo que se persigue es que los agentes implicados estén seguros de que la persona con la que se están comunicando sea quien dice ser.

El proporcionar estos servicios representa una ardua tarea ya que el protocolo se encuentra en un entorno hostil donde habita algún posible intruso que intentará realizar ataques a la seguridad del sistema. Existe mucha diversidad de ataques [22] de los cuales nombraremos solamente algunos que pueden servir de ejemplo de las debilidades que pueden presentar los protocolos si no están diseñados correctamente.

El ataque del hombre en el medio (*man-in-the-middle*) consiste en que el intruso se interpone entre los dos agentes que se están comunicando para interceptar información de importancia. Para ello necesita "engañar" a algunos de los agentes honestos. Este tipo de ataque puede evitarse si los agentes se autentican previamente.

Uno de los agentes honestos puede ser inducido a realizar la ejecución de una determinada parte del protocolo lo que ayudaría al intruso a obtener algunos datos importantes. Este tipo de ataque es llamado ataque de oráculo (*oracle*).

En el caso de un ataque de repetición (*replay*) el intruso se encuentra atento a la ejecución de un protocolo y almacenar algunos de los mensajes que se envían. En alguna ejecución posterior del mismo protocolo podría enviar uno de esos mensajes llegando a confundir a los agentes implicados.

También existen ataques a los algoritmos algebraicos utilizados para el cifrado de mensajes pero estos no son evitables mediante métodos formales mientras que los expuestos, entre otros, sí que son salvables.

3 Análisis de las propiedades de seguridad

Las propiedades básicas de seguridad expuestas en la sección anterior van a ser a continuación desarrolladas desde el punto de vista de los mecanismos de análisis.

Las técnicas formales que analizan estas propiedades describen el protocolo de seguridad y lo evalúan considerando que existe un medio controlado por el intruso o atacante [5]. Además considera que las operaciones criptográficas son perfectas, por lo que no considera ataques por criptoanálisis.

3.1 Análisis de la confidencialidad

La propiedad de confidencialidad es la cualidad de prevenir que un intruso sea capaz de deducir el texto en claro de los mensajes que se intercambian entre agentes honestos. Si en algún estado del protocolo se

produce que cierto dato aparece en la base de conocimiento del intruso, esto quiere decir que se ha vulnerado tal propiedad de dicho dato. La base de conocimiento del intruso la constituye los datos de los mensajes que se intercambian más los que el intruso puede deducir.

En principio, para analizar si un protocolo proporciona el servicio de confidencialidad de los datos, se va a considerar que se realiza un cifrado "perfecto" por lo que para deducir un dato cifrado con una determinada clave, se debe poseer dicha clave.

Se puede distinguir entre dos tipos de secreto; el primero sería el referente a elementos externos al protocolo pero que intervienen en el mismo (*strong secrecy*). Por ejemplo, una clave como entrada del protocolo que va a hacer uso de ella. Y el segundo tipo de secreto sería el concerniente a elementos creados en la ejecución del protocolo como, por ejemplo, una clave de sesión.

Para la evaluación de esta propiedad se comprueba que los datos cifrados no se pueden deducir mediante alguna regla de deducción. Las reglas de deducción principales son:

1. {{M}*sk*}*sk'* = M, *sk* es una clave simétrica y *sk'* indica que es descifrar

2. {{M}*pk*}*pk'* = {{M}*pk'*}*pk* = M., *pk* es la clave pública y pk' es la privada

3. { {M}*key1* }*key2* = { {M}*key2* }*key1*

Un ejemplo de ataque a la propiedad de secreto lo vemos en al análisis del protocolo RSA [22].

3.2 Análisis de la autenticación

La autenticación de entidades se basa en que seamos capaces de estar seguros de que todos los mensajes que recibimos son de un determinado agente (autenticación en origen). Si aceptamos mensajes que son originados por un agente deshonesto (normalmente el intruso) y finalizamos el protocolo, entonces se produce el fallo en la autenticación. A este tipo de fallo se le llama fallo de correspondencia o precedencia, ya que se modela haciendo que un agente termine el protocolo correctamente mientras que el otro agente ni siquiera lo empiece.

De este modo, un protocolo de autenticación asegura que un agente *A* ha establecido una comunicación con el agente *B*. Para asegurar la identidad se comparten algunos datos (claves, variables,...).

En la ejecución de un protocolo de autenticación se pueden producir dos situaciones:

1- El agente *A* ha completado la ejecución del protocolo con el agente *B*.

2- El agente *A* está ejecutando el protocolo aparentemente con el agente *B*.

En el proceso de autenticación hay que distinguir el punto en el que cada agente tiene total seguridad de que se está comunicando con el agente deseado.

La autenticación [24] se lleva a cabo mediante una secuencia ordenada de mensajes donde los agentes se intercambian una serie datos ordenados convenientemente. Si esa secuencia se ve alterada se puede pensar que el protocolo no se ha ejecutado correctamente. Por lo tanto, para verificar que un protocolo autentica a los agentes que se están comunicando, hay que comprobar que esa secuencia de mensajes no se altere.

Un ejemplo de protocolo de autenticación es el *Encrypted Key Exchange* (EKE) [2] (figura 2). El objetivo de este protocolo es el de asegurar que *A* está comunicándose con *B* y que *B* lo está con *A*. Al final del intercambio de mensajes tanto *A* como *B* deben conocer *Na* y *Nb*, y por lo tanto, se demuestra la autenticación de ambos.

El ataque a este protocolo se produce cuando se ejecutan dos sesiones paralelas y el intruso envía los mensajes de una sesión a la otra. Al final, termina los agentes *A* de la primera sesión y *B* de la segunda sesión, en realidad *"b"* se conecta consigo mismo. En el ejemplo de utilización de la metodología propuesta se analiza este protocolo.

3.3 Análisis del no repudio y el anonimato

En protocolos más avanzados como son los de comercio electrónico, además de las propiedades vistas hasta ahora, existen otras que se hacen necesarias para que las transacciones que se producen en la red sean totalmente seguras. Este es el caso de las propiedades de no-repudio y anonimato.

La propiedad de no-repudio [23,25] persigue que ninguna de las dos partes involucradas en una transacción se desdiga acerca de alguna acción que se haya llevado a cabo durante dicha comunicación. Es muy utilizada para el caso de transacciones comerciales en Internet como son las compras en tiendas virtuales. En este tipo de transacciones normalmente interviene alguna tercera parte confiable (*TTP*) que será la que supervise todo el proceso y muestre evidencias de los pasos llevados a cabo.

Existen distintas clases de no-repudio las cuales las describimos a continuación y para ello utilizaremos un pequeño ejemplo en el que un usuario *A* desea realizar una compra a través de Internet en una tienda *B*.

- *No-repudio de origen*. Es la prueba producida por el originario de un mensaje. De este modo, si *A* realiza una petición a la tienda *B* ésta tendrá una prueba de que ha sido *A* quien realmente le ha realizado una consulta.

- *No-repudio de recepción*. Es la prueba producida por el destinatario de un mensaje. Así, cuando *B* recibe una petición de un usuario *A* éste deberá poseer una prueba de que su petición ha sido recibida por la tienda.

Si para llevar a cabo la transacción interviene una *TTP* aparecen dos nuevos tipos de no-repudio.

- *No-repudio de envío*. Es la prueba producida por la *TTP* cuando recibe un mensaje de alguna de las dos partes involucradas. Esta prueba es entrega al originario del mensaje.

- *No-repudio de entrega*. Es la prueba producida por el destinatario del mensaje. En vez de ser entregada al originario del mismo, esta prueba se entrega a la *TTP*.

Bien es cierto que siempre se hará necesaria la intervención una tercera parte confiable ya que, por ejemplo, el no-repudio de origen mencionado se puede conseguir mediante una firma digital por parte del originario del mensaje. Para que esta firma tenga validez legal tendrá que estar respaldada por un certificado digital emitido por alguna autoridad de certificación oficial que no deja de ser una *TTP*.

La propiedad de no-repudio [22,25] se analiza garantizando que algunos eventos sucedan antes que otros al igual que ocurría para la verificación de la autenticación. Los eventos que se analizan son aquellos que representan las evidencias necesarias para que ninguna de las partes se desdiga.

Estas evidencias suelen ser mensajes cifrados por claves las cuales sólo son conocidas por los agentes implicados. De este modo, si un agente *A* recibe un mensaje cifrado con la clave que comparte con otro agente *B* entonces puede demostrar que el mensaje ha sido enviado por *B* por lo que éste último no puede negar dicha evidencia.

A diferencia con el análisis de la autenticación, el sistema debe considerar que puede ser uno de los agentes que suponemos honestos el que tiene intenciones de engañar. Debido a esto, no se podría utilizar el esquema en el que un intruso externo es el que controla el medio de transmisión.

La propiedad de anonimato consiste en la protección de la identidad de los agentes con respecto a determinados eventos o mensajes pertenecientes a la ejecución de un protocolo. Se basa en que un dato puede ser originado por uno u otro agente indistintamente.

El anonimato se puede presentar desde dos puntos de vista. Por un lado, se puede hacer que uno de los agentes o ambos sean anónimos a un observador externo. Y por otro, se puede proporcionar anonimato a un agente con respecto al agente con el que se está comunicando.

Para proporcionar el anonimato de los agentes implicados en un protocolo existen varias técnicas como son la ocultación, el enmascarado o el renombrado. El protocolo debe estar diseñado de tal modo que, aún proporcionando el anonimato de los agentes, funcione de la misma forma.

El estudio del análisis de este tipo de propiedad se encuentra muy poco desarrollado con respecto del análisis de propiedades de confidencialidad o autenticación existiendo muy pocos trabajos al respecto.

4 Metodología de diseño y análisis

En el marco del proyecto Europeo CASENET, se ha desarrollado una metodología [12] para el diseño y análisis de los protocolos de seguridad, que utiliza conjuntamente los conceptos de ingeniería de protocolos comunicaciones y de verificación de protocolos de seguridad.

En la figura 1 se describe el esquema general de la metodología. Partimos de unos requisitos funcionales y de seguridad que son formalizados mediante un lenguaje de requisitos que llamamos “Specification Requirement Security Language” (SRSL). En este lenguaje también se expresa el comportamiento del posible atacante para analizarlo convenientemente. Esta especificación es traducida automáticamente a un esquema en el lenguaje SDL donde se realiza la verificación de las propiedades de seguridad mediante la estrategia de análisis determinada. Además, el sistema SDL puede producir una código ejecutable que puede ser utilizado para el chequeo del sistema.

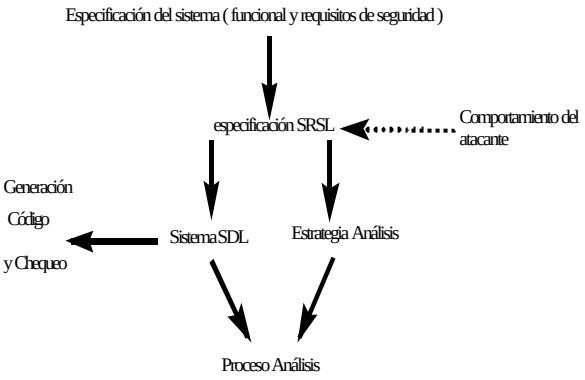


Figura 1: Esquema general de la metodología de análisis

El proceso de análisis es realizado mediante la exploración exhaustiva del sistema SDL teniendo en cuenta la capacidad del atacante.

El lenguaje de requisitos (SRSL) que utilizamos como entrada en nuestro sistema es una extensión de la Recomendación de la ITU-T Z.120, “Message Sequence Chart” (MSC) [9]. La especificación de un protocolo de seguridad en SRSL consta de dos secciones: una donde se describe el intercambio de mensaje como cualquier protocolo de comunicaciones; y la otra que define las propiedades de seguridad que se van a analizar. Actualmente se puede analizar la confidencialidad y la autenticación, aunque las demás propiedades se van a integrar próximamente.

En la figura 2 se muestra la especificación el protocolo EKE usando el lenguaje SRSL.

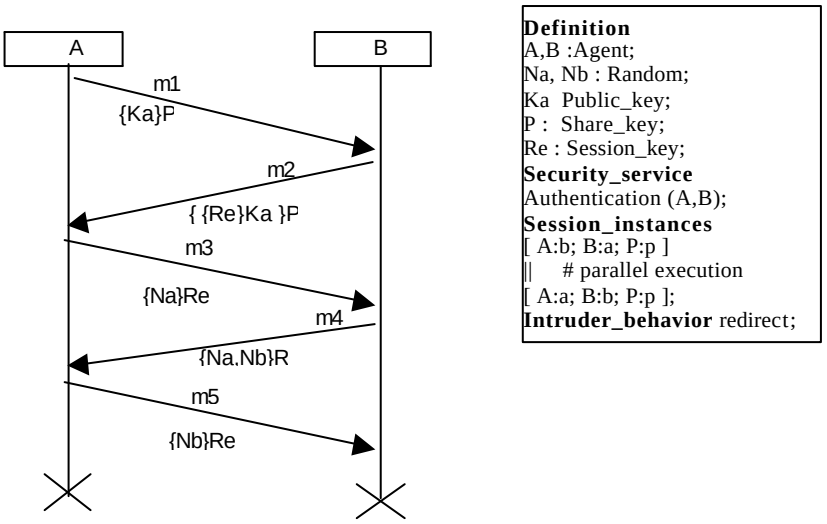


Figura 2: Especificación del protocolo EKE en el lenguaje SRSL

Una vez especificado el protocolo en el lenguaje SRSL se traduce, mediante un programa en C, escrito usando las herramientas lex/yacc, a un esquema genérico en SDL.

El SDL [16] es un lenguaje de descripción formal que se define en la Recomendación Z.100 de la ITU-T [8]. El modelo en el que se basa es un conjunto de máquinas de estado finito extendidas que se comunican entre sí y con el entorno mediante paso de mensajes. Cada máquina de estado evoluciona independientemente de las demás. La versión es la denominada SDL92 e incluye tipos abstractos de datos y otras características para el desarrollo orientado a objeto.

Telelogic TAU suite es un conjunto de herramientas que permite diseñar un sistema en SDL y, además, lo valida, verifica y simula. También genera de forma automática una implementación en lenguaje C y C++. Además también la utilizamos para crear la especificación en SRSL el editor de MSC.

La traducción produce un sistema SDL que se divide en el sistema principal, la librería de definición de los tipos de datos, la librería de definición de los tipos agente, y por último en los procesos medio-observador.

La definición de los tipos de datos se realiza en la paquete llamado “analcryptlib”. Los tipos utilizados en el lenguaje SRSL (*Agent*, *Random*, etc ...) tienen una representación en tipos de datos SDL, así como el formato de los mensajes (*m1* , *m2* , etc ..). Además se definen unos tipos de datos *conjunto* (SET) para almacenar el conocimiento del proceso intruso. En el caso de que vayamos a generar código tenemos que cambiar esta definición de datos por otra más apropiada ya que éstos están diseñados para el proceso de análisis.

Los agentes del protocolo se definen como procesos tipo (*process type*) en un paquete SDL llamado “agentlib”. La descripción es independiente del proceso de análisis, y por los tanto puede ser utilizado en una especificación como la de un protocolo de comunicaciones.

El proceso que va a analizar el comportamiento del protocolo cuando es atacado por un intruso se llama “observer”. Éste va a depender de la propiedad que pretende analizar. Se define como tipo proceso en un paquete SDL llamado “observerlib”.

El sistema principal lo compone un bloque donde se definen los procesos propiamente dichos del protocolo (instancias de los procesos tipo definidos en “Agentlib”) y los procesos tipo que van a funcionar como los diferentes medios por donde se va a transmitir los mensajes entre las diferentes entidades.

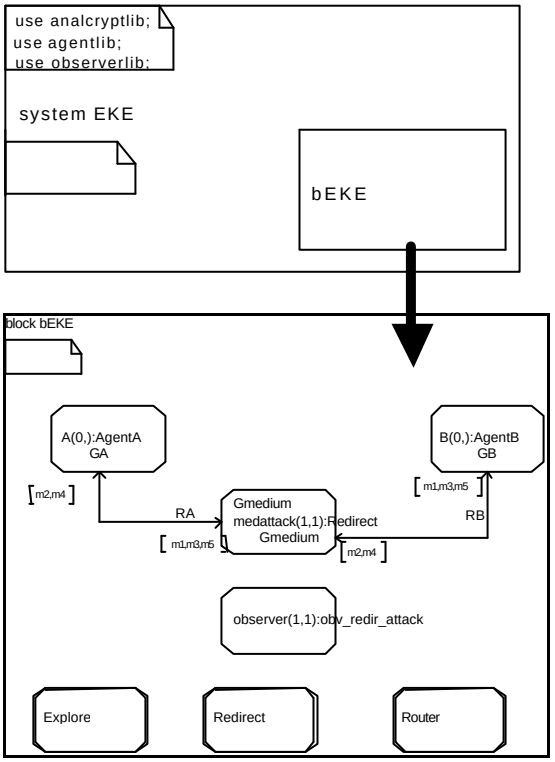


Figura 3: Esquema general de sistema SDL generado

El proceso que va a simular el medio de comunicación y a su vez en comportamiento del atacante se llama “medattack”. Este proceso va a ser instanciado a los siguientes tipos de procesos: en primer lugar el que hemos llamado “Router” que funciona como un medio fiable, sin alterar los mensajes. Sirve para verificar la especificación. Un segundo proceso llamado “Explore” que se encarga del realizar la exploración de todos los posibles mensajes en cada paso del protocolo. Este sería el menos eficiente, pero el que garantiza que se analizan un mayor número de estados. El último es un proceso creado para verificar un determinado tipo de fallo en el protocolo que llamamos “Redirect”. Éste es generado con la información que se indica en la especificación de protocolo en el lenguaje SRSL a través de las sesiones “Session_instances”, y “Intruder_behaviour”. Donde de indica la propiedad a comprobar es la sesión llamada “Security_service”.

La herramienta de análisis de SDL (TAU Validator) produce como resultado de la validación escenarios expresados en MSC. Estos escenarios van a ser los que representan los posibles fallos del protocolo.

En el caso del protocolo de autenticación e intercambio de clave EKE se ha evaluado detectando el fallo de diseño ya publicado. En la figura 4 se muestra el diagrama de secuencia resultante del ataque del protocolo mediante dos sesiones paralelas donde los mensajes del agente A de la sesión 1 son enviados al agente B de la segunda sesión. Este es el comportamiento del intruso que hemos llamado “Redirect”. Como termina el protocolo el agente A de la primera sesión y el agente B de la segunda y no los demás se considera que hay un fallo de la propiedad de autenticación, ya que A_1 cree que esta hablando con B_1, pero en realidad lo hace con B_2.

Al igual que el protocolo EKE se han analizado los protocolos más utilizados en implementaciones reales como el SSL/TLS e IPsec. Una de las ventajas de esta metodología es que es modular y se pueden especificar y analizar tanto protocolos simples como complejos.

5 Conclusiones

Los protocolos de seguridad deben garantizar aquellas propiedades de seguridad para los que estén diseñados. Las propiedades básicas que hemos considerado en este trabajo son las de confidencialidad, autenticación, no-repudio y anonimato.

La mayoría de los trabajos de basan en el estudio de la confidencialidad y la autenticación ya que son las más usuales en sistemas tradicionales como el control de acceso o el intercambio de una clave de sesión.

La metodología que se propone es este trabajo consta de una especificación de los protocolos de seguridad basada en los diagramas de secuencia MSC que llamamos SRSL. Éste es traducido a un sistema SDL donde se analizan las propiedades de seguridad indicadas. La metodología analiza las propiedades de confidencialidad y autenticación, y se está trabajando para incluir las demás propiedades.

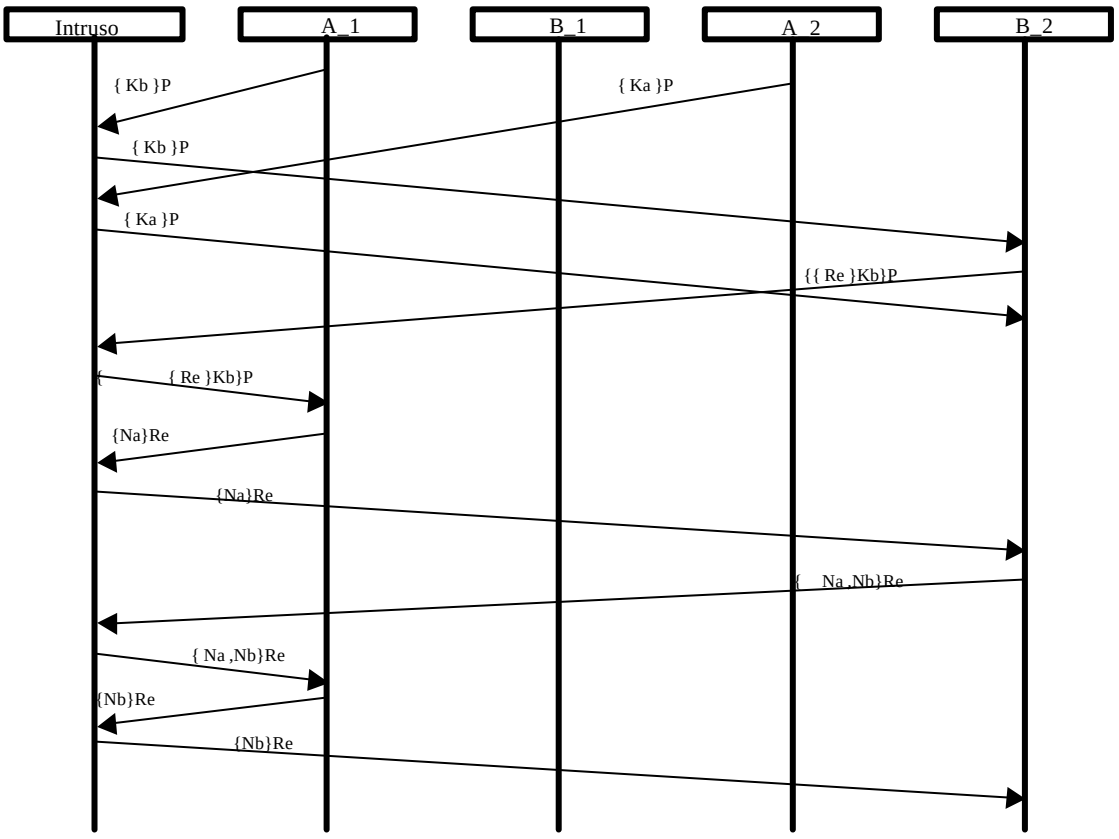


Figura 4: Resultado del análisis el protocolo EKE.

Agradecimientos

Este artículo está parcialmente subvencionado por el proyecto Europeo CASENET, IST-2001-32446.

Referencias

[1] Alur, R.; Henzinger, T.; Mang, F.; Qadeer, S.; Rajamani, S. and Tasiran, S. *"Mocha: modularity in model checking"*. In A. Hu and M. Vardi, editors, CAV 98: Computer-aided Verification, Lecture Notes in Computer Science 1427, pages 521-525. Springer-Verlag, 1998.

[2] Bellovin, S.M. and Merrit, M. *"Encrypted key exchange: Password-based protocols secure against dictionary attacks"*. In Proceedings of IEEE Symposium on Reseach in Security and Privacy, pages 72-84, 1992.

[3] Burrows, M.; Abadi, M. and Needham, R. *"A logic of authentication"*. In Proceedings of the Royal Society, Series A, 426(1871):233-271, 1989.

[4] Denker, G. and Millen, J. *"Capsl intermediate language"*. In Formal Methods and Security Protocols,1999. FLOC '99 Workshop.

[5] Dolev, D. and Yao, A. *"On the security of public key protocols"*. IEEE Transactions on Information Theory, IT-29:198{208, 1983. Also STAN-CS-81-854, May 1981, Stanford U.

[6] Gong, L.; Needham, R. and Yahalom, R. *"Reasoing about belief in cryptographic protocols"*. IEEE Symposium on Research in Security and Privacy, Oakland, California, 1990.

[7] Holzmann, G. *"Design and Validation of Computer Protocols"*. Prentice-Hall, Englewood Cliffs, 1991.

[8] ITU-T Recommendation Z.100 (11/99). "*Specification and Description Language (SDL)*", Geneva, 1999.

[9] ITU-T, Recommendation Z.120 (11/99). "*Message Sequence Charts (MSC)*". Geneva,1999.

[10] Leduc, G. and Germeau, F. *"Verification of Security Protocols using LOTOS-method and application"*. Computer Communications 23. 2000.

[11] López, J.; Ortega, J.J; Troya, J.M. *"Verification of authentication protocols using SDL-Method."* Workshop of Information Security. Abril 2002.

[12] López, J., Ortega, J.J. and Troya, J.M.. "*Protocol Engineering Applied to Formal Analysis of Security Systems*". Infrasec'02, LNCS 2437, Bristol, UK, October 2002.

[13] Lowe, G. "*Towards a Completeness Result for Model Checking of Security Protocols*". In 11th IEEE Computer Security Foundations Workshop, pages 96-105. IEEE Computer Society, 1998.

[14] Marrero, W.; Clarke, E. and Jha, S. *"Model checking for security protocols"*. DIMACS Workshop on Design and Formal Verification of Security Protocols, 1997.

[15] Meadows, C. *"A model of computation for NRL protocol analyser"*. Computer Security Foundation Workshop. 1994.

[16] Meadows, C. *"Open issues in formal methods for cryptographic protocol analysis"*. In Proceedings of DISCEX 2000,pages 237--250. IEEE Comp. Society Press, 2000.

[17] Menezes, A.;. van Oorschot, P.C.; Vanstone, S. *"Handbook of Applied Cryptography"*. CRC Press, (1997).

[18] Ortega, J.J. *Técnicas de Descripción Formal para el estudio de la Vulnerabilidad de Protocolos.* V Reunión Española de Criptología y Seguridad, Málaga 1998.

[19] Rusinowich, M. *CASRUL*. http://www.loria.fr/equipes/protheo/SOFTWARES/CASRUL/

[20] Rusinowich, M.; Jacquemard, F.; Vigneron, L. *"Compiling and Verifying Security Protocols Logic for Programming and Automated Reasonin"*. Reunion Island, November 2000. LNCS 1955.

[21] Rusinowich, M. and Turuani, M. *"Protocol Insecurity with Finite Number of Sessions is NP-complete"*. 14th IEEE Computer Security Foundations Workshop June 11-13, 2001 Cape Breton, Nova Scotia, Canada.

[22] Ryan, P. and Schneider, S. "*Modeling and Analysis of Security Protocols: the CSP Approach*". Addison-Wesley, 2001, ISBN 0 201 67471 8.

[23] Schneider, S. *"Formal analysis of a non-repudiation protocol"*. In Proceedings of the 11th IEEE Computer Security Foundations Workshop (CSFW '98), pages 54-65, June 1998.

[24] Woo T.Y.C. and Lam, S.S. *"A Semantic model for authentication protocols"*. IEEE Symposium on Research in Security and Privacity, 1993.

[25] Zhou, J. and Gollmann, D. *"Towards verification of non-repudiation protocols"*. In Proceedings of 1998 International Refinement Workshop and Formal Methods Pacific, pages 370-380, Canberra, Australia, Sept. 1998. Springer.

Estudio estadístico de sistemas de revocación de certificados mediante árboles de Merkle 2-3

Esteve Pallarès, Jordi Forné, J. L. Muñoz
Departamento de Ingeniería Telemática. Universidad Politécnica de Cataluña.
Jordi Girona 1 y 3. Campus Nord. UPC.
08034 Barcelona.
E-mail: {esteve, jforne, jlmunoz}@entel.upc.es

Abstract. *The revocation of certificates is one of the major costs in the whole PKI. Traditional revocation systems such as CRL and OCSP require high bandwidth or high computational load, respectively. A family of revocation systems based on Merkle hash trees have been proposed in the literature. Among them, the Authenticated Dictionary based on 2-3 trees seems to be one of the best options. In this paper, we present a statistical study of revocation systems based on 2-3 Merkle trees.*

1 Introducción

La seguridad en las comunicaciones es un requisito indispensable para una utilización fiable y comercial de Internet. En concreto se hace necesario proporcionar servicios de seguridad tales como la autenticación de usuarios, control de acceso, confidencialidad, integridad de la información y no repudio.

La criptografía de clave pública es ampliamente utilizada como tecnología necesaria para ofrecer dichos servicios. Se basa en la utilización de un par de claves, una pública (conocida por todo el mundo) y otra privada (mantenida en secreto por su propietario). La gran ventaja respecto a la criptografía simétrica es que la clave pública puede ser transmitida en claro por la propia red de comunicaciones, lo que facilita enormemente su distribución en entornos abiertos. Aunque la transmisión de la clave pública no requiere confidencialidad, si requiere autenticidad, es decir, el receptor debe poder verificar que dicha clave pertenece al usuario legítimo.

Para autentificar las claves públicas en un entorno distribuido, se hace necesario el uso de terceras partes de confianza (TTPs) que generen documentos firmados digitalmente que sirven de soporte a la transmisión auténtica de dichas claves. Estas terceras partes se denominan autoridades de certificación (CAs), y el documento generado recibe el nombre de certificado digital. Un certificado digital une criptográficamente el identificador de una entidad (que puede ser el nombre de un individuo, su dirección de correo electrónico, la URL de un servidor web, la dirección IP de un *host* o *router*, etc.) con la clave pública asociada. A la infraestructura que soporta la gestión de los certificados durante su periodo de vida se la conoce como Infraestructura de Clave Pública o PKI.

La ITU estandariza el formato de los certificados digitales en la recomendación X.509 [1], donde se contemplan entre otros los siguientes campos: nombre del emisor del certificado, nombre del posesor del certificado, clave pública del posesor, número de serie del certificado, fecha de activación y fecha de expiración del certificado.

El período de validez típico de un certificado digital oscila entre varios meses y varios años. Una duración corta conlleva un alto coste debido al proceso de expedición y entrega de los certificados a los usuarios, mientras que un período de duración largo aumenta el riesgo de utilización fraudulenta de un certificado. En la práctica se usan periodos de validez relativamente largos combinados con mecanismos de revocación de certificados. La revocación es el mecanismo mediante el cual el emisor del certificado puede invalidar su validez antes de su fecha de caducidad. Las posibles causas de revocación incluyen [2]: la pérdida o compromiso de la clave privada asociada, un cambio en los derechos del propietario, un cambio en la relación entre este y el emisor del certificado o simplemente como precaución frente al criptoanálisis.

La PKI presenta varios problemas, sobretodo cuando se quiere extender sus soluciones a una gran cantidad de usuarios, es decir, presenta problemas de escalabilidad. Entre estos problemas uno de los más graves es la revocación de certificados. La Figura 1 presenta el escenario típico de un sistema de revocación de certificados. La revocación de un determinado certificado la inicia una Entidad Final autorizada que puede ser el propietario del certificado, un representante autorizado o la CA emisora.

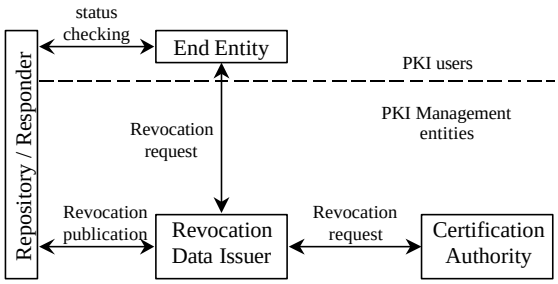


Figura 1. Modelo de referencia de un sistema de revocación de certificados

Cualquiera de las entidades anteriores puede generar una petición de revocación y esta petición a su vez genera un registro en la base de datos del *Revocation Data Issuer* (RDI). El RDI es una TTP responsable de transformar las peticiones de revocación en Datos de Estado (SD), los cuales tienen un formato adecuado para ser distribuidos a los usuarios (Entidades Finales). Sin embargo, los usuarios no obtienen los SD directamente del RDI, en realidad, el RDI publica esta información en servidores que pueden ser de dos tipos: repositorios o *responders*. La diferencia entre uno y otro es que mientras que un repositorio no es una TTP, un responder sí que lo es. En cualquier caso, la principal función de ambos, es responder a las Entidades Finales las peticiones que conciernen al estado de los certificados (*status checking*).

Las políticas de Revocación de Certificados definen la forma en la cual un usuario puede obtener información relativa al estado de un determinado certificado digital. El análisis de la forma en la cual debe ser proporcionada dicha información al usuario se ha realizado en los estudios mencionados anteriormente, de los cuales podemos concluir que las políticas de revocación de certificados se pueden agrupar en dos grandes grupos:

- Grupo de políticas de revocación basadas en distribución de listas o *Off-Line*: este tipo de políticas se caracteriza por el envío de una lista de certificados revocados o CRL al usuario, mediante la cual el usuario debe de verificar el estado del certificado.

- Grupo de políticas de revocación en línea o *On-Line*: este tipo de políticas se caracteriza por el envío de información sobre la validez de un determinado certificado o certificados que el usuario solicita en un instante en particular. Pertenecen a este grupo OCSP (*Online certificate Status protoco*l) y las basadas en árboles de Merkle :CRT (*Certificate Revocation Tree*)[3] y AD (*Autenticated Dictionary*) [4].

En [5] se presenta un estudio comparativo de los principales sistemas de revocación utilizados hasta la fecha, es decir, CRL y OCSP. En ambos casos es sencillo modelar analíticamente la longitud de la respuesta y en función de ella estimar el ancho de banda necesario para transmitir la información de revocación. De este estudio, así como de otros similares, puede deducirse que, en general, los sistemas basados en CRL requieren un mayor ancho de banda que los basados en OCSP (dado que la cantidad de información que ocupa una lista es considerablemente mayor que una respuesta autenticada sobre el estado de un certificado único), pero, en cambio, los basados en OCSP requieren una carga computacional mucho mayor para el RDI (ya que debe firmar individualmente cada respuesta). Los sistemas basados en árboles de Merkle presentan características intermedias, siendo una buena opción cuando se requieren anchos de banda ostensiblemente menores que en CRL, combinados con una carga computacional en el servidor ostensiblemente menor que en OCSP.

En los sistemas basados en árboles de Merkle la evaluación de la longitud de la respuesta no es obvia, debido principalmente a la diversidad de topologías que pueden adoptar los árboles de revocación.

El principal objetivo de este artículo es realizar un estudio estadístico de las diferentes topologías y longitudes de respuestas a que dan lugar los sistemas de revocación basados en árboles de Merkle 2-3. Los resultados aquí obtenidos permitirán evaluar el ancho de banda necesario para sistemas de revocación basados en árboles 2-3 y AD.

El resto del artículo se organiza de la siguiente forma: la sección 2 introduce los principales sistemas de revocación basados en árboles de Merkle, es decir, CRT y AD; la sección 3 muestra el entorno de simulación empleado; en la sección 4 se presentan los principales resultados obtenidos y en la sección 5 las conclusiones.

2 Sistemas de Revocación basados en árboles de Merkle

Los sistemas de revocación basados en los árboles de Merkle [6] aprovechan las propiedades de las funciones de Hash unidireccionales, el cómputo de las cuales es al menos 1000 veces más rápido que la generación de una firma digital. Al aplicar una función de Hash sobre un mensaje de longitud variable, se obtiene una salida de longitud fija denominada huella del mensaje. Además, una función de Hash se denomina unidireccional cuando a partir de una huella, no es computacionalmente posible obtener un mensaje que la genere.

Los sistemas de revocación basados en árboles de Merkle se construyen de la siguiente manera. En los elementos del nivel más bajo del árbol, denominados hojas, se almacenan los identificadores de los certificados revocados (c_i) conjuntamente con sus

huellas ($H_{0,i}=h(c_i)$), tal como se muestra en las Figuras 2 y 3. Las huellas pertenecientes a varias de estas hojas se concatenan y se les aplica nuevamente la función de Hash para obtener una nueva huella la cual se almacenará en un nodo en el nivel inmediatamente superior (p.e. $H_{1,0}= h(H_{0,0}|H_{0,1})$) Esta operación se repite sucesivamente hasta obtener en el último nivel del árbol un sólo nodo denominado raíz (H_{raiz}). En función de cómo se agrupen las hojas y los nodos podemos distinguir entre árboles CRT y AD.

En el caso de CRT los nodos se combinan en parejas por lo que se dice que CRT es un árbol binario, mientras que en AD cada nodo interno tiene dos o tres hijos, este tipo de árbol se llama árbol 2-3. La ventaja de los árboles 2-3 frente a los árboles binarios es que se pueden añadir y eliminar elementos de forma eficiente. Ambas operaciones se pueden realizar en $o(\log(n))$ [7] siendo n es el número de hojas del árbol. El nodo raíz es la única parte de los datos del árbol que se firma junto con un periodo de validez. Para demostrar la pertenencia de una determinada hoja al árbol se proporcionan la huella del nodo raíz firmada, así como las huellas necesarias para que el usuario pueda computar la huella del nodo raíz [8].

Por ejemplo, en la Figura 3 para demostrar que el certificado con identificador c_4 pertenece al árbol es necesario incluir en la respuesta al usuario los valores H_{05}, H_{10} y H_{11} con los que podrá calcular H_{raiz} utilizando la expresión (1)

$$H_{raiz} = h(H_{10} \mid H_{11} \mid h(h(c_4) \mid H_{05})) \qquad (1)$$

Así mismo es posible demostrar que un certificado no pertenece al árbol demostrando la existencia de 2 certificados adyacentes que sí pertenezcan al árbol y que además se cumpla que el certificado buscado esté entre ambos valores.

3 Detalles de la simulación

Se ha llevado a cabo una simulación mediante la generación aleatoria de árboles de Merkle 2-3. Dichos árboles se han creado a partir del subconjunto de certificados revocados los cuales han sido obtenidos aleatoriamente y de manera uniforme entre la población total de certificados. En todas las simulaciones se ha considerado que el conjunto de certificados revocados sea un 10% de la población total. Para cada uno de los árboles generados se han simulado múltiples consultas para obtener una estadística de la cantidad de nodos que contribuyen en la respuesta de dichas consultas. Cada uno de los identificadores consultados también se ha elegido de manera uniforme entre el total de la población, de manera que el 90% de los casos se realizó una consulta de un certificado no revocado y el 10% restante eran de certificados revocados.

Un árbol Merkle 2-3 es un conjunto de nodos interconectados entre sí de manera que cada nodo tiene un único padre y dos o tres hijos (izquierdo, central y derecho). El padre siempre es otro nodo mientras que los hijos pueden ser otros nodos o identificadores de certificados revocados. Existe un nodo que no tiene padre al que llamamos nodo raíz y del cual parten el resto de nodos. Los nodos más alejados del nodo raíz, a los que llamaremos nodos extremos, son los que contienen los identificadores de los certificados, los cuales están ordenados de menor a mayor dentro de la estructura del árbol. De esta forma un nodo extremo cubrirá un rango de identificadores de certificados revocados siendo el valor mínimo el identificador del hijo situado más a la izquierda y el máximo el de la derecha. Cualquier otro nodo del árbol también tiene un par de campos en los que se indican el valor máximo y mínimo del rango que cubre. De esta manera el campo del valor mínimo se corresponde con el mínimo del hijo de la izquierda y el valor máximo el máximo del hijo situado más hacia la derecha. Se ha utilizado el convenio de que cuando un nodo sólo tiene dos hijos, éstos ocupan las posiciones izquierda y central, dejando vacante la posición derecha para el caso que sea necesario insertar un nuevo nodo o identificador.

Inicialmente el árbol está formado por el nodo raíz y los identificadores de los certificados mínimo y máximo. Dichos identificadores no corresponden a certificados reales sino que simplemente acotan la población de certificados. Por ejemplo, en una población de certificados numerados del 1 al 100.000 los identificadores de los certificados mínimo y máximo serian 0 y 100.001 respectivamente. A partir de aquí se va construyendo el árbol insertando los identificadores de los nuevos certificados revocados.

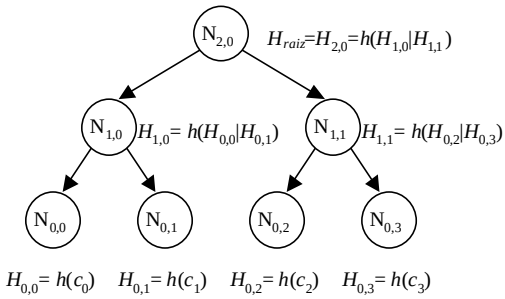


Figura 2. Árbol de Merkle

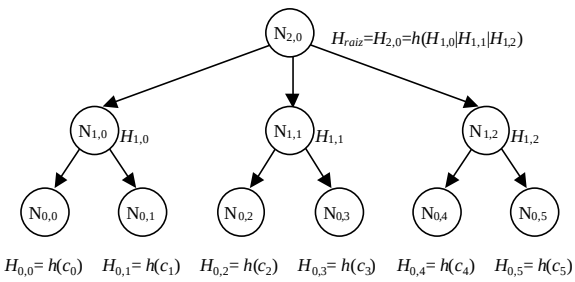


Figura 3. Árbol AD.

Primero se busca la posición dentro del árbol en la que se va a insertar el identificador. Para ello se utiliza un algoritmo de búsqueda, el cual partiendo del nodo raíz y comprobando el rango de identificadores asociado a cada nodo descendiente, encuentra la ruta hasta el identificador inmediatamente inferior que sí pertenece al árbol. Dicho algoritmo se describe más adelante. Una vez encontrado el identificador inmediatamente inferior se procede a la inserción. Para ello se comprueba el número de hijos que tiene el nodo del que cuelga dicho identificador pudiendo producirse dos situaciones:

a) Dicho nodo sólo tiene 2 hijos. En este caso se inserta el identificador en dicho nodo en la posición que le corresponda (central o derecha) y se actualiza el campo del valor máximo del rango asociado.

b) El nodo tiene 3 hijos. En este caso será necesario crear un nodo hermano, repartiendo los cuatro identificadores de certificado entre ambos nodos (dos para cada uno). Los identificadores ocuparan las posiciones izquierda y central de cada nodo, de manera que queden ordenados. También es necesario actualizar el rango asociado a cada nodo.

En este último caso vuelve a ser necesario utilizar el mismo procedimiento de inserción para insertar el nuevo nodo hermano en el árbol. De este modo se irían insertando nodos en los niveles superiores hasta encontrar un nodo con dos hijos o hasta que se llega al nodo raíz. En este último caso, después de crear un hermano al nodo raíz, es necesario crear un nuevo nodo padre a ambos, el cual será el nuevo nodo raíz.

Así pues se define un procedimiento de inserción el cual se utiliza de manera recurrente:

inserta(N_i,N_j) //inserta N_i en el nodo N_j
{
 if N_j tiene 2 hijos {
  ubica N_i en N_j
 }
 else{ // tiene 3 hijos
  $N_{hermano}$=CreaNuevoNodo()
  ubica N_j.*left*, N_j.*middle*, N_j.*right*, N_i en N_j, $N_{hermano}$
  if N_j =*raiz* {
   raiz= CreaNuevoNodo()
   ubica N_j, $N_{hermano}$ en *raiz*
  }
  else {
   inserta($N_{hermano}$, N_j.*padre*)
  }
 }
}

Para insertar un identificador de certificado en el árbol, se busca el padre del identificador inmediatamente inferior que sí pertenece al árbol y se inserta en dicho padre.

N_i=BuscaEnArbol(*Id*)
inserta (*Id*, N_i.*padre*)

Una vez generado el árbol, será posible comprobar si un certificado está revocado o no, mediante el procedimiento de búsqueda. Dicho procedimiento parte del nodo raíz y va avanzando a través del árbol hasta encontrar el nodo buscado o el inmediatamente inferior que sí pertenece al árbol. Para ello utiliza los valores máximo y mínimo del rango asociado a cada nodo. Además durante este proceso se genera una lista con aquellos nodos que contribuyen al mensaje de respuesta a una consulta de certificado revocado. Dichos nodos son aquellos que son hermanos de los nodos que forman la ruta de búsqueda. El procedimiento de búsqueda seria el siguiente:

BuscaEnArbol(*Id*) //
{
 N=*raiz*
 While *N* not nodoextremo {
  SiguienteNodo= hijo de *N* cuyo rango incluye a *Id*
  incluye *SiguienteNodo* en *ruta*
  incluye hermanos de *SiguienteNodo* en *respuesta*
  N= *SiguienteNodo*
 }
 certificado=Hijo de *N* igual o inmediato inferior a *Id*
 incluye *certificado* en *ruta*
 incluye hermanos de *certificado* en *respuesta*
 return *certificado*
}

Análogamente es posible crear un procedimiento que en caso de no encontrar el identificador buscado retorne el identificador del certificado inmediato superior, conjuntamente con la ruta a dicho certificado y el conjunto de nodos que contribuyen en la respuesta de la búsqueda de dicho certificado. De esta forma cuando se consulta si un identificador pertenece o no al árbol, inicialmente se utiliza el procedimiento de búsqueda que encuentra el propio identificador o el inmediato inferior. En caso de que el certificado pertenezca al árbol los únicos nodos que contribuyen en la repuesta son los obtenidos con este procedimiento. Si el certificado no pertenece al árbol será necesario encontrar los nodos que contribuyen a la respuesta asociada al identificador de certificado inmediatamente superior. En este caso la respuesta global estará formada por la unión de ambos conjuntos de nodos, es decir, los nodos que contribuyen a ambas respuestas.

4 Resultados obtenidos

Se han realizado simulaciones con distintas poblaciones de certificados, en concreto para 1.000, 5.000, 10.000, 50.000 y 100.000 certificados. En todas ellas se ha supuesto que el número de

certificados revocados es el 10% del total de la población. Para cada población se ha generado un total de 1.000 árboles, calculando la profundidad de los mismos y el número de nodos que los formaban. A su vez para cada árbol se han realizado 1.000 búsquedas de certificados, obteniendo en cada caso la longitud de la respuesta. En el cálculo de dicha longitud sólo se ha tenido en cuenta la cantidad de nodos que contribuyen en la repuesta, siendo la longitud real proporcional a dicha cantidad. Dicha cantidad dependerá de si la consulta corresponde a un certificado revocado o no, por ello se han separado los casos en los cuales el certificado pertenece al árbol y los que no.

4.1 Profundidad de los árboles.

La profundidad de un árbol es el número de jerarquías de nodos y certificados dentro del árbol, es decir, uno más el número de saltos necesarios que hay que dar para alcanzar un identificador partiendo desde la raíz. La profundidad mínima se obtiene para árboles ternarios, en este caso se puede calcular a partir de la cantidad de certificados revocados tal como se muestra en (2).

$$profundidad = 1 + \lfloor \log_3 revocados \rfloor \quad (2)$$

La profundidad máxima se corresponde con los árboles binarios y se obtiene a partir de la expresión (3).

$$profundidad = 1 + \lfloor \log_2 revocados \rfloor \quad (3)$$

Como se observa en la Tabla 1, fijada una población de certificados revocados, la profundidad de los árboles obtenidos varia muy poco y dicha variación es mucho menor a medida que aumenta la población, ello es debido a la variación logarítmica de este parámetro. En las pruebas realizadas para una población total de 10.000 certificados todos los árboles fueron de profundidad 11 excepto uno de profundidad 12, y para la población de 100.000 certificados todos los árboles fueron de profundidad 12.

4.2 Número de nodos de los árboles

También se ha realizado un estudio referente al número de nodos de los árboles. Para cada población se ha calculado el número medio de nodos de los árboles generados con el porcentaje de certificados revocados. En la Figura 4 se muestran los histogramas del número de nodos para las poblaciones de 1.000, 10.000 y 100.000 certificados, que se corresponden con 100, 1.000, y 10.000 certificados revocados respectivamente. En la Tabla 2 se muestra el número medio de nodos y su varianza. Se observa que el número medio de nodos se aproxima al 74.6% de los certificados revocados, siendo la aproximación mejor a medida que aumenta el número de certificados.

Tabla 1. Profundidad de los árboles en función de la población

población	mínimo	máximo	Valor medio
1.000	6	8	6,42
5.000	7	10	8,18
10.000	8	11	9,01
50.000	9	14	11,00
100.000	10	15	12,00

Tabla 2. Cantidad de nodos de los árboles en función de la población

población	Valor medio	Varianza
1.000	75,345	12,139975
5.000	373,465	54,616775
10.000	746,410	97,779900
50.000	3731,366	473,060044
100.000	7460,241	1015,450919

4.3 Longitud media de la respuesta

Se ha analizado la longitud de la respuesta a las distintas peticiones haciendo una distinción entre los casos en los cuales el certificado buscado pertenece al árbol y los casos en que no. En el primer caso la longitud de la respuesta depende de la profundidad del árbol y de la cantidad de hijos (2 o 3) de los nodos que hay en la ruta que va desde la raíz hasta el identificador buscado. En el segundo caso depende del número de hijos de los nodos que hay en las rutas que van desde la raíz hasta el identificador inmediato inferior y desde la raíz hasta el identificador inmediato superior. A su vez también depende de lo disjuntas que sean dichas rutas, es decir de la cantidad de nodos que tengan en común.

En la Figura 5 se muestran los histogramas de las longitudes de las respuestas para el caso de certificados revocados. Se observa que el valor mínimo de dicha longitud está acotado por la profundidad mínima del árbol, además los valores se distribuyen de forma similar a ambos lados del valor central. En la Tabla 3 se muestra dicho valor central y la varianza para las distintas poblaciones.

La Figura 6 muestra los histogramas de las longitudes de las respuestas para el caso en que el identificador buscado no pertenezca al árbol. En este caso el valor mínimo viene dado por la profundidad mínima del árbol mas uno, ya que la respuesta menos disjunta posible tendrá como mínimo una diferencia entre las dos rutas, esta diferencia es el certificado inmediato superior y el inmediatamente inferior. En los histogramas también se muestra que la dispersión de las longitudes es distinta a ambos lados del valor central. Ello es debido a que la respuesta mayor se producirá cuando las dos rutas sean muy disjuntas, en el peor caso sólo tendrían en común el nodo raíz.

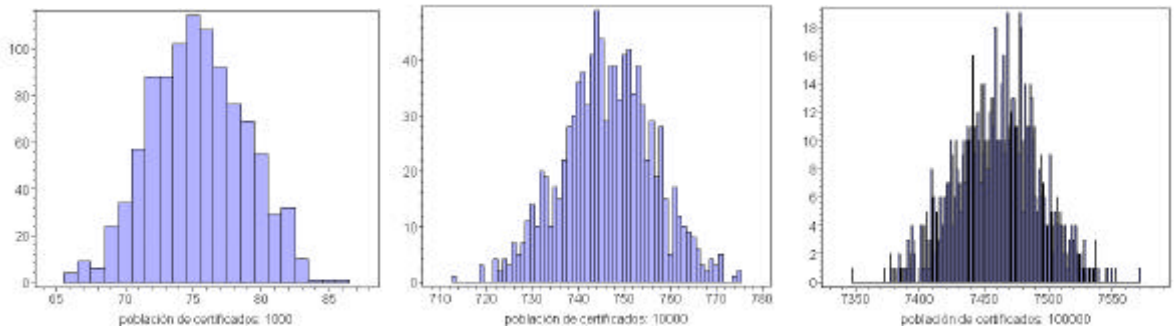


Figura 4. Histogramas del número de nodos de los árboles para distintas poblaciones

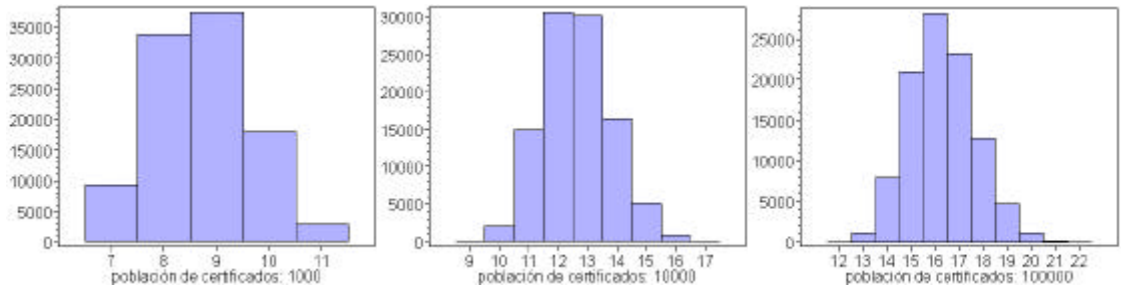


Fig. 5. Longitud de las respuestas para distintas poblaciones para certificados revocados (en número de nodos)

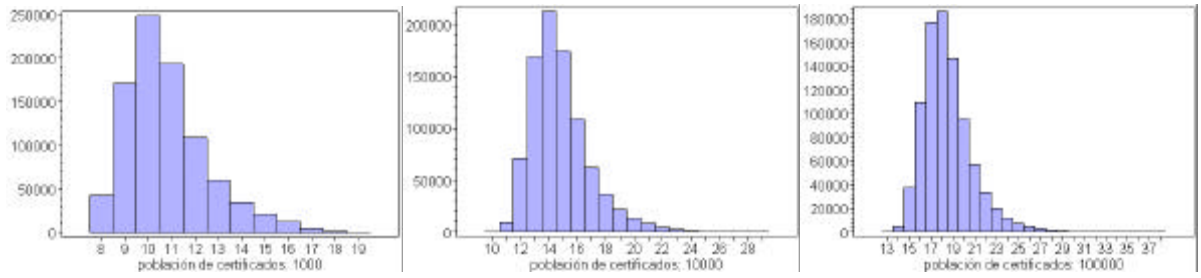


Fig. 6. Longitud de las respuestas para distintas poblaciones para certificados no revocados (en número de nodos)

Estas respuestas más largas son menos frecuentes por la propia estructura del árbol, lo cual hace se alejen del valor central. En la Tabla 4 se muestran los valores medios y la varianza de dichas longitudes.

Considerando todas las posibles consultas, tanto de certificados revocados como de certificados no revocados, se obtienen las distribuciones mostradas en la Figura 7. En éste caso se obtiene una distribución parecida a la de los certificados no revocados debido a que éstos son el 90% de la población total. Los valores medio y la varianza se muestran en la Tabla 5.

Tal y como ya se ha expresado, la longitud de las respuestas depende entre otros parámetros de la profundidad del árbol. A su vez ésta varia logarítmicamente respecto a la población de certificados revocados. Si representamos gráficamente la longitud media de las respuestas respecto a la población de certificados también se observa que se mantiene esta dependencia logarítmica. Ello se muestra en la Figura 8 en la cual el tamaño de la población se ha representado sobre un eje logarítmico.

Tabla 3. Longitud de la respuesta para identificadores de certificados revocados (en número de nodos)

población	Valor medio	Varianza
1.000	8,719326210	0,914688976
5.000	11,48081065	1,330388823
10.000	12,62404805	1,393439961
50.000	15,25623892	1,793654159
100.000	16,28052065	1,947011339

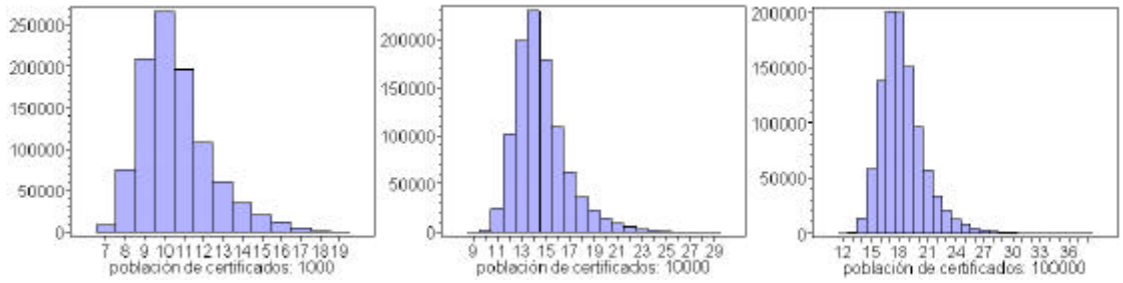


Fig. 7. Longitud de las respuestas para distintas poblaciones para todos los certificados (en número de nodos)

Tabla 4. Longitud de la respuesta para identificadores de certificados no revocados (en número de nodos).

población	Valor medio	Varianza
1.000	10,78287580	3,149708921
5.000	13,65313414	4,218127276
1.0000	14,81869387	4,493658686
50.000	17,47156967	5,104077889
100.000	18,50383333	5,301946983

Tabla 5. Longitud de la respuesta para identificadores de certificados en general (en número de nodos).

población	Valor medio	Varianza
1.000	10,57314900	3,311351224
5.000	13,43425300	4,354733332
10.000	14,59910200	4,617162794
50.000	17,25044200	5,214614805
100.000	18,28092400	5,411489706

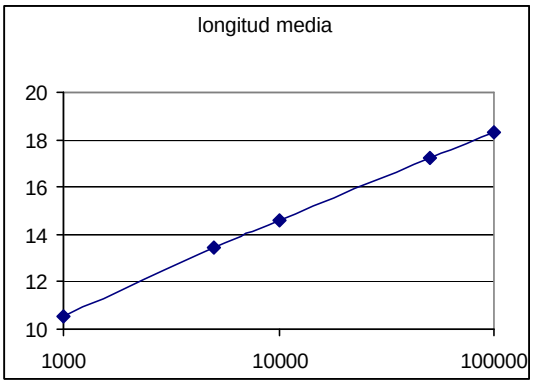


Figura 8. Longitud media de la respuesta (expresada en número de nodos) en función del logaritmo del tamaño de la población.

5 Conclusiones

En este articulo se han presentado las principales políticas de revocación de certificados. En concreto se ha estudiado la política *On-Line* basada en el árbol de Merkle 2-3 por ser un método bastante eficiente para la actualización de certificados. Para realizar dicho estudio se ha simulado la generación de este tipo de árboles y el procedimiento de consulta en los mismos para distintos tamaños de población de certificados.

Se ha observado que la profundidad de los árboles depende logarítmicamente con el número de certificados revocados, además dicha profundidad está acotada por los valores correspondientes a un árbol ternario y un árbol binario. A medida que la población aumenta es más difícil que dicha profundidad tome valores distintos a su valor medio. Por otro lado el número medio de nodos que forman el árbol es proporcional a la población de certificados revocados, pudiéndose aproximar al 74.6% de dicha población.

En cuanto a la longitud de las respuesta se han considerado dos casos, aquellas respuestas referentes a consultas sobre certificados revocados y las referentes a certificados no revocados. En ambos casos la longitud mínima de las respuestas está acotada por la profundidad de los árboles. Además para el caso de certificados no revocados se observa una dispersión de dicha longitud hacia la derecha de su valor central. Ello es debido a que la respuesta de un certificado no revocado es la unión de las respuestas de dos certificados revocados. Dependiendo de si dichas respuestas son muy disjuntas o no la longitud es mayor o menor, siendo menos probables los casos con longitudes largas.

Finalmente se ha estudiado la longitud media de todas las respuestas y se ha observado que la longitud media tiene un comportamiento logarítmico con el tamaño de la población.

Agradecimientos

Este trabajo ha sido financiado por el proyecto de investigación DISQET (CICYT TIC2002-00818).

Referencias

[1] I. R.X.509, "Information Technology Open Systems Interconnection – The Directory:Public Key and Attribute Certificate Frameworks", 1997.

[2] B. Fox and B. LaMacchia. Online Certificate Status Checking in Financial Transactions: The Case for Re-issuance. *International Conference on Financial Cryptography (FC99)*, number 1648, pages 104-117, February 1999.

[3] P.C. Kocher. "On certificate revocation and validation". *International Conference on Financial Cryptography (FC98). Lecture Notes in Computer Science*, number 1465, pages 172-177, February 1998.

[4] M. Naor and K. Nissim. "Certificate Revocation and Certificate Update". *IEEE Journal on Selected Areas in Communications*, 18(4):561-560, 2000.

[5] J.L.Muñoz, J. Castro, and J. Forné, "Estudio Comparativo de políticas de revocación de certificados: OCSP vs. Overissued-CRL", *III Jornadas de Ingeniería Telemática (JITEL 2001)*, pp. 413-420, Sept. 2001.

[6] R. Merkle, "A certified digital signature", Advances in Cryptology (CRYPTO89). Lecture Notes in Computer Science, no 435, pp. 234-246, Springer-Verlag, 1989.

[7] A. Aho, J. Hopcroft y J. Ullman, "Data Structures and Algorithms". Addison-Wesley, 1988.

[8] J. Muñoz, J. Forné, O. Esparza y M. Soriano, "Implementation of an Efficient Authenticated Dictionary for certificate Revocation", The Eigth IEEE Symposium on Computers and Communications (ISCC'2003), IEEE Computer and Communications Society. (pendiente de publicación).

Seguridad en WLAN: la Propuesta DARWIN

Josep Lluís Ferrer Gomila, Guillem Femenias Nadal, Magdalena Payeras Capellà
Departament de Ciències Matemàtiques i Informática. Universitat de les Illes Balears.
Carretera de Valldemossa, Km. 7.5, 07122 Palma de Mallorca
Teléfono: 971 17 13 90
E-mail: {dijjfg, guillem.femenias, mpayeras}@uib.es

Abstract. *Security services, an essential element for the successful implantation of any communications infrastructure, are not contemplated in an appropriate way in the current standards for broadband wireless LANs, namely IEEE 802.11 and HIPERLAN/2. A wrong design of the security architecture, a bad election of cryptographic algorithms and a lack of scalability, are among the criticism that these standards have received. In this paper, the security architecture adopted within the Spanish ICT (Information and Communication Technologies) project DARWIN (Demostrador Adaptativo Reconfigurable de una WLAN para entornos Interiores) is presented. The DARWIN approach, adopting the SSL model, incorporates all the basic security services, with a high degree of flexibility and scalability.*

1 Introducción

Durante los últimos años, el incremento en el volumen del tráfico de datos, la creciente popularidad de las aplicaciones multimedia y la tendencia a la convergencia de las comunicaciones inalámbricas y la tecnología Internet han motivado la evolución desde la segunda generación (2G) a la tercera generación (3G) de redes móviles y han propiciado la aparición de nuevos estándares WLAN, como IEEE 802.11 [6] y HIPERLAN/2 [3, 4]. Dentro de la iniciativa TIC (Tecnologías de la Información y las Comunicaciones) del MCyT, el proyecto DARWIN[1] (Demostrador Adaptativo Reconfigurable de una WLAN para entornos Interiores) tiene como objetivo la definición de un sistema de acceso radio WLAN flexible de banda ancha, basado en una plataforma de red IP.

Para el cumplimiento de los requisitos de seguridad exigidos por las capas superiores, de acuerdo con la aproximación *top down* adoptada en el proyecto, la capa DARWIN LC (*Link Control*) incorpora las funciones de seguridad. Los principales servicios considerados en la definición de la arquitectura de seguridad son: confidencialidad, integridad, control de acceso y autenticación (mutua o unidireccional). Otros servicios que también se considerarán (en DARWIN pero no en este artículo) son: no repudio en origen y en recepción, facturación, auditoría, y denegación de servicio.

El interfaz aéreo y la capa LC de DARWIN pueden considerarse como una evolución de IEEE 802.11 y HIPERLAN/2. De hecho, la arquitectura de seguridad de DARWIN se basa en el análisis de las correspondientes arquitecturas de IEEE 802.11 y HIPERLAN/2, e intenta resolver los problemas que han sido identificados en ambos estándares. Un diseño erróneo de la arquitectura de seguridad y una mala elección de los algoritmos criptográficos son algunas de las críticas que ha recibido [1] el estándar IEEE 802.11. Pensamos que HIPERLAN/2 también puede ser mejorado. En primer lugar, no contempla el servicio de integridad. En segundo lugar, no prevé la posible incorporación de infraestructuras de clave pública (PKI) basadas en certificados digitales. En tercer lugar, sólo permite el uso de RSA con tres posibles longitudes de clave (512, 768 y 1024) para el servicio de autenticación. Esto significa que si en un futuro cercano es necesario utilizar longitudes de clave mayores, el estándar HIPERLAN/2 deberá ser redefinido. Esta falta de escalabilidad puede ser generalizada a otros aspectos de seguridad de HIPERLAN/2. Finalmente, no parece apropiado que el servicio de confidencialidad se active antes de realizar el proceso de autenticación [7]. El procedimiento correcto debe empezar con la autenticación de las partes involucradas en la comunicación, para posteriormente realizar el intercambio de claves que se utilizarán para el cifrado de datos en el servicio de confidencialidad.

A continuación se presenta una propuesta que corrige los defectos de las propuestas anteriores. Como referencias para la definición de los mensajes y protocolos relacionados con los servicios de seguridad se han tomado HIPERLAN/2 y SSL [5]. SSL permite incorporar servicios de seguridad a las comunicaciones basadas en TCP/IP [2], y ha sido objeto de análisis en múltiples foros [8], llegando a la conclusión de que resulta ser un buen protocolo sin graves deficiencias.

La propuesta DARWIN permite la negociación de tres servicios de seguridad básicos: autenticación,

[1] Parcialmente financiado por la Conselleria d'Innovació i Energia de la CAIB, proyecto PRDIB-2002GC3-18, y por el Ministerio de Ciencia y Tecnología, proyecto CICYT TIC2001-0287.

integridad y confidencialidad. El diseño de los mensajes, que deben ser intercambiados entre la estación móvil (MT) y el punto de acceso (AP) permite la incorporación de nuevos algoritmos y nuevas longitudes de clave sin tener que redefinir el protocolo. Además, los algoritmos elegidos son suficientemente robustos (a diferencia de los escogidos en IEEE 802.11). Respecto a HIPERLAN/2, la propuesta DARWIN introduce, entre otras, las siguientes mejoras: permite proporcionar el servicio de integridad (no disponible en HIPERLAN/2), no establece límites en la longitud de las claves RSA, permite la autenticación basada en certificados de clave pública, la confidencialidad puede obtenerse mediante cifrado en flujo o cifrado en bloque, y la propuesta permite algoritmos de intercambio de claves diferentes del esquema Diffie-Hellman.

El resto del artículo se organiza de la siguiente manera. En la sección 2 se describe la arquitectura de seguridad de DARWIN. Los protocolos de negociación y autenticación se presentan en las secciones 3 y 4, respectivamente. La sección 5 describe el intercambio de secretos y la generación de las claves a partir de los secretos. En la sección 6 se presenta la incorporación de los servicios de integridad y confidencialidad. Finalmente, la sección 7 presenta las conclusiones.

2 Arquitectura de seguridad en DARWIN

El ámbito de DARWIN se limita a las capas inferiores del modelo de referencia OSI: capa física y capa de enlace. La capa de enlace distribuye sus funciones entre dos subcapas: la subcapa MAC (*Medium Access Control*) y la subcapa LC. A su vez, en la subcapa LC podemos encontrar un plano de usuario (DLC, *Data Link Control*) y un plano de control (RLC, *Radio Link Control*). En el plano de control se diferencian tres módulos: RRC (*Radio Resource Control*), AC (*Association Control*) y DLC *Connection Control*. En este contexto, las funciones de seguridad se han incorporado en el módulo AC (*Association Control*) del plano RLC, sobre la subcapa MAC, y por debajo de la capa IP.

Para determinar las fases de incorporación de los servicios de seguridad se ha seguido un modelo híbrido entre HIPERLAN y SSL, estableciendo cinco fases:

- Negociación: en una primera fase, durante el establecimiento de una asociación entre la estación móvil (MT) y el punto de acceso (AP), se negocian los servicios de seguridad que van a utilizarse y los algoritmos correspondientes.
- Autenticación: si así se ha decidido, MT y AP se autenticarán (unidireccional o mutuamente).
- Intercambio de material criptográfico: si se ha decidido activar los servicios de integridad y/o confidencialidad, debe procederse al intercambio de material criptográfico que permita generar las claves secretas necesarias y, si procede, los correspondientes vectores de inicialización.
- Integridad: si así se ha decidido, antes de transmitir información, debe calcularse un código de integridad criptográfico.
- Confidencialidad: si se ha negociado el servicio debe cifrarse la información, incluido el código de integridad en su caso.

Algunas funciones de seguridad no serán explicadas en este artículo, pero DARWIN sí las contempla, como son la renovación de claves, la seguridad en el proceso de *handover* y la desautenticación.

La notación que se utilizará es la siguiente:

x, y	concatenación de *x* e *y*
$H_f(m)$	*hashing* del mensaje *m* con el algoritmo *f*
$PR_i(m)$	cifrado del mensaje *m* con la clave privada de *i*
$PU_j(m)$	cifrado del mensaje *m* con la clave pública de *j*
$E_k(m)$	cifrado del mensaje *m* con la clave secreta *k*
$D_k(c)$	descifrado del criptograma *c* con la clave secreta *k*

3 Negociación

Los servicios de autenticación, integridad y confidencialidad son servicios opcionales. Por otra parte, son múltiples los algoritmos que pueden ser utilizados para cada servicio. Por tanto, es necesario que en la fase de asociación se negocie si van a utilizarse estos servicios, y con qué algoritmos concretos.

En DARWIN, tras el inicio del proceso de asociación y de asignación de un identificador MAC, empieza una negociación de la capacidad del enlace. Para el caso concreto de los servicios de seguridad, la estación móvil realiza una propuesta de los servicios deseados y de los algoritmos a utilizar, y el punto de acceso decide los servicios y algoritmos concretos. La secuencia de tramas a intercambiar es:

1.- MT $\rightarrow$ AP: serv-mt, alg-list, rand-mt

2.- AP $\rightarrow$ MT: serv-ap, alg-sel, rand-ap, [certif]

El argumento *serv-mt* enviado por la MT al AP es un octeto que contiene información sobre los servicios de seguridad deseados según se especifica a continuación (b_0 es el bit menos significativo del octeto):

b_0	el valor 1 indica que la MT solicita el servicio de integridad

b_1 el valor 1 indica que la MT solicita el servicio de confidencialidad

b_2 el valor 1 indica que la MT solicita que el AP se autentique

b_3 el valor 0 indica que la MT solicita una conexión anónima, mientras que el valor 1 indica que está dispuesto a autenticarse

b_4 el valor 1 indica que la MT dispone de un certificado de clave pública

b_5 el valor 1 indica que la MT necesitará el certificado de clave pública del AP

b_6-b_7 reservados para uso futuro

A continuación tenemos el argumento *alg-list*. Se trata de una lista de algoritmos para autenticación, intercambio de claves, confidencialidad e integridad, con un máximo de 16. Cada elemento de la lista contiene información de un algoritmo de autenticación, uno de intercambio de claves, uno de confidencialidad y, finalmente, uno de integridad.

Como algoritmos de autenticación e intercambio de claves está previsto utilizar RSA, Diffie-Hellman (DH) y clave precompartida. En función de como finalice la negociación, y para los algoritmos RSA y DH, las partes se verán obligadas a enviar un certificado de clave pública. El AP puede solicitar un certificado de firma RSA a la MT a efectos de autenticación o puede solicitar un certificado Diffie-Hellman. Todos los certificados Diffie-Hellman proporcionados por la MT deben utilizar los parámetros (grupo y generador) descritos por el AP.

Respecto de la confidencialidad, además de la opción de no utilizar este servicio, está previsto que pueda utilizarse el algoritmo DES, su variante con tres claves (3DES) y los algoritmos RC2 y RC4. En algunos casos el cifrado se realiza en flujo, mientras que en otros se realiza el cifrado en bloques encadenados (CBC, por *Cipher Block Chaining*). Finalmente, para el servicio de integridad pueden utilizarse los algoritmos MD5 y SHA.

En un futuro puede ampliarse el número de algoritmos de cada servicio, sin necesidad de tener que redefinir el contenido de las tramas. Esto confiere la propiedad de escalabilidad a la propuesta aquí realizada.

Finalmente, en la trama aparece un valor de 32 octetos, *rand-mt*. Este valor será utilizado para la generación de material criptográfico. Para obtener este valor se recomienda realizar la concatenación del valor de tiempo en formato *unix-gmt* (4 octetos) con un valor aleatorio generado de forma segura (de 28 octetos).

Respecto de la trama de respuesta del AP, el octeto *serv-ap* contiene información sobre los servicios de seguridad acordados, y tiene el siguiente significado:

b_0 el valor 1 indica que el servicio de integridad queda activado

b_1 el valor 1 indica que el servicio de confidencialidad queda activado

b_2 el valor 1 indica que el AP se autenticará

b_3 el valor 1 indica que la MT deberá autenticarse

b_4 el valor 1 indica que la MT deberá enviar un certificado de clave pública

b_5 el valor 1 indica que el AP envía en esta misma trama su certificado de clave pública

b_6-b_7 reservados para uso futuro

El siguiente argumento, *alg-sel*, contiene los algoritmos que el AP ha seleccionado. Si en la lista propuesta por la estación móvil no hubiera ninguna opción aceptable para el punto de acceso, debería rechazarse la asociación.

En el mensaje de respuesta aparece un valor de 32 octetos, *rand-ap*. Su utilidad y mecanismo de generación son análogos a la explicada para la MT.

De forma opcional el AP enviará un certificado de clave pública en el mensaje de respuesta, de forma coherente a los algoritmos escogidos, y a la información contenida en el octeto *serv-ap*.

4 Autenticación

La autenticación que introduce DARWIN es a nivel de enlace y no a nivel de usuario (que se correspondería con una autenticación de niveles más altos). La propuesta de estándar permite una autenticación mutua entre la estación móvil (MT) y el punto de acceso (AP), o la autenticación unidireccional de sólo uno de ellos (MT a AP o AP a MT). Además de la opción de no autenticarse, se prevén dos posibles métodos de autenticación: uno basado en clave precompartida y otro basado en criptografía de clave pública (concretamente RSA). El intercambio básico de información de autenticación es como sigue:

1.- MT $\rightarrow$ AP: id_type, id, [certif]

2.- AP $\rightarrow$ MT: $challenge_1$

3.- MT $\rightarrow$ AP: $challenge_2$, $response_1$

4.- AP $\rightarrow$ MT: $response_2$

En el primer mensaje la estación móvil indica el tipo de identificador de clave de autenticación que va a utilizar (ocupa 4 bits), seguido del valor de ese identificador. Cada MT debe tener asignado un identificador de clave de autenticación (de alguno de los tipos que se mostrarán a continuación) que debe presentar en este primer mensaje al AP. El AP, si es necesario, utilizará este identificador para recuperar la clave relativa al acceso. DARWIN prevé seis posibles tipos de identificadores: IEEE de 48 bits, IEEE extendido de 64 bits, NAI (*Network Access Identifier*), DN (*Distinguished Name*), identificador comprimido (un resumen MD5 del identificador) e identificador genérico (una tira de octetos no estructurada). Opcionalmente, la MT deberá enviar un certificado de clave pública.

Las informaciones $challenge_1$ y $challenge_2$ son dos valores aleatorios cada uno de 128 bits, y son utilizados por la otra parte para formular una respuesta a ese reto. La formulación de la respuesta depende del algoritmo acordado durante la fase de negociación.

Las claves de autenticación son de larga duración y pueden estar disponibles en la estación móvil y en el punto de acceso antes de empezar el proceso de autenticación. DARWIN también prevé la utilización del sistema de autenticación basado en certificados de clave pública. La estación móvil (MT) debe obtener la clave de autenticación del punto de acceso (AP), o bien gracias al certificado de clave pública que ésta hubiera enviado, o bien basándose en el identificador AP enviado en los canales de *broadcast*.

Si el proceso de autenticación acaba con éxito (las partes quedan autenticadas), es decir, si las respuestas a los retos son correctas, puede continuar el proceso de asociación. En caso contrario debe rechazarse la conexión DLC (*Data Link Control*).

Clave precompartida

Si se ha acordado utilizar el mecanismo de clave precompartida, las respuestas a los retos se calculan utilizando un código de autenticación resultado de aplicar una función de hash sobre el mensaje y un parámetro secreto. En DARWIN se utiliza la misma función que en HIPERLAN/2:

$$\text{HMAC-MD5}_k(m) =$$

$$= H_{MD5}((k \oplus opad), H_{MD5}((k \oplus ipad), m))$$

Donde el mensaje de entrada es *m*, *k* es el parámetro secreto, *opad* es el carácter 0x5c repetido 64 veces e *ipad* es el carácter 0x36 repetido 64 veces. Los valores concretos que puede tomar la variable *m* son los siguientes:

$$\text{response}_1 = \text{HMAC-MD5}_K(\text{challenge}_1, [PU_{MT}, PU_{AP},] \text{alg\_list}, \text{alg\_sel})$$

$$\text{response}_2 = \text{HMAC-MD5}_K(\text{challenge}_2, [PU_{MT}, PU_{AP},] \text{alg\_list}, \text{alg\_sel})$$

Los parámetros PU_{MT} y PU_{AP} son las claves públicas de la estación móvil y del punto de acceso, y se encuentran entre corchetes porque son opcionales (sólo aparecen si se dispone de ellas). Los parámetros *alg_list* y *alg_sel*, se corresponden con los parámetros de la fase de negociación. Para finalizar, *K* es la clave secreta precompartida entre la MT y el AP, y no debe ser de tamaño inferior a 128 bits.

RSA

En el caso de que se haya acordado utilizar el segundo tipo de algoritmo de autenticación, las respuestas se generarán a través del cálculo de una firma digital con el esquema RSA. DARWIN prevé un tamaño de clave RSA arbitrario (aunque se recomienda que esté comprendido entre 512 y 2048 bits). Las operaciones son como sigue:

$$\text{response}_1 = PR_{MT}(H_{MD5}(\text{challenge}_1, [PU_{MT}, PU_{AP},] \text{alg\_list}, \text{alg\_sel}))$$

$$\text{response}_2 = PR_{MT}(H_{MD5}(\text{challenge}_2, [PU_{MT}, PU_{AP},] \text{alg\_list}, \text{alg\_sel}))$$

5 Intercambio de claves

En este apartado se explicará el proceso de intercambio de material criptográfico y el método de generación de claves de sesión.

5.1 Intercambio para la generación de claves DARWIN

Una vez negociadas las opciones del enlace, y finalizado el proceso de autenticación, si se ha decidido utilizar cifrado y/o integridad de la información, la estación móvil y el punto de acceso deben intercambiar tramas con datos para generar las claves de cifrado simétrico y/o cálculo de códigos de integridad. En DARWIN pueden utilizarse tres posibles métodos para el intercambio de material criptográfico: Diffie-Hellman, RSA y clave precompartida.

Diffie-Hellman

Si se utiliza el método de intercambio de claves Diffie-Hellman, el proceso empieza de la siguiente manera:

/* características del enlace ya negociadas */

1.- MT $\rightarrow$ AP: $PU_{MT} = g^x \bmod n$

2.- AP $\rightarrow$ MT: $PU_{AP} = g^y \bmod n$

De los parámetros necesarios hay dos que son comunes para emisor y receptor:

g base generadora

n un número primo de valor elevado

Por otra parte cada usuario debe generar un valor secreto y aleatorio sólo conocido por él (en el esquema *x* e *y*).

En DARWIN hay unos valores de *g* y *n* que están fijados en las especificaciones y que pueden ser preconfigurados en la estación móvil y en el punto de acceso. También existe la posibilidad de que en la fase de autenticación hayan intercambiado los valores necesarios para poder llevar a cabo las operaciones indicadas previamente.

Una vez acabado el intercambio, el cifrado ya está activo y cada usuario dispone del material necesario para generar la clave de sesión (proceso que se explicará en el apartado siguiente). Este material secreto, denominado *presecreto maestro*, consiste en el resultado de realizar la siguiente operación:

$$PSM = g^{xy} \bmod n$$

Esta operación es muy sencilla para los dos usuarios implicados en el intercambio, pero computacionalmente imposible para cualquier usuario malicioso.

RSA

Si se utiliza el método de intercambio de claves RSA, el proceso empieza de la siguiente manera:

/* características del enlace ya negociadas */

1.- MT $\rightarrow$ AP: PU_{AP}(random)

2.- AP $\rightarrow$ MT: PU_{MT}(random)

La MT genera de forma segura un valor aleatorio de 48 octetos, *random*, y lo cifra con la clave pública RSA del AP. Para proporcionar mayor seguridad al intercambio, el AP envía el mismo valor *random* cifrado con la clave pública de la MT, con el objeto de que ésta pueda verificar que se ha recibido correctamente y de forma segura ese valor.

Como en el caso Diffie-Hellman, una vez acabado el intercambio, el cifrado ya está activo y cada usuario dispone del material necesario para generar la clave de sesión (proceso que se explicará en el apartado siguiente). Ahora el *presecreto maestro* es el valor *random*:

PSM = random

La operación de descifrado para obtener el valor *random* es muy sencilla para los dos usuarios implicados en el intercambio, pero computacionalmente imposible para cualquier usuario malicioso.

Clave precompartida

Si se utiliza el método de intercambio basado en clave precompartida, *K*, el proceso es de la siguiente manera:

/* características del enlace ya negociadas */

1.- MT $\rightarrow$ AP: E_K(AP, random)

2.- AP $\rightarrow$ MT: E_K(MT, random)

La MT genera de forma segura un valor aleatorio de 48 octetos, *random*, y concatenado con un identificador del AP, los cifra con la clave compartida con el AP. Para proporcionar mayor seguridad al intercambio, el AP envía el mismo valor *random* concatenado con un identificador de la MT, cifrados con la clave compartida, con el objeto de que ésta pueda verificar que se ha recibido correctamente y de forma segura ese valor.

Como en los casos anteriores, una vez acabado el intercambio, el cifrado ya está activo y cada usuario dispone del material necesario para generar la clave de sesión. Ahora el *presecreto maestro* es el valor *random*:

PSM = random

La operación de descifrado para obtener el valor *random* es muy sencilla para los dos usuarios implicados en el intercambio, pero computacionalmente imposible para cualquier usuario malicioso.

5.2 Generación de claves en DARWIN

Una vez que la MT y el AP han intercambiado el *presecreto maestro* (sea utilizando RSA, Diffie-Hellman o clave precompartida), cada uno de ellos debe generar de forma independiente (pero llegando a los mismos resultados) las claves de sesión para el cifrado a efectos de confidencialidad y/o las claves para el servicio de integridad. DARWIN utiliza claves distintas en los dos sentidos de la comunicación. Por ello es necesario generar dos claves y, según el algoritmo de cifrado escogido, dos vectores de inicialización.

En primer lugar debe generarse el *secreto maestro*, *SM*, a partir del *presecreto maestro*, y de los valores aleatorios, *rand-mt* y *rand-ap*, estos últimos intercambiados en la fase de negociación:

$$SM = H_f(PSM, H_f('A', PSM, rand\text{-}mt, rand\text{-}ap)), H_f(PSM, H_f('BB', PSM, rand\text{-}mt, rand\text{-}ap)), H_f(PSM, H_f('CCC', PSM, rand\text{-}mt, rand\text{-}ap))$$

En la anterior expresión (y en la posterior) *f* puede ser la función MD5 o SHA. Una vez que se ha calculado el secreto maestro, *SM*, debe borrarse de la memoria el *presecreto maestro*, *PSM*. A continuación debe

generarse el material de clave, *KB*, necesario para obtener las claves de sesión y los vectores de inicialización:

$KB = H_f(SM, H_f('A', SM, rand\text{-}ap, rand\text{-}mt)),$
$H_f(SM, H_f('BB', SM, rand\text{-}ap, rand\text{-}mt)),$
$H_f(SM, H_f('CCC', SM, rand\text{-}ap, rand\text{-}mt)),$
...

Este proceso se realizará hasta que se haya generado material suficiente. Posteriormente se particionará el material, *KB*, en el siguiente orden: *clave-MT*, *clave-AP*, *IV-MT*, *IV-AP*, *IC-MT* y *IC-AP*. El material sobrante será descartado. El tamaño de cada elemento depende de los algoritmos acordados durante la fase de negociación. Aparecen dos valores, *IC-MT* e *IC-AP*, que serán utilizados a efectos de cálculos de integridad como se explicará posteriormente.

Como en HIPERLAN, pudiera darse el caso de que alguna clave generada fuera una clave débil o semidébil. Si es así, debe descartarse y tomar el siguiente bloque de *KB*, y si fuera necesario, generar más material criptográfico.

Si el algoritmo de cifrado acordado es 3DES, las tres claves generadas deben ser diferentes, y ninguna de ellas debe ser débil o semidébil. Si no es así debe procederse de forma análoga a como se ha explicado en el párrafo anterior.

6 Integridad y confidencialidad

El cifrado de la información permite proporcionar el servicio de confidencialidad respecto de los datos transmitidos, mientras que las funciones *keyed-hash* permiten conseguir el servicio de integridad. Se trata de unas funciones de transporte de datos básicas de la capa DLC.

Si se negocian uno de los servicios o ambos durante la fase de asociación o *handover*, este cifrado y/o *hash* se empezarán a utilizar inmediatamente después de realizar el necesario intercambio de "claves" durante los procedimientos de asociación o *handover*. Las tramas de información se cifran y se protege su integridad completa e individualmente.

Todas las tramas se protegen utilizando los algoritmos de cifrado y de integridad definidos en el *alg-sel* vigente. Siempre hay un *alg-sel* activo, que inicialmente será de contenido nulo.

Cuando finaliza el proceso de negociación, las dos partes comparten secretos que se utilizarán para cifrar las tramas y calcular los códigos de integridad criptográficos de su contenido. Las técnicas para realizar las operaciones de cifrado y códigos de integridad quedan definidas por el *alg-sel*.

6.1 Integridad

El cálculo del código de integridad se realiza como se indica a continuación:

$IC = H_f(IC\text{-}sec, pad2, H_f(IC\text{-}sec, pad1, seq\text{-}num, inf))$

En la anterior expresión, *f* puede ser la función MD5 o SHA (la que se haya acordado en la fase de negociación). *IC-sec* fue generado a partir del material secreto, y la MT y el AP disponen de su correspondiente valor (*IC-MT* y *IC-AP*, respectivamente). El campo *seq-num* es el número de secuencia para esta trama. El argumento *inf* es la información a ser protegida. Finalmente, *pad1* y *pad2* toman los siguientes posibles valores:

pad1 es el carácter 0x36 repetido 48 veces para MD5 y 40 veces para SHA

pad2 es el carácter 0x5c repetido 48 veces para MD5 y 40 veces para SHA

Puede comprobarse que por las características de la función de cálculo del código de integridad, también puede obtenerse indirectamente un segundo servicio: la autenticidad de las partes.

6.2 Confidencialidad

DARWIN prevé el posible uso de DES, 3DES, RC4 y RC2 para permitir proporcionar distintos niveles de seguridad. El algoritmo 3DES se utiliza en su modo EDE (*Encryption - Decryption - Encryption*), es decir, primero se cifra la información con la primera clave, el resultado se descifra con la segunda clave, y finalmente el anterior resultado se cifra con la tercera clave:

$$c = E_{clave3}(D_{clave2}(E_{clave1}(m)))$$

Las operaciones de cifrado pueden realizarse en modo flujo (*stream cipher*) o en modo bloque encadenado (*CBC block cipher*). A continuación se describirán los dos posibles casos.

Cifrado en flujo

La operación de cálculo del código de integridad, si debe realizarse, es previa al cifrado. El cifrado en flujo se realiza sobre la trama entera, incluyendo, en su caso, el código de integridad calculado. Para los cifradores en flujo que no utilizan un vector de sincronización (como RC4), el estado del cifrador en flujo del final de una trama se utiliza en la siguiente trama.

Para realizar el cifrado de los datos es necesario, además de esta información y de la clave de sesión (*K*, que tomará el valor *clave-MT* o *clave-AP*, según sea la MT o el AP el que realice la transmisión), un

vector de inicialización (*IV*, que tomará el valor *IV-MT* o *IV-AP* para la MT y el AP, respectivamente).

Cifrado en bloque

Antes de cifrar la información, como en el cifrado en flujo, se realiza, si así se ha negociado, el cálculo del código de integridad, que también será cifrado. Antes de realizar la operación de cifrado se añade una secuencia de octetos que garantice que el bloque que debe ser cifrado sea múltiplo del tamaño de entrada del cifrador. Se añade un último octeto que indica el número de octetos de *padding* que se han introducido.

El vector de inicialización (IV) para el encadenamiento de bloques CBC del primer bloque a cifrar es el acordado en la fase de negociación. El IV para los subsiguientes bloques es el último texto cifrado del anterior bloque. Ejemplos de cifradores que pueden utilizarse en bloque son el RC2 y el DES.

7 Conclusiones

DARWIN incorpora todos los servicios de seguridad básicos, con un elevado grado de flexibilidad, permitiendo negociar los servicios deseados y los algoritmos a utilizar. El modelo que se ha adoptado en DARWIN permite la escalabilidad de los servicios de seguridad. DARWIN puede incorporar nuevos algoritmos y diferentes longitudes de clave, sin la necesidad de redefinir las especificaciones.

Como elemento claramente diferencial respecto de HIPERLAN e IEEE 802.11, DARWIN prevé utilizar, para el servicio de autenticación, la infraestructura de clave pública (cada día más extendida). Esta autenticación puede ser bidireccional, o únicamente en una dirección (MT a AP, o AP a MP). MT y AP pueden utilizar autenticación basada en certificados, pero el envío de estos certificados no es obligatorio (si las partes ya disponen de esta información).

En relación al intercambio de claves, además de Diffie-Hellman y clave precompartida, DARWIN establece un esquema basado en RSA.

Para el servicio de confidencialidad se han elegido algoritmos robustos (a diferencia de IEEE 802.11) y procesos de generación de claves seguros. DARWIN proporciona cifrado en flujo y cifrado en bloque, y establece claves separadas para las dos direcciones de la comunicación.

Se ha seguido como modelo el establecido en SSL, por tratarse de un protocolo ampliamente analizado y que ha demostrado ser un adecuado protocolo de seguridad.

Finalmente, el servicio de integridad no fue establecido en HIPERLAN/2, y en IEEE 802.11 está vinculado al servicio de confidencialidad. En la propuesta DARWIN, la integridad y la confidencialidad son dos servicios independientes. Es muy útil para la detección de usuarios maliciosos una vez ha finalizado la fase de autenticación (especialmente si las partes no utilizan el servicio de confidencialidad).

Referencias

[1] W. Arbaugh: "Wireless Research: 802.11 Security Vulnerabilities"; February 2002.

http://www.cs.umd.edu/~waa/wireless.html

[2] S. Blake-Wilson, M. Nystrom, D. Hopwood, J. Mikkelsen and T. Wright: "Transport Layer Security (TLS) Extensions"; Internet Engineering Task Force (IETF), Transport Layer Security Working Group, Internet Draft, February 2003.

http://www.ietf.org/internet-drafts/draft-ietf-tls extensions-06.txt

[3] ETSI TS 101 761-1 V1.3.1: "Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Data Link Control (DLC) Layer; Part 1: Basic Data Transport Functions"; December 2001.

[4] ETSI TS 101 761-2 V1.3.1: "Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Data Link Control (DLC) Layer; Part 2: Radio Link Control (RLC) sublayer"; January 2002.

[5] A. Frier, P. Karlton and P. Kocher: "The SSL Protocol Version 3.0".

http://home.netscape.com/eng/ssl3

[6] IEEE Std 802.11b-1999/Cor 1-2001 (Corrigendum to IEEE Std 802.11b-1999): IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Amendment 2: Higher-speed Physical Layer (PHY) extension in the 2.4 GHz band— Corrigendum 1; IEEE Computer Society, November 2001.

[7] S. Kent and R. Atkinson: "Security Architecture for the Internet Protocol"; Internet Network Working Group, RFC-2401, November 1998.

http://www.ietf.org/rfc/rfc2401.txt

[8] D. Wagner and B. Schneier: "Analysis of the SSL 3.0 Protocol"; 2nd USENIX Workshop on Electronic Commerce, 1996.

LKH mejorado para gestión de claves en grupos multicast

Josep Pegueroles, Francisco Rico-Novella.
Departamento de Ingeniería Telemática. Universitat Politècnica de Catalunya.
Jordi Girona 1 y 3. Campus Nord, Mód C3, UPC. 08034 Barcelona
Teléfono: 934 016 012 Fax: 934 015 981
E-mail: josep.pegueroles@entel.upc.es / telfrn@entel.upc.es

Abstract*. The most important efficiency parameters in Multicast Security are storage, delivery and update of cryptographic keys. Traditionally these actions are performed by a centralized trusted third party called the Key Server (KS). Different works have been presented that address the issue of minimizing storage for KS and required bandwidth for updating keys. We present a method for multicast rekeying using a broadcast encryption technique and pseudo-random functions in order to reduce number of sent messages for rekeying and minimize the number of keys to store by the KS.*

1 Introducción

La tecnología multicast ha permitido superar las principales dificultades en cuanto a uso ineficiente de ancho de banda que presentaban los servicios de multivideoconferencia o vídeo quasi-bajo demanda. El estudio y estandarización de nuevos protocolos multicast para una transmisión fiable de la información permitirá el despegue comercial de estos servicios multimedia. Como en cualquier aplicación en red comercial, los aspectos de seguridad toman especial relevancia. En concreto, debe garantizarse que un usuario malintencionado no tenga acceso a los paquetes enviados a una dirección multicast. De igual forma, debe evitarse que se pueda suplantar la identidad de cualquiera de los emisores autorizados.

Es obvio que este tipo de comunicación debe limitarse sólo al grupo autorizado. El cifrado de la información que circula entre los miembros del grupo es, por tanto, un servicio esencial. Todos los miembros de un grupo multicast deben compartir la misma clave de sesión, ya que todos reciben "el mismo paquete cifrado" que viaja por la red y, por tanto, deben ser capaces de descifrarlo. De este modo, el punto crítico de seguridad en multicast es la gestión de claves de grupo o GKM (*Group Key Management*).

La GKM estudia la forma de distribuir y actualizar claves criptográficas durante el tiempo de vida del grupo haciendo especial hincapié en los problemas derivados del dinamismo de dichos grupos multicast [1]. La clave de sesión debe actualizarse siempre que un miembro abandone o se dé de alta en el grupo. Así se consiguen los denominados servicios de seguridad hacia adelante y hacia atrás (*Forward and Backward Secrecy* -FS y BS). FS significa que ningún miembro saliente puede obtener información sobre cómo descifrar las comunicaciones de grupo posteriores a su baja. Equivalentemente, por BS se entiende que ningún miembro entrante puede obtener información sobre cómo descifrar comunicaciones de grupo anteriores a su alta [2].

En los últimos años han ido apareciendo en la literatura distintas propuestas de soluciones a estos problemas. Entre todas ellas, el algoritmo LKH o *Logical Key Hierarchy* [3] se ha impuesto como el estándar de facto. Este algoritmo consigue reducir el número de mensajes necesario para actualizar las claves de grupo a log_2N, donde N es el número de miembros del grupo multicast. A pesar de ello, LKH presenta aún algunos problemas si se desea implementarlo en aplicaciones reales. Entre estos problemas debe destacarse la memoria necesaria en el gestor de claves, la latencia y el proceso por lotes.

Como no todos los miembros reciben los paquetes de cambio de clave en el mismo instante, el emisor no puede usar la clave de cifrado actualizada hasta pasado cierto tiempo desde que se dio la orden. Este es el principal problema de latencia a solucionar. Por su parte, se entiende por proceso por lotes la actualización de claves cuando los miembros se dan de alta o de baja en grupos de más de uno. Este mecanismo resulta imprescindible en ciertos escenarios como *WebTV*, donde es habitual que los usuarios lleguen a ráfagas (por ejemplo al inicio de un evento deportivo), pero no está resuelto en LKH y añade cierta dificultad al algoritmo.

Este trabajo presenta una mejora del algoritmo LKH que minimiza la cantidad de memoria necesaria en el gestor del grupo para mantener las claves, reduce la latencia y es compatible con algoritmos de proceso por lotes.

El resto del artículo se estructura de la siguiente manera. En la sección 2 se profundiza en el problema de gestión de claves en multicast, se hace un breve repaso a las propuestas aparecidas en la literatura y se explica con más detalle el algoritmo LKH. La sección 3 presenta la propuesta de mejora

de dicho algoritmo. En un primer subapartado se propone una variación del algoritmo LKH para minimizar la cantidad total de claves a guardar por el gestor de claves. Seguidamente se añaden los mecanismos necesarios que consiguen reducir la latencia. En el apartado 4 se realiza el análisis de seguridad del algoritmo. La sección 5 se encarga de la evaluación. Finalmente, en la sección 6 se resumen las conclusiones y apuntamos líneas futuras de investigación.

2 Gestión de claves en multicast

Los aspectos más importantes a tener en cuenta cuando se requiere comunicaciones seguras en grupo son: la privacidad, la autenticación tanto de miembro como de grupo, y la gestión de políticas de grupo. La utilización de una clave común (un secreto compartido por todos los integrantes de la comunicación) solventa dos de dichos problemas: privacidad y autenticación de grupo. Sólo los poseedores de la clave de grupo tendrán acceso a los datos de la comunicación (privacidad) y sólo ellos podrán generar mensajes válidos (autenticidad de grupo).

La gestión de claves de grupo o *Group Key Management* (GKM) es de especial interés cuando se quiere dotar de seguridad a los servicios multimedia sobre redes multicast.

Uno de los problemas más importantes de los que se debe encargar la GKM es del acuerdo de la clave de grupo entre todos los participantes en la comunicación. Este proceso tiene sus puntos débiles en la escalabilidad (los grupos pueden llegar a tener miles o decenas de miles de miembros) y dinamismo (los grupos pueden cambiar su constitución varias veces durante una única sesión).

El IRTF y el IETF, a través de sus grupos de investigación y trabajo GSEC [4] y MSEC [5] dedican grandes esfuerzos a la estandarización de una arquitectura de seguridad multicast común. Una arquitectura que permita el desarrollo homogéneo de nuevos protocolos criptográficos para prestar los servicios de seguridad mencionados en [6].

Dicha arquitectura está formada por tres entidades funcionales distintas y sus respectivas interfaces (véase Fig. 1): el Servidor de Políticas de Grupo o *Policy Server*, encargado de establecer los criterios de seguridad a seguir por todos los miembros del grupo; el Gestor de Grupo y Servidor de Claves o *Group Controller* (GC) *and Key Server* (KS), que distribuye las claves entre los miembros del grupo y los da de alta y de baja; y los miembros, que pueden ser tanto emisores como receptores.

Normalmente GC y KS son una misma entidad (a partir de ahora los mencionaremos indistintamente), y junto con los miembros de grupo son las

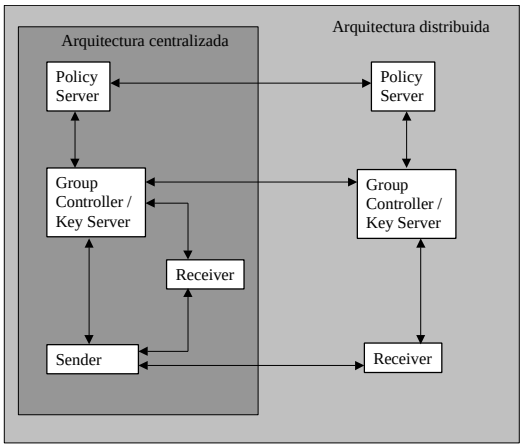


Figura 1. Arquitectura de referencia del grupo de Seguridad en Multicast del IRTF

entidades funcionales que intervienen propiamente en la gestión de claves en grupo.

2.1 El problema de Nueva Clave o *Rekeying*

Como se ha comentado anteriormente, el GC debe garantizar la existencia de confidencialidad “hacia adelante” y “hacia atrás” cada vez que se produce algún cambio en los integrantes del grupo. Estos servicios se denominan en la literatura con los términos acuñados del inglés: *Forward Secrecy* (FS) y *Backward Secrecy* (BS). Por FS se entiende que la clave de sesión no proporciona ninguna información útil sobre futuras claves de sesión, o lo que es lo mismo, que ningún miembro que abandone el grupo puede obtener información de la comunicación a posteriori de su baja. BS significa que la clave de sesión no proporciona información significativa sobre claves pasadas de la misma sesión, es decir, ningún nuevo miembro del grupo puede obtener información intercambiada dentro del grupo previamente a su adhesión. A su vez, todos los miembros del grupo deben estar seguros de que la información que están leyendo proviene única y exclusivamente de otro miembro autorizado del grupo.

La manera de conseguir los mencionados servicios de seguridad es cambiando la clave de sesión cada vez que se produce una baja o una alta en el grupo.

2.2 Estado del arte

Las propuestas que se han presentado en la literatura con el fin de solucionar el problema comentado han sido diversas. En [7] se discute el método trivial de establecer una conexión segura para cada miembro del grupo. Este método tiene una complejidad de O(N) y se ve inviable para grandes grupos. En [8] se realiza una mejora realizando subdivisiones de grupo basadas en la topología de la red, pero los problemas de gestión de los subgrupos que añade este método son

mayores que las ventajas en cuanto a reducción de mensajes que consigue. Finalmente [3,9,10] proponen estructuras de árboles de claves que cifran claves (o *Key Encryption Keys*). Estos métodos son los más aceptados actualmente y consiguen una reducción de los números de mensajes hasta llegar a órdenes logarítmicos, O(logN). En el siguiente apartado se describen con más detalle los métodos con árboles lógicos de claves y se presenta el LKH como su mayor exponente.

2.3 Árboles lógicos de claves: LKH

Los esquemas basados en árboles lógicos de claves utilizan dos tipos de claves de cifrado: Claves de Sesión o *Session Encryption Keys* (SEK) y claves que cifran claves *o Key Encryption Keys* (KEK).

Las SEKs se utilizan para cifrar los datos que se intercambian dentro de un grupo multicast, por ejemplo flujos de vídeo en una multivideoconferencia. Las KEKs se utilizan para cifrar las claves (o secretos) que los distintos miembros del grupo van a necesitar para obtener la SEK. Normalmente las KEKs se organizan en árboles binarios. La raíz de dichos árboles son claves que conocen todos los miembros de un grupo multicast seguro. Cada uno de esos miembros se "sitúa" en una hoja de dicho árbol, cuya clave sólo conoce el miembro que reside en ella.

Denominaremos a los distintos nodos del árbol siguiendo el criterio (nivel del nodo, posición en el nivel). De esta forma, el nodo raíz será el (1,1); los hijos de éste se corresponderán con (2,1) y (2,2) y así sucesivamente. En la Fig. 2a se muestra un ejemplo de árbol lógico de claves. La clave correspondiente al nodo (X,Y) la denominaremos $K_{(X,Y)}$.

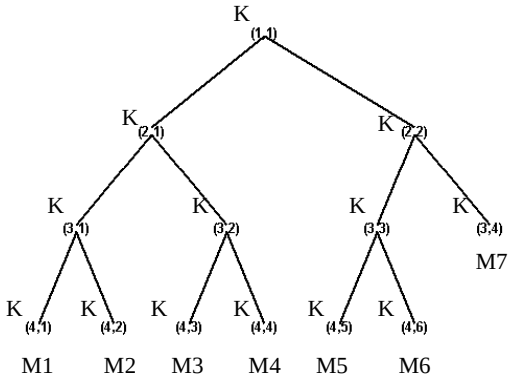
Considere el grupo de 7 usuarios de la Fig. 2a. El árbol tiene 13 nodos, cada nodo se corresponde con una KEK. Los miembros se "localizan" en los nodos hoja. Las claves de las hojas son sus claves únicas (secretos sólo compartidos entre ellos y el KS), mientras que $K_{(1,1)}$ se revela a los 7 usuarios. Además, cada una de las claves en los nodos intermedios se reparte a los usuarios hijos del nodo. Por ejemplo, $K_{(3,1)}$ se revela sólo a los usuarios en las hojas (4,1) y (4,2), y $K_{(2,2)}$ se revela a los usuarios en los nodos (4,5), (4,6) y (3,4).

El método de gestión de claves LKH opera del siguiente modo. Considere de nuevo la Fig.2a con N=7 miembros (M1..M7) y servidor de claves centralizado (KS). Cada miembro conoce un subconjunto de claves del árbol, gestionadas por el KS. El subconjunto de KEKs de cada usuario debe permitirle recuperar la nueva SEK cada vez que ésta se cambie. Un miembro genérico (M_j) almacena el subconjunto de claves correspondiente al camino que existe desde su hoja hasta la raíz. En nuestro ejemplo, M1, situado en (4,1), conocerá $K_{(4,1)}$,$K_{(3,1)}$,$K_{(2,1)}$ y $K_{(1,1)}$.

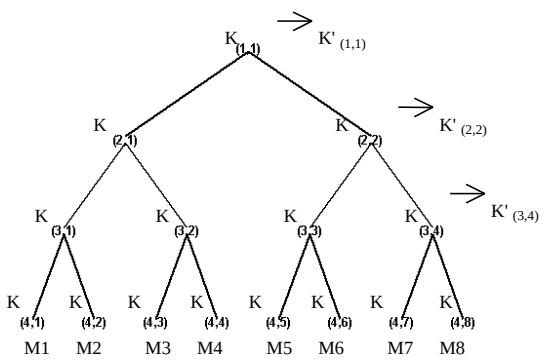
Altas de usuarios

Cuando un nuevo miembro quiere añadirse al grupo (M8) debe primero contactar con el KS vía un canal unicast seguro. En esa fase negociarán una clave compartida sólo entre ellos dos que les servirá para comunicarse de forma secreta ($K_{(4,8)}$). Después de esto, el KS deberá actualizar todas las claves que existan en el camino desde la hoja donde ha situado a M8 hasta la raíz. De no ser así, posibles comunicaciones anteriores que hubieran sido cifradas con dichas claves, y que M8 podría tener grabadas, serían vulnerables en este momento. En la Fig. 2b puede observarse dicho comportamiento, donde las claves actualizadas se han marcado como $K'_{(x,y)}$.

El KS debe repartir las nuevas claves a los usuarios pertinentes. Para ello utiliza la jerarquía de claves existente, junto con protocolos de transporte multicast, que por simplificación, supondremos fiable. En primer lugar envía a cada miembro en los nodos (4,7) y (4,8) todas las claves de su subconjunto correspondiente. Esta transmisión se realiza usando sólo sus claves únicas y mediante una conexión unicast ya que el uso del canal multicast no supondría ningún ahorro de ancho de banda.



a) Árbol lógico de claves con 7 miembros



b) Miembro M8 se añade al grupo

Figura 2

Seguidamente, envía un mensaje multicast con las claves $K'_{(2,2)}$ y $K'_{(1,1)}$ cifradas con $K_{(3,3)}$, de forma que sólo los miembros en los nodos (4,5) y (4,6) puedan descifrarlas.

Finalmente, se envía un mensaje multicast con la nueva clave raíz $K'_{(1,1)}$ cifrada con $K_{(2,1)}$, de forma que los miembros en los nodos (4,1) a (4,4) puedan recuperarla. Llegados a este punto, los 8 miembros en el grupo multicast conocen el subconjunto de claves actualizado que hay desde su posición en el árbol hasta la raíz. Todos los miembros conocen la clave raíz así ésta se puede usar para cifrar un mensaje multicast que contendrá la nueva clave de sesión o SEK'.

Baja de miembros

Ahora, asuma que el miembro M4 abandona el grupo. Todas las claves que poseía deben ser actualizadas. $K_{(4,4)}$ es simplemente eliminada. Vea Fig. 3. Las demás KEKs son actualizadas por el KS y enviadas al resto de miembros del grupo cifradas con las claves situadas en nodos inferiores a los nodos hermanos de las claves actualizadas.

En nuestro ejemplo, en primer lugar el KS envía al miembro M3, usando su clave única y por un canal unicast, el subconjunto de claves actualizadas. Seguidamente, envía un mensaje multicast que contendrá las claves $K'_{(2,1)}$ y $K''_{(1,1)}$ cifradas con $K_{(3,1)}$, de esta forma sólo M1 y M2 podrán recuperarlas. Finalmente, envía un mensaje multicast que contendrá la clave $K''_{(1,1)}$ cifrada con $K'_{(2,2)}$, de forma que los miembros en los nodos (4,5) a (4,8) podrán recuperarla. Llegados a este punto, todas las claves que conocía M4 cuando era miembro autorizado del grupo han sido actualizadas, por lo que se puede asegurar que M4 no tendrá acceso a ninguna comunicación futura de dicho grupo.

De acuerdo al ejemplo, es fácil observar como los métodos basados en árboles binarios de claves pueden realizar las operaciones de actualización usando únicamente $0(\log_2(N))$ mensajes, donde N es el número de miembros en el grupo multicast.

3 Propuesta de mejora de LKH

Como se ha comentado anteriormente, a pesar del avance que supuso LKH respecto a métodos triviales de renegociación de claves, los problemas de latencia y ancho de banda persisten, sobretodo en entornos muy dinámicos y con gran número de usuarios.

Aprovechando el trabajo de Wang Bin en [11], a continuación se propone una modificación del LKH que reduce el número de claves que el KS debe guardar en memoria así como el número de mensajes a enviar por cada *rekeying*, con su consecuente reducción de latencia.

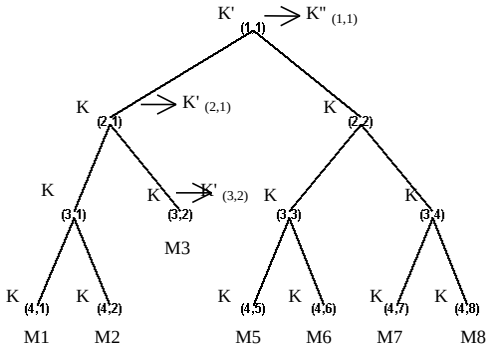


Figura 3 M4 abandona el grupo.

3.1 Mejora en cuanto a memoria necesaria en el KS

Uno de los parámetros de eficiencia que se debe mejorar en LKH es la cantidad de memoria que necesita el KS para almacenar claves durante toda la sesión. Para ello se propone el uso de funciones pseudoaleatorias con una semilla que sólo conozca el KS, evitando así tener que guardar la totalidad del árbol binario de claves.

Cuando se requiere un *rekeying*, el KS no envía las claves actualizadas sino que distribuye a cada miembro la información que necesita para actualizarlas. Para facilitar su comprensión, a continuación se detalla un ejemplo sencillo.

Considere el árbol binario de claves en el que las claves de las posiciones (i,j) se generan siguiendo la expresión (1). Fig. 4.

$$K_{(i,j)} = F_{r1}(2^i + j) \oplus r \qquad (1)$$

Fr1 es una función pseudoaleatoria con semilla *r1* y *r* es otro número aleatorio necesario para actualizar las claves. El símbolo $\oplus$ denota la función XOR.

Cuando la naturaleza del grupo cambia (un miembro se da de alta o de baja) el KS manda a todos los miembros que necesitan actualizar alguna de sus claves, el parámetro en (2).

$$P = r \oplus r' \qquad (2)$$

Con este parámetro, cada miembro sólo tiene que calcular (3) para actualizar sus claves.

$$K'_{(i,j)} = K_{(i,j)} \text{ Å } P = F_{r1}(2^i + j) \text{ Å } r' \qquad (3)$$

Como nadie a parte del KS conoce ni $F_{r1}(2^i + j)$ ni *r*, ningún usuario normal podrá calcular ni las claves que se usaran en el futuro ni las pasadas, ya que sólo conocen (i,j) y *P*. Para repartir *P* a los miembros que quedan en el grupo se utilizan las reglas típicas de LKH.

El sistema propuesto reduce la cantidad de memoria necesaria en el KS pero sigue necesitando $O(\log_2 N)$ mensajes para realizar el *rekeying* y, por tanto, la latencia sigue siendo del mismo orden.

3.2 Mejora de la latencia

A continuación se propone combinar el método anterior con técnicas empleadas en entornos de cifrado broadcast o *broadcast encryption*. Con ello se consigue reducir aun más el ancho de banda necesario (y el número de mensajes) y por consiguiente la latencia. Además, igual que en el método discutido anteriormente, el KS sólo debe almacenar dos semillas de funciones pseudoaleatorias (además de la clave de sesión).

Broadcast encryption

El problema de *rekeying* que se ha presentado hasta el momento es, en realidad, una ligera modificación del problema clásico de cifrado broadcast (denominado en inglés *broadcast encryption*). En él se aborda la problemática de permitir a un nodo centralizado el envío de información por un canal broadcast a un conjunto arbitrario de usuarios autorizados. Esta transmisión debe realizarse en un número mínimo de pasos o mensajes [12].

La principal diferencia entre los problemas broadcast y multicast *encryption* radica en que en el cifrado broadcast el KS no conoce la identidad de los usuarios que quiere "evitar". En el problema de multicast *encryption*, en cambio, el KS sí que conoce esta identidad, ya que, por definición, los usuarios son limitados y conocidos.

Los métodos de broadcast *encryption* se basan principalmente en propiedades matemáticas. Nuestra propuesta aborda la renegociación de claves multicast desde este enfoque. A continuación se detallan las características genéricas que debe tener un sistema de este tipo para adecuarse a nuestros requerimientos.

Requisitos matemáticos

Sea $M = \{M_1, M_2, ..., M_N\}$ un grupo de N miembros que se comunican mediante multicast. Considere un conjunto de funciones $F = \{f_1, f_2, ..., f_N\}$ generadas por el KS. Cada miembro M_i sólo comparte el secreto f_i con el KS. Sea $A \subset M$ el subconjunto de miembros autorizados a quien se quiere repartir el secreto.

Se desea que las funcione tengan las siguientes propiedades:

- Dado un secreto, k, es fácil obtener el parámetro p tal que $f_i(p) = f_j(p) = k \quad \forall i, j \in A$.

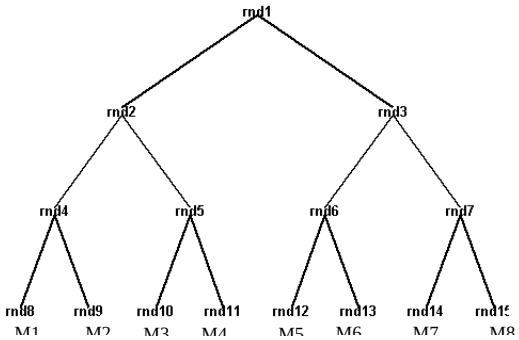


Figura 4 Árbol de números aleatorios.

- El conocimiento de p no revela nada de k a ningún miembro no perteneciente al grupo autorizado .

Si se cumplen estas propiedades, el envío multicast de un único mensaje p bastaría para repartir la nueva clave de sesión k . El algoritmo propuesto se basa en este comportamiento.

Algoritmo propuesto

Considere un grupo multicast dinámico de tamaño N y un KS que comparte un secreto con cada uno de los miembros de este grupo. El KS construye un árbol lógico de claves, al estilo LKH según las siguientes reglas. Cada nodo en el árbol es un número aleatorio generado según la expresión (1). Como en LKH, cada miembro se sitúa en una hoja del árbol y conoce las claves que hay desde su hoja hasta la raíz.

Cuando se debe acordar una nueva clave, el KS calcula un parámetro público según la expresión (4).

$$P = r_2 \prod_{i \subset S} rnd_i + (r \oplus r') \qquad (4)$$

r_2 es el resultado de una función pseudoaleatoria, diferente a F_{r1}, y usada para evitar ataques por confabulación. La necesidad de este parámetro se discutirá más adelante en el análisis de seguridad. $\prod_{i \subset S} rnd_i$ es el producto de todos los números aleatorios del subconjunto S, siendo S todos los los nodos hermanos (*siblings*) al camino de claves actualizado. $(r \oplus r')$ es el parámetro cegado o ofuscado que necesitan los miembros que aún pertenecen al grupo para actualizar el árbol.

Como ejemplo, considere que el miembro M3 de la Fig. 4 abandona el grupo. El parámetro P se calcularía según la expresión (5)

$$P = r_2 \cdot rnd_{11} \cdot rnd_4 \cdot rnd_3 + (r \oplus r') \qquad (5)$$

Para recuperar el factor cegado, cada miembro realizará una reducción modular de *P* módulo alguno de los números aleatorios que conoce, correspondientes al camino desde su hoja hasta la raíz. Como este número aleatorio estará incluido en el producto de números usado para generar *P*, la reducción modular tendrá como resultado $(r \oplus r')$. Observe la expresión (6).

$$P = (r_2 \cdot rnd_{11} \cdot rnd_4 \cdot rnd_3 + (r \oplus r')) \bmod rnd_4 = (r \oplus r') \quad (6)$$

Conocido $(r \oplus r')$, cada usuario puede actualizar sus claves, al igual que en el algoritmo propuesto anteriormente siguiendo la expresión (3).

4. Análisis de Seguridad

A continuación se discute el nivel de seguridad que ofrece el método propuesto y se compara con los de otros métodos existentes, como el LKH.

Tradicionalmente, los algoritmos de renegociación de claves multicast tienen tres posibles atacantes: un miembro revocado, una confabulación de miembros y un miembro autorizado durante sucesivos r*ekeyings*. La única forma que se tiene de romper el sistema es encontrando alguno de los factores de $\prod_{i \subset S} rnd_i$. Si algún atacante llegara a saber cualquiera de esos rnd_i sería capaz de realizar la reducción modular y obtener el secreto $(r \oplus r')$.

4.1 Ataques y atacantes

Miembro revocado

Un único miembro revocado del sistema no conoce ninguno de los factores ya que, por definición, las claves que conoce no se incluyen en el mensaje de *rekeying*. Además, los secretos que él conocía mientras era miembro autorizado del grupo ya no se usarán más en el futuro ya que serán actualizadas en cuanto él abandone el grupo

Confabulación de miembros revocados

Al igual que en los miembros individuales, una confabulación de miembros revocados no puede obtener ninguna información sobre los secretos usados, ya que todos los parámetros que ellos conocían no serán de nuevo usados en el mensaje de *rekeying*, y todos fueron actualizados.

Confabulación de miembros autorizados

La forma más simple de confabulación de miembros autorizados consiste en que un miembro autorizado revele cualquiera de sus secretos conocidos a otro miembro autorizado. Es obvio que dicho agujero de seguridad es insolventable ya que nadie puede impedirnos que un usuario autorizado reparta su clave entre usuarios no autorizados. De cualquier forma, este caso no es significativo, y por supuesto también se da en LKH.

Si no se usara el parámetro r_2 mencionado en la sección 3, dos atacantes autorizados podrían obtener algún factor que no les corresponda mediante el análisis de mensajes de *rekeying* y sus propios secretos.

De nuevo, considere que M3 abandona el grupo. Suponga que r_2 no se incluye en la expresión (5). Dos miembros autorizados confabuladores, por ejemplo M1 y M4, podrían colaborar para conocer el secreto rnd_3 mediante el análisis del mensaje de *rekeying*, según los siguientes pasos:

i. M1 obtiene el secreto $(r \oplus r')$ mediante la reducción modular con su secreto (rnd_4)

ii. M1 obtiene $\prod_{i \subset S} rnd_i$ restando $(r \oplus r')$ de $P = rnd_{11} \cdot rnd_4 \cdot rnd_3 + (r \oplus r')$.

iii. M1 divide rnd_4 de $\prod_{i \subset S} rnd_i$ y le da el resultado a M4.

iv. Finalmente M4 divide rnd_{11} de los datos que le ha dado M1 y obtiene rnd_3. Que es un secreto que no pertenecía ni a M4 ni a M1

Tanto M1 como M4 conocen el secreto $(r \oplus r')$. Por tanto, podrán actualizar rnd_3 y así descifrar la nueva clave de sesión cuando abandonen el grupo.

Este agujero de seguridad se supera con la introducción del número aleatorio r_2 a la expresión (5). De esta forma, la única forma que tiene M4 de obtener rnd_3 es factorizando $rnd_3 \cdot r_2$. Este problema es considerado de difícil resolución si el resultado del producto está compuesto por factores primos suficientemente grandes. Más adelante en esta sección discutiremos cómo se pueden asegurar que rnd_i estén compuestos por primos grandes.

Miembros autorizados a durante sucesivos rekeyings

Otra amenaza importante para nuestro sistema es lo que un usuario autorizado puede averiguar analizando la información de la que dispone durante sucesivos *rekeyings*. Si los números aleatorios fueran reutilizados para distintos mensajes de *rekeying* un miembro del grupo podría fácilmente encontrar un factor con sólo aplicar al algoritmo de máximo común divisor. Se puede ver un ejemplo en la expresión (7).

Mensaje 1 (M3 de la Fig 3, revocado):

$$P = rnd_{11} \cdot rnd_4 \cdot rnd_3 + (r \oplus r')$$

Mensaje 2 (M2 de la Fig 3, revocado):

$$P = rnd_8 \cdot rnd_5 \cdot rnd_3 + (r' \oplus r'') \quad (7)$$

Con estos dos mensajes, cada miembro autorizado puede calcular fácilmente $rnd_{11} \cdot rnd_4 \cdot rnd_3$ y $rnd_8 \cdot rnd_5 \cdot rnd_3$ con sólo encontrar el secreto y restándoselo de P. Con estos dos productos, es fácil encontrar rnd_3 como el mcd$(rnd_{11} \cdot rnd_4 \cdot rnd_3, rnd_8 \cdot rnd_5 \cdot rnd_3)$.

Para evitar este ataque, todos los números aleatorios en el árbol de claves deberían ser actualizados cada vez que se requiere un *rekeying*. Si sólo se usasen mecanismos LKH para actualizar estos aleatorios, se produciría un incremento considerable del número de mensajes y por tanto del ancho de banda necesario. Para evitar este efecto se utiliza el mecanismo de mínima memoria en KS descrito anteriormente.

En cada *rekeying* se actualiza todo el árbol y se recalculan todos los números aleatorios. De esta forma distintos mensajes de *rekeying* correspondientes a distintas renegociaciones de clave tendrán distintos factores como componentes y el algoritmo de máximo común divisor no revelará nada sobre esos factores. Además, el mecanismo de actualización del método descrito delega las funciones de actualización del árbol a los usuarios individuales, de esta forma no se requieren mensajes de actualización adicionales.

4.2. Restricciones

Por descontado, para un funcionamiento correcto del algoritmo propuesto se requiere tener en cuenta algunas restricciones que existen sobre los parámetros a usar.

En primer lugar, *(rÅr') mod* rnd_i *= (rÅr')* implica que *(rÅr')*< rnd_i para todo número aleatorio rnd_i. Esta condición es de fácil cumplimiento ya por diseño se pueden fijar, por ejemplo, las longitudes en número de bits de cada uno de estos parámetros.

Por otra parte, como se comentó anteriormente, los números rnd_i deberían estar formados por como mínimo un primo grande. Esta condición hace que la factorización sea un problema considerado difícil en criptografía, y sobre el cual se basan numerosos algoritmos robustos como el RSA.

Es importante remarcar que el método propuesto no requiere que los números rnd_i sean primos ellos mismos. El hecho de no ser un único número primo puede incluso hacer más seguro el sistema ya que la factorización de $\prod_{i \subset S} rnd_i$ no revelaría nada sobre rnd_i sino que daría como resultado sus factores, y encontrar un rnd_i válido sólo se podría hacer mediante combinaciones de esos factores. En cualquier caso en [13] se puede encontrar la información necesaria de cómo se puede asegurar que un número aleatorio con una longitud mínima de bits contenga como mínimo un primo de otra determinada longitud.

5. Evaluación

Finalmente, en la Tabla 1 mostramos los valores de los parámetros más significativos para renegociación de claves en multicast. Queda de manifiesto como el método propuesto mejora en gran medida la eficiencia en casi todos los parámetros. Debe resaltarse que tanto latencia como el número de mensajes necesarios para la actualización de claves se han reducido al mínimo.

La Fig. 5 muestra el comportamiento de la latencia en número absoluto de paquetes. Es preciso comentar que aunque sólo se utilice un mensaje para el *rekeying*, éste es de mayor longitud ya que resulta del producto de $\log_2 N$ factores. Este el motivo por el cual en la Fig. 6 se compara el uso real de ancho de banda normalizado por el tamaño de paquete. De cualquier forma, a pesar de ello, el uso de un solo paquete reduce el *overhead* necesario para el *rekeying*, y por tanto, la cantidad total de bits a transmitir.

Tabla 1. Parámetros de eficiencia de distintos algoritmos de renegociación de claves en multicast

Parámetros de eficiencia	Método de Rekeying			
	Trivial	LKH	Mejora 1	Mejora 2
Claves a repartir	N	O(2N)	O(2N)	O(2N)
Mensajes de actualización	N	$O(\log_2 N)$	$O(\log_2 N)$	1
Claves guardadas en KS	N	O(2N)	3	3
Claves guardadas en User	1	$O(\log_2 N)$	$O(\log_2 N)$	$O(\log_2 N)$
Latencia	N	$O(\log_2 N)$	$O(\log_2 N)$	1

6. Conclusiones y líneas futuras

En este trabajo hemos presentado un nuevo método de renegociación de claves en entornos multicast. El método se basa en el uso de un único mensaje de renegociación para actualizar la clave de sesión. Este hecho reduce la latencia que presentan otros métodos de renegociación como el LKH.

Para reducir la carga de renegociación a un solo mensaje se han utilizado las propiedades matemáticas de la reducción modular que permiten seleccionar a un subconjunto arbitrario de miembros autorizados a los cuales se les quiere hacer inteligible un mensaje. Para evitar los posibles ataques realizables al sistema, el árbol entero de claves se actualiza de forma sencilla cada vez que se realiza un *rekeying*. Además, gracias al uso de funciones pseudoaleatorias para generar las claves, el KS sólo debe almacenar 3 parámetros.

Como línea futura inmediata se apunta la aplicación y adecuación de este algoritmo a mecanismos de renegociación en bloque o *batch rekeying algorithms* en los que los miembros no se dan de alta y de baja uno a uno sino en lotes de grupos de muchos usuarios.

Agradecimientos

Este trabajo ha sido soportado por los proyectos DISQET y CREDO [CICYT TIC2002-00818 TIC2002-00249], dentro del Plan Nacional de I+D.

Referencias

[1] I-D irtf-smug-taxonomy-01 A taxonomy of multicast security issues, R. Canetti, B. Pinkas. Aug 2000

[2] T. Hardjono, G. Tsudik. IP Multicast Security: Issues and Directions. Technical Report unkown, University of Southern California, September 1999.

[3] I-D. Harney-sparta-lkhp-sec-00. Logical Key Hierarchy Protocol (LKH). H. Harney, E. Harder, Mar 99

[4] The Group Security (GSEC) Research Group. http://www.securemulticast.org/gsec-index.htm

[5] MSEC Working Group. http://www.securemulticast.org/msec-index.htm

[6] RFC2094 Group Key Management Protocol Architecture. H. Harney, C. Muckenhirn, July 1997

[7] A Secure Scalable Multicast Key Management Protocol (MKMP), D. Harkins, N. Doroswamy.

[8] Mittra. Iolus: A Framework for Scalable Secure Multicasting. In Proc. ACM SIGCOMM, pages 277-288, Cannes, France, September 1997

[9] I-D. Irtf-smug-groupkeymgmt-oft-00 Key Management for Large Dynamic Groups: One-Way Function Trees and Amortized Initialization. D. Balenson, D. McGrew, A. Sherman. Aug 2000

[10] Canetti, Malkin, Nissim. Efficient Communication Storage Tradeoffs for Multicast Encryption. Eurocrypt'99 p 456-470 1999

[11] Bin, Jian-Hua. Optimal Key Storage for Secure Multicast. Department of Electronic Engineering, Shangai Jiaotong University.

[12] A. Fiat and M. Naor. Broadcast Encryption. Advances in Cryptology. CRYPTO 93., 1993.

[13] Menezes, Oorschot , Vanstone. Handbook of Applied Cryptography. CRC Press. 1996. ISBN: 0-8493-8523-7

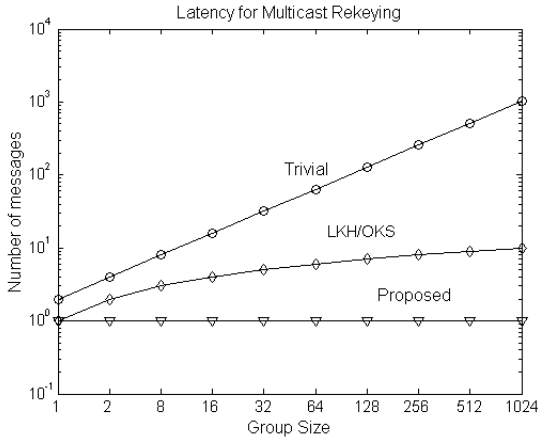


Figura 5. Latencia en número de mensajes

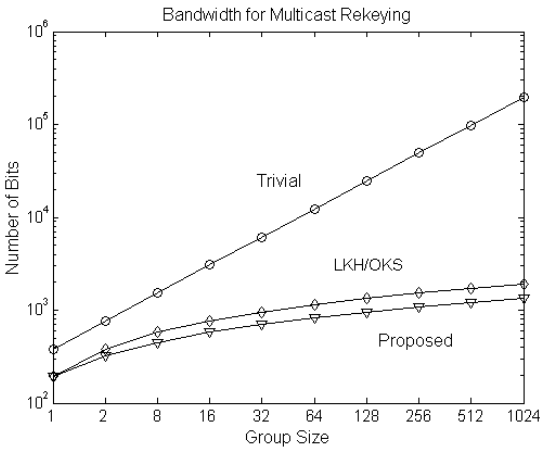


Figura 6. Ancho de banda en número de bits.

Protección de la propiedad intelectual basada en juegos de adivinanza

Marcel Fernandez y Miguel Soriano
Departamento de Ingeniería Telemática. Universidad Politécnica de Cataluña
C/ Jordi Girona 1 i 3. Campus Nord, Mod C3, UPC.
08034 Barcelona. España.
Teléfono: 934 016 023 Fax: 934 015 981
E-mail: {marcelf,soriano}@entel.upc.es

Abstract *In the guessing secrets problem defined by Chung, Graham and Leighton in [4], a player* **B** *has to unveil a set of $k > 1$ secrets that a player* **A** *has chosen from a pool of N secrets. To discover the secrets, player* **B** *is allowed to ask a series of boolean questions. For each question asked* **A** *can adversarially choose one of the secrets but once he has made his choice he must answer truthfully. In this paper we first present a solution to the guessing secrets problem consisting in an error correcting code equipped with a tracing algorithm that efficiently recovers the secrets. Next, we show how with a slight modification in the tracing algorithm our approach to the guessing secrets problem also provides a solution to the collusion secure fingerprinting problem.*

1. Introducción

En la versión original del juego de identificación de secretos, tal como se presentó en el programa de TV "I've got a secret", cuatro personas intentaban adivinar el secreto guardado por una quinta persona. Esta quinta persona recibía una recompensa económica si entre los cuatro encuestadores no podían adivinar el secreto. En este artículo se presenta una variante del juego definida por Chung, Graham y Leighton [4], basada en la presencia de dos jugadores A y B. El jugador A extrae un subconjunto de $k \geq 2$ secretos de un conjunto de N secretos. El jugador B realiza una serie de preguntas con el objetivo de descubrir los secretos. Para cada pregunta, A puede elegir un secreto cualquiera entre los k, pero no puede falsear el resultado. Utilizando la misma aproximación que Alon, Guruswami, Kaufman y Sudan analizaron en [1], los autores de este artículo presentan una solución al problema de adivinar secretos elegidos basado en la concatenación de un código dual de Hamming y un código Reed Solomon. Además, se ha diseñado un algoritmo de identificación que identifica los secretos usando técnicas de descodificación soft-decision. El problema de adivinar secretos está relacionado con distintos aspectos telemáticos como son la entrega eficiente de contenidos en Internet [4] y el diseño de esquemas para la protección del copyright en documentos digitales [1]. Los autores aprovechan la relación con los esquemas de protección de copyright para mostrar como usando pequeñas modificaciones la solución propuesta para el problema de adivinar secretos puede ser usada en esquemas de fingerprinting. La técnica de fingerprinting consiste en empotrar marcas en un objeto digital con el fin de obtener protección ante redistribuidores ilegales. El artículo se organiza como se indica a continuación. En la Sección 2 se describe formalmente el juego de adivinar secretos para el caso de $k = 2$. La Sección 3 muestra una visión conceptual de los códigos IPP y de los (2,2) separables. En la Sección 4 se ofrece una primera aproximación para resolver el problema de adivinar secretos usando códigos binarios dual de Hamming. La solución puede ser mejorada usando concatenación de códigos, tal como se muestra en la Sección 5. Finalmente, en la Sección 6 se detalla como esta última propuesta puede ser transformada para su adopción en esquemas de fingerprinting seguros contra confabulaciones.

2. Adivinando dos secretos con respuestas binarias

En esta sección se presenta una descripción formal del juego para el caso de $k = 2$ secretos. Sea $S = \{s_1, s_2\}$ el subconjunto de dos secretos elegidos por A de entre el conjunto de N secretos. En primer lugar debemos indicar que no se puede garantizar que el jugador B pueda identificar los dos secretos ya que si todas las respuestas están relacionadas con uno de los dos secretos, B no podrá tener ninguna información del otro. Debemos ser conscientes también que B nunca podrá tener la certeza que un determinado secreto ha sido elegido por A. Esto es así porque A puede disponer de tres secretos $\{s_1, s_2, s_3\}$, y responder por mayoría. En este caso B consigue una solución factible para los tres conjuntos de secretos $\{s_1, s_2\}, \{s_1, s_3\}$ y $\{s_2, s_3\}$. Usando el razonamiento anterior, se

puede ver que para una respuesta dada, se tienen las siguientes posibles configuraciones: a) una configuración en estrella en la que todos los pares de secretos tienen un elemento común; b) una configuración en estrella degenerada, en la que hay un único par de secretos y c) una configuración en triángulo, en la que hay tres posibles pares de secretos disjuntos. La solución para el problema de $k = 2$ secretos consiste en encontrar la configuración adecuada para una secuencia de respuestas dada. Cuando se considera el conjunto de cuestiones a formular, hay dos posibles estrategias a adoptar: adaptativa y preliminar. En el caso de estrategia adaptativa, cada pregunta depende de las respuestas previas. En el caso de estrategia preliminar, las preguntas son establecidas al principio del juego. Aunque las estrategias adaptativas parecen más razonables, se puede conseguir una solución sorprendentemente satisfactoria usando estrategia preliminar [1], y por ello serán utilizadas en nuestro estudio.

3. Códigos IPP y (2,2) separables

En esta sección se pretende dar una visión conceptual de códigos IPP y códigos separables (2,2), a partir de un enfoque basado en códigos correctores de errores.

Un código C es un subconjunto de un espacio vectorial $\mathbf{F}_q^n$. El conjunto de escalares $\mathbf{F}_q$recibe el nombre de alfabeto del código. Un código C es lineal si constituye un subespacio de $\mathbf{F}_q^n$. La dimensión k del código es la del subespacio constituido por C, y n es la longitud del código. La nomenclatura habitual para C es $[n, k, d]$, siendo d la distancia mínima del código (distancia de Hamming mínima entre dos palabras código cualesquiera).

Dadas dos palabras cualesquiera $\mathbf{a}$ y $\mathbf{b}$ de $\mathbf{F}_q^n$, se define el conjunto de descendentes $D(\mathbf{a}, \mathbf{b})$ como Puede observarse que $\mathbf{a}$ y $\mathbf{b}$ son elementos de dicho conjunto $D(\mathbf{a}, \mathbf{b})$. Dado un código C, el código descendente C^* se define como : $C^* := \bigcup_{\mathbf{a}\in C, \mathbf{b}\in C} D(\mathbf{a}, \mathbf{b})$.

Si $\mathbf{c} \in C^*$ es un descendente de $\mathbf{a}$ y $\mathbf{b}$, o sea, $\mathbf{c} \in D(\mathbf{a}, \mathbf{b})$, decimos que $\mathbf{a}$ y $\mathbf{b}$ son padres de $\mathbf{c}$. Un código IPP (identifiable parent property) es aquél en que para todo elemento $\mathbf{c} \in C^*$, al menos uno de sus padres puede ser identificado con absoluta certeza Habitualmente una palabra $\mathbf{c} \in C^*$ tiene varias parejas posibles de padres; pero si el código es IPP la intersección de todas esas posibles parejas no puede ser el conjunto vacío. Los algoritmos de decodificación IPP se basan precisamente en eso, dado un elemento $\mathbf{c} \in C^*$ buscar todas las posibles parejas que puedan constituir sus padres y posteriormente se efectúa la intersección. El siguiente teorema proporciona una forma explícita de construir un código IPP.

Teorema 1 *([7]) . Sea q la potencia de un primo p. Si $q \geq n - 1$, existe un código Reed Solomon (acortado, o extendido) sobre $\mathbf{F}_q$con parámetros $[n, n/4, n - n/4 + 1]$ y es IPP.*

El siguiente teorema da una condición necesaria para que una palabra código sea padre de un descendente.

Teorema 2 *Sea C un código Reed-Solomon 2-IPP. Si una palabra código coincide en más de $2(n-d)$ posiciones con un descendente dado, puede garantizarse que esa palabra código es un padre del descendente.*

Una forma no tan robusta para identificar a los padres es haciendo uso de los códigos (2,2) separables [8]. Un código C es (2,2) separable si para dos subconjuntos disjuntos de palabras código de tamaño 2, $\{\mathbf{a}, \mathbf{b}\}$ y $\{\mathbf{c}, \mathbf{d}\}$ donde la intersección entre $\{\mathbf{a}, \mathbf{b}\}$ y $\{\mathbf{c}, \mathbf{d}\}$ es nula, sus respectivos conjuntos de descendentes son también disjuntos, $D(\mathbf{a}, \mathbf{b}) \cap D(\mathbf{c}, \mathbf{d}) = \emptyset$.

4. Uso de códigos dual de Hamming para adivinar secretos

En esta sección mostraremos como el uso de códigos (2,2) separables puede establecer un conjunto de preguntas que resuelve el problema de acertar secretos con $k = 2$. Asimismo se presentará una estrategia basada en el algoritmo de decodificación de Chase que desvela de forma eficientemente los secretos elegidos.

4.1. Construcción explícita de la estrategia

Siguiendo la notación de [1], denotamos las preguntas en un esquema preliminar como la secuencia G de n funciones booleanas $g_i : \{1, \ldots, N\} \to \{0, 1\}$. Para un secreto concreto $\mathbf{x}$, la secuencia de respuestas a las preguntas g_i serán $C(\mathbf{x}) = \langle g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_n(\mathbf{x})\rangle$.

Podemos asumir sin pérdida de generalidad que $\log_2 N$ es un entero. En este caso, usando la representación binaria para $\{1, \ldots, N\}$ podemos redefinir C como el mapeo $C : \{0, 1\}^{\log_2 N} \to \{0, 1\}^n$. Desde ese punto de vista, C puede considerase como un código corrector de errores. De ahora en adelante, nos referiremos a una estrategia a partir de su código asociado C y a la secuencia de respuestas a un secreto dado como su palabra código asociada. La cuestión siguiente es: ¿Qué propiedades debe tener el código corrector de errores para poder solventar el problema de identificar secretos?

A partir de la definición de códigos (2,2) separables presentado en la sección 3, inferimos que si un código es (2,2) separable, para cada cuatro cualesquiera y distintas palabras código (secretos) s_1, s_2, s_3 y s_4, existe al menos un valor

$i \in \{1,\ldots,n\}$ denominado índice discriminante para el que $C(s_1)i = C(s_2)_i \neq C(s_3)_i = C(s_4)_i$. Si B plantea cuestiones de acuerdo a un esquema (2,2) separable, usando la respuesta a la i-ésima pregunta es capaz de excluir un par de secretos de otro par cualquiera de secretos disjuntos. Más formalmente:

Lema 1 *([1]) Existe un código (2,2) separable* $C : \{0,1\}^{log_2 N} \rightarrow \{0,1\}^n$ *si y sólo si existe una estrategia preliminar para B que resuelve el problema de identificar 2 secretos de un conjunto de N utilizando n preguntas.*

Del lema anterior se deduce que el problema de acertar secretos se reduce a construir códigos (2,2) separables. El siguiente corolario (presentado en [5]) da las condiciones suficientes para que un código lineal sea (2,2) separable.

Corolario 1 *([5]) Todos los códigos lineales equidistantes son (2,2) separables*

En consecuencia, el código dual binario de Hamming S_r $[2^r - 1, r, 2^{r-1}]$ que consta del vector 0 y $2^r - 1$ palabras código de peso de Hamming 2^{r-1}, caracterizado porque todas las palabras código están a la misma distancia, es apto para resolver el problema de adivinar secretos.

4.2. Identificación eficiente de secretos

En este momento nos enfrentamos a como identificar secretos de forma eficiente usando códigos dual de Hamming. Para identificarlos, necesitamos un procedimiento que vincule la palabra asociada a una secuencia de respuestas con las palabras códigos asociadas a esos secretos. El siguiente lema muestra como puede establecerse esta relación.

Lema 2 *Supongamos que se usa un código dual de Hamming como estrategia para resolver el problema de* $k = 2$ *secretos. Sean* s_1 *y* s_2 *un par de secretos y sean* $\mathbf{x}_1$ *y* $\mathbf{x}_2$ *sus palabras código asociadas. El conjunto de posibles secuencias de respuestas de A, de acuerdo a los secretos* s_1 *y* s_2 *es precisamente* $D(\mathbf{x}_1,\mathbf{x}_2)$*, es decir, el conjunto de descendentes de* $\mathbf{x}_1$ *y* $\mathbf{x}_2$*.*

Si denotamos por $\mathbf{v}$ la palabra correspondiente a la secuencia de respuestas, tal como se puede ver en la sección 2, podemos obtener la siguiente casuística:

1.- Una configuración de estrella (llamemos $\mathbf{x}$ al secreto común). En tal caso tenemos que $\mathrm{d}(\mathbf{x},\mathbf{v}) \leq 2^{r-2} - 1$.

2.- Una configuración en estrella degenerada, siendo $\mathbf{x}$ e $\mathbf{y}$ el único para de secretos. Entonces $\mathrm{d}(\mathbf{x},\mathbf{v}) = \mathrm{d}(\mathbf{y},\mathbf{v}) = 2^{r-2}$.

3.- Una configuración en triángulo para los tres posibles pares de secretos $\{\mathbf{x},\mathbf{y}\}$, $\{\mathbf{x},\mathbf{z}\}$ y $\{\mathbf{y},\mathbf{z}\}$. Se tiene que $\mathrm{d}(\mathbf{x},\mathbf{v}) = \mathrm{d}(\mathbf{y},\mathbf{v}) = \mathrm{d}(\mathbf{z},\mathbf{v}) = 2^{r-2}$.

En consecuencia, se precisa un algoritmo cuya salida sean todas las palabras códigos de un código dual de Hamming con distancia 2^{r-2} de $\mathbf{v}$. El algoritmo de decodificación que presentamos se basa en el de Chase [3]. Tal como se asume en [3], suponemos que se dispone de un decodificador binario que corrige hasta $\lfloor (d-1)/2 \rfloor = 2^{r-2} - 1$ errores. Si la palabra código más cercana a $\mathbf{v}$ está a una distancia $\geq 2^{r-2}$, el decodificador binario falla en la decodificación. Pero en tal caso, la palabra $\mathbf{v}'$, obtenida aplicando un test de patrones $\mathbf{p}$ de peso 1 a $\mathbf{v}$ ($\mathbf{v}' = \mathbf{v} \oplus \mathbf{p}$) está a distancia $2^{r-2} - 1$ de una palabra código. Por lo tanto usando el test de patrones adecuado, podemos ser capaces de corregir 2^{r-2} errores. La idea del algoritmo es encontrar de forma eficiente el test de patrones adecuados, usando los secretos ya encontrados. Debe tenerse en cuenta que una vez se ha encontrado una palabra código asociada a un secreto se hace uso de este resultado jugando con las posiciones en las que hay coincidencias entre la palabra código asociada al secreto y la palabra asociada a las respuestas.

Algoritmo de Chase simplificado: El algoritmo usa: Una funcion llamada *binary_decoder*, ($b\_d\mathbf{v}$) cuya salida es la única palabra código a distancia $2^{r-2} - 1$ de $\mathbf{v}$, en caso que exista.

Una función llamada $right\_shift$, $r\_s(\mathbf{p})$ que toma como entrada un patrón $\mathbf{p}$ de peso 1, y como salida un patrón desplazado una posición a la derecha respecto a $\mathbf{p}$.

Una lista *list* que mantiene los modelos ya usados y que no han permitido decodificar $\mathbf{v}$.

Tomamos $\mathbf{u} = (u_1,\ldots,u_r)$ y $\mathbf{v} = (v_1,\ldots,v_r)$.

Entonces $\mathbf{u} \oplus \mathbf{v}$ denota la función XOR aplicada a todos las posiciones $\mathbf{u} \oplus \mathbf{v} = (u_1 \oplus v_1,\ldots,u_r \oplus v_r)$.

Entrada: S_r, código dual de Hamming, de dimensión r; palabra $\mathbf{v}$ asociada a un secreto

Salida: Todas las palabras código a distancia 2^{r-2} de $\mathbf{v}$

1.- Asignar $\mathbf{u}_1 := b\_d(\mathbf{v})$. Si $\mathbf{u}_1 \neq 0$, la salida es $\mathbf{u}_1$ y finalizar.

2.- Inicializar $\mathbf{p} := (1,0,0,\ldots,0)$, $list := \{\emptyset\}$.

3.- Asignar $\mathbf{v}' := \mathbf{v} \oplus \mathbf{p}$ y ejecutar $b\_d(\mathbf{v}').\mathbf{u}_1 := b\_d(\mathbf{v}')$.

4.- Si $\mathbf{u}_1 \neq \emptyset$, ir al paso 5. Sino añadir $\mathbf{p}$ a *list*. $\mathbf{p} := r\_s(\mathbf{p})$. Ir al paso 3

5.- Construir un nuevo test de patrones $\mathbf{p}$ de peso 1, tal que

- sea distinto a todos los patrones de *list*
- su soporte sea una de las posiciones coincidentes entre $\mathbf{v}$ y $\mathbf{u}_1$.

6.- Asignar $\mathbf{v}' := \mathbf{v} \oplus \mathbf{p}$ y ejecutar $b\_d(\mathbf{v}')$. $\mathbf{u}_2 := b\_d(\mathbf{v})$.

7.- Si $\mathbf{u}_2 \neq \emptyset$, ir al paso 8 Sino añadir $\mathbf{p}$ a list. Ir al paso 5

8.- Construir un nuevo test de patrones $\mathbf{p}$ de peso 1, tal que:

- sea distinto a todos los patrones de *list*
- su soporte sea una de las posiciones coincidentes entre $\mathbf{v}$, $\mathbf{u}_1$ y $\mathbf{u}_2$.

Si no hay más patrones disponibles, la salida es $\mathbf{u}_1$, $\mathbf{u}_2$ y finalizar.

9.- Asignar $\mathbf{v}' := \mathbf{v} \oplus \mathbf{p}$ y ejecutar $b\_d(\mathbf{v}')$. $\mathbf{u}_3 := b\_d(\mathbf{v})$.

10.- Si $\mathbf{u}_3 \neq \emptyset$, ir al paso 11 Sino añadir $\mathbf{p}$ a *list*. Ir al paso 8

11.- La salida es $\mathbf{u}_1$, $\mathbf{u}_2$ y $\mathbf{u}_3$ y finalizar.

5. Identificando secretos usando códigos concatenados

En codificación de canal, el uso de códigos concatenados permite obtener mejores tasas. En esta sección, se mejora la tasa del código (2,2) separable presentado en la sección anterior usando concatenación de códigos. En consecuencia disminuye el número de preguntas en el problema que tratamos de resolver. Un código concatenado es la combinación de un código interno $[n_i, k_i, d_i]$ q_i-ario ($q_i \geq 2$) con un código externo $[n_o, k_o, d_o]$ sobre $\mathbf{F}_{q_i^{k_i}}$. La combinación se basa en el mapeo de palabras código del código interno a elementos de $\mathbf{F}_{q_i^{k_i}}$, dando origen a códigos q_i-arios de longitud $n_i n_o$ y dimensión $k_i k_o$. El tamaño del código concatenado es el mismo que el del código externo, y por eso a partir de este punto identificaremos a cada secreto con su palabra código asociada del código externo. Para construr un código separable binario (2,2), usando concatenación tomando:

- como código interno un $[2^r - 1, r, 2^{r-1}]$ dual de Hamming S_r
- como codigo externo se elige un código $[n, \lceil n/4 \rceil, n - \lceil n/4 \rceil + 1]$ IPP Reed-Solomon sobre $\mathbf{F}_{2^r}$ una función de mapeo $\phi : \mathbf{F}_{2^r} \to S_r$.

Las palabras códigos de C se obtienen como se indica a continuación. Tomamos una palabra código $\mathbf{x} = (x_1, \ldots, x_n)$ del código Reed Solomon y calculamos $\mathbf{y}_i = \phi(x_i)$, $1 \leq i \leq n$. La concatenación de $\mathbf{y}_i$'s constituye la palabra código $\mathbf{y} \in \mathcal{C}$, siendo $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n)$ tal que $\mathbf{y}_i = \phi(x_i)$.

Antes de presentar el algoritmo que identifica los secretos, mostraremos el algoritmo de decodificación por listas soft-decision de Guruswami y Sudan.

5.1. Algoritmo de decodificación por listas soft-decision de Guruswami y Sudan

El concepto de decodificación por listas [6] es una extensión de la decodificación clásica. En lugar de intentar entregar una única palabra código, un decodificador por listas obtiene una pequeña lista de todas las palabras código que están a una distancia superior a la capacidad correctora del código. La decodificación soft-decision es aplicable cuando el decodificador aprovecha toda la "información residual" generada por el receptor y en lugar de usar los símbolos recibidos, el decodificador usa información de fiabilidad (probabilidades) de esos símbolos recibidos. El siguiente teorema es un resultado de gran alcance propuesto por Guruswami y Sudan [6] que establece la existencia de un algoritmo de decodificación por listas soft-decoding. Dicho algoritmo permite obtener en tiempo polinómico una lista pequeña de candidatos a ser la palabra código enviada a partir de un conjunto de pesos asociados a la fiabilidad de los símbolos de la palabra recibida.

Teorema 3 *([6]). Consideremos un código Reed Solomon* $[n, k, n-k+1]$ *cuyos mensajes son polinomios* f *sobre* $\mathbf{F}_q$ *de grado máximo* $k-1$. *Sea* f *la función codificadora* $f \rightarrowtail \langle f(x_1), f(x_2), \ldots, f(x_n) \rangle$, *donde* $x_1, \ldots, x_n$ *son elementos distintos de* $\mathbf{F}_q$. *Sea* $\epsilon > 0$ *una constante arbitraria. Para* $1 \leq i \leq n$ *y* $\alpha \in \mathbf{F}_q$, *sea* $r_{i,\alpha}$ *un número racional no negativo. Entonces, existe un algoritmo determinista en tiempo de ejecución polinómico en* n, q *and* $1/\epsilon$ *que teniendo como entrada los pesos* $r_{i,\alpha}$ *para* $1 \leq i \leq n$ *y* $\alpha \in \mathbf{F}_q$, *encuentra una lista de todos los polinomios* $p(x) \in \mathbf{F}_q[x]$ *de grado* $\leq k-1$ *que satisface*

$$\sum_{i=1}^{n} r_{i,f(x_i)} \geq \sqrt{(k-1)\sum_{i=1}^{n}\sum_{\alpha \in \mathbf{F}_q} r_{i,\alpha}^2} \qquad (1)$$

5.2. Identificación eficiente de secretos

Ya estamos en condiciones de describir completamente el algoritmo. Como entrada toma una palabra $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n)$ correspondiente a las respuestas del jugador B y como salida genera las palabras código correspondientes a los secretos. En primer lugar ejecutaremos la decodificación interior consistente en la decodificación de cada subpalabra $\mathbf{y}_i$ usando el algoritmo de Chase simplificado. La salida, como se puede ver en la Sección 4, será una única palabra código $\{\mathbf{h}_1\}$, un par de palabras código $\{\mathbf{h}_1, \mathbf{h}_2\}$, o tres palabras código $\{\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3\}$.

Entonces para $i = 1, \ldots, n$, usamos el mapeo $\phi(s_m) = \mathbf{h}_m$ para obtener el conjunto de $S_i^{(j)} = \{s_{i_1}, \ldots, s_{i_j}\}$, donde el superíndice $j \in \{1, 2, 3\}$ indica la cardinalidad del conjunto. Nótese que los elementos de $S_i^{(j)}$ son símbolos de $\mathbf{F}_{2^r}$. Denotamos por $\mathcal{S}^{(1)}$ el conjunto de $S^{(1)}$'s, por $\mathcal{S}^{(2)}$ el conjunto de $S^{(2)}$'s y por $\mathcal{S}^{(3)}$ el conjunto de $S^{(1)}$'s. Finalmente, definimos la n-tupla de conjuntos $\mathcal{S} = (S_1^{(j)}, \ldots, S_n^{(j)})$ usada para establecer los pesos que serán la entrada del algoritmo de decodificación GS soft decisión. De la lista de salida del algoritmo, podemos identificar fácilmente los secretos.

1. Para $i := 1$ a n:
   a) Decodificar la palabra interna $\mathbf{y}_i$ usando el Algoritmo Simplificado de Chase

para obtener una lista de como mucho 3 palabras código $\{\mathbf{h}_1, \ldots, \mathbf{h}_j\}$, $j \in \{1, 2, 3\}$.

b) Definir $S_i^{(j)} = \{s_{i_1}, \ldots, s_{i_j}\}$, donde $\phi(s_m) = \mathbf{h}_m$, $1 \leq m \leq j$ y $j \in \{1, 2, 3\}$ dependiendo de la salida del paso 1*a*.

2. Inicializar $\mathcal{S} = (S_1^{(j)}, \ldots, S_n^{(j)})$. Definir los subconjuntos $\mathcal{S}^{(1)} = \{S_p^{(j)} \in \mathcal{S} : j = 1\}$, $\mathcal{S}^{(2)} = \{S_p^{(j)} \in \mathcal{S} : j = 2\}$ y $\mathcal{S}^{(3)} = \{S_p^{(j)} \in \mathcal{S} : j = 3\}$.

3. Para $p := 1$ to n $(1 \leq m \leq j)$, asignar los pesos r_{p,α_l} como:

$$r_{p,\alpha_l} := \begin{cases} \frac{1}{j} & \text{if } s_{p_m} = \alpha_l, \\ & s_{p_m} \in S_p^{(j)} \\ 0 & \text{en otro caso} \end{cases}$$

4. Ejecutar el algoritmo GS usando r_{p,α_l}, obteniendo una lista de palabras código U.

5. Si $(|\mathcal{S}^{(1)}| + |\mathcal{S}^{(2)}|) > 2(n-d)$ calcular $U_{1,2} = \{\mathbf{u} \in U : |\{p : u_p \in S_p^{(1)} \vee u_p \in S_p^{(2)}\}| > 2(n-d)\}$. Notar que $|U_{1,2}| \leq 2$.

   - Si $|U_{1,2}| = 2$ entonces devolver $U_{1,2}$ y finalizar.
   - Si $U_{1,2} = \{\mathbf{u}^1\}$ entonces definir $S_p^{(1)'} = \{s_{p_i} : s_{p_i} \in S_p^{(2)} \wedge s_{p_i} \neq u_p^1\}$ y construir el conjunto $\mathcal{P} = \{S_p^{(1)} : u_p^1 \notin S_p^{(1)}\} \cup \{S_p^{(1)'}\}$.
     - Si $|\mathcal{P}| = 0$, entonces devolver $\mathbf{u}^1$ y finalizar.
     - Si $|\mathcal{P}| \geq k$, entonces tomar cualesquiera k de los símbolos en $\mathcal{P}$ y recodificar para encontrar $\mathbf{u}^2$. Devolver $\{\mathbf{u}^1, \mathbf{u}^2\}$ y finalizar.
     - Si $|\mathcal{P}| < k$, tomamos los símbolo de $\mathcal{P}$, y se usa cualquier posible combinación del alfabeto de símbolos para las restantes posiciones hasta k y recodificar, obteniendo todas las posibles palabras código asociadas a un secreto, una para cada combinación, que junto con u1 constituyen una configuración en estrella.

6. Encontrar una lista U_3, de todas las palabras código $\mathbf{u}^l \in U$ tales que $|u_p^l \in S_p^{(3)}| \geq 2(n-d)+1$. Notar que $|U_3| \leq 3$. Devolver $|U_3|$ y finalizar.

6. Un esquema de fingerprinting basado en la identificación de secretos

Un código de fingerprinting [2] es un conjunto de palabras código ("fingerprints"), de forma que cada palabra código es empotrada en una copia distinta de un documento digital. Las palabras código deben ser elegidas de forma que sea posible identificar al menos a un usuario culpable en caso de un ataque de confabulación. En un ataque de confabulación, una coalición de usuarios compara sus copias y crea una nueva copia pirata cambiando algunas de las marcas que pueden detectar. Bajo la premisa que la coalición sólo puede modificar las marcas en las que las copias difieren, el conjunto de copias piratas potenciales que la coalición es capaz de crear es precisamente el conjunto de descendentes de las palabras códigos pertenecientes a los miembros de la coalición. En consecuencia, la tarea de un algoritmo de rastreo es identificar a los padres de un descendente dado. El algoritmo de rastreo es prácticamente idéntico al algoritmo presentado en la Sección 5.2 que identifica secretos a partir de una secuencia de preguntas. Sin embargo, dado que un esquema de fingerprinting no se desea acusar a usuarios inocentes, por ello debemos establecer las siguientes restricciones: en caso de una configuración en estrella, la salida sólo puede ser la palabra código común de la estrella (padre) . En caso de una configuración en triángulo, el resultado sería el de un rastreo (decodificación) fallido.

El algoritmo toma como entrada una palabra $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n)$ del código concatenado y da como salida los padres seguros de esta palabra, vistos como palabras código del código externo. Sólo necesitamos reemplazar los pasos 5 y 6 del algoritmo de la sección 5.2 con los siguientes pasos.

5. Si $(|\mathcal{S}^{(1)}| + |\mathcal{S}^{(2)}|) > 2(n-d)$ entonces calcular

$U_{1,2} = \{\mathbf{u} \in U : |\{p : u_p \in S_p^{(1)} \vee u_p \in S_p^{(2)}\}| > 2(n-d)\}$.

Notar que $|U_{1,2}| \leq 2$.

Si $|U_{1,2}| = 0$ entonces ir al Paso 6, en otro caso la salida es $|U_{1,2}|$ y finalizar.

6. Encontrar una lista U_3, de todas las palabras código $\mathbf{u}^l \in U$ tales que $u_p^l \in S_p^{(3)}$ para todo $S_p^{(3)} \in \mathcal{S}$.

a. Para cada $S_p^{(2)} \in \mathcal{S}$ hacer:

Si existe un $S_p^{(2)} = \{s_{p_1}, s_{p_2}\}$ para el que hay exactamente 2 palabras código $(\mathbf{u}^1, \mathbf{u}^2) \in U_3$ tales que $u_p^1 = s_{p_1}$ and $u_p^2 = s_{p_2}$, entonces devolver $\mathbf{u}^1$ and $\mathbf{u}^2$ y finalizar.

b. Para cada $S_p^{(1)} \in \mathcal{S}$ hacer:

Si existe un $S_p^{(1)} = \{s_{p_1}\}$ para el que hay exactamente 1 palabra código $\mathbf{u}^1 \in U_3$ tal que $u_p^1 = s_{p_1}$, entonces devolver $\mathbf{u}^1$ y finalizar.

c. Decodificación fallida

Intuitivamente, la razón por la que los códigos concatenados de la Sección 5 son aptos en esquemas de fingerprinting es porque las palabras código están empotradas en objetos digitales y son desconocidas para los confabuladores, de forma que sólo pueden detectar las posiciones donde difieren. En este caso, puede conseguirse que la probabilidad de crear descendentes cuya decodificación desemboque en una configuración en triángulo sea arbitrariamente pequeña, incrementando la longitud del código. Nótese que si el algoritmo no finaliza en una decodificación fallida la salida identifica sin ningún genero de dudas a uno de los padres y nunca acusa a usuarios inocentes.

7. Conclusiones

A lo largo de este artículo se presenta en primer lugar un conjunto de cuestiones para resolver el problema de adivinar secretos $k = 2$ con un algoritmo eficiente para poder identificarlos. Dado que el conjunto explícito de preguntas se basa en la concatenación de un código binario (2,2) separable, donde ambos, el código interno y el externo se decodifican por encima de su capacidad correctora. Para la decodificación del código interno, se presenta una modificación del algoritmo de Chase, que aprovecha el hecho que la estructura del codigo descendente es un código dual Hamming y permite una búsqueda eficiente de todas las palabras código que están a distancia 2^{r-2}. El código externo es decodificado con el algoritmo de decodificación por listas de Guruswami y Sudan. Además, usando la relación conceptual entre el juego de adivinar secretos y las confabulaciones de usuarios deshonestos en esquemas de fingerprinting, se ha adaptado la solución del primer problema para su utilización en el rastreo de usuarios fraudulentos. El algoritmo presentado nunca acusa a un usuario inocente y la probabilidad que no pueda identificar a ninguno de los culpables se puede hacer arbitrariamente pequeña.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Ciencia y Tecnologia (CICYT) Proyecto TIC2002-00818 (DISQET).

Referencias

[1] N. Alon and V. Guruswami and T. Kaufman and M. Sudan. Guessing secrets efficiently via list-decoding. *Proceedings of the XIII ACM-SIAM SODA*, 254–262, 2002.

[2] D. Boneh and J. Shaw. Collusion-secure fingerprinting for digital data. *Lecture Notes in Computer Science*, 963:452–465, 1995.

[3] D. Chase. A class of algorithms for decoding block codes with channel measurement information. *IEEE Trans. Inform. Theory*, 18:170–182, 1972.

[4] F. Chung and R. Graham and T. Leighton. Guessing secrets. *The Electronic Journal of Combinatorics*, 8(1):R13, 2001.

[5] G. Cohen, S. Encheva, and H. G. Schaathun. On separating codes. Technical report, ENST, Paris, 2001.

[6] V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and algebraic-geometry codes. *IEEE Trans. Inform. Theory*, 45(6):1757–1767, 1999.

[7] H. D. L. Hollmann, J. H. van Lint, J.P. Linnartz, and L. M. G. M. Tolhuizen. On codes with the Identifiable Parent Property. *J. Combinatorial Theory*, 82(2):121–133, May 1998.

[8] Y. L. Sagalovich. Separating systems. *Probl. Inform. Trans.*, 30(2):14–35, 1994.

Sesión 5A

Redes de Área Local Inalámbricas y ad-hoc

Implementación de un Servicio de Microlocalización WLAN[1]

Xavier Garrido Riu, Miquel Oliver i Riera, Boris Bellalta i Jiménez
E-mail: xgarrido@telecom.upf.es, miquel.oliver@upf.edu, boris.bellalta@upf.edu
Ingeniería de Telecomunicación. Universitat Pompeu Fabra (UPF)
Edificio Estació de França. Passeig de la Circumval·lació, 8
Teléfono: 935422910-2945
08003 Barcelona

Abstract*. Location Systems and Services will be one of more successfully applications in wireless networks. User's location and information directed to the user, filtered according his position, offers an opportunity of enhancing the services of a 802.11b WLAN with no hardware changes. In this work, we present a microlocation system based on signal strength information loaded from the mobile device. The system estimates the user location with the help of a database of typical signal strength for each physical point of interest using algorithms that compare the real-time readings with stored points. The results show a good performance measures in the estimation of the user position in 3D indoor buildings.*

1 Introducción

En este artículo se presenta la implementación y evaluación de sistemas y servicios de localización utilizados en Redes de Datos Inalámbricas (WLAN) [1,2].

Una WLAN es una red local de conmutación de paquetes que emite en la banda de libre emisión situada en 2.4 GHz. Actualmente opera a velocidades entre 11 y 54 Mbps. Se trata de una tecnología estandarizada por el IEEE desde 1997. Inicialmente tuvo una aceptación baja debido a su alto coste y la falta de compatibilidad entre dispositivos, pero en los últimos dos años ha experimentado un crecimiento espectacular hasta el punto de llegar a pensar que puede competir con la telefonía móvil de tercera generación en escenarios de reducida movilidad y elevado ancho de banda, también conocidos como *hot-spots*.

Los Sistemas de Localización se están convirtiendo en uno de los servicios más atractivos y que más éxito pueden tener dentro de las redes móviles en general, y en particular, las WLAN. No sólo facilitan la ubicación física de los terminales, sino que también permiten ofrecer contenidos personalizados en función de cada perfil de usuario.

Los primeros Servicios de Localización que aparecieron para redes WLAN trabajaban a nivel de punto de acceso (AP, *Access Point*). A estos servicios se los conoce como Macrolocalización al ofrecer una resolución dependiente del área de cobertura del propio AP [3]. Estos servicios se basan en aproximar la posición del usuario móvil por las coordenadas físicas del AP al cual está asociado. La Macrolocalización ofrece una gran fiabilidad pero su precisión viene determinada por el tamaño de las celdas que definen los puntos de acceso. Es evidente que para una gran parte de aplicaciones la baja resolución que ofrecen estos sistemas puede resultar insuficiente y por esta razón surgió la necesidad de los sistemas de mayor precisión.

Estos sistemas de mayor precisión, conocidos como sistemas de Microlocalización [3] se basan en la capacidad del terminal móvil para obtener de su tarjeta wireless la potencia de señal recibida de todos los APs que detecta. Esta técnica también es conocida como *Indoor Location* puesto que el entorno ideal para desplegar este tipo de servicios es el interior de edificios. Esto es debido a la competencia con sistemas de posicionamiento tradicionales como el GPS (*Global Positioning System*) que ya representa un sistema de localización de similares prestaciones para exteriores, pero que pierde eficacia en interiores debido a la falta de visibilidad directa con los satélites que sustentan este servicio.

Para obtener un sistema de localización en interiores, el sistema propuesto trabaja a partir de información de lecturas de potencia en la tarjeta de cliente del dispositivo móvil. Utilizando un algoritmo de localización, se procesan las medidas de señal obteniéndose como resultado la ubicación más probable del usuario. El algoritmo de localización utiliza como información una base de datos con la potencia media que se recibe en un conjunto de puntos geográficos (Radiomapa). En consecuencia, el sistema necesita un estudio previo detallado del entorno para construir el mapa de medidas de potencia.

[1] Este artículo se ha realizado en el marco del proyecto MOBICAT – i2CAT (http://www.i2cat.net/)

A continuación se presenta la implementación y evaluación de una solución de Microlocalización (arquitectura del Sistema de Localización, Radiomapa y algoritmos) que aprovecha la propia infraestructura de la red WLAN. El presente trabajo se basa en las líneas de investigación marcadas en el proyecto Radar [4] desarrollado por Microsoft Research [5] y la Universidad de California.

2 Entorno de desarrollo

El escenario utilizado en el presente trabajo está formado por una red experimental con cuatro APs que cubren varias plantas interiores, así como los alrededores del edificio. Los cuatro APs son de diferentes proveedores, así como los adaptadores de red, con el objetivo de estudiar la compatibilidad entre ellos y plantear el problema en un escenario lo más realista posible.

El Servicio de Microlocalización desplegado se centra en un Servidor de Localización (Fig. 1) que recoge y procesa la información de todos los usuarios que controla. Al mismo tiempo, genera a través de un servidor web una representación gráfica de la situación de cada móvil sobre un mapa del edificio, y que a priori cualquier usuario del servicio puede descargar.

No obstante, la implementación también puede funcionar autónomamente y con total privacidad. Si por motivos de privacidad, el usuario del terminal conectado a la WLAN no desea enviar sus lecturas de potencia al Servidor de Localización, el propio terminal móvil puede ejecutar él mismo los algoritmos de localización para hallar su propia ubicación, sin necesidad de comunicarse con el Servidor de Localización.

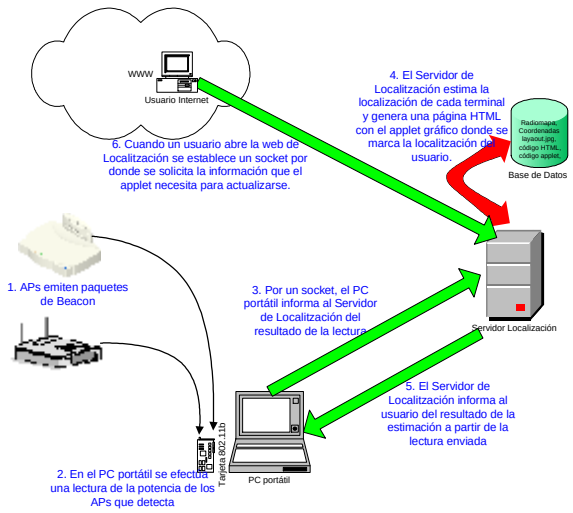


Figura 1: Arquitectura del sistema global de Microlocalización

3 Características

El Sistema de Microlocalización propuesto opera de forma probabilística, al no poder afirmar con total certeza la posición exacta del terminal móvil. Esta característica es intrínseca a todos los sistemas de localización. La información de salida que genera el servicio es el punto del Radiomapa, establecido previamente, en que la probabilidad de encontrar al usuario sea máxima.

Independientemente del algoritmo utilizado, la parte del sistema que se ejecuta en el terminal móvil realiza periódicamente una lectura de potencias de los APs recibidos desde su posición. A partir de esta información, bien sea en un servidor centralizado o en el propio terminal, se ejecuta el algoritmo de comparación entre la lectura in situ y todos los posibles puntos del Radiomapa.

Se necesita, por tanto, la colaboración del usuario ya que una parte del servicio residirá en el propio dispositivo móvil (como mínimo la realización de las lecturas de potencia). Este software que se encarga de realizar las lecturas de potencia es una aplicación codificada en C++ que usa funciones proporcionadas por la librería WRAPI [6]. Todas las otras partes de la implementación se han desarrollado en Java.

Este sistema tiene por el momento ciertas restricciones hardware y software que afectan al terminal móvil (tarjeta WLAN) y al sistema operativo, pero la creciente estandarización de estos productos hará, sin duda, reducir estos problemas iniciales en futuros dispositivos. Concretamente, la implementación desarrollada viene limitada por las condiciones que impone la librería WRAPI que trabaja en entorno Windows XP y con tarjetas adaptadoras con controlador compatible NDIS 5.1 (Network Driver Interface Specification).

4 El Radiomapa

El Radiomapa se estructura en una base de datos con todas las posibles localizaciones. Cada punto está caracterizado por la potencia media observada o estimada en el lugar que representa. Así, por ejemplo, si la zona que queremos supervisar esta cubierta por un conjunto de **N** APs, cada entrada de la tabla del Radiomapa podría tener la forma (x, y, z, SS_1, SS_2, ..., SS_N) donde x, y, z serían las coordenadas físicas absolutas o relativas del punto en 3D y las SS_i el nivel de señal que se recibiría de cada AP.

Una de las claves para que el Sistema de Localización funcione eficientemente es que el Radiomapa sea lo más preciso posible. Para crear un Radiomapa se plantean dos alternativas: medir la potencia RF in situ que se recibe de todos los APs en los sitios de interés (empíricamente); y emulación del entorno tratando de obtener la potencia que se recibiría de cada AP a partir de simulaciones con modelos de propagación en la banda de trabajo.

Por simplicidad, en este trabajo se ha optado por la primera opción, por parecer más viable, y por trabajar con medidas reales en el mismo escenario en que después estarán los usuarios móviles (así se evita

arrastrar los errores introducidos por los modelos de propagación).

Cabe destacar que el Radiomapa es el nexo de unión entre el software de localización y el entorno en el que tiene que localizar. Es por esta razón que cada zona de supervisión necesitará un Radiomapa particular que sólo será válido mientras se mantengan unas determinadas condiciones de propagación, las ubicaciones de los APs, que no se altere la estructura interior de paredes del edificio o que no se añada nuevo mobiliario que afecte al entorno RF.

Crear un Radiomapa preciso es un factor decisivo para el éxito posterior del Sistema de Localización. Para caracterizar cada punto se tiene que registrar la potencia media típica que se detecta de cada AP, pero esta señal sufre fuertes fluctuaciones (Fig. 2) provocadas por la propia naturaleza aleatoria de los canales RF, por el ruido radioeléctrico y por las propagaciones multicamino.

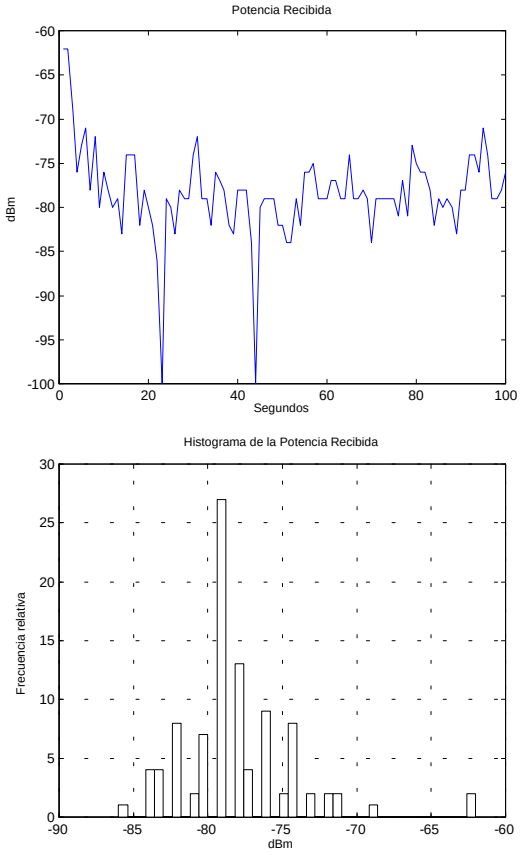


Figura 2: Distribuciones de la potencia de señal recibida de un único AP por un cliente en un punto del Radiomapa.

Por tanto, para establecer qué valores de potencia caracterizan un punto del Radiomapa se deberá realizar un estudio estadístico de la potencia recibida y de sus márgenes dinámicos.

Como se desprende de la Fig. 2, uno de los resultados remarcables consiste en que la desviación estándar se sitúa sobre los 4 dBm para la gran mayoría de puntos. Esto significa que es conveniente que todos los puntos del Radiomapa estén separados entre sí por una distancia Euclídea superior en el espacio de señales, de manera que se reduzca la probabilidad que, en un momento dado, a causa de una fluctuación profunda, se realice una lectura más propia de otro de los puntos del Radiomapa.

5 Estrategias y algoritmos para calcular las coordenadas

Para poder localizar se parte del principio que el nivel de potencia recibido es función de la distancia. Lógicamente, cuanto más próximos nos encontremos a un AP más fuerte será el señal recibido. Pero esto no es siempre cierto en un ambiente de interiores, ya que sólo conoceríamos la distancia respecto al AP y no la dirección, y además esta distancia sería poco fiable ya que el nivel de señal recibido también depende de otros factores, como el número de obstáculos que deberá atravesar y las diferentes trayectorias en las propagaciones multicamino.

Por esta razón, para localizar a los usuarios móviles, se ha optado por un sistema en que el dispositivo obtiene la potencia de señal que recibe de cada AP detectado y, posteriormente, usando los algoritmos adecuados busca en el Radiomapa cual de los puntos posibles es más similar a la lectura realizada utilizando criterios de distancia Euclídea.

La ubicación de los APs en la zona a cubrir es determinante. Para obtener un Radiomapa con los puntos bien caracterizados es importante que exista una cierta superposición de coberturas suficiente para procurar que en todos los puntos se reciba señal de varias estaciones base. Supongamos que recibimos en dos puntos diferentes un nivel de potencia parejo. La única manera de distinguirlos es que reciban diferentes potencias de otros APs. Esto supone un desaprovechamiento de los recursos ya que si nos propusiésemos cubrir la mayor área posible con el mismo número de APs no necesitaríamos estos solapamientos de coberturas.

A continuación se presentan los algoritmos que se han utilizado en este sistema. Primero se han implementado dos algoritmos (NNSS o *Nearest Neighbour in Signal Space*, y HBA o *History Based Algorithm*) siguiendo las directrices marcadas en el proyecto Radar [6,7], para finalmente presentar un tercer algoritmo propio al que hemos llamado NNSSMR (*Nearest Neighbour in Signal Space with Movement Restrictions*) que es una evolución de los primeros, reduciendo el efecto de los errores más habituales en los algoritmos originales, especialmente en entornos 3D, pues no habían sido diseñados para ello.

Definimos el espacio de señales como un espacio vectorial de dimensión **N** (donde **N** es el número de APs que dan cobertura) . En este espacio vectorial se representan todos los puntos de Radiomapa.

5.1 Nearest Neighbour in Signal Space (NNSS)

Este algoritmo decide la ubicación del usuario como el punto más próximo dentro del espacio de señales tomando como referencia las medidas instantáneas de señal, obteniendo, para cada lectura de potencia realizada in situ, la distancia Euclídea (o geométrica) con todos los puntos del Radiomapa. La localización del usuario se estima como la del punto que minimiza tal distancia.

Al efectuarse cada medida de señal los resultados se organizan en forma de vector de dimensión N, de manera que se identifica cada elemento del vector con uno de los APs que definen el espacio vectorial de señales, de la misma forma que están definidos los puntos del Radiomapa. A continuación se ofrece la expresión de la distancia entre la lectura efectuada (SS) y el punto i-ésimo del Radiomapa (1). El funcionamiento es muy parecido al criterio de máxima verosimilitud que se usa en algoritmos de codificación-modulación en los canales de comunicaciones digitales.

$$d_i = \sqrt{\sum_{j=0}^{N-1} (SS[j] - p_i[j])^2} \qquad (1)$$

Una variante del algoritmo anterior es el NNSS-AVRG. En ciertas ocasiones una lectura de nivel de señal puede tener muy próximos varios puntos del Radiomapa. En NNSS se optaría por una mínima diferencia. En cambio, en esta variante AVRG se propone quedarse con una lista reducida de los puntos más probables y estimar la localización del usuario como el resultado de promediar las coordenadas de los puntos seleccionados (Fig. 3).

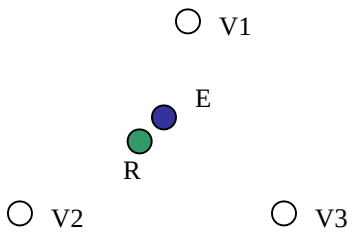


Figura 3: En NNSS-AVRG promediando varios puntos vecinos (V1,V2,V3) se puede estimar un punto (E) más próximo a la posición real del usuario (R) que cualquiera de los otros puntos V.

En este trabajo se ha descartado el algoritmo NNSS-AVRG debido a su bajo rendimiento. La razón está en que como el resultado de la estimación se obtiene al promediar las coordenadas de los tres puntos más probables y como en nuestro caso los puntos no son siempre equidistantes, ni tan siquiera siguen ninguna simetría, el punto estimado puede alejarse bastante del correcto. No obstante, el método se reaprovecha posteriormente en el algoritmo HBA, que a su vez se puede concebir como una versión mejorada del NNSS, porque no sólo tiene en cuenta los puntos más probables en una iteración, sino que utiliza información de lecturas posteriores (historia).

5.2 History Based Algorithm (HBA)

El HBA está pensado para registrar trazas de los usuarios móviles. Cada vez que el terminal efectúa una lectura, mediante la rutina NNSS-AVRG determina los *k* puntos más probables del espacio de señales que constituye el Radiomapa. Pero en este caso no se estima la localización inmediatamente, sino que se continúan realizando lecturas hasta tener una historia de profundidad *h*, es decir calcularemos la posición a partir de estos *k*h* puntos obtenidos.

Entre todas las parejas de puntos consecutivos se asigna un peso. Para modelar este peso se ha elegido de nuevo la distancia Euclídea, así si el peso es grande estos dos puntos son muy distantes, y si el peso es pequeño están muy próximos en el Espacio de Señales.

El camino más corto entre cada pareja de lecturas consecutivas representa la trayectoria más probable del usuario. Partiendo del punto más probable de la lectura actual (se puede calcular con la ayuda de otro algoritmo), se estima el punto de localización como el punto resultante después de haber recorrido toda la profundidad de *h* lecturas (Fig. 4).

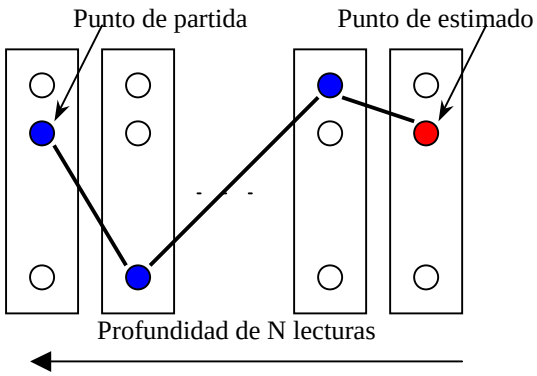


Figura 4: Algoritmo HBA

Cada vez que se realiza una lectura, se actualiza el vector de la historia añadiendo los nuevos *k* puntos más probables del Radiomapa para esa lectura y eliminando los *k* puntos más antiguos respecto a la nueva medida. Como se observa, el funcionamiento es similar al Algoritmo de Viterbi, utilizado en los receptores para determinar el mensaje más probable emitido en un canal ruidoso.

Como inconveniente, este sistema sufre un retardo de *h-1* lecturas de potencia para ser capaz de decidir la localización, es decir, no sabremos donde se encuentra el usuario hasta que se hayan realizado *h* lecturas de más.

Para el algoritmo HBA se ha tomado los valores de k=3 y h=6 (se consideran los tres puntos más probables de las últimas seis lecturas).

El valor de k proviene de NNSS-AVRG y debe ser mayor que 1 para aportar más información que NNSS, pero tampoco puede serlo mucho más, ya que si incluye un gran número puntos no filtraría los poco probables.

A priori, a mayor valor de h el algoritmo ofrecerá mejores prestaciones ya que tendrá una mayor profundidad, pero recordemos que h significa un retardo, ya que no podremos saber la estimación de una lectura hasta que no se realicen $h-1$ medidas de más. Es por este motivo que al aparecer estos retardos, este tipo de algoritmo sólo es válido para aplicaciones del tipo registro de trazas y no para servicios virtuales instantáneos como mapas *on line*.

5.3 Nearest Neighbour in Signal Space with Movement Restrictions (NNSSMR)

En este apartado se propone un nuevo algoritmo que hemos denominado 'NNSSMR'. El algoritmo se basa en imponer una serie de restricciones que condicionan los posibles puntos resultantes, mejorando su precisión.

El principal inconveniente de los algoritmos sin restricciones es que se ven seriamente mermados por los desvanecimientos o *fadings* profundos que afectan a este tipo de señales. Por ejemplo, si al efectuar una lectura de potencia, ésta se ve afectada por un desvanecimiento de señal en alguna de las dimensiones del vector de lectura o si simplemente no se detecta un AP de los que caracterizan el punto donde realmente se encuentra el usuario, el algoritmo puede dar fácilmente como resultado un punto erróneo del Radiomapa, realizando un cambio de posición anormal, incluso de planta a planta cuando se evalúa el sistema en un entorno 3D. Otra clase de error frecuente consiste al pasar por la vertical de un punto situado en otra planta, ya que el sistema se suele confundir con mucha facilidad, y no hay duda que equivocarse de planta supone un error no aceptable para un sistema de Microlocalización en interiores.

Por estos motivos, este algoritmo se ha diseñado de forma que sólo permita cambios de posición entre puntos vecinos del Radiomapa. Así, si un usuario está en un punto determinado, después de efectuar la siguiente lectura podrá únicamente o bien continuar en el mismo punto o bien cambiar a un punto vecino, pero nunca podrá repentinamente aparecer en el otro extremo del mapa o cambiar de planta sin pasar antes por zonas de acceso entre plantas diferentes.

El algoritmo funcionará correctamente si las lecturas se realizan con suficiente rapidez de manera que imposibiliten que en el tiempo entre dos iteraciones, el usuario haya atravesado todo un punto vecino. El tiempo de lecturas se debe tomar teniendo en cuenta la distancia media entre puntos vecinos del Radiomapa. En el entorno de trabajo, se ha tomado una separación entre puntos de 13 metros, considerando que un usuario se moverá en general a velocidades inferiores a los 2 m/s, tomando medidas periódicas cada 4 segundos es suficiente.

Para conseguirlo, se ha añadido a la Base de Datos del Radiomapa una extensión que contiene las restricciones de cada punto especificando hacia que estado tiene cada uno permitido evolucionar.

Además, en el Radiomapa se ha agregado un punto identificando *señal muy débil* que se caracteriza por recibir señal nula de todos los APs (punto origen en el Espacio de Señales) y no tener restricción alguna. De este modo al iniciarse el algoritmo se supone que el usuario se encuentra en este punto y una vez realiza la primera lectura de potencia, empiezan a aplicarse las restricciones.

No obstante, durante la fase de pruebas se detectó que debido a la propia fluctuación de las señales RF en interiores, al cambiar de posición, el sistema puede equivocarse al decidir y estimar la posición anterior o un punto vecino erróneo. En el momento de la siguiente lectura el usuario podía haber cambiado de posición nuevamente y, el sistema se desincronizaba. Esto ocurría porque el usuario podía estar en un punto para el cual el algoritmo no tenia permitido saltar a causa de las propias restricciones (suponía saltar 2 o más puntos vecinos de golpe). En este caso, el sistema en función de la lectura, permanecía atascado en un punto o bien trataba de llegar al punto real del usuario saltando de vecino en vecino cada iteración, pero sin seguir el camino correcto.

Para solucionar este problema, en cada lectura se realiza un seguimiento de la distancia Euclídea o geométrica en el Espacio de Señales entre el punto estimado y la lectura efectuada. Si durante dos lecturas consecutivas esta distancia es mayor que un cierto umbral (12 dB es un valor empírico que recomendamos), para que el algoritmo se reajuste se libera de las restricciones por una iteración y calcula la estimación por NNSS.

Esta resincronización sólo se hace cuando hay dos lecturas consecutivas con un error superior al umbral. Se ha observado que sólo una lectura catalogada de errónea puede haber sido provocada por un decaimiento temporal del señal y precisamente lo que se busca con este algoritmo es evitar, o cuanto menos minimizar estos saltos fugaces de ida y vuelta que provocan los *fadings*.

6 Resultados

Para realizar una primera evaluación aproximada del rendimiento del Sistema de Microlocalización, se ha desarrollado una aplicación que cada vez que efectúa la estimación, se le confirmaba manualmente la

posición del usuario. De esta manera se han podido medir las prestaciones de cada algoritmo. Únicamente se ha considerado como estimación errónea si la estimación era incorrecta y se ha descartado el error perteneciente a la distancia existente entre la posición exacta del usuario dentro de la microcelda y las coordenadas exactas que definen el punto o micro celda en el Radiomapa. Cuando una estimación ha sido errónea se ha aproximado el error como la distancia física entre las coordenadas del punto estimado y el correcto.

Los resultados corresponden a medidas tomadas por usuarios móviles, ya que en el caso de usuarios estáticos, el éxito del sistema es muy alto con todos los algoritmos, a menos que el usuario se sitúe en una zona frontera donde, debido a la propia fluctuación del señal radioeléctrico, el resultado de la estimación oscila entre dos puntos del Radiomapa.

En las pruebas todos los usuarios realizaban el mismo recorrido, pasando al menos una vez por todos los puntos del Radiomapa, según se probase en un entorno de una planta (2D) o de varias (3D). Por tanto, aunque en la Fig. 5 aparezcan conjuntamente los datos de los algoritmos 2D y 3D no son comparables entre sí. Para realizar el procesado estadístico, se ha dispuesto de más de 100 muestras para cada punto de estudio.

Una vez se dispone de la información del error en forma de distancia, se procesa esta información para obtener la media (media del error esperado), la mediana (valor central del conjunto de todas las muestras consideradas erróneas) y el percentil (cota de error para el 80% de los casos).

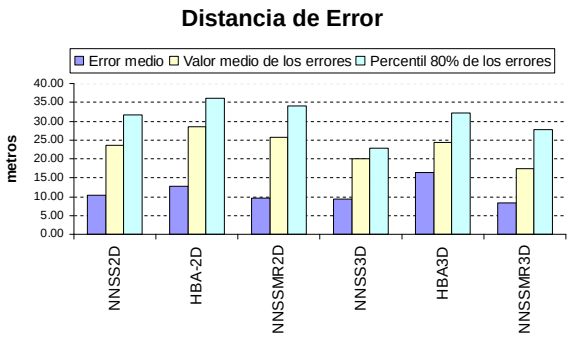


Figura 5: Distancia de error

Como se observa en la Fig. 5 el algoritmo HBA es el que ofrece peores resultados, sobre todo en 3D arrojando valores de error medio superiores a los 16 metros. Creemos que esto se debe a que este algoritmo sufre graves inconvenientes: el principal es que le cuesta detectar cambios entre celdas consecutivas porque al cambiar l usuario de punto en el Radiomapa, la celda antigua continúa apareciendo entre las *k* celdas más probables y entonces el sistema estima que no ha habido cambio. Esto se soluciona al hacer un cambio de punto hacía uno nuevo diferente de los dos anteriores, pero el algoritmo siempre conlleva un retardo intrínseco.

Entre los algoritmos NNSS y NNSSMR la diferencia es pequeña (9 y 7 metros de error medio, respectivamente), aunque siempre favorable al segundo.

Al analizar los errores puede parecer que la probabilidad de error sea muy elevada pero se debe tener en cuenta que todas las medidas se han realizado con usuarios móviles que cambiaban continuamente de microcelda. Esto supone el peor de los escenarios posibles ya que a medida que el usuario se desplaza pasa por muchos puntos de transición. Además el sistema se ha evaluado en una planta en forma de "L" de 139 x 35 m donde los puntos del Radiomapa están separados una media de 13 m (ver Fig. 6). La densidad de APs es demasiado baja, ya que algunos de los puntos únicamente recibían potencia de un único AP. Por esta razón, el error se reducirá ostensiblemente aumentado el número de APs.

7 Futuras líneas de trabajo

A continuación se presentan posibles líneas de continuación del trabajo de cara a mejorar las prestaciones del sistema de Microlocalización propuesto:

Adaptación al entorno: [5,7] Se propone disponer de varios Radiomapas y usar uno u otro en función de ciertas condiciones ambientales como por ejemplo la cantidad de gente que hay en el edificio, el nivel de la calefacción o la cantidad de luz. También sería interesante que los Radiomapas se fueran auto corrigiendo a partir de su propio uso, aunque para ello se requiere algún sistema corrector de las posiciones.

Light APs: [5] Con este tipo de estaciones base se pretende aumentar el área de solapamientos de cobertura. Estos dispositivos sólo emitirían mensajes de *beacon* pero no tendrían capacidades de red, así se abarataría mucho su coste y no se tendría que malgastar APs para obtener superposición de cobertura.

Encriptación entre dispositivos: [7] En determinados entornos corporativos donde la privacidad y la seguridad sean capitales sería necesario encriptar los sockets del sistema con una clave conocida únicamente por las dos partes (usuario y servidor).

Configuración automática: Se podría pensar en hacer el sistema más versátil y cuando que el usuario móvil entrase en una nueva WLAN con Servidor de Localización, automáticamente y de manera transparente al usuario, el Servidor enviase al dispositivo móvil la información necesaria para poder establecer las comunicaciones correspondientes,

liberando al usuario del problema de reconfigurar su aplicación de Localización para cada nueva WLAN.

Orientación del usuario: [7] Se está trabajando para tener en cuenta la orientación del usuario ya que la tarjeta wireless acostumbra a ir en unos de los laterales del portátil lo que produce que en determinadas orientaciones el propio dispositivo atenúe el señal del AP.

Dispositivos que no son PCs portátiles: Evolucionar la implementación para que seamos capaces de obtener lecturas de potencia de todos los APs que se detectan en terminales ligeros como iPAQ.

Diseño de una arquitectura de nivel medio (Middleware): [8] Estándar que agrupe todas las funciones de localización, de esta manera se facilitaría el diseño de aplicaciones al proporcionar estructuradamente toda la información disponible relativa a la localización en una red WLAN.

Integración con GPS: Creación de un sistema dual de manera que se pudiera obtener una localización de gran precisión con GPS en exteriores y cubrir con Servicio de Microlocalización las zonas como interiores de edificios y calles de grandes ciudades donde GPS no funciona correctamente.

8 Conclusiones

Los resultados de este trabajo demuestran la viabilidad de ofrecer un Servicio de Localización de gran precisión para usuarios móviles conectados a una red WLAN en un entorno de interiores sin que se requiera ninguna estructura especializada de hardware adicional. A pesar de la naturaleza variable de los radiocanales dentro de un edificio, es posible usar los propios recursos de una WLAN caracterizando cada ubicación posible a partir de la potencia recibida da cada uno de los APs próximos.

El éxito del sistema se basa en dos pilares. Por un lado, un Radiomapa preciso que represente con la mayor fidelidad posible el entorno a supervisar. Debe tener la mayor densidad de puntos posibles, pero sin que haya ningún par de puntos demasiado parejos entre sí (demasiado cercanos en el Espacio de Señales).

El segundo eje del sistema se apoya en el uso del algoritmo más conveniente para cada situación. Se han implementado diferentes algoritmos capaces de determinar en que punto del Radiomapa es más probable que el usuario esté ubicado. Entre todos los probados, el algoritmo que mejores prestaciones ha ofrecido es el NNSSMR.

Los resultados indican que es posible crear aplicaciones basadas en el reconocimiento del entorno por medio de la Microlocalización como navegar por el edificio, encontrar la impresora más cercana o ubicarse a sí mismo o a otros usuarios.

Como novedad se ha propuesto, y evaluado un nuevo algoritmo y se ha implementado el sistema en 3D, sin degradar apreciablemente las prestaciones respecto a un sistema 2D, aunque la precisión global cambia mucho dependiendo si se trata de un usuario mayoritariamente estático o de uno en constante movimiento.

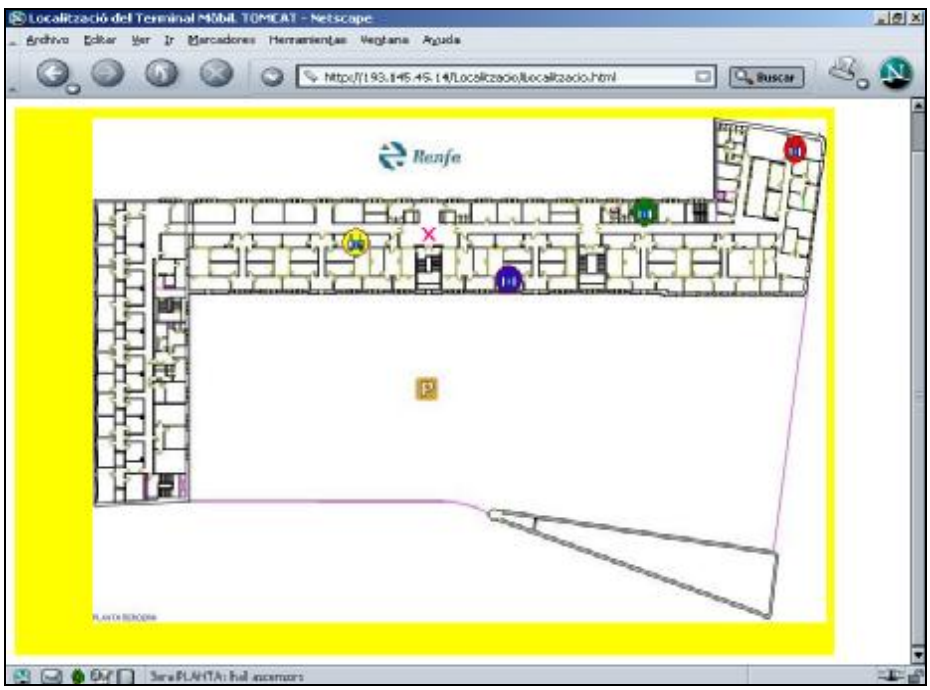


Figura 6: Applet generado por el Servidor de Localización

El mayor inconveniente que se ha apreciado es que cada vez que se quiere desplegar el sistema sobre una nueva zona, requiere un análisis meticuloso para generar un nuevo Radiomapa, que sólo será válido mientras no varíen las condiciones en que se ha realizado inicialmente.

Finalmente, resaltar que no debemos olvidar que los Servicios de Localización representan una herramienta que permite desarrollar aplicaciones en función de la ubicación del terminal móvil, y eso implica interactuar con el medio que rodea a cada usuario.

Referencias

[1] Gil Held, "Deploying Wireless LANs", McGraw-Hill Telecom, 2002

[2] Eric Ouellet, Robert Padjen, Arthur Pfound, "Building a Cisco Wireless LAN", Syngress Publishing Inc, 2002.

[3] Garrido X, Oliver M., "Implementació d'un Servei de Localització per a un entorn WLAN", Febrero 2003. Proyecto Final de Carrera – ETSETB (UPC).

[4] P. Bahl, V. N. Padmanabhan and A. Balachandran. "A Software System for Locating Mobile Users: Design, Evaluation, and Lessons". Microsoft Research Technical Report MSR-TR-2000-12. April 2000.

[5] Microsoft Research: Location Determination and Services. Radar Project http://www.research.microsoft.com/~bahl/MS_Projects/projects.html#radar

[6] Wireless Research API (WRAPI). http://ramp.ucsd.edu/pawn/wrapi/

[7] Blake M. Harris, "Amulet: Approximate Mobile User Location Tracking System". http://www.blakemharris.com

[8] The Parlay Group (Specifications): http://www.parlay.org/

Nuevo Protocolo de Encaminamiento para la Mejora de la Supervivencia en Redes Ad Hoc

Mari Carmen Domingo Aladrén, David Remondo Bueno y Olga León
Departamento de Ingeniería Telemática. Universidad Politécnica de Cataluña (UPC)
Escuela Politécnica Superior de Castelldefels
Av del Canal Olímpic s/n. 08860 Castelldefels (Barcelona)
Teléfono: 93 413 70 51 Fax: 93 413 70 07
E-mail: {cdomingo, remondo, olga}@ mat.upc.es

Abstract. Energy conservation will be one of the most significant aspects of ad hoc networks in the long term. This article presents a new version of the Dynamic Source Routing (DSR) protocol for mobile ad hoc networks that favours the selection of routes containing nodes with relatively high battery levels, in order to maximize the lifetime of the network as a whole. This proposal implies little modifications of the original algorithm, it can be easily implemented in practice and it does not require any centralized mechanism. After a quantitative analysis, we see extensive simulation results in relatively large ad hoc networks with high mobility. The results indicate that the proposed scheme, called SEADSR (Simple Energy Aware DSR), outperforms the standard DSR in terms of network survivability without any reduction of the network capacity. Besides, the signalling overhead is negligible and the amount of packets that arrive timely to their destinations is maintained.

1 Introducción

Las redes ad hoc están formadas por dos o más dispositivos que son capaces de comunicarse entre sí sin la necesidad de recurrir a una infraestructura de red preexistente. Dichas redes pueden estar constituidas formando grupos de terminales móviles independientes, aunque también cabría la posibilidad de que alguno de estos dispositivos estuviera conectado a un sistema celular o a una red fija. Las redes ad hoc son capaces por definición de configurarse a sí mismas, prescindiéndose de la intervención de un administrador del sistema.

Nuestro contexto se sitúa en lo que serían las redes ad hoc entendidas como un conjunto de dispositivos sin infraestructura formados por terminales inalámbricos y basados en radio enlaces.

En las redes ad hoc es posible que dos nodos inalámbricos se puedan comunicar entre sí, incluso cuando se hallan fuera de su alcance radio gracias a la presencia de nodos intermedios que actuarán como routers y reenviarán los paquetes de datos de la fuente al destino. Una red con este modo de funcionamiento recibe el nombre de red inalámbrica multisalto.

Los terminales móviles en las redes ad hoc se alimentan típicamente de baterías. La tecnología de las baterías ha experimentado un progreso muy lento si lo comparamos con los resultados alcanzados en la tecnología de circuitos integrados, donde se ha logrado un crecimiento espectacular en la velocidad de las comunicaciones. Por este motivo tiene especial importancia la conservación de la energía, que se conseguirá gracias al ahorro de potencia de transmisión y será un factor destacado a largo plazo en el rendimiento de los sistemas inalámbricos.

Los terminales en las redes ad hoc pueden funcionar no sólo como sistemas finales (ejecutando aplicaciones, enviando información como nodos fuente y recibiéndola como nodos destino), sino también como sistemas intermedios (reenviando paquetes de otros nodos). Por consiguiente, el consumo de potencia está muy íntimamente relacionado con el encaminamiento.

DSR (Dynamic Source Routing) [1] es un protocolo de encaminamiento para redes ad hoc, donde la topología varía dinámicamente debido a la movilidad de los nodos. No obstante, este protocolo no tiene en consideración el tiempo de vida de las baterías de los nodos en la red ad hoc. Con el fin de introducir este parámetro hemos procedido a modificar el protocolo DSR y hemos creado una nueva versión denominada SEADSR (Simple Energy Aware DSR).

En este trabajo hemos efectuado simulaciones de DSR y SEADSR utilizando IEEE 802.11b como capa de enlace de datos. Puesto que este estándar no implementa control de potencia, no tiene sentido optimizar el protocolo de encaminamiento en relación a la potencia de transmisión. De hecho, la supervivencia de la red mejora distribuyendo el tráfico de acuerdo con los recursos de batería disponibles.

La segunda sección explica el funcionamiento básico del protocolo DSR. En la tercera sección presentamos un estudio de la bibliografía relacionada para proceder a introducir y analizar posteriormente el protocolo SEADSR. La sección cuarta presenta los

resultados de nuestros experimentos. Finalmente, en la sección 5 se presentan las conclusiones.

2 DSR

DSR es un protocolo de encaminamiento bajo demanda, lo cual significa que se crean rutas únicamente en el caso de que un nodo fuente necesite enviar datos a un nodo destino.

Cada nodo mantiene una caché con rutas válidas que han sido establecidas en el pasado.

El protocolo utiliza dos mecanismos fundamentales: Descubrimiento (Route Discovery) y Mantenimiento de Ruta (Route Maintenance).

Cuando un terminal móvil necesita enviar paquetes a un destino concreto, lo primero que deber hacer es consultar su caché para ver si encuentra alguna ruta particular hacia ese destino y, si existe, hace uso de ella enviando paquetes.

Si no encuentra dicha ruta, el nodo fuente inicia un proceso de Descubrimiento de Ruta enviando un paquete broadcast denominado petición de ruta, RREQ (Route Request), con la dirección del nodo fuente, la dirección del nodo destino y un identificador RREQ. Cada nodo dentro del alcance de transmisión recibe este paquete y comprueba si es el destino o bien si su caché contiene una ruta activa hacia el destino. En estos casos, el nodo retorna un mensaje de petición de respuesta, RREP (Route Reply), que contiene una copia del registro de ruta acumulado en el RREQ. En el caso de que dicho nodo sea el destino, el registro de ruta representa todas las direcciones de los nodos intermedios que el RREQ ha atravesado en su camino desde la fuente hasta ese nodo concreto. Si este nodo no es el destino pero se trata de un nodo particular que conoce una ruta hacia el destino, añade la ruta contenida en su caché al registro de rutas y así genera el RREP de acuerdo con la información que posee.

Si este nodo ni es el destino ni conoce una ruta hacia el destino, entonces consulta si ya ha recibido un paquete con la misma fuente, destino e identificador de RREQ, o bien si su propia dirección ha aparecido en el registro de rutas. En este caso, con el fin de limitar el número de peticiones de ruta que se propagan a través de la red, el nodo elimina el RREQ. Por lo tanto, los RREQ que se propagan durante el Descubrimiento de Ruta son los que llegan primero a un determinado nodo. Se puede afirmar entonces que la ruta seleccionada es la más rápida que hay en ese momento para llegar al destino. Si no se cumple ninguna de las condiciones anteriores, el nodo retransmite el paquete, añadiendo su propia dirección al registro de rutas [2].

Cuando un nodo debe devolver el mensaje RREP al iniciador del Descubrimiento de Ruta, lo que hará será consultar su caché en busca de una ruta hacia el nodo fuente y la utilizará en caso de que exista. Por otro lado, si los enlaces son simétricos, el nodo puede invertir la ruta almacenada en el registro de rutas del RREQ. Si los enlaces no son bidireccionales, el nodo puede realizar su propio procedimiento de Descubrimiento de Ruta hacia el nodo origen y hacer 'piggyback' de su RREP en un nuevo RREQ.

Cuando un paquete sigue una ruta hacia un destino, cada nodo a lo largo de dicha ruta debe asegurarse de que el paquete ha sido recibido por el nodo siguiente del registro de ruta. Este objetivo se consigue gracias al empleo de paquetes de reconocimiento. Si un nodo intermedio no recibe dichos paquetes de reconocimiento, llega a la conclusión de que ha habido algún problema en el enlace y elimina las rutas que contienen dicho enlace de su caché de rutas. Entonces envía un paquete de error de ruta, RERR (Route Error) [3] a cada emisor que ha estado enviando paquetes a dicho nodo vecino desde la última vez en que se recibieron los reconocimientos. Este mecanismo se conoce como Mantenimiento de Ruta. Cuando se recibe un RERR, el emisor original utiliza otros caminos contenidos en su caché para reenviar los datos o bien inicia un nuevo procedimiento de Descubrimiento de Ruta.

Como se ha explicado anteriormente DSR selecciona la ruta más rápida al destino cuando utiliza el proceso de Descubrimiento de Ruta. No obstante, con DSR la ruta con el mínimo número de saltos es seleccionada únicamente en el caso ideal de que los paquetes RREQ sufran el mismo retardo de transmisión en cada enlace y que no se produzcan retardos por otros motivos (por ejemplo procesamiento o contienda MAC). En situaciones reales, habrá una alta probabilidad de que DSR seleccione rutas con el mínimo número de saltos, pero no existe ninguna garantía.

DSR es un protocolo relativamente sencillo, puesto que no utiliza una función de coste (un parámetro dependiente de varias variables que se utiliza para el encaminamiento). Este hecho descarga a los nodos de la ardua tarea de tener que calcular una función semejante cada vez que se reenvía un paquete.

3 Simple Energy Aware DSR

3.1 Encaminamiento y Disponibilidad de Energía

Han surgido varias propuestas que relacionan el encaminamiento con la disponibilidad de energía. En [4] se estudian dos protocolos de encaminamiento que ajustan la potencia de transmisión de forma dinámica teniendo en cuenta las tasas de error de la capa de enlace de datos y las consiguientes retransmisiones. Estas consideraciones motivan que en [5] aparezca un protocolo de encaminamiento basado en una función de coste que considera tanto la tasa de error del enlace como la energía necesaria

para realizar un único intento de transmisión a través del enlace.

En [6] se presenta un protocolo que conserva la capacidad de batería de los nodos 'apagando' aquellos nodos que no se dedican a la transmisión y recepción activa de paquetes.

En [7] se propone introducir características de la batería directamente en el protocolo de encaminamiento utilizando la capacidad de batería sobrante como métrica del tiempo de vida de cada terminal.

En [8] se realiza una comparación entre distintos esquemas de encaminamiento que consiguen minimizar la potencia de transmisión a la hora de seleccionar una ruta y otros esquemas que tratan de maximizar el tiempo de vida de los nodos de toda la red.

Los dos objetivos de minimizar la potencia total de transmisión para una ruta y para toda la red pueden llevarnos a una contradicción, por ejemplo, en el supuesto caso de que varias rutas de energía mínima tuvieran un terminal común, la capacidad de batería de ese terminal se agotaría rápidamente.

En [8] se presenta un nuevo esquema de encaminamiento que consigue satisfacer estos dos requisitos simultáneamente.

El algoritmo Minimum Battery Cost Routing (MBCR) consigue encontrar una ruta con la mayor capacidad total de batería sobrante. Definimos $f_i(c^t_i)$ como la función de coste de batería del nodo n_i, donde c^t_i representa la capacidad de batería del nodo en el instante t. Es posible escoger una función de coste como ésta:

$$f_i(c^t_i) = 1/ c^t_i \quad (1)$$

El coste de batería R_J para una ruta seleccionada J será entonces:

$$R_J = \sum_{i=0}^{D_J-1} f_i(c^t_i), \quad (2)$$

donde D_J representa el número de nodos que pertenecen a la ruta J. Con el fin de seleccionar una ruta con la máxima capacidad total de batería remanente, es necesario escoger una ruta m que tenga el mínimo coste de batería:

$$R_m = min\{R_J | J \in A\}, \quad (3)$$

donde A es el conjunto que contiene todas las rutas posibles.

3.2 Descripción del SEADSR

En este trabajo se asume que la potencia de transmisión es fija e igual para todos los nodos. El protocolo DSR no puede incorporar ninguno de los algoritmos anteriormente mencionados con potencia de transmisión fija porque no podemos introducir una función de coste directamente en dicho protocolo(ya que el RREQ se retransmite dependiendo de si no ha llegado anteriormente otro RREQ similar al mismo nodo). El motivo es que DSR selecciona la ruta dependiendo de los tiempos de llegada a los nodos intermedios de los paquetes que tienen el mismo identificador RREQ en el proceso de Descubrimiento de Ruta.

Los autores [9] especifican un protocolo de encaminamiento que mejora la supervivencia de la red, manteniendo al mismo tiempo la sencillez del DSR: Simple Energy Aware Dynamic Source Routing (SEADSR).

La idea básica detrás de este algoritmo es la siguiente: Cuando un nodo intermedio en una red ad-hoc decide reenviar un mensaje de RREQ (en el modo DSR) que ha recibido, se introduce un retardo adicional antes de retransmitir dicho mensaje:

$$\tau = [C_{max} - C(t)]\ \tau_{max} / C_{max}, \quad (4)$$

donde C_{max} es la capacidad de la batería, $C(t)$ es el nivel de batería actual y τ_{max} hace referencia a un parámetro de diseño que representa el máximo retardo introducido. Podemos apreciar que τ toma un valor entre 0 y τ_{max} y es directamente proporcional a la energía consumida por el nodo.

La selección de ruta dependerá, al igual que en DSR, de los factores previamente mencionados, pero este retardo adicional introducido establece además una interdependencia entre la selección de una ruta y los niveles de batería de los nodos. El parámetro τ_{max} desempeña un papel importante en el proceso de selección de ruta. Cuanto mayor sea el parámetro τ_{max}, mayor será la influencia de los niveles de batería en comparación con otros factores. Si escogemos un valor de τ_{max} grande, el algoritmo tenderá a seleccionar rutas que comprenden a nodos con altos niveles de batería pero que posiblemente tendrán más saltos. Por el contrario, si se escoge un valor pequeño para τ_{max}, los niveles de batería de los nodos intermedios ya no serán un factor decisivo y el algoritmo seleccionará con una mayor probabilidad rutas con menos saltos y niveles de batería más bajos.

Es importante resaltar que el funcionamiento del algoritmo será diferente dependiendo de la función de retardo seleccionada. No es posible encontrar la función de retardo óptima debido a la variabilidad de tráfico y de la topología. No obstante, otras funciones de retardo alternativas pueden mejorar el rendimiento

general del sistema [10], aunque estos métodos se alejan del verdadero propósito de nuestro artículo.

En [9] los autores presentan simulaciones relacionadas con la supervivencia de red para redes con un número significativo de flujos de tráfico. Los resultados mostraron una pequeña mejora sobre DSR para los valores de escenario y parámetros seleccionados. El esquema en [9] se basa también en el mecanismo básico explicado en esta sección. No obstante, podemos esperar que los valores de timeout de las cachés sean decisivos en el proceso de selección de rutas y por consiguiente en el rendimiento del algoritmo. Cuando una ruta ha permanecido activa durante un tiempo, los nodos que forman parte de la ruta tendrán niveles de batería bajos y la ruta ya no será la óptima. A pesar de todo, como dicha ruta se halla contenida en la caché, será escogida preferiblemente antes que otras a la hora de establecer un nuevo flujo de datos hacia el mismo destino y no será posible iniciar un proceso de Descubrimiento de Ruta para localizar otras rutas con mejores propiedades desde el punto de vista energético. Por este motivo, nosotros creemos que la caché de rutas perjudica al buen funcionamiento del algoritmo en cuanto a energía se refiere. Por consiguiente, el algoritmo de encaminamiento SEADSR propuesto en este artículo no tiene caché de rutas. SEADSR es, por tanto, una modificación de DSR que suaviza los requerimientos de memoria.

En [10] se sugiere introducir también un retardo en el Procedimiento de Descubrimiento de Ruta de DSR. No obstante, nuestro algoritmo es diferente debido a la inexistencia de cachés. Además, nosotros hemos hecho simulaciones considerando redes mayores (escalabilidad) y mucho más móviles.

3.3 Análisis del SEADSR

Considérese una topología de red como la ilustrada en la Fig. 1. En t = 0 s un nodo fuente llamado S quiere enviar paquetes a un nodo destino D. Por lo tanto el protocolo de encaminamiento debe encontrar el mejor camino hacia el destino. Nosotros consideraremos dos posibles rutas llamadas SABD que atraviesa los nodos A y B, que tienen una capacidad de batería de 30 J, y la ruta SCD que atraviesa el nodo C con una capacidad de batería de 14 J. Cuando un nodo actúa como nodo intermedio o router, consume una potencia de transmisión de 5 W. Para facilitar el análisis hemos supuesto que las potencias de recepción y procesamiento son insignificantes. Hemos asumido que en t = 1 s otra fuente llamada S' desea enviar paquetes al mismo nodo destino, de forma que es necesario establecer una nueva ruta. Hemos supuesto que la red utiliza diferentes protocolos de encaminamiento para poder estudiarlos y compararlos. Los tres protocolos propuestos son MMBCR [5], DSR y SEADSR.

En MMBCR (Min-Max Battery Cost Routing) el coste de batería R_J para una ruta J sería:

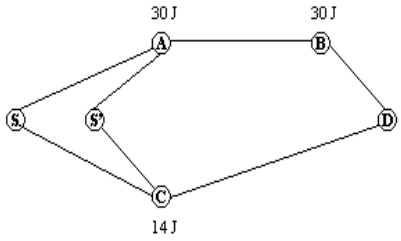


Fig. 1. Red ejemplo. Se muestran los valores iniciales de energía.

$$R_J = \max_{i \in J} f_i(c_i^t), \quad (5)$$

Se selecciona la ruta m con el mínimo coste de batería:

$$R_m = min \{ R_J \mid J \in A \}, \quad (6)$$

donde A es el subconjunto que contiene todas las rutas posibles.

Es importante destacar que con este protocolo de encaminamiento no hay garantía de que sea seleccionada la ruta con la mínima potencia total de transmisión.

En la Fig. 2 hemos comparado la energía consumida por los nodos intermedios en función del tiempo para los tres esquemas de encaminamiento.

Podemos apreciar que DSR consigue los peores resultados en términos de supervivencia de red, porque las dos fuentes siempre seleccionan la ruta a través de los nodos CD, de forma que en t = 1,9 s el nodo C ha agotado su energía y únicamente permanece disponible la otra ruta. Por otro lado, el protocolo de encaminamiento MMBCR consigue en estos casos rutas con un tiempo de vida mayor en comparación con DSR porque las dos fuentes siempre usan la ruta a través de los nodos ABD que contiene nodos intermedios con mayor capacidad de batería. Por lo tanto es cierto que con este protocolo de encaminamiento se consume más energía pero al mismo tiempo la supervivencia de red mejora y los nodos A y B agotan sus reservas de energía en t = 3,5 s.

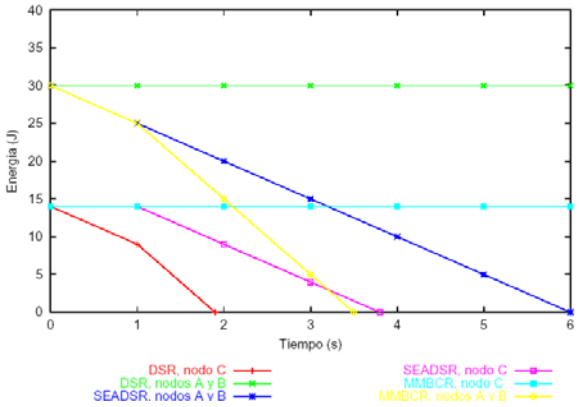


Fig. 2. Energía con DSR, SEADSR y MMBCR para la red de la Fig. 1 (C_{max} = 40 J).

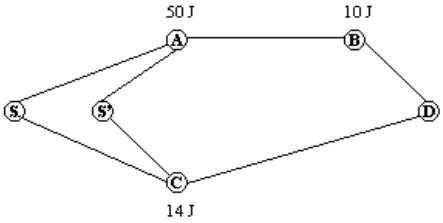


Fig. 3. Red ejemplo. Se muestran los valores iniciales de energía.

Finalmente, podemos apreciar que SEADSR logra un funcionamiento mejor. En este caso cada fuente selecciona una ruta diferente (SABD es escogida por la fuente S y S'CD por la fuente S'). Así, la energía total consumida es mayor que en DSR y menor que en MMBCR, pero a la vez se extiende el tiempo de vida de todos los nodos de la red más que en los otros dos protocolos de encaminamiento. El nodo C agota su energía en t = 3,8 s. Podemos considerar este caso como el típico, porque los niveles de energía en los diferentes nodos no son muy dispares.

Aunque no sea fácil introducir una función de coste en SEADSR debido al funcionamiento del protocolo de encaminamiento, podemos apreciar que en realidad estamos seleccionando el mejor camino hacia el destino de acuerdo con la siguiente expresión: $\min[\Sigma 1/B_i]$, donde B_i representa la capacidad de batería actual.

Ahora consideraremos el ejemplo de la Fig.3 donde tenemos la misma topología y condiciones de red que en la Fig. 1, pero en este caso se asume que los nodos A y B tienen niveles de batería de 50 J y 10 J respectivamente.

Podemos apreciar en la Fig. 4 que DSR y SEADSR experimentan los peores resultados en términos de supervivencia de red, porque la ruta SCD es siempre seleccionada por las dos fuentes, de forma que en t = 1,9 s el nodo C ha agotado su capacidad de batería y sólo la otra ruta permanece disponible. Este es el peor caso para el protocolo SEADSR: Únicamente puede funcionar igual que DSR en las peores condiciones y topología de red. Podemos apreciar que las condiciones de esta red benefician al protocolo MMBCR, el cual selecciona la ruta SCD para la primera fuente y la ruta S'ABD para la segunda. A pesar de que la energía consumida es mayor, el consumo se distribuye a lo largo de las dos rutas de forma que los nodos B y C agotan sus reservas de energía en t = 3 s y t = 2,8 s respectivamente.

4 Simulaciones

El simulador usado en este trabajo para la evaluación de los protocolos de encaminamiento es ns-2 [11].

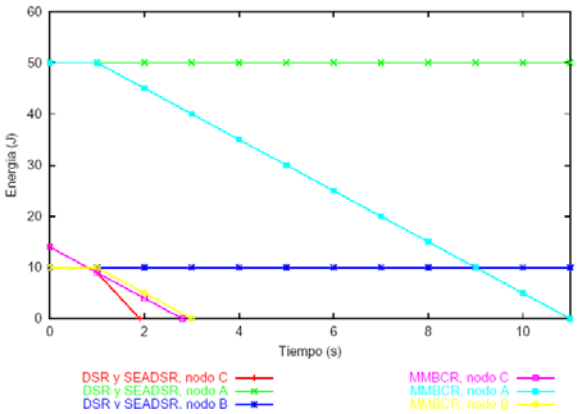


Fig. 4. Energía con DSR, SEADSR y MMBCR para la red de la Fig. 3 (C_{max} = 60 J).

Cien terminales móviles que usan IEEE 802.11b se distribuyen aleatoriamente de acuerdo con una distribución uniforme en una región cuadrada de 600 m por 600 m. Cada nodo elige un punto de destino al azar dentro del área y se mueve hacia él a una velocidad uniformemente distribuida entre 0 y 3 m/s. El tamaño del área y el número de nodos han sido seleccionados de forma que en media se necesitan varios saltos para llegar de la fuente al destino.

Una vez el nodo ha alcanzado su destino, hace una pausa durante un periodo fijo de 20 segundos, escoge otro destino y repite el proceso.

A diferencia de los resultados obtenidos en [10], la gran cantidad de flujos de tráfico que hemos estudiado nos permite determinar como se comportan las cachés.

Se envían paquetes UDP (User Datagram Protocol) de 512 bytes, espaciados 100 ms y generados por fuentes de tráfico de tasa constante (CBR).

Un nodo cualquiera puede ser seleccionado como nodo fuente con una probabilidad de 0,16 y tratará de encontrar una ruta hacia el destino durante un intervalo de tiempo entre 0 y 180 s. El parámetro de diseño τ_{max} es 1 s para el SEADSR.

Puesto que nuestro objetivo es analizar el impacto de los protocolos de encaminamiento en la supervivencia de la red, hemos considerado las perdidas de energía de los nodos de la red ad hoc debidas únicamente a su funcionamiento como nodos intermedios, es decir, debido a la energía necesaria para reenviar paquetes originados por otros nodos. Por esta razón, hemos modelado los nodos con dos baterías: Una de estas baterías proporcionará energía permanentemente cuando el nodo funcione como nodo fuente; la otra capacidad de batería disminuirá cada vez que el nodo reenvíe un paquete, de forma similar a lo que ocurre en un nodo intermedio. La variable *C(t)* en (4), utilizada por el algoritmo de encaminamiento, representa el nivel de la segunda de

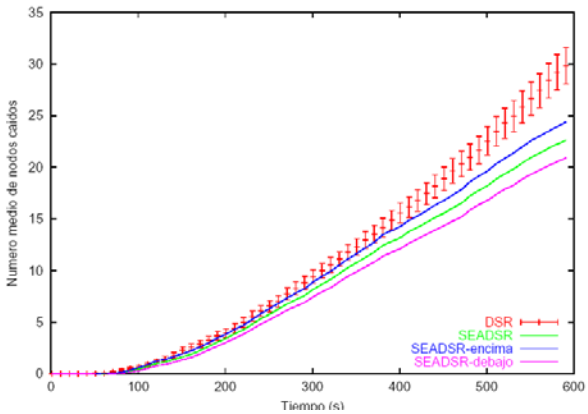


Fig. 5. Fallo de los nodos debido al agotamiento de sus reservas de energía en función del tiempo. Se muestran los intervalos de confianza del 90%.

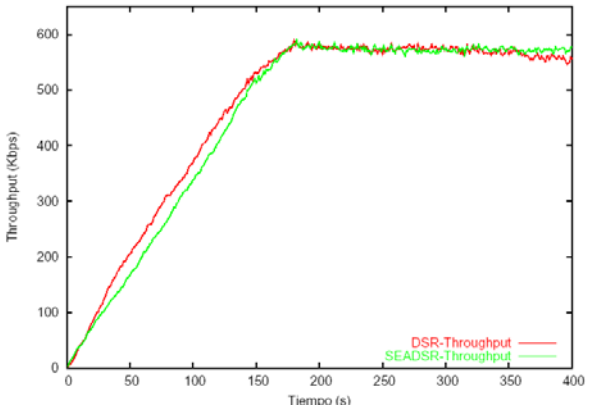


Fig. 6. Throughput en función del tiempo.

estas baterías. Aunque este modelo pudiera considerarse poco realista, la alternativa produciría una rápida disminución del nivel energético de algunos nodos (las fuentes), impidiendo así el análisis del protocolo.

La Fig. 5 muestra las pérdidas de nodos debido al agotamiento de su energía en función del tiempo para el DSR estándar y el SEADSR. Hemos considerado que todos los nodos tienen una capacidad de batería de 0,8 J al principio de la simulación. En la Fig. 5 encontramos el tiempo medio de fallo del nodo enésimo usando DSR y SEADSR. Los intervalos de confianza son del 90% para 80 experimentos independientes. Las simulaciones tienen una duración de 600 s.

En la Fig. 5 puede observarse que en el estándar DSR hay más nodos que han agotado su capacidad de batería en función de tiempo que en SEADSR. Hasta el segundo 300 los intervalos de confianza son demasiado grandes para extraer conclusiones. No obstante, a partir de ese punto puede apreciarse que el número de nodos caídos aumenta en el estándar DSR más rápidamente que en SEADSR. Por consiguiente, SEADSR supera a DSR en cuanto a la supervivencia de la red.

Para comparar nuestro protocolo de encaminamiento con DSR en relación al throughput y el retardo de paquetes, se hicieron simulaciones de un sistema con suficiente capacidad de batería de forma que durante el tiempo de simulación no fallaran los nodos debido al agotamiento de su energía. Después de algunas pruebas, descubrimos que todos los nodos de la red con una capacidad de batería de 5 J permanecen con vida al menos 400 s.

La Fig. 6 muestra el throughput en función del tiempo para el estándar DSR y SEADSR, obtenido a partir de 80 experimentos independientes. Los primeros 180 s de la Fig. 6 corresponden al periodo durante el cual las fuentes comienzan a enviar tráfico.

Cuando t<150s, DSR es superior a SEADSR, puesto que selecciona las mejores rutas en relación al retardo extremo a extremo. Por el contrario, cuando el sistema está completamente cargado, el mejor balanceo de carga para el caso del SEADSR compensa la sobrecarga de señalización adicional debida al descubrimiento de rutas.

La Fig. 7 representa el histograma de retardo de paquetes de datos utilizando parámetros con los mismos valores. Muestra el número de paquetes de datos en función del retardo que han experimentado en su camino desde la fuente hacia el destino a través de la red. Cada simulación tiene una duración de 400 s y los retardos de los paquetes son medidos desde el segundo 180 (cuando todas las fuentes están activas) hasta el segundo 400. Hemos hecho 30 experimentos para DSR y para SEADSR. Cada punto de la abcisa en la Fig. 7 corresponde a un intervalo de retardo de 10ms. Por ejemplo, el valor de la abcisa 2 indica que aproximadamente 2000 paquetes de datos han sufrido un retardo entre 20 y 30ms.

Los resultados muestran que el número de paquetes de datos con retardos inferiores a 10 ms es sólo un 5% más pequeño en SEADSR que en el estándar DSR. La situación cambia para retardos entre 10 ms y 50 ms. En este caso, encontramos más paquetes de datos cuando se usa SEADSR que con el estándar DSR. Se observa, no obstante, que el número total de paquetes de datos que llegan al destino con un retardo aceptable (por ejemplo menor que 30 ms) se mantiene. Aunque SEADSR introduce un retardo adicional en el mecanismo de Descubrimiento de Ruta, esto influirá en un número relativamente pequeño de paquetes de datos en comparación con el número de paquetes favorecido mediante el balanceo de carga.

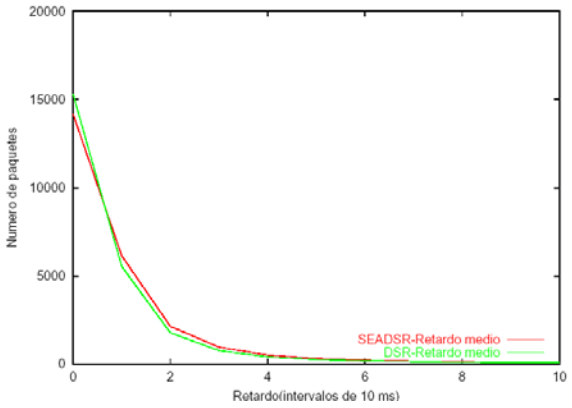


Fig. 7. Histograma del retardo de paquetes de datos.

5 Conclusiones

Hemos efectuado simulaciones en redes ad hoc con condiciones exigentes para los protocolos de encaminamiento (número de nodos grande, alta movilidad y número medio de saltos elevado) para mostrar que el protocolo propuesto SEADSR supera al estándar DSR en relación a la supervivencia de red sin que se vea reducida la capacidad del sistema.

En simulaciones de sistemas con poblaciones de nodos estables (ningún nodo muere debido a agotamiento de sus reservas de energía), los resultados indican que a pesar del retardo adicional de encaminamiento y del recargo de señalización introducidos por SEADSR, el número de paquetes de datos que llegan puntualmente al destino (retardos inferiores a 30 ms) sigue manteniéndose.

SEADSR alarga el tiempo de vida de los terminales inalámbricos en conjunto, lo cual repercute en el throughput total de la red, en el caso de que los recursos energéticos sean escasos.

Referencias

[1] D.B. Johnson and D.A. Maltz, "DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks," in Ad Hoc Networking, C.E. Perkins, ed. Addison Wesley, 2001.

[2] D.B. Johnson, D.A. Maltz, Y. Hu and J.G. Jetcheva, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-07.txt , 21 Feb. 2001. IETF Internet Draft.

[3] C-K. Toh, Ad hoc Mobile Wireless Networks: Protocols and Systems, Prentice Hall PTR, 2002.

[4] A. Misra and S. Banerjee, "MRPC: Maximizing Network Lifetime for Reliable Routing in Wireless Environments," IEEE Wireless Commun. and Networking Conf. (WCNC), Orlando, Florida, U.S.A., March 2002.

[5] S. Banerjee and A. Misra, "Minimum Energy Paths for Reliable Communication in Multi-hop Wireless Networks," Mobihoc'02, Lausanne, Switzerland, June 2002.

[6] S. Singh. and C.S. Raghavendra, "PAMAS-Power Aware Multi-Access protocol with Signalling for Ad hoc Networks," ACM Comm. Review, Jul. 1998.

[7] S. Singh and C.S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks," Proc. of MobiCom'98, Dallas, Texas, U.S.A., Oct. 1998.

[8] C.-K. Toh, H. Cobb and D. Scott, "Performance Evaluation of Battery-Life-Aware Routing Schemes for Wireless Ad Hoc Networks," Proc. IEEE Int. Conf. on Communications (IEEE ICC), Helsinki, Finland, June 2001.

[9] M.C. Domingo, O. León and D. Remondo, "On the Extension of Battery Life with Dynamic Source Routing," IFIP WG6.7 Workshop and EUNICE Summer School on Adaptable Networks and Teleservices (EUNICE' 2002), Trondheim, Norway, Sep. 2002.

[10] W. Yu and J. Lee, "DSR-based Energy-aware Routing Protocols in Ad Hoc Networks," Int. Conf. on Wireless Networks (ICWN), 2002.

[11] Ns-2: Network Simulator, http://www.isi.edu/nsnam/ns/

Localización en redes WLAN 802.11: desarrollo e implementación de una solución basada en traps SNMP

Eduard Garcia Villegas, Rafael Vidal Ferré
Departament d'Enginyeria Telemàtica. Universitat Politècnica de Catalunya.
EPSC, Avda. del Canal Olímpic, sn. 08860 Castelldefels
Teléfono: 934 137 055 Fax: 934 137 007
E-mail: eduardg@entel.upc.es, rafael.vidal@entel.upc.es

***Abstract**. The growth of WLANs and hence, the increase of mobile data terminals, bring about a visible proliferation of location-based services. This scenario also carries the need for location management systems that are device independent. In this paper, after an overview of several different solutions, we present a location discovery system for WLANs 802.11, based on SNMP traps. As a result of our work, two location aware applications have been implemented. One of these applications is a web service for location discovery that can be used to suggest the nearest points of interest (noteworthy tourist spots, shops, libraries, etc.). The other is a centralized application proposed for WLAN administrators to localize mobile users on their current position or to track those users on the past. The bulk of location-based services don't have the need of a big granularity, therefore our solution simply identifies to which AP a mobile user is currently connected to.*

1 Introducción

Durante el curso 2001/02, a iniciativa del grupo de comunicaciones inalámbricas del Departamento de Ingeniería Telemática, se llevo a cabo el despliegue de una red WLAN con tecnología IEEE 802.11b en la EPSC (Escuela Politécnica Superior de Castelldefels) formada en una primera fase por tres puntos de acceso (APs) conectados a la red de la Escuela [1]. El objetivo del despliegue era doble. Por un lado se pensaba en ofrecer a toda la comunidad que forma la EPSC de un acceso sin hilos de calidad a su red y, por extensión, a la de la UPC y a Internet. Por otro, se buscaba disponer de una plataforma sobre la que desarrollar y probar soluciones para el soporte de la movilidad, como por ejemplo Mobile IP y Cellular IP, aplicaciones que sacarán provecho de ella, además de desarrollar herramientas que permitiesen una fácil monitorización de la propia red WLAN y sus usuarios. Es precisamente en este último punto donde se centra el presente artículo.

1.1 Localización en redes móviles

Para la monitorización de la red WLAN se tomo como punto de partida el protocolo SNMP (*Simple Network Management Protocol*) [2] y la información contenida en las MIBs (*Management Information Base*) de los APs utilizados. A partir de esta información se desarrolló una web de acceso restringido que permite monitorizar el correcto funcionamiento de la red a través de estadísticas actualizadas de determinados parámetros de la misma y la visualización de las alarmas producidas en ella. Todo ello mediante la utilización de MRTG (*Multi Router Traffic Grapher*) [3].

El siguiente paso era la monitorización de los usuarios y en concreto de su localización. La motivación para conseguir este objetivo era constatar cómo el acceso a la información de localización supone un valor añadido de cara a la gestión de una red con usuarios móviles, a la vez que abre las puertas al desarrollo de nuevos servicios como ya se está comprobando para el caso de las redes de comunicaciones móviles celulares de segunda generación. Algunos ejemplos de estos nuevos servicios son: el control de flotas, las campañas de marketing vía mensajes cortos dirigidas a zonas concretas de especial interés o la búsqueda de un determinado servicio (restaurantes, farmacias, hospitales, museos,...) en función a la posición del usuario. Aunque la precisión con que se conoce la posición de un usuario no es comparable a la de sistemas de localización vía satélite, es suficiente para el desarrollo de muchas aplicaciones. Además, las redes celulares presentan algunas ventajas muy significativas respecto a las de satélite. En primer lugar superan en cobertura a los sistemas satélite en entornos urbanos y muy especialmente en el interior de edificios; y sobretodo, el usuario de la red celular no necesita de ningún hardware ni software adicional a su terminal telefónico para ser localizado.

En el momento de plantearse una solución para la localización en redes WLAN se ha tomado precisamente este último aspecto como condición irrenunciable: para localizar un usuario de la red WLAN no se necesitará más hardware que su tarjeta de red ni ningún software adicional más allá de los controladores de ésta. En el segundo capítulo del artículo se clasifican y comparan las diferentes soluciones de localización en redes WLAN existentes, a la vez que se introduce la desarrollada por los autores basada en el envío y tratamiento de *traps* SNMP. La arquitectura funcional de esta solución se describe con detalle en el tercer capítulo.

El trabajo realizado culmina con la implementación de esta arquitectura en la WLAN de la EPSC y el

desarrollo de aplicaciones que pudiesen mostrar las posibilidades de esta solución. En el momento de buscar utilidades a la información de localización en redes WLAN es necesario observar que estas redes presentan algunas particularidades respecto a las celulares, que hacen pensar en otras aplicaciones. En primer lugar, su ámbito de explotación es privado en cuanto a sus usuarios y reducido en cuanto a cobertura. En nuestro caso, por ejemplo, el acceso se limita a la comunidad de alumnos, profesores y personal de servicios que forma la Escuela y la cobertura al edificio donde ésta se ubica. En segundo lugar, los dispositivos utilizados, típicamente portátiles y PDAs (*Personal Digital Assistants*), disponen de mayores capacidades de proceso, almacenamiento y multimedia. Y finalmente, la velocidad de acceso es mucho más elevada, con 802.11b, entre 1 y 11Mbps. Mientras que las mayores prestaciones de los terminales y la mayor velocidad de acceso permitirán el desarrollo de aplicaciones más sofisticadas que podrán enviar y/o recibir volúmenes de datos más elevados en un menor tiempo, las limitaciones de su ámbito de explotación nos llevaran a pensar en soluciones a medida de las necesidades de los administradores y usuarios de una determinada red WLAN. Por ejemplo: museos, campus universitarios, edificios de oficinas,...

Todo esto se ha tenido en cuenta en la implementación del sistema que se ha realizado y que se describe en el cuarto capítulo. Esta implementación se ha traducido en dos aplicaciones. Una de ellas dirigida a los administradores de red permite rastrear y localizar a los usuarios de ésta. La otra, ofrece a los usuarios información a medida según su localización mediante un servicio vía web. Respecto a esta última aplicación, en el quinto capítulo se comentan las posibles causas de error en la localización que puede introducir el sistema y se describen las pruebas realizadas para estudiar como afectan a los usuarios. Finalmente, se cierra el artículo con las conclusiones en las que se comentan los principales logros conseguidos y las líneas futuras de trabajo que piensan seguirse.

2 Localización en redes 802.11

Las estrategias de localización de dispositivos móviles en redes WLAN 802.11 se pueden agrupar en dos grandes bloques: las que sólo utilizan información de nivel de enlace y superiores y las que también se fijan en parámetros de nivel físico, en concreto en el de potencia de señal recibida.

Las soluciones del primer tipo permiten determinar el AP al que un nodo móvil se encuentra asociado. Es decir, tienen una resolución igual al área de cobertura del AP. Esta resolución puede mejorarse completándolas con soluciones del segundo bloque, es decir, con estrategias basadas en la medida de niveles de potencia. Básicamente, estas soluciones, realizan conversiones de niveles de señal, medidos desde diferentes puntos, en longitudes, a partir de las cuales, mediante un algoritmo, el sistema devuelve las coordenadas x, y, z del dispositivo [4]. De hecho, muchas de las soluciones existentes proponen métodos más sencillos en dos dimensiones, haciendo la (falsa) suposición de que los APs y el cliente se encuentran en un mismo plano [5][6][7]. Para la conversión nivel de señal/distancia, la teoría dice que hay una relación $1/r^2$, pero los resultados empíricos ponen de manifiesto que la respuesta real es difícil de describir con ecuaciones. En entornos diferentes a espacios abiertos, las conversiones de potencias a distancias pueden resultar engañosas debido a la atenuación producida por paredes de hormigón, ficheros metálicos, ascensores, etc. Por este motivo se usan modelos empíricos de propagación particulares que no son válidos de un entorno a otro.

La complejidad introducida por las soluciones basadas en medidas de señal es en muchos casos injustificable, pues como ocurre en el caso de las redes celulares, conocer la zona de cobertura en la que se encuentra un usuario puede ser más que suficiente para muchos servicios de localización. Por este motivo, nos centraremos en el estudio del primer bloque de soluciones basadas en la información de nivel de enlace y superiores.

A priori, la forma más evidente de realizar una localización de este tipo, sería utilizar la información que el dispositivo cliente conoce, pues éste sabe al instante la identidad del AP al que se conecta, es más, lo sabe con antelación ya que debe mantener estadísticas a partir de medidas de potencia para decidir el traspaso de un AP a otro. El problema de esta solución es que no es independiente del sistema operativo (SO) o de la arquitectura del dispositivo al servirse de ciertas llamadas a sistema.

Otra posibilidad sería aprovechar que normalmente los APs trabajan haciendo de *bridge* transparente, por lo que mantienen una tabla (*bridge learn table*) con información de los dispositivos que han estado asociados al AP recientemente. Una consulta periódica a estas tablas mediante el protocolo SNMP, proporciona la información necesaria para el funcionamiento de otro sistema para localizar usuarios [8]. Un inconveniente importante es que el AP mantiene dicha información sobre un usuario durante un periodo entre 15 y 30 minutos después de que éste se haya desconectado o se haya asociado a otro AP, por eso, se debe buscar la información más reciente de entre todos los APs y mantener, un fichero *log* con el resultado de las consultas SNMP a los diferentes APs. Otro inconveniente es que para obtener una medida más precisa, se deben realizar consultas con más frecuencia, lo que supone un aumento significativo del tráfico en la red.

Otra solución es el uso de RADIUS (*Remote Authentication Dial-In User Service*) [9], que, junto con EAP (*Extensible Authentication Protocol*) [10][11], aparte de proporcionar control de acceso a los APs, puede usarse para mantener información sobre la situación de usuarios de una red WLAN. Cada AP actúa como un cliente RADIUS que comprobará la autenticación de los usuarios dentro de

su área de cobertura. Si la autenticación es correcta, se guarda la información (hora, identidad del AP y MAC de cliente) en un fichero log que luego se inspecciona para obtener el AP al que un determinado usuario, conocida su dirección física MAC, está conectado [8]. El principal problema de esta solución es el hecho de que, aunque el uso de EAP (sobre 802.1x) está ampliamente aceptado en equipos fijos (AP, *switch*, etc.), no todos los fabricantes de tarjetas cliente 802.11 apuestan por ese método para realizar un control de acceso seguro en redes sin cables [12].

Finalmente, el método que se quiere exponer en este artículo, se basa en una funcionalidad del protocolo de gestión de redes SNMP diferente a la vista hasta ahora. Sin los inconvenientes de las consultas periódicas a las tablas de todos los APs, se obtienen mejores resultados que con la solución RADIUS (ver tabla 1), ya que no se requiere una autenticación previa tipo cliente-AP-RADIUS, solución más costosa en términos de tiempo y tráfico. Este método se basa en el envío de *traps* SNMP para la gestión de eventos. Un *trap* (o *notification*, como se rebautiza en SNMPv2) es un mensaje enviado por una entidad SNMP a otra, para indicar la ocurrencia de un evento significativo, como una condición definida específicamente, o un umbral que ha sido alcanzado. Esto significa que, ante ciertos eventos, el dispositivo no espera a que hagan una consulta SNMP para mostrar la información de dicho evento, lo que hace es informar inmediatamente de éste a un determinado nodo de la red. En el caso que nos atañe, cuando un AP detecta la conexión o desconexión de un cliente, automáticamente envía un *trap*. De esta manera, en cuanto un usuario se conecta a la WLAN o cambia su punto de acceso, se dispone de la información sobre su localización, es decir se sabe en todo momento, a qué AP está conectado el cliente, sabiendo su dirección física MAC.

Una vez descartada la solución basada en medidas de potencia, ampliando los datos de [8], en la tabla 1 podemos ver las ventajas del último método sobre los demás:

- Se trata de un sistema rápido
- Genera poco tráfico adicional
- No requiere software adicional en terminales
- Independiente del SO y de la arquitectura del cliente

Como contrapunto, el sistema estará limitado a redes cuyos APs sean capaces de generar *traps* al detectar asociaciones de usuarios.

3 Arquitectura del sistema

La arquitectura del sistema de localización basado en *traps* SNMP, versiones uno, dos y/o tres, se compone de cinco entidades lógicas:

- **Usuario móvil**: dispositivo con interfaz de red 802.11 cuya dirección física se conoce y que se desplaza por el área de un ESS (*Extended Service Set*).
- **AP**: punto de acceso de los usuarios móviles a la red de área local. Estos dispositivos deben ser capaces de generar *traps* SNMP ante eventos de conexión o desconexión de usuarios.
- **Servidor de localización (SLOC)**: proceso que recibirá *traps* SNMP y que guardará su información en un fichero log o en una base de datos.
- **Log**: fichero o base de datos donde se almacena la información recibida de los *traps* SNMP.
- **Servidor de aplicaciones (SAP)**: proceso que lee e interpreta la información del log para mostrarla a los usuarios de la aplicación de localización.

En la figura 1 puede verse un ejemplo de la interrelación de estos elementos. En primer lugar el AP1 informa al servidor de localización mediante el envío de un *trap* sobre la asociación del usuario con MAC 1. SLOC guarda esta información y le añade la hora en que llegó el mensaje. A continuación, el usuario se mueve y pasa a asociarse a un nuevo AP, AP2. Esto provoca el envío de un nuevo *trap* por parte del AP2 a SLOC, que lo procesa tal como se ha comentado.

Tabla 1: comparación entre sistemas de localización en WLANs

	En cliente	RADIUS	SNMP	Traps SNMP
Servicios de red adicionales	NO	Cliente RADIUS en AP	AP con SNMP	AP con SNMP y generación de traps
Señalización adicional	NO	Queries RADIUS cada asociación	Consultas periódicas	Traps SNMP por cada asociación
Software adicional en cliente	Sí	No	No	No
Independiente de S.O.	No	Sí	Sí	Sí
Independiente de hardward	No	Si	Si	Si
Dificultad de implementación	Alta	Baja	Media	Baja
Tiempo de respuesta	Inmediato después de asociación	3 a 5 segundos	10-30 segundos (según periodo consulta)	1 a 2 segundos

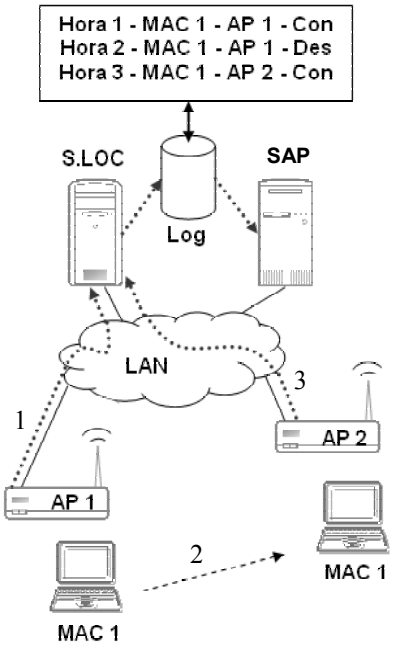


Figura 1: Arquitectura del sistema de localización

4 Aplicaciones desarrolladas

Sobre la arquitectura descrita en el apartado anterior se han desarrollado dos aplicaciones que sacan partido a la información de localización de los usuarios en una red 802.11 desde dos puntos opuestos:

- **XLOC**. Herramienta para administrador que permite, desde un punto central de la red, conocer el área de cobertura donde se encuentra un determinado usuario y rastrear sus movimientos.
- **Servicio de páginas web inteligente**. Las páginas servidas son personalizadas según la localización del usuario que hace la petición.

En los siguientes subapartados se explicará el detalle del funcionamiento de estas aplicaciones, realizadas y probadas sobre la red WLAN 802.11b de la EPSC.

4.1 Implementación

Los *traps* SNMP no están estandardizados y el formato de la información varía según el fabricante del AP. La aplicación ha sido implementada con soporte para los *traps* SNMPv1 de los APs 3COM AirConnect, y los *traps* SNMPv1 y SNMPv2 de los APs Cisco Aironet 340/350, pero puede ser modificada fácilmente para admitir formatos de otros APs. En ambos casos, cuando un AP detecta un evento de tipo ***MU state change*** (cambio de estado de una unidad móvil), ya sea provocado por la asociación de un nuevo usuario al AP o por la pérdida de un usuario que estaba asociado, envía un trap SNMP al servidor de localización.

La información que incluyen los *traps* SNMP que recibe el servidor de localización consta de:

- Dirección IP del AP que ha detectado el evento
- Dirección física (MAC) del AP que ha detectado el evento
- Dirección física (MAC) del cliente, el estado del cual ha cambiado
- Nuevo estado del cliente (asociado, desasociado, perdido, etc.)

El trato dado a la información contenida en el trap es dependiente de la implementación. En nuestro caso, por sencillez, se guarda en un fichero de texto que reside en el mismo servidor de localización, pero una vez obtenida, la información puede almacenarse en cualquier tipo de base de datos, lo que facilitaría su posterior análisis y el poder ser accedida desde cualquier tipo de aplicación (PHP, servlet, JSP, etc.).

Así pues, lo único que conocen los APs sobre un cliente asociado es su dirección MAC. Por este motivo, en las dos aplicaciones se realiza una búsqueda basada en direcciones MAC para saber en qué AP está asociado y así conocer el área donde se encuentra el usuario. Esta idea se recoge en el diagrama de la figura 2, donde se puede ver el funcionamiento simplificado del sistema. Por un lado, hay un proceso continuamente esperando la llegada de *traps* SNMP, cuando eso sucede, se limita a añadir esa información al sistema de **log**. Dejando de lado temas de concurrencia, el esquema es válido para representar los procesos que siguen las dos aplicaciones. Ambas esperan una petición de búsqueda por parte del usuario. Partiendo de una situación en que el usuario a localizar está registrado, es decir, conocemos el mapeo IP → MAC, se obtiene del sistema de almacenamiento de *traps* la información sobre el AP al que está asociado. El último paso consiste en generar la información personalizada según la localización y mostrarla al usuario. Si, de entrada, no se tiene información sobre el cliente a localizar o sobre el AP al que se ha asociado un cliente, la aplicación mostrará un error.

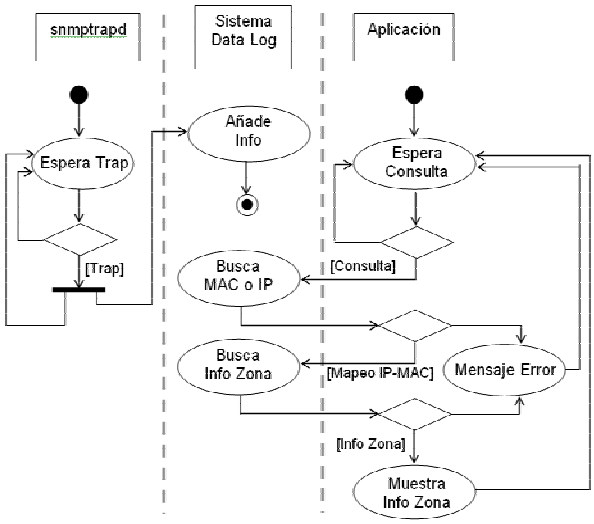


Figura 2: Diagrama de actividad del sistema

La importancia del mapeo IP → MAC se comentará en los siguientes apartados.

4.2 Servidor de localización

En nuestra maqueta, el servidor de localización se monta sobre un PC Pentium con sistema operativo (SO) Linux Red Hat 7.3 (*kernel* 2.4.18). Para recibir correctamente los *traps* que envían los APs se requiere la instalación del demonio ***snmptrapd***. Este demonio forma parte del paquete *ucd-snmp* [13]. En nuestra maqueta se ha instalado la versión 4.2.4-3 del paquete, incluida en la distribución del SO instalado, aunque cualquier versión posterior, actualmente bajo el nombre de *net-snmp*, es válida.

Los mensajes que el demonio ***snmptrapd*** recibe, pueden guardarse directamente en un fichero o ser enviados al demonio ***syslogd*** de la misma máquina o de otra diferente. ***Syslogd*** es el encargado de tratar los mensajes del sistema.

En nuestro caso, el servidor de localización, además de ser la máquina preparada para recibir los *traps* SNMP que envían los AP, es donde se ejecuta un CGI para la aplicación de páginas web personalizadas y donde se ejecuta la aplicación del administrador. Por ello, además de lo anterior, se ha instalado la versión 1.3.23 del servidor HTTP ***Apache*** [14] para atender las peticiones de clientes de la aplicación de páginas web personalizadas. Para compilar y ejecutar la aplicación de administrador **XLOC**, es necesario instalar las librerías GTK+ [15], versión 1.2.0 o superior, y *gnome* [16], versión 1.0.0 o superior.

4.3 Páginas web inteligentes

Como ya se ha comentado, la información que sobre un cliente móvil lleva un *trap* SNMP se limita a su dirección MAC. Sin embargo, a nivel HTTP, para distinguir usuarios sólo podremos usar direcciones IP. Por ello, es necesario conocer el mapeo IP – MAC de los usuarios del sistema y tener guardada esa información en una base de datos o en un fichero de configuración. En nuestro caso, se utiliza un fichero de configuración donde, además, se guarda información de los APs y de las zonas que éstos cubren.

Para que un cliente pueda hacer uso de la aplicación, bastará con que haga una petición web al servidor de localización, pidiendo el ejecutable CGI. Si el cliente está registrado en el sistema (conocemos su mapeo IP – MAC), verá una página HTML como la que se muestra en la figura 3, donde se puede ver cuál es el área donde se encuentra en ese momento, información sobre el AP al que está asociado, fecha y hora de su asociación e información sobre la zona donde se encuentra (tiendas, restaurantes, bibliotecas, etc.)

Para mostrar siempre la información sobre la localización actualizada, se puede provocar que la página se refresque de manera automática con más o menos frecuencia, haciendo uso del *Meta Tag* [17] "Refresh", que, aunque en un principio sólo era válido para navegadores Netscape, actualmente es interpretado correctamente por la inmensa mayoría de los navegadores existentes.

4.4 Aplicación Administrador (XLOC)

Se trata de una aplicación programada en C, que hace uso de las librerías GTK+ 1.2 y *gnome-libs* para proporcionar una interfaz gráfica amigable. Está pensada para ser ejecutada en la misma máquina donde se guarda la información sobre los *traps*, es decir, el servidor de localización. Si se desea ejecutar esta aplicación en cualquier otra máquina, se puede configurar el *demonio* ***snmptrapd*** o el ***syslogd*** para

Figura 3: Aplicación de web personalizada

redirigir los *traps* hacia ella.

Como en el caso de la aplicación de páginas web personalizadas, se necesita un fichero de configuración donde guardar información de los APs y de los clientes. Dicho fichero es prácticamente igual al fichero de la aplicación CGI.

La ejecución del programa hará aparecer la ventana que se muestra en la figura 4. En la parte superior de ésta aparece una entrada de texto para escribir la dirección física del cliente a localizar o seleccionarla del menú desplegable donde aparecen las direcciones que la aplicación lee del fichero de configuración. La parte inferior está ocupada por un plano de la Escuela en el que se marcan las diferentes zonas de cobertura, para facilitar la ubicación espacial de los usuarios.

Hay dos modalidades de búsqueda, la primera, mediante el botón ***Histórico***, permite hacer un seguimiento de los *traps* que han llegado al servidor de localización referentes a un determinado cliente. Las flechas permiten avanzar o retroceder en el tiempo y así conocer los movimientos realizados por el cliente. Dicha modalidad nos facilita la información retenida a lo largo del tiempo en el fichero de *traps*, pero no nos asegura si el cliente que buscamos está conectado en ese mismo instante. Para ello se utiliza la otra modalidad, botón ***Localiza***, con la que se fuerza el envío de mensajes *ICMP echo request* (ping) al cliente, de esta manera, si se recibe respuesta, podemos estar seguros de que el cliente está conectado y ver en qué AP.

5 Pruebas de campo

Las pruebas que se explican en este apartado fueron realizadas para determinar el grado de fiabilidad que tienen los datos sobre localización que muestran las aplicaciones. Entendemos como error cuando, sabiendo nuestra posición, es decir, a qué AP estamos asociados, la aplicación nos muestra unos datos incorrectos.

El error puede ser debido a tres causas, pudiendo presentar las dos primeras una varianza significativa. Como primera causa aparece el tiempo que transcurre desde que el cliente decide asociarse a un determinado AP hasta que se genera el *trap*. El tiempo de asociación (o des-asociación) del cliente al AP puede variar de 100 a 600 ms dependiendo de la combinación de modelo de dispositivo cliente con diferentes modelos de AP [18]. La segunda causa es el tiempo de envío y recepción del *trap*. Estos tiempos dependerán de la capacidad del AP para generar el *trap* en el momento que se produzca el evento que lo provoca y de la red para transportarlo. En cualquier caso se trata de valores de ms dentro de una red LAN. Finalmente, la tercera fuente de error es el tiempo de refresco de la página en el navegador del cliente que es del orden de segundos.

Las zonas donde hay cobertura de más de un AP, y por tanto, los traspasos son frecuentes, son las zonas donde el comportamiento del sistema puede ser más conflictivo y donde, por tanto, se han realizado las pruebas. Dichas zonas se representan en la Fig. 5 mediante líneas numeradas (de la forma X-Y) en el punto donde se detectan traspasos del AP X al AP Y.

Para determinar la máxima exactitud del sistema las pruebas de error se han realizado minimizando la influencia de las fuentes de error controlables:

- Refresco de la página con la máxima frecuencia posible. El cliente recibe la

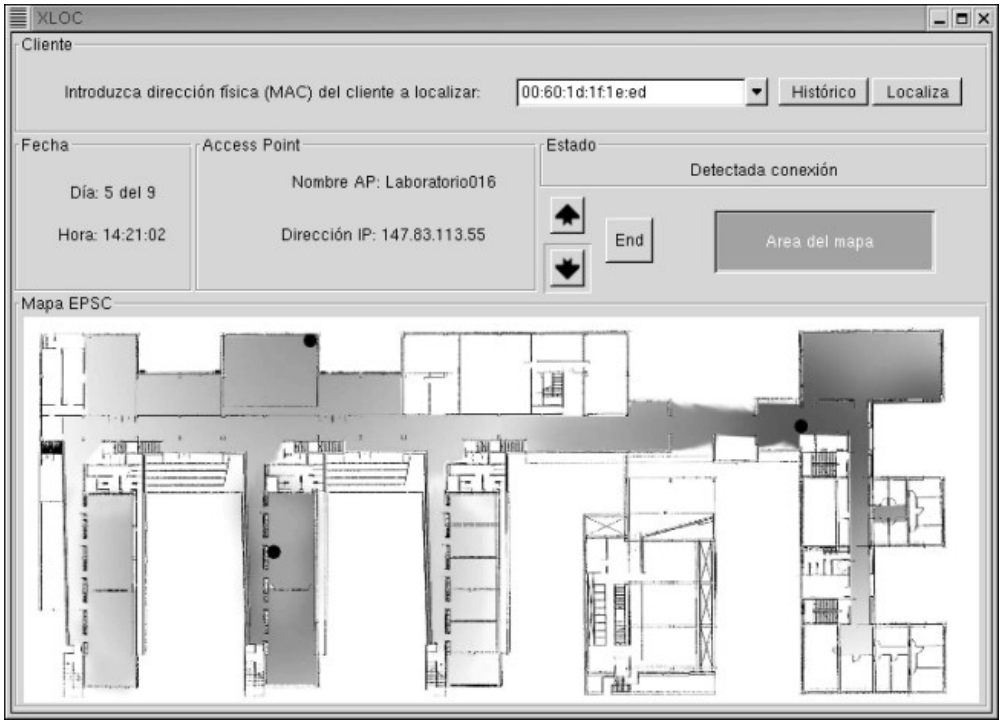


Figura 4: Apariencia de la aplicación de administrador

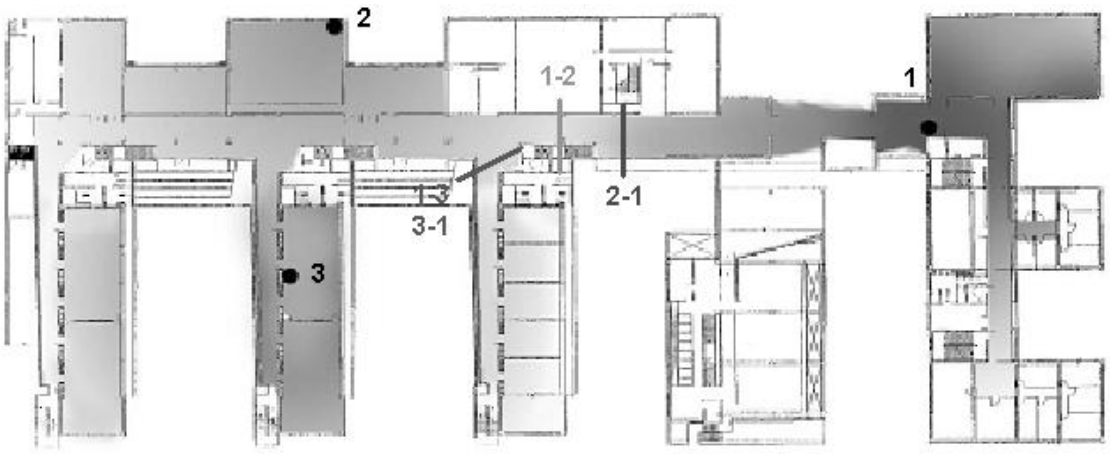


Figura 5: detalle de los límites entre las áreas de cobertura de los tres APs

información cada segundo, i.e. la precisión de la medida está limitada a 1s.

- Versión reducida de la página web (4KB). El tiempo que dura la transferencia de los datos no es significativo ya que a la velocidad mínima (1Mbps), representa tan sólo un retardo de unos 30 ms.
- Red con poca carga. No tendremos tampoco en cuenta el retardo entre el AP que genera un trap y el servidor que lo recibe (orden de ms), ya que no influye en la medida si trabajamos con valores dentro de un orden de magnitud de segundos.

La información obtenida de la aplicación es comparada con la obtenida de las utilidades incluidas con los controladores de las tarjetas 802.11, gracias a las cuales, se puede conocer de manera instantánea la información del AP al que el cliente está asociado.

Por norma general, el cliente recibe la información actualizada con un retardo de alrededor de 1s. Sin embargo, en el peor caso, es decir, cuando el traspaso se realiza justo después de la última actualización de la página CGI, el error es cercano a los 2 segundos. Ello es debido a que la información se actualiza en el servidor de localización dentro del primer segundo después del traspaso y la primera actualización de la página CGI llega aún con la información antigua, por lo que, para tener la nueva información deberá esperar un segundo más. Ese retardo puede suponer, traducido en longitud, un error de entre 1 y 2 metros dentro de una área de cobertura adyacente a la que muestra la aplicación como situación del cliente, suponiendo una velocidad pedestre típica (unos 4 ó 5 km/h).

Finalmente, debe añadirse, que al enviarse los *traps* SNMP sobre UDP (modo datagrama, sin acuse de recibo), podría llegar a producirse la pérdida de un *trap*. En esta situación, el usuario recibiría información errónea hasta que volviera a asociarse a otro AP. De todos modos, para un sistema de localización basado en *traps* SNMP instalado en una red de área local sin congestión y con pocos saltos entre APs y servidor, la probabilidad de perder un paquete pequeño, como un trap SNMP, es prácticamente nula, no habiéndose producido ni una sola vez en las pruebas realizadas.

6 Conclusiones y trabajo futuro

En este artículo se ha descrito una solución que permite localizar usuarios de una red WLAN 802.11 sin que sea necesario hardware o software adicional en los dispositivos móviles. Dicha solución depende solamente del soporte, por parte de los APs, al envío de *traps* SNMP relacionados con la asociación de usuarios.

Para demostrar la utilidad del sistema, se han implementado dos aplicaciones: una orientada a administradores de redes WLAN y otra a los usuarios de éstas. Las aplicaciones han sido probadas y actualmente están en funcionamiento en la red WLAN de la EPSC. Las pruebas realizadas y aquí expuestas, demuestran que el tiempo de respuesta del sistema, concretamente para el caso del servicio de páginas web inteligentes, es adecuado para usuarios que se desplazan dentro del área de cobertura de la red a velocidades pedestres.

Como contrapunto, el sistema presenta un punto débil al depender de UDP para el transporte de los *traps* entre APs y servidor de localización, ya que la pérdida de un *trap* puede resultar en que las aplicaciones muestren información errónea a la hora de ubicar el usuario que provocó el envío del *trap* perdido. Tal pérdida se produce con una probabilidad prácticamente nula cuando el sistema se utiliza en una LAN, pero es un asunto muy a tener en cuenta si se pretende implantar en una WAN, caso que puede ser muy interesante como se comenta más adelante. Este contrapunto puede resolverse a medida que el uso de SNMPv2 se normalice y los fabricantes doten a los APs de la capacidad de enviar mensajes *inform* en lugar de *traps*. Aunque también viajan sobre UDP, su recepción debe ser confirmada por el destinatario.

La proliferación de redes WLAN abiertas y organizadas sin ánimo de lucro así como la figura de los WISP (*Wireless Internet Service Providers*), con acuerdos de *roaming* incluidos, está rompiendo con

la idea de que las redes WLAN son de ámbito reducido y para usuarios privados. Una sola organización puede controlar un número significativo de redes y usuarios con lo que la información de localización aumenta de valor, acercándose más al modelo de negocio de los operadores celulares. Este modelo pasa por separar los datos de localización de los servicios que pueden ser prestados por otras empresas. Para conseguir este propósito y manteniendo la arquitectura descrita en el tercer capítulo, se está modificando la implementación presentada de manera que el servidor de localización escriba la información de localización en una base de datos SQL residente en otro equipo, mediante una modificación del código de ***snmtrapd***. Esta información podría ser recuperada por red desde cualquier otro equipo en el que se quisiera ofrecer un servicio basado en localización y tratada en local para generar estadísticas de uso de la red asociadas a localización de sus usuarios.

Otro aspecto a resolver es el mapeo MAC-IP. Este par puede ser conocido de antemano como en el caso de la WLAN de la EPSC o el de un museo con WLANs en sus salas que alquile PDAs a sus visitantes para que obtengan, vía web, información actualizada según la sala en que se encuentren, por citar un posible caso de aplicación comercial. Sin embargo, en el escenario descrito en el párrafo anterior, la asignación de direcciones suele ser dinámica por lo que se hace necesario obtener información de los servidores que realizan la asignación de direcciones para poder utilizar la función **localiza** de **XLOC** y el servicio de páginas web inteligentes**.**

Finalmente, comentar que se está trabajando en una versión web de **XLOC** que permita su integración en la web de monitorización de la WLAN de la EPSC para facilitar su utilización desde cualquier equipo con el único requisito de que disponga de un navegador.

Agradecimientos

El trabajo presentado en este artículo se enmarca dentro del proyecto TIC2000-1041-C03-01 financiado por Comisión Interministerial de Ciencia y Tecnología (CICYT)

Referencias

[1] X.Bordoy, R.Vidal. “Despliegue de una WLAN en la EPSC”. Buran (pendiente de publicación.)

[2] J. Case, et. al. “A Simple Network Management Protocol (SNMP)”. IETF RFC 1157. Mayo 1990.

[3] MRTG: The Multi Router Traffic Grapher: http://www.mrtg.org

[4] Alois Ferscha, Wolfgang Beer, Wolfgang Narzt. “Location Awareness in Community Wireless LANs”. GI/ÖGC-Jahrestagung, pp. 190-195, vol. 1. 2001

[5] P. Bahl, N. Padmanabhan. “RADAR: An In-Building RF-based User Location and Tracking System”. INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, pp. 775-784, vol. 2. Marzo 2000.

[6] Bhasker, Ezekiel. Griswold, Bill. Location Detection in a Wireless 802.11b Network Environment. 2001

[7] “A Practical Approach to Identifying and Tracking Unauthorized 802.11 Cards and Access Points”. Interlink Networks White Paper, URL: http://www.interlinknetworks.com. 2002.

[8] Koo, Simon, et. al. “Location Discovery in Enterprise-based Wireless Networks: Implementation and Applications”. Second IEEE Workshop on Applications and Services in Wireless Networks (ASWN 2002), Paris. Julio 2002.

[9] C. Rigney, et. al. “Remote Authentication Dial In User Service (RADIUS). IETF RFC 2865. Junio 2000.

[10] L. Blunk, et. al. “PPP Extensible Authentication Protocol (EAP)”. IETF RFC 2284. Marzo 1998.

[11] L.Blunk, et. al. “Extensible Authentication protocol (EAP)”. Internet Draft. Enero 2003.

[12] Erik Dobbelsteijn. “WLAN authentication and authorisation methods”. 2nd Mobility Meeting, Amsterdam. Octubre 2002.

[13] The NET-SNMP Project Home Page: http://net-snmp.sourceforge.net/

[14] The Apache HTTP Project: http://httpd.apache.org/

[15] GTK+ - The Gimp Tool Kit: http://www.gtk.org

[16] GNOME: http://www.gnome.org

[17] Dave Ragget. “HTML 3.2 Reference Specification”. W3C Recommendation. URL: http://www.w3.org/TR/REC-html32.html. Enero 1997.

[18] Arunesh Mishra, Minho Shin, William Arbaugh. “An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process”. 2002. CS Tech Report Number CS-TR-4395. UMIACS Tech Report Number UMIACS-TR-2002-75.

Abriendo el camino hacia la Cuarta Generación: Una Nueva Arquitectura de Redes de Área Personal Inalámbricas

J.A. Irastorza, J. Choque, R. Agüero, L. Muñoz
Departamento de Ingeniería de Comunicaciones,
ETSII y Telecomunicación
Universidad de Cantabria
39005 Santander, España
E-mail: {angel, jchoque, ramon, luis}@tlmat.unican.es

***Abstract**. Several proposals for wireless personal area networks (WPANs) have been made. However, they fall short in terms of scalability, mobility and reconfigurability. Based on typical user scenarios, this paper proposes a novel WPAN architecture that addresses the current shortcomings and identify key areas of research required to enable its implementation. The main focus of this architecture is that it addresses the scalability by a three-layered network: (1) a small network of very basic terminals, with a star topology, which we call a Virtual Device (VD); (2) a small network comprising Virtual Device(s) and more capable devices named Advanced Terminals, in a mesh topology, which we call personal area network (PAN); (3) a larger network comprising any number of PANs, communicating together through a meshed network, and possibly linked to the external world through gateways; this network is called a Community Area Network (CAN) in this paper.*

1 Introducción

El dominio impuesto por la tecnología Internet en las redes de comunicaciones de datos ha sido el motor principal para el inicio de una nueva era en las redes de telecomunicación. Este hecho ha estimulado la evolución de los protocolos Internet, en especial el dominante Internet Protocol (IP). Su nueva versión, denominada IPv6, reduce muchas de las limitaciones de la versión actual (IPv4), especialmente en lo concerniente al número de direcciones IP disponibles, soporte de calidad de servicio (Quality of Service, QoS) y mecanismos de seguridad. Al mismo tiempo, y sobre todo en la última década, las redes de comunicaciones inalámbricas han experimentado un fuerte crecimiento, enfocado tanto en los servicios de voz como de datos. Producto de ello han ido apareciendo nuevas soluciones, como las denominadas redes de área local inalámbricas (Wireless Local Area Network, WLAN), que han llegado a ser muy populares en comunicaciones interiores y exteriores, debido a su cada vez menor coste y mayor velocidad de transmisión. En un futuro cercano se espera una tendencia similar para la tecnología WPAN (Wireless Personal Area Network). Una WPAN es una infraestructura de comunicación de corto alcance, que soporta comunicaciones en el área de la persona, esto es, en un radio de 5 metros alrededor del usuario. La comunicación se realiza entre una o varias personas y sus dispositivos, aunque también podemos imaginarnos comunicaciones entre dispositivos WPAN pertenecientes a diferentes personas que estén separadas más de 5 metros. En este caso, será necesario que exista una red ad hoc, o alguna clase de infraestructura adicional, como una WLAN, red celular o quizás satélite, que permita comunicar esos dispositivos.

En este contexto se presenta un escenario de una futura generación WPAN (Next Generation WPAN, NG-WPAN), fundamentada en la fuerte sinergia entre la tecnología Internet, basada en IPv6, y las nuevas infraestructuras WPAN, que reemplazarán lo que se conoce como la primera generación de la tecnología WPAN, representada por Bluetooth [1].

Este artículo está organizado de la siguiente manera. La sección 2 contiene una descripción de los futuros escenarios que posiblemente se crearán en base a nuevos desarrollos en la tecnología WPAN. La sección 3 incluye una descripción de la arquitectura propuesta para la nueva familia de WPANs; la sección 4 propone la pila de protocolos adecuada para la NG-WPAN, concebida en el contexto del proyecto de la Unión Europea denominado PACWOMAN [2] (Power Aware Communications for Wireless OptiMised personal Area Networks). En la sección 5 se describe un ejemplo práctico de aplicación de la arquitectura propuesta; y finalmente en la sección 6 se resumen las principales innovaciones planteadas y los retos a que se enfrenta la investigación para desarrollar la arquitectura propuesta.

2 Futuros Escenarios

Aunque las aplicaciones NG-WPAN pueden trabajar sobre un amplio rango de escenarios diferentes, se ha identificado tres ejemplos principales como muestra de las capacidades de NG-WPAN desde el punto de vista de negocio.

- Servicios personales (por ejemplo, en el hogar y entorno hospitalario);
- Servicios de negocios (por ejemplo, gestión de flotas);
- Entretenimiento/Ocio (por ejemplo, juegos, aplicaciones de vídeo de alta velocidad en vehículos).

El escenario de servicios personales está enfocado principalmente a la monitorización remota, tele-presencia y seguridad. Un usuario con sensores médicos (por ejemplo, electrocardiograma, temperatura, presión) puede moverse libremente mientras que está siendo continuamente monitorizado por profesionales de la salud. El escenario de servicios de negocios puede emplear, junto a las clásicas comunicaciones de voz y datos, redes de sensores que comuniquen diferentes parámetros de una flota, por ejemplo en camiones la velocidad, el nivel del aceite, el nivel de combustible, etc. Esta red de sensores también puede ser utilizada en edificios para monitorizar los parámetros ambientales (por ejemplo, temperatura, luz). El escenario de ocio considera la distribución de vídeo de alta velocidad para entretenimiento público en vehículos, en edificios, en el hogar o para uso individual (por ejemplo, empleando lentes con visor de video).

En estos escenarios, se identifican tres categorías principales de aplicaciones en función de la tasa de transferencia de información:

- Servicios de telecontrol de sensores y adquisición de datos de muy baja velocidad, localizados en el entorno de la persona para aplicaciones de tele-monitorización o localizados en vehículos para aplicaciones de gestión de flotas.

- Servicios de negocio o tele-monitorización médica de baja a media velocidad, esos aparecen casi en todas partes, para transferir datos de sensores, voz, video de baja calidad o fotos y servicios relacionados con ordenadores, tales como el servicio de impresión.

- Servicios de vídeo interactivo y multimedia de media a alta velocidad, o transferencia de datos entre ordenadores con requerimientos de tiempo real.

Finalmente, basándose en las aplicaciones NG-WPAN y la visión de los futuros escenarios, se desarrolla una arquitectura general que engloba los diferentes requerimientos y necesidades de las nuevas redes WPAN, abriendo de esta forma un camino hacia las futuras redes de Cuarta Generación.

3 Arquitectura General

En las siguientes secciones se describe la arquitectura propuesta para la NG-WPAN en base a dos observaciones principales que se pueden obtener del planteamiento realizado en la sección anterior. Una primera observación que se puede obtener de los escenarios mostrados es que deberán soportar una amplia variedad de dispositivos de diferentes funcionalidades, capacidades, potencias, velocidades de transmisión y costes. Dispositivos sencillos de uso personal (por ejemplo, sensores) que normalmente serán de muy bajo coste o incluso desechables. Otros dispositivos tendrán mayor capacidad y un mayor coste lo que les permitiría incorporar funcionalidades de bridge, router o incluso gateway. Sin embargo el costo adicional que implica añadir funcionalidades WPAN a los dispositivos de mayor capacidad no debería ser más que una pequeña fracción del coste básico del dispositivo. Por consiguiente se propone para NG-WPAN una estrategia inherentemente de bajo coste, basada en una arquitectura jerárquica y escalable empleando una o más opciones para la interfaz física, adaptada a la clase de servicio.

Una segunda observación derivada de los escenarios analizados es que las comunicaciones ocurren en tres diferentes áreas geográficas. 1) un área interior, cuyo centro esta constituido por la persona, 2) un área local exterior y 3) una segunda área exterior mucho más amplia. Estas tres diferentes áreas se traducen lógicamente en tres posibles redes: PAN, CAN y WAN (Personal, Community and Wide Area Network). En este artículo, principalmente se dará un mayor enfoque a la PAN y la CAN mientras que la WAN sólo se describirá brevemente.

3.1 Primer Nivel: PAN

En el caso más simple, la PAN puede ser una red capaz de operar independientemente. Debe soportar un amplio rango de velocidades de transmisión, posiblemente con un gran número de dispositivos sencillos, soportando tasas de datos de muy baja velocidad, así como un número reducido de dispositivos de mayor capacidad, como PDAs (Personal Digital Assistant), cámaras, etc. Por lo tanto, la PAN se divide en dos redes: una formada por los dispositivos de baja velocidad y otra por los dispositivos de alta velocidad.

PAN de baja velocidad: Un Dispositivo Virtual

Los dispositivos de baja velocidad y reducida capacidad son los denominados terminales basicos (basic Terminals, bTs), los cuales están bajo el control de un dispositivo de mayor capacidad denominado el dispositivo Maestro (M). Los bT se encuentran a poca distancia del Maestro (alrededor

de 2 metros) y sólo mantienen comunicación directa con él, razón por la cual conforman una

topología estrella. Para fines de comunicación e interoperabilidad con otros dispositivos externos, el conjunto de los bTs y el Maestro actúan como un dispositivo único denominándose dispositivo virtual (Virtual Device, VD). El Maestro coordina las comunicaciones entre los bT y otros dispositivos que se encuentran alrededor del dispositivo virtual. Como se muestra en la Fig. 1, el VD actúa como un concentrador de baja a alta velocidad a través del Maestro y puede ser visto como tal desde la perspectiva de la PAN de alta velocidad.

PAN de alta velocidad: Una Red de Área Personal

Aparte de los pequeños sensores, una persona podría también llevar una cámara digital, un PDA, o quizás un ordenador portátil. Para acomodar tasas de datos de mayor velocidad sin desperdiciar ancho de banda, lo cual ocurrirá si los datos que envía la cámara hacia el PDA pasan a través del Maestro, se utiliza una topología que posibilite la comunicación directa entre los dispositivos de alta velocidad. De manera optimista se puede decir que la red estará completamente conectada, aunque esto no siempre se pueda garantizar. Por lo tanto, la figura global de una PAN es una topología mallada donde uno de los nodos puede ser un dispositivo virtual, que agrupa a los dispositivos de baja velocidad, y los otros nodos son terminales avanzados (Advanced Terminals, aTs), como se muestra en la Fig. 2. Notar que la reconfiguración dinámica de la red y de los aspectos de seguridad son menos cruciales en este nivel que en los de CAN o WAN.

3.2 Segundo Nivel: CAN

Para ampliar un grado más la conectividad presentada por el nivel PAN y permitir que dispositivos de diferentes PANs se comuniquen, introducimos un concepto novedoso denominado CAN, como se muestra en la Fig. 3. La formación de una CAN implica la realización de los siguientes procedimientos: descubrimiento de servicios, enrutamiento en un entorno ad hoc, gestión de la seguridad y finalmente intercambio de datos.

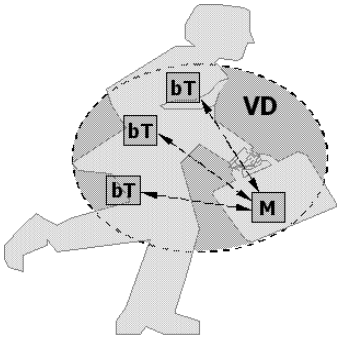


Figura 1: El Dispositivo Virtual, una pequeña red de dispositivos básicos

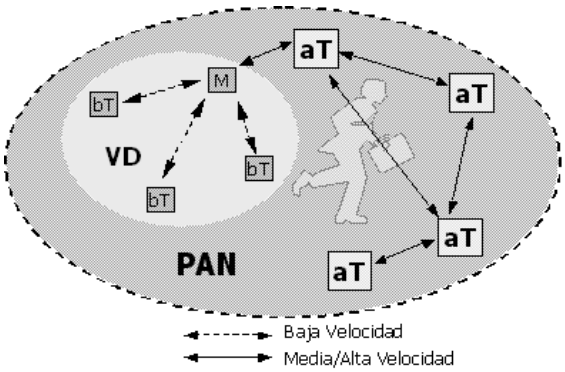


Figura 2: La Red de Área Personal, una red de terminales avanzados

El *procedimiento de descubrimiento de servicios* aporta un mecanismo automático que permite a los dispositivos saber qué servicios están disponibles en un determinado instante y lugar. Por otra parte, la necesidad de formar redes en cualquier momento y cualquier lugar suscita la inclusión de capacidades de redes ad hoc [3] en el nivel CAN, por lo que será imprescindible implementar en cada dispositivo un *procedimiento de mantenimiento de rutas*. La red en éste nivel será una red de tipo malla y, para que sea compatible con otras redes, utilizará la tecnología IP. Como paso previo al *intercambio de datos* será necesario establecer un procedimiento que *gestione la seguridad* de manera que se cumplan determinadas políticas de acceso y compartición de recursos específicas para cada CAN formada. De esta manera, podemos ver una CAN como una asociación entre dos dispositivos pertenecientes a distintas PANs que intercambian datos asociados a un determinado servicio, regida por una determinada política de acceso y por parámetros que regulan el uso del servicio.

3.3 Tercer Nivel: WAN

Para completar lo descrito anteriormente, el sistema tiene que proporcionar al usuario posibilidades de comunicación global, lo cual exige el uso de clásicos sistemas WAN (inalámbricos o no). El acceso a esos tipos de redes se realizan a través del Gateway (G). Las comunicaciones pueden ser entre dos PANs o entre un nodo dentro de la PAN y un servidor externo (por ejemplo, un servidor web), como se muestra en la Fig. 4.

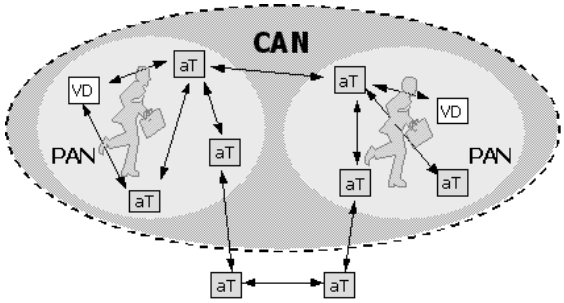


Figura 3: La Red de Área Comunitaria, una red local de PANs

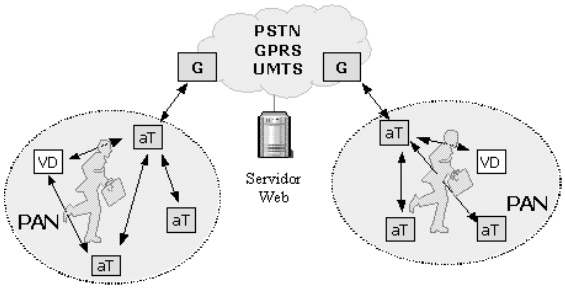


Figura 4: La Red de Área Comunitaria, PANs/CANs comunicándose a través de redes externas

Aunque la comunicación entre una PAN y el Gateway podría ser vista como una particularización de una CAN, sus características son muy diferentes especialmente en lo que concierne a aspectos de QoS extremo a extremo y seguridad.

4 Arquitectura de Protocolos

Esta sección especifica la pila de protocolos que se ajusta a la arquitectura de red de tres niveles descrita en la sección anterior. Para este propósito, la sección ha sido dividida en tres partes. La primera muestra una visión de la pila de protocolos propuesta desde el punto de vista de la escalabilidad. La segunda, especifica las modificaciones que se necesitan hacer en esta pila de protocolos para optimizar su rendimiento en un escenario de red ad hoc inalámbrico. Por último se describe el marco de gestión de red, que completa la pila de protocolos, integrándose y adaptándose a la arquitectura de red propuesta.

4.1 Escalabilidad del Protocolo

Resulta evidente que una adecuada arquitectura de protocolos que acomode todos los servicios soportados por NG-WPAN debe ser escalable. La diversidad de terminales abarca dispositivos básicos, dispositivos Maestro, terminales avanzados y el gateway. A continuación se presenta una descripción de las especificaciones de cada tipo de dispositivo.

Terminal Básico

Los requerimientos de este tipo de dispositivos incluyen la transmisión de pequeñas cantidades de datos hacia un dispositivo Maestro mediante una conexión punto a punto, controlados por eventos disparados automáticamente cada cierto intervalo de tiempo o mediante técnicas basadas en sondeo. Esto, unido a su reducida inteligencia y al hecho de requerir un bajo precio, da lugar a proponer una sencilla pila de protocolos de baja velocidad, denominada pila LDR (Low Data Rate), constituida por las Capas Física (PHY), Enlace de Datos (DLC) y Aplicación. La Capa Física puede estar basada en la tecnología UltraWideBand (UWB), Bluetooth o el estándar IEEE 802.15.4. La Capa de Aplicación esta constituida por un conjunto de programas que implementan la comunicación propietaria entre el bT y el Maestro.

Dispositivo Maestro

Es un elemento de la red que permite dar visibilidad al VD hacia los diferentes niveles de la arquitectura NG-WPAN: PAN, CAN y WAN. Por consiguiente, dentro del dispositivo Maestro se definen dos interfaces: primero, una interfaz de baja velocidad LDR asociada a una pila LDR, similar a la descrita en la sección anterior, empleada para la comunicación con los bTs que conforman el correspondiente VD. Segundo, una interfaz de media/alta velocidad M/HDR (Medium and High Data Rate) asociada con una pila M/HDR, la cual esta constituida todas las capas desde el nivel físico hasta el nivel de aplicación basada en una arquitectura IP, usada para la comunicación con los dispositivos que conforman la red PAN, CAN o WAN, como se muestra en la Fig. 5.

La pila LDR requiere que se incluyan funcionalidades asociadas con las Capas OSI Física y Control de Enlace de Datos. De esta manera, la pila LDR debe implementar una aplicación que asegure un nivel físico fiable, un enlace lógico de datos libre de errores y un simple mecanismo de direccionamiento. La fiabilidad del nivel físico estará basado principalmente en las características que dispone el interfaz físico a utilizar.

Por otra parte, un dispositivo Maestro puede comunicarse con terminales M/HDR usando la pila M/HDR. Por lo tanto, se requiere que la pila M/HDR soporte capacidades de reenvío de paquetes en un escenario ad hoc y permita, a su vez, proteger a los protocolos de nivel superior (por ejemplo TCP/IP) de las imperfecciones del enlace inalámbrico, esto último se explicará con más detalle en la sección 4.2. La Capa Física M/HDR propuesta estaría basada en IEEE 802.15.1 para tasa de datos de media velocidad y 802.15.3 para tasas de datos de alta velocidad.

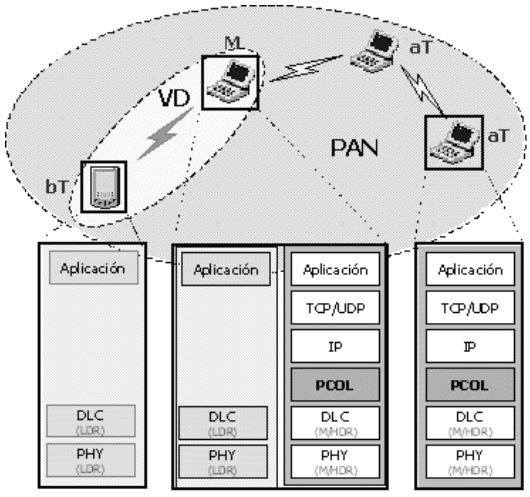


Figura 5: Arquitectura de Protocolos de los diferentes tipos de dispositivos

Terminal Avanzado

Este tipo de terminales son nodos en la red inalámbrica ad hoc que soportan una interfaz M/HDR y por consiguiente incorporan una única pila de protocolos. Este terminal avanzado cumple tres funciones principales:

- Soporta aplicaciones con altos requerimientos de ancho de banda.
- Actúa como un nodo intermedio reenviando paquetes dirigidos a otros nodos.
- Compensa las imperfecciones del enlace inalámbrico mediante el uso de módulos de optimización del enlace descritos en la siguiente sección.

4.2 Capa de Optimización PAN/CAN

Es muy conocido que el rendimiento de la pila de protocolos TCP/IP se reduce drásticamente debido a la alta tasa de error de bit que existe en los enlaces inalámbricos [4]. Para hacer frente a este comportamiento del canal inalámbrico se propone la incorporación de una capa que implemente una serie de módulos diseñados para mejorar el rendimiento en cada enlace. Esta propuesta sigue la filosofía planteada por los Performance Enhanced Proxy (PEP) utilizada para mejorar el rendimiento de protocolos Internet en rutas de la red que sufran un bajo rendimiento debido a errores, reducido ancho de banda o mala calidad del enlace [5].

Por lo tanto, en lugar de rediseñar los protocolos Internet que ya existen adaptándolos al entorno inalámbrico, nuestra arquitectura propone mejorar el rendimiento TCP/IP mediante una nueva capa, denominada Capa de Optimización PAN/CAN (PAN/CAN Optimization Layer, PCOL), que reside entre la capa IP y la infraestructura de red subyacente. La PCOL también es responsable de gestionar los mecanismos de enrutamiento ad hoc y de controlar los aspectos de potencia durante las comunicaciones, haciendo uso para tales propósitos del marco de gestión de red.

4.3 Gestión de Red

La arquitectura de red propuesta se asienta sobre una estructura de red ad hoc inalámbrica, cuyas características intrínsecas de movilidad y dependencia del medio inalámbrico, así como la anteriormente comentada de la escalabilidad (representada por una gran diversidad de dispositivos LDR, M/HDR) conducen a la necesidad de dotar a esta arquitectura de capacidades de gestión de red. El marco de gestión deberá adaptarse al entorno de red sobre el que va a desarrollar las distintas tareas de gestión (fallos, rendimiento, configuración, seguridad, contabilidad) maximizando así su eficiencia y productividad. No todas las tareas de gestión van a tener la misma relevancia, siendo las más importantes las de configuración, fallos y rendimiento.

A continuación se comentan brevemente aquellas propiedades básicas en redes ad hoc que deben considerarse en un escenario de gestión:

- Diversidad en cuanto a la complejidad de los diferentes nodos que forman parte de una red ad hoc, desde sencillos sensores hasta ordenadores portátiles de última generación. Esta diversidad se verá reflejada en el marco de gestión, de manera que la contribución de los dispositivos a las tareas correspondientes será la adecuada a sus capacidades.
- Movilidad de los nodos en la red, lo que supone cambios de topología, que deberán ser controlados o monitorizados por el entorno de gestión de red. Dependiendo del grado de movilidad de los nodos en la red, esta monitorización puede suponer un gran aumento de los mensajes de gestión, hecho que habrá que controlar para no sobrecargar o incluso colapsar la red.
- Dependencia del estado de la carga de las baterías de los nodos, que limita el exceso de envió de tráfico de gestión, de forma que la energía consumida por los nodos sea minimizada. Esto supone un tratamiento inteligente de la información de gestión al dotar a los agentes de capacidades para generar información elaborada y enviarla a la estación gestora en momentos donde la carga de la red sea baja.
- La variabilidad de la calidad de la señal es un aspecto habitual en entornos ad hoc, que deberá controlarse para que desvanecimientos puntuales de la señal no sean entendidos como variaciones de la topología de red, como los que anteriormente se han comentado debidos a la movilidad de los nodos.

El entorno de gestión propuesto se adapta a las características señaladas anteriormente al igual que se enmarca dentro de la arquitectura mostrada en la sección 3. En esta arquitectura se describen dispositivos de baja, media y alta velocidad de datos que interaccionan entre ellos en entornos PAN, CAN o WAN. Como resultado de todas estas premisas se propone un modelo jerárquico para la gestión. Dicho modelo asume la existencia de gestores intermedios que permitan una cierta distribución en las tareas de gestión. Cada gestor intermedio recogerá información del nivel inmediatamente inferior, lo procesará y lo enviará al gestor de nivel superior si fuese necesario. Este modelo no recoge unas comunicaciones directas

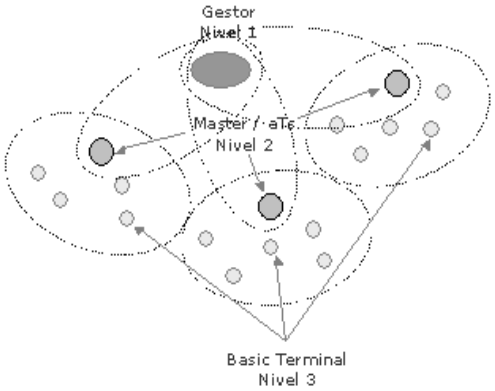


Figura 6: Niveles en la arquitectura jerárquica de gestión

entre gestores intermedios. Una de las principales ventajas que presenta este modelo es que reduce el número de mensajes de gestión intercambiados en la red. La implementación del mismo sobre la arquitectura propuesta implica la definición de tres niveles en la jerarquía de gestión, tal y como se muestra en la Fig. 6:

- Nivel 3, es el nivel más bajo en el modelo jerárquico de gestión, y se implementa sobre dispositivos LDR, que en la arquitectura propuesta se denominan bTs. Estos dispositivos, normalmente sensores, estarán agrupados por cercanía y pertenencia a un determinado VD formando lo que se ha llamado una PAN. Determinados bT podrían incluso no implementar la pila TCP/IP en aras de ahorrar su coste.

- Nivel 2, es el nivel intermedio en este modelo jerárquico de gestión y se implementa sobre dispositivos M/HDR que en la arquitectura propuesta podrán ser tanto los denominados Maestros o los dispositivos aTs. Mientras los primeros pertenecen a un determinado VD los aTs únicamente son nodos en la red ad hoc del tipo M/HDR.

- Nivel 1, es el nivel más alto del modelo jerárquico y será implementado sobre un nodo M/HDR que estará dedicado a realizar las tareas de gestor de la red. Siguiendo el concepto de CAN descrito en secciones anteriores, podemos decir que la aplicación de gestión implementa una CAN (CAN de gestión) entre la estación gestora y los dispositivos gestionados.

Una vez descrita la arquitectura jerárquica de gestión y pensando en su implementación hay que elegir la tecnología que mejor se adapta a los requerimientos hasta ahora presentados.

SNMP (Simple Network Management Protocol) es un protocolo que se adapta adecuadamente a los requerimientos de partida. SNMP ha evolucionado desde su aparición como estándar de gestión en redes TCP/IP con la versión 1 y un modelo totalmente centralizado, hasta la actualidad, con la versión 3 (estándar desde diciembre 2002) [6] y un modelo de gestión jerárquico que permite la comunicación entre gestores e introduce alto grado de seguridad. Esta tecnología es apropiada para la gestión en los 3 niveles, aunque debería ser complementada con una gestión propietaria en el nivel 3 en el caso que los bTs a gestionar no implementasen la pila TCP/IP. Además, dada la popularidad, de dicha tecnología existe un software de desarrollo de libre distribución sobre Linux, llamado NET-SNMP [7] que facilita la implementación final de la arquitectura de gestión propuesta.

A continuación se detallan los componentes del entorno de gestión y su ubicación en la arquitectura de red propuesta, ver Fig. 7 :

Nivel 1: Sobre un equipo M/HDR se implementa la Estación Gestora, para lo cual se desarrolla el componente *Gestor de Nivel 1* que gestionará, directamente, los dispositivos MTs y aTs, mientras que los dispositivos bTs serán gestionados vía un proxy implementado en el nivel 2. Adicionalmente se implementa otro componente, *Web Server*, que permite a una estación de monitorización situada detrás del gateway, es decir, fuera de la red ad hoc, acceder vía el protocolo HTTPS (Secure Hypertext Transfer Protocol) a la información de gestión.

Nivel 2: Se implementa sobre dos dispositivos diferentes: MTs y aTs. Sobre el MT se desarrollan tres componentes distintos (1) el *Gestor de Nivel 2,* que informa al Gestor de Nivel 1 sobre los bT que controla; (2) un *Agente de Nivel 2* que informa al Gestor de Nivel 1 sobre información de gestión del propio MT; (3) y, por último, un *Proxy* que traduce las operaciones SNMP dirigidas a los bT en operaciones de gestión de un protocolo propietario. El dispositivo aT implementa solamente un *Agente de Nivel 2* que envía la información de gestión del propio dispositivo aT a la estación de gestión

Nivel 3: Cada dispositivo bT implementa un *Agente de Nivel 3* que interacciona con los gestores de nivel 1 y 2 vía el Proxy.

Por último cabe recalcar la existencia en cada uno de los dispositivos gestionados de unas bases de información de gestión "Management Information Bases" (MIBs) que recogen la información a gestionar en cada dispositivo. Así en el nivel 1, la

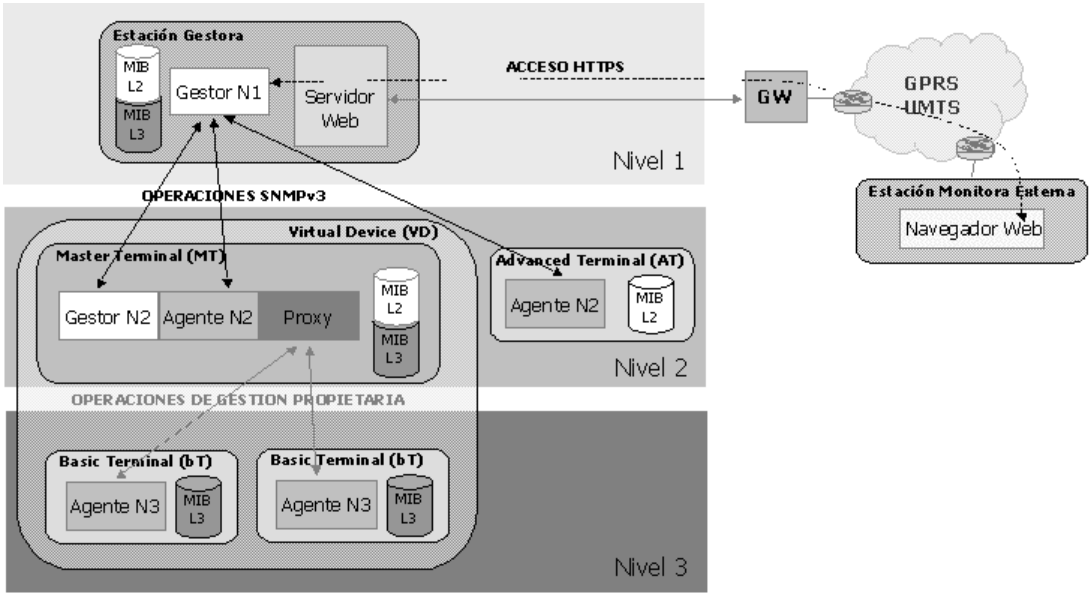


Figura 7: Componentes del entorno de gestión y su ubicación en la arquitectura de red propuesta

estación de gestión manejará toda la información gestionable de los dispositivos situados en los niveles 2 y 3. En el nivel 2; la MIB del MT recogerá tanto la información del propio dispositivo MT como información procesada y evaluada de los bTs a su cargo; la MIB del aT recogerá información de gestión del propio equipo aT y de su implicación en la estructura de la red ad hoc a la que pertenece. En el nivel 3, la MIB del bT recogerá información básica del dispositivo LDR que controla, por ejemplo, si esta operativo o no, el valor o valores que, en caso de ser un sensor, controle, el estado de sus baterías, y alguna característica reseñable del nodo.

5 Escenario de aplicación

En esta sección se presenta un escenario de aplicación en el que se muestran los conceptos principales de la arquitectura general propuesta.

El escenario que se presenta se inscribe en el entorno hospitalario en el cual un doctor necesita consultar alguna información de un paciente haciendo uso de un dispositivo inalámbrico del tipo PDA. Para ello partimos del supuesto que el paciente tiene controlados una serie de parámetros médicos mediante una serie de sensores conectados a un dispositivo Maestro que los controla reenvía la información requerida por el doctor.

Este escenario, como muestra la Fig. 8, se adapta perfectamente a la arquitectura general propuesta en la sección 3. En él, podemos identificar dos PAN, la primera centrada el entorno personal del doctor y básicamente formada por el dispositivo PDA, y la segunda centrada en el entorno personal del paciente y constituida por un VD (sensores conectados con el Maestro). Dado que la PAN del doctor (PAN 1) necesita obtener información de la PAN del paciente (PAN 2) se formará una CAN entre las dos PAN. Esta comunidad formada entre las dos PAN fijará una serie de reglas en cuanto a seguridad, permisos y tipo de información a intercambiar. Puede darse el caso que la conexión entre las dos PAN necesite hacer uso de nodos intermedios (aTs) por no haber una conexión física directa entre ellas, en este caso los nodos aTs intermedios no pertenecerán a la CAN sino que únicamente serán nodos de enlace en la red ad hoc.

6 Conclusiones

Este artículo presenta una nueva arquitectura de redes WPAN, para ello se parte de unos futuros escenarios de aplicación y se demuestra como la arquitectura propuesta se adapta a dichos escenarios aportando una solución escalable y gestionable que optimiza los recursos del medio inalámbrico.

La arquitectura WPAN propuesta representa un concepto centrado en la persona, que le permite comunicarse con sus dispositivos próximos personales (PDA, web pads, agendas personales, cámaras, ordenadores personales, sensores corporales, etc...) y prolongar las comunicaciones inalámbricas mas allá del entorno local.

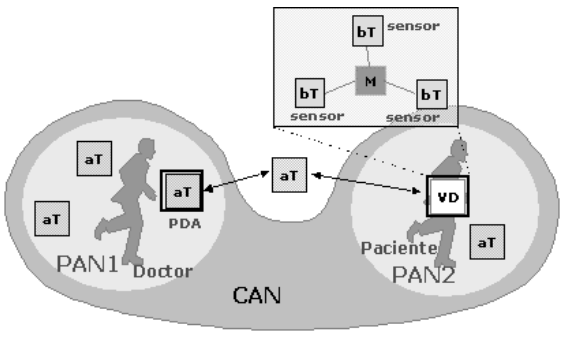


Figura 8: Escenario de aplicación en el entorno hospitalario

En aras de alcanzar las metas propuestas para esta nueva generación WPAN se presentan a continuación una serie de retos:

- Diseño de sistemas de bajo coste y bajo consumo, así como integración de la parte radio.
- Definición de capas físicas adecuadas, así como técnicas de acceso mejoradas.
- Avances en redes Ad hoc.
- Arquitecturas Middleware.
- Seguridad.
- Aspectos Humanos.

Referencias

[1] J. Bray, and Charles F. Sturman, "Bluetooth 1.1, Connect Without Cables", 2nd edition, Prentice Hall PTR, Upper Saddle River, New Jersey, 2002.

[2] Power Aware Communications for Wireless OptiMised personal Area Network (PACWOMAN), contract No IST-2001-34157 http://www.imec.be/pacwoman/.

[3] Advances in Mobile Ad Hoc Networking (Special Issue), IEEE Personal Communications, Vol. 8, Nº.1, February 2001.

[4] George Xylomenos, George C. Polyzos, Petri Mähönen and Mika Saaranen, TCP performance issues over wireless links, IEEE Communications Magazine, vol. 39, no. 4, April 2001, pp.52-58.

[5] L. Muñoz, M. García, J. Choque, R. Agüero, P. Mähönen, "Optimizing Internet Flows over IEEE 802.11b Wireless Local Area Networks: A Performance-Enhancing Proxy Based on Forward Error Correction", IEEE Commun. Mag., vol. 25, Nº. 12, Dec. 2001, pp. 60-67.

[6] Grupo de trabajo SNMPv3 perteneciente al área de Operación y gestión del IETF http://www.ietf.org/html.charters/snmpv3-charter.html

[7] Página principal del proyecto Open Source Net-SNMP http://www.net-snmp.org

Abbreviated Dynamic Source Routing: Protocolo DSR abreviado para máquinas con pocos recursos

Miguel Angel Ortuño Pérez, Vicente Matellán Olivera,
Luis Rodero Merino, José Centeno González
{mortuno,vmo,lrodero,jcenteno}@gsyc.escet.urjc.es
Grupo de Sistemas y Comunicaciones
Departamento de Informática, Estadística y Telemática
Universidad Rey Juan Carlos

Abstract *DSR is one of the best known protocols for datagram routing in ad-hoc networks. It uses source routing, so each datagram must carry the addresses of all the machines in its path. Under some circumstances, specially with the use of IPv6, headers size can become very big. In this paper we propose ADSR protocol, a modification of DSR that drastically reduces its headers size by using abbreviated addresses for the routing. This mechanism can lead to two different nodes having the same address, fact that we call a* **collision** *and analyze in this work. We will also show some results about this protocol performance obtained by simulations implemented using the ns-2 network simulator.*

1. Redes Ad-Hoc

Las redes *ad-hoc* se definen como redes de comunicaciones que se forman cuando se necesitan, compuestas por las estaciones que están en determinado momento en determinado lugar y que no precisan de ninguna infraestructura externa. Los enlaces suelen ser inalámbricos: Esta tecnología junto con la alimentación mediante baterías, además de los algoritmos adecuados permite prescindir de cableado, puntos de acceso, *routers* pre-existentes o alimentación externa [7]. Estas redes están formadas por nodos similares entre sí que cooperan, normalmente no jerarquizados donde todos son al tiempo encaminadores y estaciones finales. Al menos idealmente, tampoco deben necesitar administración por parte de ningún usuario, ni usuario normal ni súper-usuario.

Si en el nivel de enlace todos los nodos fueran visibles entre sí, el problema del encaminamiento estaría resuelto. Pero esto es muy infrecuente y probablemente conlleve un derroche de energía (emisión a gran potencia) y reducción de ancho de banda (aumento de estaciones compitiendo por el medio). En general, cada nodo será capaz de acceder por sus propios medios sólo a los nodos más próximos, confiando en la colaboración de la red para llegar a todos los demás. Actuando de buena fe todos se comportarán como encaminadores además de como productores y consumidores de paquetes.

1.1. Protocolos de Encaminamiento para redes Ad-Hoc

Desde mediados de los años 90 se han desarrollado una serie de protocolos para redes Ad-Hoc.

Actualmente destacan de forma clara dos: DSR Dynamic Source Routing Protocol [4] y AODV (Ad-hoc on-demand distance vector routing) [6]. Ambos trabajan bajo demanda. DSR hace encaminamiento en origen (*source routing*), mientras que AODV emplea tablas de enrutamiento convencionales que actualiza frecuentemente y con números de secuencia para determinar su frescura.

Para nuestro trabajo partimos de DSR. Estudios recientes [1] [2] apuntan a que DSR sobrecarga menos la red con información de enrutado y ofrece mejores resultados en situaciones de moderada y media *tensión* en la red (carga, número de nodos y movilidad). Si bien a partir de ciertos niveles, AODV puede resultar más adecuado.

1.2. Descripción del protocolo DSR

DSR es un protocolo de encaminamiento en origen compuesto de dos mecanismos que trabajan coordinadamente: *Descubrimiento de ruta* y *mantenimiento de ruta*. Trabaja *bajo demanda*, no hay ninguna operación que se realice periódicamente: Cuando el nodo emisor se mueva o cuando cambie la topología de la red, el algoritmo percibe los cambios y se adapta a ello, pero

sólo para las rutas que estén en uso.

1.2.1. Descubrimiento de Ruta (*Route Discovery*)

Supongamos una red como la de la figura 2. El nodo A dispone de alguna tecnología inalámbrica que le permite alcanzar B pero no más allá. B llega hasta A, C y E, etc. Si A quiere enviar un paquete al nodo D y no sabe por dónde encaminarlo debe hacer un descubrimiento de ruta, que a grandes rasgos podemos describir así:

- A radia una *petición de ruta hasta D* a todos sus vecinos. Si quien lo recibe no es D, reenvía la petición. Esta es la conocida técnica de encaminamiento por inundación [8]. Para controlar la inundación, cada petición tiene un identificador único, de forma que cada nodo reenvía esta petición sólo una vez.

- En cada reenvio del datagrama con la petición, el nodo añade su propia dirección, de tal forma que queda registrada la ruta por la que ha ido pasando.

- Si la petición llega a D, éste extrae la ruta del datagrama (ABCD) y se la envía a A en un *Route Reply*. Usando encaminamiento en origen, este *Route Reply* volverá por el mismo camino *deshaciendo lo andado*.

- Una vez que A conoce la ruta para D, la incluirá en todos los paquetes que le envíe, empleando de nuevo encaminamiento en origen.

1.2.2. Mantenimiento de Ruta (*Route Maintenance*)

Cada nodo se hace responsable de que el paquete que recibe llegue al siguiente nodo en la ruta. En la red de la figura 2, cuando B recibe un paquete de A, se asegura de que llegue a C. (Protocolos de enlace como IEEE-802.11 ya ofrecen este servicio, con lo que no supone esfuerzo adicional).

Así, si mientras A envía sus paquetes a D, el nodo C se mueve fuera del alcance de B, este lo descubrirá y se lo comunicará a A con un un mensaje *Route Error* indicando que esta ruta no es válida.

- El protocolo es *best effort*, en caso de que una ruta se *caiga* no se reenvía el paquete, eso lo harán, si procede, niveles superiores.

- La siguiente vez que A deba enviar un datagrama a D, si conoce un camino alternativo, lo usará. En otro caso lanzará una nueva *Route Request*.

2. Máquinas con recursos limitados

A pesar de las continuas mejoras y abaratamientos del hardware, siempre habrá dispositivos capaces de comunicación sin cables pero con pocos recursos, probablemente por restricciones en su precio o tal vez por limitaciones de ocupación del espacio radioeléctrico o de consumo de la energía de sus baterías: PDAs, electrodomésticos, juguetes, equipos industriales, etc.

Tomamos como ejemplo los equipos con los que los estudiantes de la asignatura de Robótica en nuestra universidad hacen prácticas: Lego Mindstorm RCX. Cuenta con un procesador Hitachi H8/300, ROM de 16K y RAM de 32 k. Podemos comparar sus prestaciones con las de un micro-ordenador de los años 80, el Sinclair ZX-Spectrum. Puede además comunicarse mediante infrarojos con otros RCX o con un PC. Su protocolo de comunicaciones, LegOS, usa una trama de 256 bytes. Dispositivos similares tienen tramas de este orden o incluso menores.

Si intentamos llevar el protocolo DSR sobre IPv4 a una máquina de estas características o similares, una buena parte del datagrama estará ocupado por las cabeceras. La longitud de las cabeceras es variable: tomando como referencia la implementación de DSR disponible para el simulador de redes ns-2 [3] serían necesarios 88 bytes: Un tercio del total disponible.

Si DSR se usa bajo IPv6 se requerirían 288 bytes, lo que resultaría inviable con la arquitectura que proponemos como ejemplo. En estas misma condiciones, el protocolo que presentamos precisaría de 46 y 65 bytes respectivamente (figura 1).

3. Solución Propuesta

Proponemos el protocolo denominado *Abbreviated Dynamic Source Routing*, o ADSR. Es una modificación de DSR basada en construir las rutas usando direcciones abreviadas: Cada ruta no contiene la dirección de los nodos que la componen, sino un nuevo identificador que se construye a partir de la dirección original y que tendrá tamaño menor o igual. Esto supone romper la idea de una dirección que identifique de forma única a una estación: Podrá haber más de una máquina con la misma dirección abreviada, hecho al que

denominamos **colisión**. Modificaremos el protocolo DSR para que tolere estas colisiones. Esta modificación a DSR puede verse como la aplicación de técnicas de hashing sobre las direcciones de los nodos, o también, en cierta forma, un algoritmo de compresión con pérdida sobre las rutas.

Si R es una ruta convencional como las que usa DSR, podremos abreviarla con cualquier función $Abb()$ que satisfaga lo siguiente:

1. Dadas una ruta convencional cualquiera:

   $R1 = (D_1, D_2, ...D_n)$

   y su ruta abreviada

   $Abb(R1) = (d1, d2, ..., d_n)$

   Debe cumplirse:

   $\forall i, 1 \leq i \leq n$

   $size(d_i) \leq size(D_i)$

   donde $size(d)$ es el tamaño en bytes de una dirección.

2. Dadas dos rutas convencionales cualquiera:

   $R1 = (D_1, D_2, ...D_n)$

   $R2 = (E_1, E_2, ...E_m)$

   Sean sus rutas abreviadas

   $Abb(R1) = (d1, d2, ...d_n)$

   $Abb(R2) = (e1, e2, ...e_m)$

   Debe cumplirse:

   $d_i = e_j \wedge D_i \neq E_j \Rightarrow$

   $i < n \wedge j < m$

   Esto es, si dos direcciones colisionan, no son las últimas de una ruta. O en otras plabras, la última dirección de cada ruta se construye de forma que no se produzcan colisiones (O que la probabilidad de una colisión sea despreciable).

El propósito de esta segunda condición no es tanto evitar que un nodo reciba paquetes que no le corresponden (puesto que el nivel de red superior lo percibiría y podría eliminarlos) como impedir que un nodo crea disponer de una ruta para determinada máquina, cuando en realidad lleva a otra cuya dirección colisiona con la deseada.

Tras estudiar otras alternativas [1], se propone una función Abb(R) muy sencilla:

- Para $1 \leq i \leq n-1$, d_i será el último byte de D_i
- Para $i = n$, $d_i = D_i$

A partir de estos principios en ADSR se modifica el protocolo DSR lo mínimo necesario para permitir su funcionamiento con este tipo de rutas. Enumeraremos estas modificaciones posteriormente.

4. Clasificación de las colisiones

El ahorro de espacio en las cabeceras que hemos descrito en el apartado 2 supone un único problema: Lo que hemos denominado *colisiones*, dos máquinas distintas cuya dirección abreviada coincide. A continuacion analizamos y clasificamos las colisiones, describiendo el comportamiento del protocolo ADSR atendiendo al tipo de colisión.

Para mayor claridad en las figuras, en este apartado no haremos referencia a una ruta genérica $r_1 = (d_1, d_2, ..., d_{n-1}, d_n)$ sino que tomaremos una ruta concreta (la misma en todos los casos): Un nodo a buscando una ruta hasta d que será $r = a, b, c, d$.

La notación a , a' indicará dos direcciones que colisionan ($d_i = e_j$ con $D_i \neq E_j$).

- Colisión indiferente

  Obviamente, si la colisión se produce en un nodo que no es visible por *a,b,c* ni *d*, esta no afecta de ninguna manera.

- Colisión en destinatario.

  Situaciones como la representadas en la figura 3 no se darán nunca, por la segunda condición de la función $Abb()$.

- Colisión distante

  La figura 4 representa una colisión que no plantea problemas. La petición de ruta que inunda la red, tras pasar *c'* será descartada, al ser imposible alcanzar *d* desde *c'* (a menos que la petición de ruta volviese por *a* , pero esto lo impide el control de la inundación).

  Cuando el paquete sea enviado por *b* a *c*, *c'* no lo recibe y por tanto no interfiere. Llamamos a este caso *colisión distante* porque los nodos cuya dirección colisiona están distanciados en la red.

[1] Buscar el que la probabilidad de colisión sea nula es tanto como decir que buscamos hashing perfecto. El hashing perfecto tiene un coste computacional elevadísimo [5] . Además exige conocer las claves sobre las que se aplica en el momento de definir la función hash, lo que es inviable: implicaría conocer en todo momento las direcciones de todas las estaciones en la red. Y aún consiguiéndolo, con direcciones abreviadas de 1 byte estaríamos limitando el tamaño de la red a 255 nodos.

- Colisión adyacente

  El único caso de colisión conflictivo está representado en la figura 5. La petición de ruta que inunda la red, cuando pase por *c'* no generará respuesta alguna, cuando lo haga por *c* sí, con lo que *d* devolverá una respuesta con la ruta *a,b,c,d* como en todos los casos anteriores.

  Consideremos ahora el paquete con la ruta *a,b,c,d* tras atravesar *b*: tanto *c* como *c'* lo interpretarán como dirigido a ellos, esto genera dos copias del paquete: Una *legítima* que llegará correctamente a *d*. La que atraviesa *c'* no podrá alcanzar su destino y acabará generando un mensaje *Route Error*, diremos entonces que *c'* **hace sombra** a *d*. Este es el caso peor, el mensaje de error provocará que se deje de usar una ruta correcta, si bien la ruta que atraviesa la copia legítima del paquete seguirá en funcionamiento el tiempo que transcurra hasta que se genere el error, llegue a *a* y sea procesado. Algunos paquetes tendrán oportunidad de llegar a su destino. Entonces *a* hará una nueva petición, si los nodos no cambian su posición se generarán las mismas respuestas y todo el proceso se repetirá de nuevo.

5. Características de ADSR

ADSR es esencialmente igual a DSR, excepto en todo aquello que resulta incompatible con el uso de direcciones abreviadas.

A continuación se enumeran los aspectos que difieren en ambos protocolos.

Construcción de Rutas

- DSR: En la fase de *descubrimiento básico de ruta* el paquete de petición de ruta se distribuye por inundación y va almacenando la dirección de cada nodo por el que pasa.
- ADSR: El paquete de petición de ruta almacena la dirección abreviada de cada máquina que recorre, teniendo en cuenta que el nodo final de una ruta es un caso especial que exige la ausencia de colisiones.

Múltiples destinatarios en el nivel de enlace

- DSR: Para un paquete con la ruta $R1 = (D_1, D_2, ..., D_i, D_{i+1}, ...D_n)$ que llega a D_i, alcanzar D_{i+1} es inmediato: es un envio *unicast* a una dirección conocida. Bajo DSR habrá un nivel de enlace que probablemente emplee un esquema de direccionamiento distinto, pero entre la dirección de red y la de enlace habrá una correspondendia uno a uno que podrá resolverse con técnicas como ARP o similares.
- ADSR: Dada una ruta $r_1 = (d1, d2, ..., d_i, d_{i+1}, ..., d_n)$ el datagrama debe transmitirse desde d_i hasta d_{i+1}, pero d_{i+1} no identifica de forma única un nodo, por lo que este envío debe llegar a todas las máquinas cuya dirección abreviada coincida con d_{i+1}, con tal de que sean visibles desde d_i.

  Así, cada envio desde el punto de vista del nivel de enlace se convierte potencialmente en una transmisión *multicast*. Esto en general no estará previsto, con lo que inevitablemente habrá que convertir este multicast en una de estas dos opciones

  - Varios *unicast*, lo que exige conocer las direcciones completas de todas las estaciones que deban recibir el envio.
  - Un *broadcast*. Cada receptor, una vez que haya recibido el paquete lo descarta si su dirección abreviada no coincide con la de los destinatarios. Cabe destacar que una consecuencia importante del uso de broadcast es la imposibilidad de usar asentimientos:

Rutas Parciales

- DSR: una ruta $R_1 = (D_1, D_2, ...D_n)$ indica cómo alcanzar D_n, pero también puede usarse para encaminar paquetes hasta $D_2, ...D_{n-1}$.
- ADSR: Una ruta $r_1 =$

  $(d_1, d_2, ...d_{n-1}, d_n)$ es válida sólo para alcanzar d_n. Si se requiere enviar un datagrama a D_i (con $i < n$) es necesario solicitar una nueva ruta, puesto que d_i podría corresponderse con D_i o con cualquier otro nodo cuya dirección abreviada coincida con esta.

Inversión de Rutas

- DSR: Sea una petición de ruta que llega a su destino $R_1 = (D_1, D_2, ...D_n)$ Si el nivel de enlace es bidireccional, el nodo d_n puede construir una una ruta para d_1 directamente a partir de esta: Basta invertirla $(D_n, ...D_2, D_1)$ para almacenarla en caché.

- ADSR. Si la ruta $r_1 =$

  $(d1, d2, ...d_{n-1}, d_n)$ se invierte resulta $(d_n, ..., d_2, d_1)$ que no es una ruta válida para d_1 al no satisfacer la segunda condición de Abb(), con lo que será necesaria una nueva petición de ruta.

  Aunque no resultará extraño que ADSR está bajo un protocolo de red, tal vez IP. Si se tiene acceso a la cabecera del nivel de red, es posible conocer la dirección D_1 y generar de nuevo una dirección abreviada cuya probabilidad de colisión con la dirección de otro nodo sea despreciable. De esta forma se ahorra una petición de ruta.

Control de Inundación

- DSR: La petición de ruta se extiende por inundación. Cada petición tiene un identificador. Para evitar ciclos, cuando un nodo recibe una petición de ruta, si la ha procesado previamente la descarta, en otro caso la reenvía. Para saber si debe descartarse se hacen dos comprobaciones

  - Se mantiene una caché de identificadores de peticiones tratadas recientemente.
  - Además, el nodo comprueba si su dirección está incluida en la ruta seguida por el paquete.

- ADSR: Sólo podrá aplicarse la primera de las comprobaciones descritas en el párrafo anterior: La segunda no es válida, un nodo verá su dirección abreviada en una ruta si el paquete ha visitado otro nodo cuya dirección abreviada colisione con la suya.

  Como consecuencia, en casos extremos podrían producirse ciclos (aunque siempre finitos).

Simplificación de rutas

- DSR: Cualquier ruta

  $(D_1, D_2, ..., D_i, ..., D_j, D_{j+1}..., D_n)$ con $D_i = D_j$ indica un bucle, y por tanto puede simplificarse resultando $(D_1, D_2, ..., D_i, D_{j+1}, ..., D_n)$

- ADSR:

  Una ruta

  $r1 = (d_1, d_2, ..., d_i, ..., d_j, d_{j+1}..., d_n)$ con $d_i = d_j$ no puede simplificarse, pues direcciones abrevidas iguales pueden referirse a distintos nodos.

6. Experimentación

Para analizar el rendimiento de ADSR lo implementamos sobre el simulador de redes ns-2, una herramienta muy extendida, libre y que implementa una gran cantidad de protocolos, incluyendo DSR.

Ofrecemos a continuación unos resultados preliminares. Debemos indicar que partimos de la implementación de DSR existente en ns-2 a la que hacemos las modificaciones descritas en el apartado 5. Esta implementación tiene una serie de optimizaciones que sin duda habría que suprimir al llevar el protocolo a una máquina real de las características de las citada en el apartado 2, pero eso lo obviaremos ahora: queremos comparar una implementación de DSR con una de ADSR.

La configuración de la red, la carga de trabajo y los escenarios sobre los que se podría usar el protocolo ADSR pueden ser extremadamente diversos, es difícil hablar de una configuración *típica* o unos parámetros extraídos de la *realidad*. Además, los resultados dependen mucho de las condiciones iniciales, y son muy variables en función de estos. Así que hemos tomado como punto de partida los parámetros de entrada usados por Broch *et al* [1] en su comparativa del redimiento de varios protocolos ad-hoc:

Sobre un escenario de 1500x300 metros, durante un tiempo simulado de 900 segundos se situarán arbitrariamente 50 nodos, de los cuales 30 a la vez establecerán conexiones a una velocidad constante de 100 bytes/s. Cada nodo hará una pausa antes de moverse en dirección aleatoria con velocidad aleatoria, de media 10 m/s. Esta pausa variará entre 0 (movimiento continuo) y el tiempo total de la simulación (nodos estáticos). Cada simulación se repite 10 veces con la misma configuración y se toma la media aritmética del resultado.

Los resultados se representan en la figura 6 donde tomamos como métrica el tanto por uno de paquetes entregados a su destinatario. Como era de esperar, los mejores resultados corresponden a DSR (el ahorro de espacio visto en la figura 1 tiene un precio). En esta misma figura 6 se observan los resultados de ADSR cuando forzamos el porcentaje de colisiones a un 20 %, 50 % y 100 % (valores muy altos, extremadamente poco probables en situaciones reales).

Se observa que a pesar de someterse al protocolo a situaciones de elevada *tensión* en cuanto a número de colisiones, la pérdida de rendimiento no es excesivamente acusada, obteniendo a cambio un gran ahorro de espacio en el datagrama.

7. Conclusiones

Se han mostrado unas condiciones donde es difícil o imposible aplicar el protocolo DSR. Como respuesta se propone el protocolo ADSR, que basado en el anterior reduce drásticamente el tamaño de las cabeceras; con un gran ahorro en el tamaño de los paquetes, supone una moderada pérdida de rendimiento.

Referencias

[1] Broch, J., Maltz, D. A., Johnson, D. B., Hu, Y.-C., and Jetcheva, J. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Mobile Computing and Networking* (1998), (ACM MOBICOM'98), pp. 85–97.

[2] Das, S. R., Perkins, C. E., and Royer, E. E. Performance comparison of two on-demand routing protocols for ad hoc networks. In *Proceedings of IEEE INFOCOM - The Conference on Computer Communications* (2000), pp. 3–12.

[3] Fall, K., and Varadhan, K. The ns manual. http://www.isi.edu/nsnam/ns/doc. UC Berkeley and Xerox PARC.

[4] Johnson, D., Maltz, D., and Broch, J. *DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks.* Addison-Wesley, 2001, ch. 5, pp. 139–172.

[5] Lewis, T. G., and Cook, C. R. Hashing for dynamic and static internal tables. *IEEE Computer 21* (1988), 45–56.

[6] Perkins, C. Ad hoc on demand distance vector routing. citeseer.nj.nec.com/article/perkins99ad.html, 1997.

[7] Perkins, C. E. *Ad Hoc Networking.* Addison-Wesley, 2001.

[8] Perkins, C. E., Belding-Royer, E. M., and Das, S. R. IP flooding in ad hoc mobile networks. www.ietf.org/proceedings/01dec/I-D/draft-ietf-manet-bcast-00.txt, 2001. IETF Internet Draft.

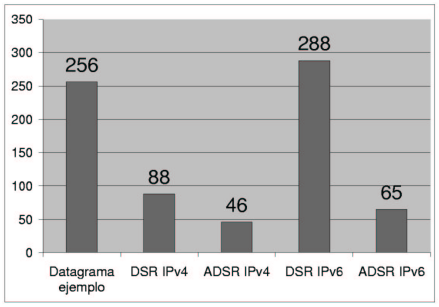


Figura 1: Comparación del tamaño del datagrama (bytes)

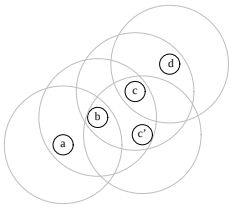


Figura 5: Colisión adyacente

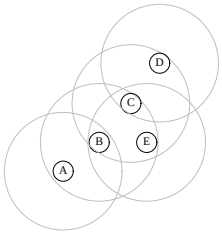


Figura 2: red ad-hoc

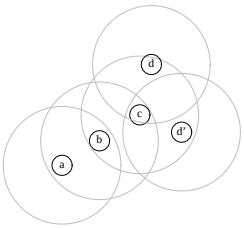


Figura 3: Colisión en destinatario

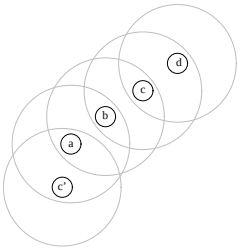


Figura 4: Colisión distante

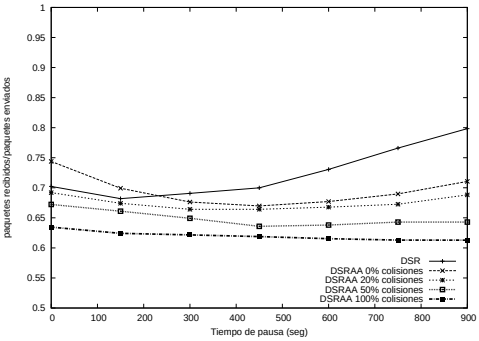


Figura 6: Paquetes Entregados: Media

Mejora de las Prestaciones de la Pila TCP/IP en Entornos Inalámbricos Multisalto

R. Sanz, R. Agüero, L. Sánchez, J. Choque, L. Muñoz
Departamento de Ingeniería de Comunicaciones
Grupo de Ingeniería Telemática. Universidad de Cantabria
39005 Santander
E-mail: [roberto, ramon, lsanchez, jchoque, luis]@tlmat.unican.es

Abstract. *This paper is focused towards the design and the development of machineries to leverage optimizations for heterogeneous multihop wireless networks. It is framed inside the IST Project 6HOP in which the core of the optimization machinery is the so-called Wireless Adaptation Framework (WAF). WAF is proposed as a remedy to the performance degradation problem that Internet protocols (and in particular IPv6-based) face when operated over a multihop environment realized with heterogeneous wireless technologies. 6HOP assumes a hybrid architecture with the following characteristics: access point devices are present, some wireless nodes are combined multihop routers and hosts, and wireless devices communicate directly with each other. Furthermore, 6HOP considers the integration of existing WLAN and WPAN technologies, bridging the gap between theoretical wireless ad hoc technology and viable multihop network solutions for commercial deployment. In this article, key features of the proposed framework are discussed and a high-level architectural description is given.*

1 Introducción y Objetivos

La presencia de las comunicaciones móviles en la vida cotidiana está en constante crecimiento desde finales del siglo XX. La telefonía móvil celular es únicamente un ejemplo, pero es asimismo apreciable el auge de tecnologías de redes inalámbricas de área local (WLAN, Wireless Local Area Network) y personal (WPAN, Wireless Personal Area Network). Este aumento se enmarca en un escenario en el que el acceso a Internet se ha convertido en un elemento cotidiano más.

El apogeo de las tecnologías móviles ha servido de catalizador para la aparición de numerosas y novedosas aplicaciones, entre las que destacan las redes multisalto (en la literatura frecuentemente denominadas redes ad hoc), por el amplio abanico de posibilidades que abren. Un ejemplo de este tipo de redes se puede encontrar en una sala de reuniones sin una infraestructura de comunicaciones apropiada, en la que un grupo de usuarios forma, de manera espontánea, una red temporal para comunicarse entre ellos.

Sin embargo, el ejemplo anterior es únicamente una pequeña muestra de las posibilidades que estas topologías tienen. Ciertos operadores se plantean emplear este tipo de redes para ampliar sus zonas de cobertura de manera rápida y rentable.

Evidentemente, y a pesar de este interés, queda un número considerable de aspectos técnicos relacionados con este tipo de topologías de red que se han de resolver cuando, por ejemplo, la pila de protocolos empleada es la bien conocida TCP/IP. En este sentido, son destacables los problemas que dicha pila experimenta en entornos inalámbricos con un solo salto [1], por lo que es de esperar que aquellos se vean magnificados por la presencia de un escenario multisalto.

En este contexto, el proyecto 6HOP ("Protocols for Heterogeneous Multi-Hop Wireless IPv6 Networks") perteneciente al V Programa Marco de la UE persigue la optimización de la pila TCP/IP sobre redes multisalto heterogéneas.

El artículo se estructura en las siguientes secciones: primeramente se dará un breve repaso al estado del arte de los aspectos más relevantes para la arquitectura que se propone; posteriormente, se recogerán los requerimientos más importantes de las topologías multisalto que habrá que considerar a la hora de afrontar el diseño de la arquitectura. A continuación se describirán las diferentes topologías que se emplearán para validar el diseño. Por último, se proporcionará una especificación de alto nivel de dicha arquitectura, introduciendo los mecanismos de señalización y transporte de datos, antes de concluir y bosquejar las siguientes fases que se afrontarán en el proyecto.

2 Desarrollos Previos

Las redes inalámbricas de múltiples saltos han sido recientemente objeto de una actividad investigadora muy importante. Destaca sobremanera la labor desarrollada por el grupo de trabajo MANET [2] que se encarga de la especificación de un conjunto de protocolos para el enrutamiento. Seguidamente, se dará un breve repaso a las características más importantes de estos protocolos, enumerando los aspectos más relevantes de las especificaciones que actualmente se encuentran recogidas dentro de ese grupo de trabajo. Adicionalmente, se hará una somera

descripción del resultado principal del proyecto WINE, ya que proporcionará el punto de partida del diseño de la arquitectura de 6HOP.

2.1 Protocolos de Enrutamiento Ad hoc

Los protocolos de enrutamiento de redes ad hoc se pueden dividir atendiendo a diversos parámetros; dos de los más importantes son: su política para adaptarse a los cambios topológicos de la red y la utilización de divisiones jerárquicas dentro de la misma. En cuanto al primero de ellos, los protocolos pueden ser de dos tipos: reactivos, cuando los procesos de descubrimiento de ruta a un destino en particular se ejecutan únicamente en el momento en que sean necesarios (por ejemplo al comenzar una comunicación); y preventivos, en los que el propio protocolo trata de adelantarse a los cambios de la red y mantiene, mediante el uso de mensajes periódicos, su topología en todos los componentes de la misma. El retardo a la hora de comenzar las comunicaciones es menor en el segundo de los grupos, que, sin embargo, presenta unas características de eficiencia y de consumo de energía peores que los reactivos, debido al gran número de mensajes que se tienen que transmitir.

El segundo de los criterios mencionados anteriormente hace referencia a la manera en la cual los nodos se disponen en una única red. Si no existe jerarquía alguna en la organización de los diferentes nodos y todos tienen, por tanto, el mismo rol, se trataría de un protocolo plano. Sin embargo, también se contempla la posibilidad de que se formen agrupaciones lógicas entre nodos que se encuentren en la misma zona. Esta última opción añade complejidad al protocolo, pero restringe la propagación de mensajes por toda la red, y posibilita la escalabilidad de la misma. Además, estas asociaciones lógicas pueden emplearse asimismo para implementar mecanismos adicionales (por ejemplo, seguridad) en este tipo de redes.

Aunque dentro de MANET se había examinado un amplio abanico de aspectos de rendimiento relacionados con el enrutamiento en redes ad hoc y se había trabajado en un número elevado de protocolos candidatos, tras una reciente revisión, se ha seleccionado un conjunto de únicamente cuatro protocolos en los que se centrará la atención a partir de ahora. Estos cuatro protocolos cubren por igual las aproximaciones reactiva y preventiva. A continuación, se dará una breve descripción de cada uno de ellos:

- AODV (Ad Hoc On Demand Distance Vector Routing), se trata de un protocolo reactivo, por lo que no lleva asociada una gran pérdida de eficiencia, ofreciendo buenas características en lo referente al gasto de energía. Su adaptación a los cambios topológicos de la red es rápida y dinámica y, además, no tiene unos requerimientos muy elevados en cuanto a memoria ni a complejidad en su algorítmica. Se trata del protocolo de enrutamiento sobre el que más se ha trabajado, tanto desde el punto de vista de evaluación como de implementación.
- DSR (Dynamic Source Routing Protocol), comparte la característica principal de AODV en cuanto a que se trata de un protocolo reactivo, que se adapta automáticamente al ritmo en el que se producen los cambios topológicos en la red. La mayor diferencia entre ambos es que en este caso es el nodo origen de un datagrama el que decide la ruta por la que debe viajar; por este motivo, además de unas necesidades mayores de memoria, la implementación es bastante más compleja que en el caso AODV. La consecuencia directa de este hecho es que el número de implementaciones disponibles de DSR sea bastante menor. Una de sus principales desventajas frente a AODV es que, al incluir la ruta completa en cada datagrama, se produce una pérdida de eficiencia considerable, sobre todo al emplear la nueva versión de IP (en la que el tamaño de las direcciones es bastante mayor); para evitar esta sobrecarga, se ha incluido recientemente una opción adicional a la recomendación, que se basa en el mantenimiento de un indicador de flujo y que evita mandar la ruta completa en todos los datagramas.
- OLSR (Optimized Link State Routing Protocol), se trata de un protocolo preventivo, basado en la técnica del estado de enlace (link-state). Para minimizar el número de mensajes que se envían a la red, optimizando, de ese modo, tanto el gasto de energía como la sobrecarga asociados a este tipo de protocolos preventivos, se seleccionan un conjunto de nodos que son los encargados de mandar, periódicamente, información acerca de la topología de la red.
- TBRPF (Topology Broadcast based on Reverse-Path Forwarding), otro ejemplo de protocolo preventivo, basado en el algoritmo de Dijkstra y en el que se trata de minimizar el impacto de los mensajes con información de la topología de la red haciendo que cada elemento únicamente transmita un subconjunto de la información que posee.

Dentro del proyecto 6HOP y debido, mayoritariamente, a las limitaciones de energía en los componentes de la arquitectura, se optó por emplear protocolos reactivos. Se acometerá la implementación de ambas opciones (tanto AODV como DSR), validando su correcto funcionamiento, para poder realizar una comparación exhaustiva entre ellas y dejar abierta, como área de investigación, la posibilidad de explotar la información que se proporcionará a través de la arquitectura de 6HOP para su modificación y optimización futuras.

2.2 Proyecto WINE

El principal resultado del trabajo desarrollado durante el proyecto WINE (Wireless Internet Networks) se

trata de un PEP (Performance Enhancing Proxy) de nivel 2, destinado a la mejora de las prestaciones de la pila TCP/IP cuando trabajen sobre un único salto inalámbrico [3]. Se busca ofrecer una solución local, sin afrontar la modificación de los protocolos de capas superiores. Dicho PEP, denominado capa de adaptación inalámbrica (WAL, Wireless Adaptation Layer), proporciona un marco genérico en el que se pueden incluir un conjunto de soluciones (módulos) para la mejora de diversos tipos de tráfico en un escenario que únicamente contempla un enlace inalámbrico (como muestra la Fig. 1). La solución diseñada no sólo se adapta al tipo de tráfico, sino que también se ajusta a las características del canal inalámbrico, medidas en términos de la relación señal a ruido (SNR, Signal to Noise Ratio), tasa de paquetes erróneos, etc.

En el proyecto 6HOP se pretende adaptar este PEP para que ofrezca un comportamiento adecuado en entornos multisalto. Para ello, será necesario, al menos: (1) reducir la señalización, aunque no muy relevante en el caso de un único enlace inalámbrico, introduce una complejidad elevada en el escenario multisalto de 6HOP; (2) la arquitectura que se empleó en WINE era claramente centralizada, ya que existía un elemento, el Punto de Acceso (AP, Access Point), que se encargaba de gestionar las comunicaciones con todos los terminales móviles (WT, Wireless Terminal) que se encontraran en su área de cobertura; en 6HOP esto no es posible, debido a la intrínseca característica de descentralización de las redes ad hoc, por lo que se tiene que diseñar una arquitectura cuyo funcionamiento sea eminentemente distribuido.

3 Requerimientos

A continuación, se mencionan los principales requerimientos a tener en cuenta para la optimización de las comunicaciones en un entorno inalámbrico multisalto.

3.1 Enrutamiento Multisalto

En una red inalámbrica multisalto algunos nodos deben poseer la funcionalidad de reenvío de paquetes como medio para posibilitar la comunicación entre nodos que no estén dentro del mismo área de cobertura. De hecho, el enrutamiento es una de las funciones más importantes en el diseño de una red inalámbrica multisalto. Éste se puede llevar a cabo utilizando diferentes alternativas como, por ejemplo, enrutamiento a nivel IP (Apartado 2.1), bridging de nivel 2 y esquemas de acceso celular.

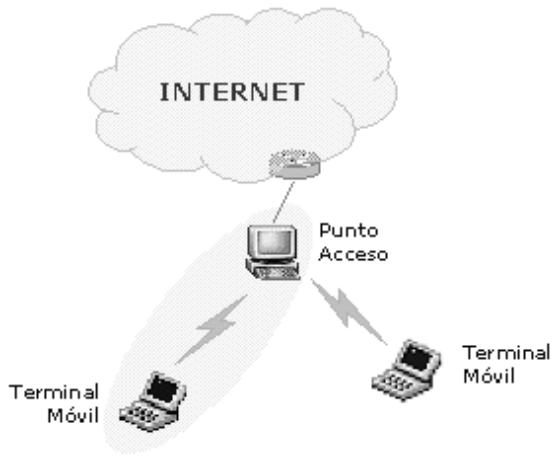


Figura 1. Topología de referencia del proyecto WINE

3.2 Compensación del Canal Radio

Los enlaces inalámbricos de una red multisalto se caracterizan por presentar una elevada pérdida de paquetes, alta latencia y bajo rendimiento. Estas características se deben a las imperfecciones del canal radio y el resultado obvio será la degradación de las prestaciones finales de las aplicaciones, en particular las basadas en la arquitectura de protocolos TCP/IP.

Es recomendable la introducción de nuevos mecanismos que compensen todos estos efectos negativos. Estas funciones se implementan mediante los denominados "protocol boosters", que residen generalmente en la capa de enlace o entre la capa de red y el driver inalámbrico [4] de manera transparente a las capas superiores.

3.3 Interfaz Inalámbrica Uniforme

Existen interfaces de red inalámbricas con diferentes características desde el punto de vista de nivel físico y MAC. Habitualmente, las aplicaciones y protocolos de capa superior requieren acceder a éstas a través de su driver. Para que una arquitectura soporte la presencia de tecnologías heterogéneas, se requiere una interfaz común que posibilite el acceso uniforme a los protocolos de capa inferior y drivers inalámbricos. Dicha interfaz común debería soportar, al menos, primitivas para la configuración, obtención de estadísticas y manejo de eventos. Además, facilitará la operación de aplicaciones inalámbricas así como la de protocolos de gestión de rutas y de movilidad.

3.4 Mitigación de la Interferencia

En un entorno inalámbrico multisalto y heterogéneo, pueden existir diferentes enlaces correspondientes a equipos de comunicaciones operando en la misma banda de frecuencias. Los casos más característicos en el marco de este artículo, son los de las tecnologías Bluetooth e IEEE 802.11b, ambas operando en la banda no regulada de 2.4 GHz, siendo necesarias técnicas que mitiguen los efectos de la interferencia entre ambas. Por otro lado, existen otros remedios, como alternar la transmisión entre ambas tecnologías (división temporal) o utilizar tecnologías inalámbricas que operen en bandas de frecuencia diferentes.

3.5 Consumo de Energía

Los terminales móviles, tales como ordenadores portátiles o agendas electrónicas de bolsillo (PDA) ven limitada su vida de operación debido a la dependencia de su alimentación a través de una

batería. La demanda de energía estará relacionada tanto con el número de procesos como con la potencia radiada que conlleva el mantenimiento de los enlaces de comunicaciones. Se deberá implementar una pila de protocolos eficiente, utilizando esquemas de recuperación de errores que eviten las retransmisiones innecesarias, limitar el número de colisiones, evitar el modo de operación promiscuo de las tarjetas de red y reducir la utilización de mensajes periódicos que realizan, por ejemplo, los protocolos de enrutamiento preventivos. Puede ser asimismo beneficioso reducir la potencia de transmisión cuando las condiciones del canal sean favorables.

3.6 Seguridad

Las redes inalámbricas son tradicionalmente consideradas como redes fácilmente vulnerables, al no existir la posibilidad de aislar el medio radio. Los riesgos existentes incluyen: escuchas ilegales, intrusión en la red de hardware no permitido, uso dañino de servicios, etc. Sin la presencia de ningún nivel de seguridad, una arquitectura inalámbrica puede llegar a ser inviable, independientemente de la presencia de mecanismos para la mejora de prestaciones.

A pesar de no tratarse un objetivo prioritario del proyecto 6HOP (marco de este artículo) se tendrán en consideración las implicaciones que los diferentes mecanismos de seguridad puedan imponer a la arquitectura diseñada.

3.7 Optimización Inter-Capa de Protocolos

Las pilas de protocolos tradicionales se basan en la comunicación vertical entre capas adyacentes. Sin embargo, debido a las condiciones variantes del canal radio, la movilidad de los terminales y las limitaciones de consumo de energía, las redes inalámbricas multisalto requieren que las diferentes capas de la pila de protocolos interactúen entre ellas.

El concepto de optimización inter-capa de protocolos (cross-layer) en el marco de las redes inalámbricas ad hoc fue propuesto inicialmente en [5]. Básicamente, se define un entorno de comunicación vertical dentro de la pila de protocolos; es decir, cada protocolo adapta su modo de operación en función de las condiciones del canal, las modificaciones topológicas de la red, los requerimientos de las aplicaciones, etc. La información asociada a la adaptación de una entidad particular se exporta a otros protocolos de la pila, con objeto de que éstos se ajusten o no a dichas variaciones, consiguiendo una optimización global. Esta aproximación constituye el núcleo del diseño de la WAL [3].

3.8 Soporte para IPv6

La introducción del protocolo IPv6 en una red multisalto inalámbrica posee, desde el punto de vista técnico, las siguientes ventajas: (1) número muy elevado de direcciones (se asume que en un futuro el número de terminales inalámbricos será alto), eliminando la necesidad de traductores de direcciones de red (NAT, Network Address Translation); (2) IPv6 se adapta bien a los protocolos de enrutamiento debido a su estructura intrínseca de cabeceras (cabeceras básica y extendida); (3) facilidad en la utilización de mecanismos de compresión; (4) IPv6 presenta un completo formato de cabecera para implementar seguridad y calidad de servicio.

Además, se identifican otros requerimientos, como son el direccionamiento (el modo en que se asignan las direcciones IPv6 a los nodos) y mecanismos de transición (la integración de una red inalámbrica multisalto basada en IPv6 sobre una existente basada en IPv4).

4 Topologías

4.1 Adaptación y Optimización Inalámbrica Multisalto

A la hora de afrontar el desarrollo de mecanismos de adaptación y optimización en una red inalámbrica multisalto dentro del marco del proyecto 6HOP, se considerarán un conjunto de restricciones: (1) los nodos se caracterizan por presentar una movilidad baja o nula; (2) el número de saltos es limitado; (3) pueden existir múltiples rutas entre una pareja de nodos; (4) cada salto puede utilizar una técnica inalámbrica diferente (heterogénea); (5) un nodo con baja movilidad puede cambiar sus puntos de conexión con nodos vecinos (handoff).

La Fig. 2 muestra los diferentes escenarios de investigación que se consideran en 6HOP: (a) un único salto, un nodo con baja movilidad, este fue el escenario del proyecto WINE [3]; (b) multisalto fijo, distancias variables entre nodos, ruta única; (c) multisalto con baja movilidad, ruta única; (d) red mixta, nodos con baja movilidad pueden cambiar su punto de conexión (cambio de topología), múltiples rutas; (e) red ad hoc mallada, múltiples rutas posibles, topología cambiante.

En el proyecto 6HOP, el escenario (a) no será considerado, pues ya lo fue en [3]. El escenario (e) está fuera del alcance en lo que a implementación se refiere. El escenario (b) es el caso más básico que se abordará; (c) resulta más complicado puesto que las condiciones del canal pueden variar rápidamente, y

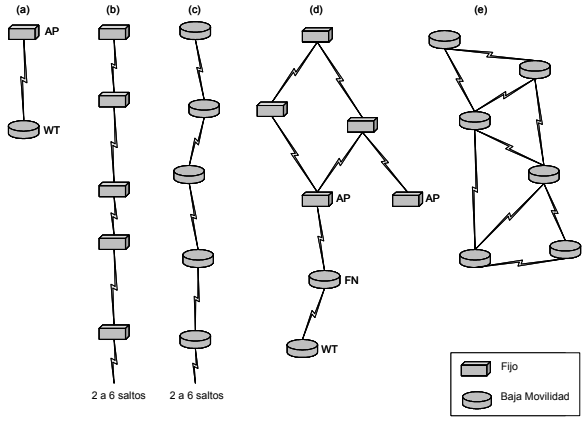


Figura 2: Escenarios de investigación multisalto

(d) es la situación más realista para una red de acceso inalámbrica en la que existen múltiples rutas y cambios topológicos.

La investigación analítica partirá de los escenarios (b) y (c) para comprobar cómo influyen la movilidad de los nodos, la distancia entre ellos, el número de saltos y las condiciones del canal en el comportamiento del sistema. Dependiendo de estos factores y modificando los parámetros de configuración del modelo de simulación (longitud del paquete, estados del canal, etc), se identifican diferentes técnicas que pueden mejorar las prestaciones del sistema. El escenario (d) es el que más se se acerca lo más posible a un entorno real en el que los nodos poseen diferentes niveles de movilidad, soportan tecnologías físicas heterogéneas (IEEE 802.11a/b, Bluetooth) y cambian su punto de conexión. Las técnicas evaluadas mediante simulaciones se verificarán con medidas reales.

4.2 Enrutamiento Mejorado

En la medida de lo posible, se debe elegir el mecanismo de descubrimiento de ruta más apropiado a los requerimientos de baja complejidad e interoperabilidad. Desde una perspectiva general, el enrutamiento consiste en el transporte de información de un origen a un destino a través de la red, involucrando dos actividades básicas: determinación de las rutas óptimas y transporte de la información a través de la red. El enrutamiento puede realizarse tanto a nivel 3 (routing) como a nivel 2 (bridging), siendo la diferencia entre ambos la información manejada y, por tanto las tareas realizadas para enrutar. En el presente trabajo, se consideran diferentes escenarios sobre los que se compararán diversos mecanismos de enrutamiento con el fin de elegir la solución que mejor se adapte al entorno de una red multisalto. En este entorno, la información proporcionada por la interfaz inalámbrica uniforme (tasa de paquetes erróneos, calidad del enlace radio, consumo de energía o la notificación de traspaso explícita – EHN, Explicit Handoff Notification), pueden mejorar las prestaciones globales de la red.

4.3 Escenarios de Prueba

Se considerarán los siguientes escenarios de prueba: (1) bridging: el enrutamiento se realiza a nivel 2; en la Fig. 3 los Puntos de Acceso (AP, Access Point) y los Nodos Intermedios (FN, Forwarding Node) actúan como puentes que implementan el protocolo Spanning Tree; (2) enrutamiento MANET a nivel IP; en la Fig. 3 se realiza en las subredes que forman el AP_2 y el AP_3 mientras que los AP_2 y AP_3 emplean enrutamiento IP tradicional en dirección hacia el AP_1); (3) integración de soluciones de macro-movilidad (Mobile IP). En este caso, el enrutamiento multisalto se realiza mediante técnicas de MANET dentro de las subredes que forman el AP_2 y el AP_3, mientras que la gestión de movilidad intra-dominio controla el enrutamiento entre AP_1, AP_2 y AP_3 dando soporte a los traspasos de un terminal móvil entre APs (entre AP_2 y AP_3).

5 Arquitectura

En el seno del proyecto 6HOP se define el concepto de WAF (Wireless Adaptation Framework) como una arquitectura diseñada específicamente para redes inalámbricas multisalto heterogéneas. El objetivo de WAF es servir de mecanismo que potencie las comunicaciones multisalto basadas en IPv6 sobre WLAN/WPAN existentes y venideras, lo que implica el acceso tradicional a servidores remotos y comunicaciones peer-to-peer. WAF abarcará un subconjunto de los requerimientos definidos en el Apartado 3, además de posibilitar la interacción con otro tipo de tecnologías.

5.1 Requerimientos y Características Principales de WAF

La optimización de las comunicaciones en entornos multisalto impone una serie de requerimientos: (1) descubrimiento, mediante primitivas de señalización, de las capacidades del nodo vecino con respecto a los mecanismos de optimización con objeto de negociar las correspondientes funcionalidades soportadas por cada uno; (2) algunos mecanismos demandarán la operación en modo extremo a extremo a través de múltiples saltos inalámbricos, lo que implica una señalización entre los nodos origen y destino así como la inteligencia necesaria en los nodos intermedios; (3) la señalización que coordine la operación de los protocolos deberá ser ligera, introduciendo la menor sobrecarga posible (número de mensajes del protocolo reducido y eliminación de los avisos periódicos); (4) las técnicas de enrutamiento multisalto deberán estar correctamente integradas con el resto de mecanismos de optimización, los cuales exportarán información (a través de la interfaz inalámbrica uniforme) a cualquier protocolo de enrutamiento multisalto; (5) las aplicaciones futuras deberán asimismo integrarse correctamente con las técnicas de optimización.

Las principales características de WAF son: (1) ser un mecanismo que permita el control de los drivers inalámbricos y los protocolos de nivel de enlace, así como la operación de protocolos de otros niveles; (2) soportar protocol boosters que mejoren las prestaciones del canal radio; (3) proporcionar una

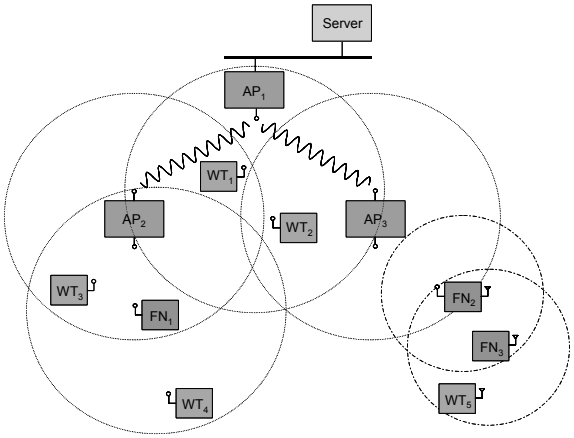


Figura 3. Red inalámbrica multisalto heterogénea

interfaz inalámbrica uniforme para cualquier protocolo de nivel de enlace y tecnología inalámbrica; (4) soportar la cooperación entre nodos-WAF para permitir el descubrimiento de las capacidades de los nodos vecinos; (5) configuraciones de red eficientes – gestor de conexión o de consumo de energía; (6) gestión remota – utilizando el protocolo estándar SNMP y nuevas MIBs; (7) extensibilidad – un conjunto de protocolos y agentes de control residentes en diferentes niveles de la pila de protocolos utilizando la interfaz inalámbrica uniforme; (8) soportar enrutamiento multisalto aunque sin depender del protocolo particular implementado.

5.2 Arquitectura de Alto Nivel de WAF

La Fig. 4 muestra la arquitectura de alto nivel de WAF. Los principales componentes de la arquitectura WAF son:

- Wireless API – es el componente central de WAF – proporciona una interfaz uniforme a las aplicaciones y protocolos para posibilitar su operación. También, facilita la configuración, la obtención de estadísticas y ofrece servicios para la gestión de eventos mediante la enmascaración de los drivers inalámbricos subyacentes, los protocol boosters y las tablas de enrutamiento. Es decir, el acceso a componentes de niveles inferiores se lleva a cabo a través de la API propuesta, que sigue un enfoque similar al utilizado por [6]. Además, la Wireless API puede emplearse para optimizar el enrutamiento multisalto, combinando las decisiones de enrutamiento con las características de los niveles inferiores.
- Gestor(es) – emplean la Wireless API para realizar funciones de control muy concretas. Algunos ejemplos de gestores son WAM (Wireless Adaptation Manager) y CPM (Connection and Power Manager).
- Protocolo(s) – básicamente son protocolos de enrutamiento y de gestión de movilidad que operan empleando la información recibida de la Wireless API que podría utilizarse, además, por algunas aplicaciones.
- Agente(s) – son agentes de gestión de red basados en SNMP.
- Protocol boosters – se dedican al procesado de paquetes durante la transmisión/recepción a/desde el driver inalámbrico para mejorar el rendimiento del sistema. Los protocol boosters se aplican en el transmisor y/o en el receptor en función de los requerimientos de tráfico demandados por las aplicaciones y las características del dispositivo inalámbrico particular. Algunos ejemplos de protocol boosters son: control de errores FEC (Forward Error Correction), TCP snooping [7] y RoHC (Robust Header Compression) para TCP [8].
- Interfaces – permiten a la Wireless API acceder y manipular el driver inalámbrico, los protocol boosters y las tablas de enrutamiento.

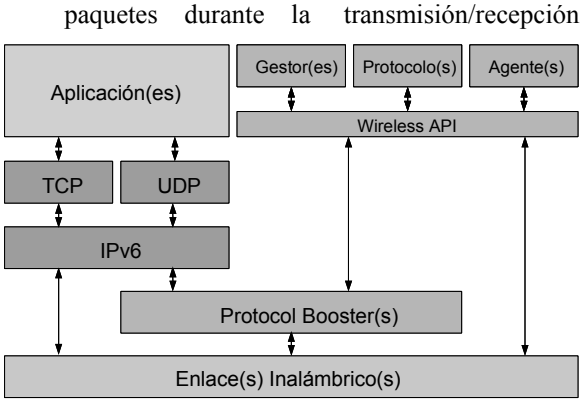


Figura 4: Arquitectura de alto nivel de WAF

6 Tráfico WAF

6.1 Plano de datos

Como ya se ha mencionado anteriormente, uno de los principales requisitos que se cubre en el proyecto 6HOP es la descentralización de su esquema de comunicaciones, sin la presencia de ningún tipo de jerarquía implícita en la red. Una consecuencia directa de este enfoque es que cada nodo debe decidir (de manera independiente) los mecanismos con los que procesar cada paquete de datos. Además, la primitiva correspondiente deberá ser lo suficientemente auto contenida, para que el receptor pueda invertir el proceso. De esta manera no es necesario establecer ningún tipo de conexión (como sucedía en el proyecto WINE), consiguiendo reducir considerablemente la señalización. Para ello se propone el formato de trama que se muestra en la Fig. 5. Cada módulo añade su cabecera que incluye, al menos, información acerca de: (1) el propio módulo, mediante un identificador unívoco; (2) la longitud de la cabecera; (3) parámetros de operación del módulo, si fuera necesario; y (4) siguiente módulo que la primitiva de datos tiene que visitar. Teniendo en cuenta que un error en la cabecera tiene un impacto elevado en la operación de todo el protocolo, es apropiado proteger esta información con un código CRC. Por último, por cuestiones pragmáticas en la implementación, se propone añadir un campo que contenga la longitud total de la información WAF de la trama.

6.2 WAM

WAM se encarga de la gestión de los protocol boosters que ocultan las deficiencias de los canales inalámbricos a las capas superiores mediante la utilización de un conjunto de parámetros, como las características del canal, el tipo de tráfico, las capacidades de los nodos vecinos, nivel de energía disponible, etc. Para asegurar el correcto funcionamiento del esquema de comunicaciones

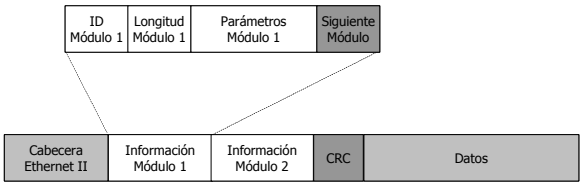


Figura 5. Formato de la trama de datos WAF

distribuido es necesario que cada nodo sea consciente de sus vecinos (aquellos que se encuentren en su área de cobertura) y de las capacidades que tienen, por lo que es necesario incluir un mecanismo de señalización entre entidades WAM. Por otro lado, para poder adaptar su operación a las características del canal, al nivel de energía disponible en un nodo, etc. es necesario definir un conjunto de interfaces (a través de la Wireless API) que le permita el acceso a esta información.

En lo referente a la señalización entre entidades WAM se distinguen dos posibilidades diferenciadas: (1) al activarse, un nodo realizará un procedimiento con el fin de conocer las capacidades de todos sus vecinos y viceversa, como aparece reflejado en la Fig. 6; (2) si algún nodo (activo) que se está moviendo, se introduce en el área de cobertura de otro terminal, la topología de la red se verá modificada, aspecto detectado por los mecanismos de enrutamiento, que informarán a WAM; éste, al percatarse de tal situación, tratará de descubrir las capacidades del nuevo vecino, para lo que le mandará un mensaje similar a los mostrados en la Fig. 6.

Como se puede observar, el primero de los mecanismos tiene características preventivas, mientras que el segundo presenta un comportamiento reactivo. Todo el tráfico de señalización se transmitirá sobre UDP, para reducir la sobrecarga asociada al tráfico de control.

6.3 CPM

Uno de los objetivos más importantes entre los que se persiguen en el diseño de la arquitectura de 6HOP es proporcionar un comportamiento que tenga en cuenta los requerimientos de energía de los nodos. Algunas interfaces de red permiten la modificación de ciertos parámetros de operación, que facilitarían este tipo de comportamiento. Por ejemplo se podría reducir la potencia de transmisión cuando el estado del canal inalámbrico es lo suficientemente bueno, optimizando el consumo de energía.

Un terminal también podría deshabilitar su capacidad de FN, en función de la energía disponible en cada momento, pasando de comportarse como un WT; en este caso, CPM empleará cierta señalización externa para anunciar esta nueva situación a sus vecinos, como se muestra en la Fig. 7. Funcionalidades adicionales incluirían la gestión del estado de latencia de las tarjetas inalámbricas o la modificación del número máximo de retransmisiones.

Para que el conjunto de funcionalidades descrito con anterioridad se pueda llevar a cabo, es necesario que CPM acceda a un conjunto de datos, a través de interfaces proporcionadas por la Wireless API. Por último, todo el tráfico de control entre entidades CPM se realizará, al igual que en el caso de WAM, sobre el protocolo UDP, para mantener al mínimo la sobrecarga introducida por la señalización.

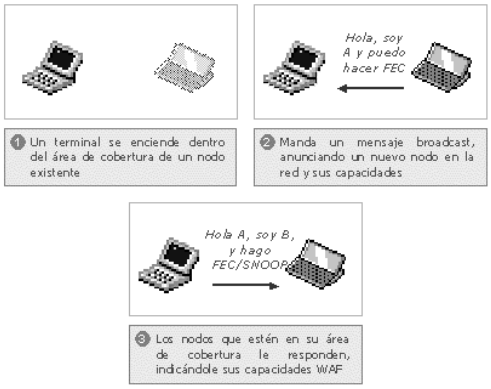


Figura 6. Descubrimiento de nuevos terminales y de las capacidades WAF de un nodo

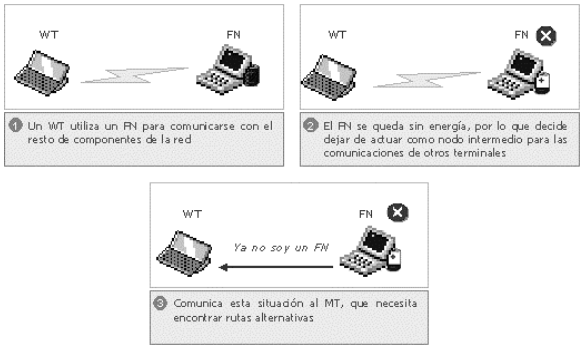


Figura 7. Ejemplo de modificación de capacidades WAF

7 Conclusiones

Las redes inalámbricas multisalto poseen el potencial de mejorar las existentes redes inalámbricas de una manera eficiente y robusta. Sin embargo, es necesario realizar ciertas optimizaciones para alcanzar una solución viable, a la vez que se afronta la integración de las redes existentes (WLAN) con las redes futuras (WPAN). Mediante el proyecto 6HOP se desea obtener una solución factible que cumpla con los requerimientos planteados, enfocándolo hacia aplicaciones reales que demandan la interconexión de terminales en modo ad hoc.

La solución que plantea 6HOP se pone en práctica a través del concepto de WAF, orientado a potenciar las prestaciones de las redes inalámbricas multisalto heterogéneas. Con el fin de soportar aplicaciones reales, 6HOP asume una topología a pequeña escala y semi-dinámica, esto es, un número de saltos limitado y baja movilidad.

El principal objetivo de 6HOP es la optimización de las redes inalámbricas multisalto, las cuales están formadas, a su vez, por nodos que emplean tecnologías inalámbricas heterogéneas, particularizando la evaluación de las prestaciones a los protocolos Internet y en concreto al transporte basado en IPv6. Además, se debe considerar la naturaleza híbrida de la estructura de la red puesto que una red multisalto cuya formación es espontánea (sin cambios rápidos en la topología) se conecta, de manera transparente, con dispositivos que forman parte de una infraestructura.

En este artículo se plantean los requerimientos de comunicación multisalto demandados por WAF así como la arquitectura propuesta y sus principales componentes desde una visión de alto nivel. También se introducen los mecanismos de transporte de datos y de señalización entre entidades WAF.

Agradecimientos

Este trabajo ha sido realizado en el marco del proyecto IST-2001-37385 "Protocols for Heterogeneous Multi-Hop Wireless IPv6 Networks" (6HOP) financiado por la Unión Europea dentro del programa IST (Information Society Technologies) y del proyecto TIC2002-02817 "Red de Acceso Celular IP Multisalto" (RACIMUS) financiado por el Ministerio de Ciencia y Tecnología.

Referencias

[1] L. Muñoz, M. García, J. Choque, R. Agüero, and P. Mähönen, "Optimizing Internet Flows over IEEE 802.11b Wireless Local Area Networks: A Performance Enhancing Proxy Based on Forward Error Correction," IEEE Communications Magazine, Vol. 39, No. 12, pp. 60-67, diciembre 2001.

[2] IETF, Mobile Ad-Hoc Networks (MANET), `http://www.ietf.org/html.charters/manet-charter.html`

[3] P. Mähönen, T. Saarinen, N. Passas, G. Orphanos, L. Muñoz, M. Carcía, A. Marshall, D. Melpignano, T. Inzerilli, F. Lucas, and M. Vitiello, "Platform-Independent IP Transmission over Wireless Networks: The WINE Approach," IEEE Personal Communications, vol. 8, no. 6, pp. 32-40, diciembre 2001.

[4] D.C. Feldmeier, A.J. McAuley, J.M. Smith, D.S. Bakin, W.S. Marcus, and T.M. Raleigh, "Protocol Boosters," IEEE Journal on Selected Areas in Communications, vol. 16, no. 3, pp. 437-444, abril 1998.

[5] A.J. Goldsmith and S.B. Wicker, "Design Challenges for Energy-Constrained Ad Hoc Wireless Networks," IEEE Wireless Communications, vol. 9, no. 4, pp. 8-27, agosto 2002.

[6] J. Tourrilles, "Wireless Extensions for Linux," Linux Wireless LAN Howto, enero 1997.

[7] H. Balakrishnan, S. Seshan, and R.H. Katz, "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks," ACM Wireless Networks, diciembre 1995.

[8] L.E. Jonsson, "Requirements on ROHC TCP/IP Header Compression," Internet Draft, <draft-ietf-rohc-tcp-requirements-05.txt>, octubre 2002, trabajo actualmente en desarrollo, expira en abril de 2003.

Sesión 5B

Comunicaciones multimedia

Análisis de un Servicio de Vídeo Bajo Demanda Basado en Tecnología de Streaming

Xabiel G. Pañeda, David Melendi, V. García, R. García, A. Neira, E. Riesgo
Departamento de Informática, Universidad de Oviedo
Campus Universitario de Viesques. Sede Departamental Oeste
33204 Xixón-Gijón Asturies, Spain
Teléfono: 985 18 33 70 Fax: 985 18 19 86
E-mail: {xabiel, melendi, victor, roberto}@correo.uniovi.es

Abstract. *This paper describes the ongoing work in the analysis process of video streaming service. The aim of this analysis is to get the optimum configuration of the service from two points of view, the service provider and the communication operator. Configuring the caches, distributing the content, developing redistribution routes, creating new contents in the most popular subject and increasing or decreasing the length of information depending on the subscribers behaviour can improve the quality of the service. To carry out this analysis a web tool has been developed and added to the service. This tool provides an automatic way to know the behaviour of the service, by using graphical representations and textual information. Media transmission time, session length, and pause periods histograms as well as Zipf–like, fast leaving and failed access studies are generated. The information obtained will be used to develop a performance model of the service and to improve the configuration. The studied service is the www.lanuevaespana.es multimedia section, which is one of the most successful digital news sites in Spain. The section offers a large number of videos on demand with several subjects, length and quality. This work is included in a project about analysis, modelling and configuring of interactive multimedia services.*

1 Introducción

No han pasado muchos años desde que Internet era simplemente una red académica. Sus usuarios utilizaban navegadores de texto para acceder a información relacionada con sus investigaciones y estudios. La aparición del World Wide Web y de los navegadores gráficos, a principios de los noventa, ha llevado a la red de redes a convertirse en el más importante medio de comunicación de masas. Con el incremento del número de usuarios han aparecido los intereses comerciales, y grandes empresas de todos los sectores (comunicaciones, información, distribución) han realizado inversiones millonarias para tratar de atraer a sus sitios ese potencial mercado global. Uno de estos grupos de compañías es el que engloba a los medios de comunicación, que han apostado por situar copias de sus diarios y publicaciones en la red. Debido a la gran competencia que se ha creado, la simple colocación de información en la red no ha sido suficiente para conseguir captar clientes, y ha sido necesario un gran trabajo que los convirtiera en atractivos. Para lograrlo han ido incorporado fotos, animaciones y videos que atrajeran al público.

El vídeo bajo demanda permite que el usuario mediante un simple clic pueda visualizar un vídeo en el instante que desee, e incluso interaccionar con él. Utilizando la tecnología de *streaming,* el vídeo puede ser reproducido por el cliente sin necesidad de descargas previas. Únicamente serán cargados unos pocos segundos en un buffer del cliente para amortiguar deficiencias en la transmisión. Las ventajas de esta tecnología y las expectativas generadas en los usuarios han provocado la proliferación de secciones multimedia en las publicaciones digitales. Sin embargo, el video bajo demanda presenta algunos problemas. Uno de los más importantes es el consumo de recursos que realiza tanto en los equipos que proveen el servicio como en la red, donde además de necesitar anchos de banda importantes requiere que la calidad de servicio se mantenga constante. Con objeto de mantener los parámetros de calidad y seleccionar los contenidos más interesantes resulta necesario realizar análisis que determinen los parámetros de configuración más correctos. Para alcanzar una calidad de servicio adecuada debe trabajarse desde dos puntos de vista diferentes, el del operador de red y el del proveedor del servicio. El operador de red deberá conseguir mejorar el servicio trabajando sobre las arquitecturas de servicio, el enrutado y el ancho de banda. Por otro lado el proveedor del servicio intentará mejorar el servicio trabajando sobre los contenidos, variando su calidad (definición, ancho de banda consumido), temática, duración, etc.

En este artículo se presenta el proceso de análisis seguido en el servicio de vídeo *streaming* de La **Nueva España Digital**. Este servicio inició su andadura a principios del año 2001, siendo desarrollado por el área de Ingeniería Telemática de la Universidad de Oviedo. Con el objeto de conseguir una configuración adecuada se le dotó de una

herramienta automática de análisis. Sus resultados han permitido mejorar tanto los elementos de configuración propios del operador de red, como los elementos propios del proveedor del servicio. Para facilitar la configuración del servicio, por parte del proveedor, a partir de los resultados proporcionados por la herramienta se ha desarrollado una metodología que incluye tanto los tests necesarios para identificar la acogida por parte de los usuarios de los contenidos, como la posible solución en caso de haber sido esta mala. Para facilitar la configuración al operador de red se ha desarrollado un modelo de prestaciones que permite comprobar el funcionamiento bajo todo tipo de condiciones, pudiendo prever la evolución del servicio.

El resto del artículo esta organizado de la siguiente forma: en la sección 2 se hará un recorrido por otros trabajos relacionados con el tema. El sistema que ha sido objeto de estudio será presentado en la sección 3. El proceso de análisis seguido será descrito en la sección 4. La sección 5 se centrará en la herramienta de análisis desarrollada. La sección 6 recorrerá los elementos fundamentales de la metodología de análisis y configuración diseñada para los proveedores de servicios de vídeo bajo demanda. La sección 7 describirá las principales características del modelo de simulación desarrollado. La sección 8 comentará los resultados obtenidos con el proceso de análisis en el caso de estudio. Por último, las conclusiones y los trabajos futuros serán expuestos en las secciones 9 y 10.

2 Trabajos Relacionados

El análisis de servicios de video bajo demanda es un campo relativamente joven dentro del mundo de la investigación. No hace muchos años que estos servicios han comenzado a utilizarse y por tanto los trabajos en este campo aún no son muy numerosos. Sin embargo, si son abundantes los estudios realizados sobre sistemas Web clásicos como [1,2,3]. Estos estudios pueden servir como base para investigaciones en el campo del vídeo, si bien debe tenerse en cuenta la diferencia sustancial que supone el trabajo con peticiones de tipo discreto como sucede con las páginas Web, y las peticiones de tipo continuo como en el caso del vídeo y del audio.

En los últimos años han aparecido diversos artículos que abordan el tema de análisis de servicios multimedia, algunos de ellos se han centrado en parámetros generales de servicio sin centrarse en elementos propios de servicios de tipo continuo como [4,5] y otros [6,7,8,9] sin embargo, han tocado elementos como, tiempos entre interacciones, duración de las sesiones de los usuarios, saltos en la línea de reproducción, etc, más propios de estos servicios. En todos los casos el análisis se realizaba mediante el estudio de los *logs* con herramientas *off-line* que unos casos se habían desarrollado a tal efecto y en otros casos eran de propósito general. En ningún caso, estos trabajos afrontaron el análisis como un elemento continuo que mejorara la configuración día a día. También se han publicado algunos trabajos sobre prototipos y sobre generadores de cargas sintéticas [10,11] con conclusiones interesantes. Otro aspecto incipiente es la generación de métricas de calidad para estos servicios de tipo continuo. En este campo ha aparecido algún artículo [12], que da una nueva óptica al análisis que hasta ahora se limitaba a chequear la calidad en función de los paquetes perdidos. No se ha encontrado ningún estudio en el campo de la configuración del servicio desde el punto de vista del proveedor, todos los estudios anteriormente citados se han centrado en obtener conclusiones para la optimización de recursos, tanto en equipos como en red.

Este trabajo pretende aportar novedades sobre los trabajos anteriores en varios puntos. El primero es convertir el análisis en un sistema automático y continuo que permita realizar los ajustes necesarios en cada momento. Los servicios de vídeo bajo demanda están actualmente siendo analizados con herramientas adaptadas de los sistemas Web [13,14] que no permiten estudiar ciertos elementos característicos de los medios continuos. En el caso de estudios como los anteriormente relatados, que han profundizado más, nunca se ha asumido el análisis como un proceso continuo.

Otra aportación es la de afrontar el análisis desde dos puntos de vista, el del proveedor del servicio y el del operador de red. Los trabajos publicados hasta el momento han prestado escasa atención a la configuración que el proveedor del servicio puede realizar para mejorar la calidad y aumentar el número de usuarios. En el campo del análisis de prestaciones no se tiene constancia de ningún modelo, hasta ahora todos los trabajos se han encaminado a la medición [13,14].

Por último cabe destacar que este estudio ha detectado patrones de comportamiento significativamente diferentes a los presentados en trabajos como [6], donde el tiempo de vídeo transmitido podía representarse con distribuciones del tipo Lognormal, Pareto o Gamma. En este estudio se ha observado un comportamiento similar a la suma de distribuciones Gamma y Lognormal.

3 Descripción del Servicio

Esta sección describe el servicio multimedia de la **Nueva España Digital**, una de las 10 publicaciones digitales más visitadas del país, según la empresa de control de visitas OJD. El servicio multimedia de la www.lanuevaespana.es inició su andadura a principios de 2001. En la actualidad el número de vídeos y de visitas se ha ido incrementando hasta alcanzar una gran reputación debido fundamentalmente al nivel de producción propia.

3.1 Arquitectura del Servicio

La sección multimedia de la Nueva España Digital tiene una arquitectura formada por tres servidores. El primero de ellos es el que soporta el servicio de páginas Web desde el cual se accede a los vídeos. Los otros dos servidores se encargan, uno de servir el vídeo y otro del sistema de análisis. La sección puede ser accedida desde dos enlaces en la página principal (uno en el menú y otro en un *banner*) y mediante un enlace en cada una de las otras páginas. Para reproducir los vídeos existen dos posibilidades: una mediante una página Web en la que se incrusta el *plug-in* del reproductor de vídeo y otra mediante el programa reproductor directamente. La tecnología utilizada para transmitir el vídeo es Real Server 8 [11], que lo servirá bajo demanda cuando los clientes lo soliciten. El sistema de análisis obtendrá del servidor de vídeo los *logs* de funcionamiento y los procesará para que mediante la herramienta Web de la que dispone se puedan visualizar los informes deseados.

3.2 Descripción de los Contenidos

El servicio multimedia de La Nueva España Digital contiene información variada clasificada en 7 secciones según la temática. Las secciones son: noticias, música, turismo, conferencias, cortometrajes, visitas y otros. La sección de ***Noticias*** contiene todo tipo de información de actualidad (ruedas de prensa, reportajes, entrevistas, etc). Su duración varía entre los 30 segundos de las más cortas hasta los 20 minutos de las más largas. La sección de ***Música*** abarca todo tipo de información relacionada con la misma, siendo dos los elementos fundamentales, los video clips (de corta duración) y las entrevistas a músicos y productores (rondan los 20 minutos). La sección de ***Turismo*** abarca diversos vídeos relacionados con Asturies, desde gastronomía hasta naturaleza, pasando por turismo. Los vídeos tienen una duración entre los 30 y los 45 minutos y están producidos por la Productora de Programas del Principado. La sección de ***Conferencias*** recoge las grabaciones del Ciclo de Conferencias de Ciencia y Cultura de la Universidad de Oviedo. Aunque son de temática científica tratan de tocar los diversos temas con un aire de divulgación. Su duración está entre la hora y las dos horas, y disponen de un índice interactivo que permite al usuario posicionarse en partes concretas de la exposición. La sección de ***Cortometrajes*** recoge películas, cuya duración en ningún caso excede los 15 minutos. La sección de ***Visitas*** agrupa a las visitas que diversos colegios realizan a la sede del periódico. Su duración ronda los 2 minutos. Por último está la sección de ***Otros*** que enmarcaría a vídeos que no se adaptan a ninguna de las demás secciones. Tanto la temática como la duración de los mismos es variada.

Todos los vídeos están disponibles en tres calidades que permiten al usuario seleccionarlos según el tipo de red de acceso de la que disponga. Las calidades son: módem, banda ancha, plus. Estas calidades difieren en el ancho de banda que consumen y en el tamaño de pantalla en el que se proyectan. Así, la calidad módem consume un ancho de banda de 56kbps y se presenta en un formato de 160x120. La calidad banda ancha tiene actualmente 280x160 *pixels* de resolución y consume un ancho de banda de 90kbps. Por último, la calidad plus tiene una resolución de 320x240 y necesita un ancho de banda medio de 200 kbps. Las características de las calidades banda ancha y plus han ido variando desde el inicio del servicio debido al proceso de análisis y configuración reflejado en este artículo.

Fig. 1. Visualización de un vídeo de la sección de Turismo

Actualmente, el servicio multimedia de la Nueva España cuenta con 187 vídeos disponibles en todas las calidades.

4 Proceso de Análisis

El proceso de análisis llevado a cabo en este trabajo se ha dividido en dos partes independientes aunque claramente relacionadas. La primera de las partes se ha centrado en desarrollar una metodología de análisis y configuración del sistema para proveedores de servicio. Esta metodología se ha ido contrastando con la evolución del sistema real y su objetivo es mejorar la calidad del servicio y aumentar la satisfacción del usuario. La otra parte del proceso se ha centrado en la elaboración de un modelo de prestaciones del servicio. Su objetivo es el de permitir al operador de comunicaciones predecir el comportamiento del sistema en condiciones diferentes a las del sistema actual y poder configurar el servicio bajo previsiones fiables. Actualmente, el modelo se centra en la elaboración de previsiones en cuando al consumo de anchos de banda, tanto en el servidor como en los usuarios, si bien, en un futuro próximo se afrontará el estudio de otros parámetros de servicio.

Para conseguir un proceso de análisis continuo se ha desarrollado una herramienta que a partir de los datos proporcionados por el servidor (o servidores) realiza diversos estudios de forma automática. Los tests realizados por la herramienta serán acordes con los análisis diseñados por la metodología propuesta, y además proveerán los datos necesarios para configurar el modelo del servicio.

5 Herramienta Desarrollada

En esta sección se presentarán las principales características de la herramienta que se ha desarrollado para realizar el análisis del servicio de forma automática. Esta herramienta es un potente instrumento capaz de generar tanto informes gráficos mediante su interfaz web, como informes textuales. Actualmente la herramienta está capacitada para trabajar sobre los *logs* de los servidores de streaming de la familia Real (Real Server y Helix), aunque se pretende que un futuro próximo sea capaz de trabajar con otros sistemas como: WindowsMedia y QuickTime.

5.1 Arquitectura de la Herramienta

La herramienta desarrollada presenta una arquitectura compleja como se muestra en la figura 2. Está compuesta por varios módulos, donde el elemento central será una base de datos donde se almacenará la información extraída de los *logs* de los servidores a los cuales se esté monitorizando. La base de datos será cargada con información mediante dos módulos de adquisición dependiendo del tipo de carga, *on-line* o *off-line*. Los informes generados por la herramienta serán de dos tipos: unos gráficos exportados a través de una página web y otros textuales para posibilitar su utilización en otro tipo de herramientas.

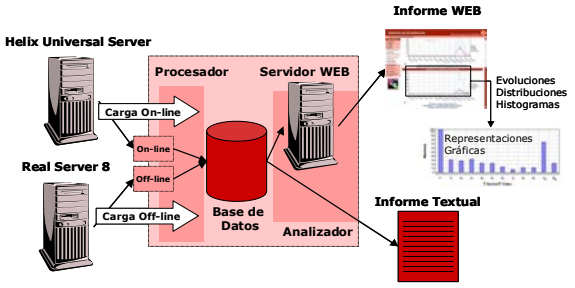


Fig. 2. Herramienta de Análisis

5.2 Proceso de Carga

El sistema de análisis permite dos modos de carga de la base de datos. El primero de ellos es la carga en línea (*on-line*). Este método solamente es posible con servidores de *streaming* de la gama Helix Universal Server. Mediante la utilización de un socket es posible recibir la información sobre los accesos de los usuarios según se producen. Este modo de carga hace que el sistema de análisis disponga de información actualizada en todo momento. El otro método de carga es el modo fuera de línea (*off-line*), este método es posible utilizarlo con cualquier servidor con tecnología RealNetwork. En este caso la herramienta dispone de un módulo de procesamiento para los ficheros de *log* del servidor. Este proceso se realiza fuera de línea, si bien, puede automatizarse para que se ejecute diariamente y de forma incremental. Mediante un proceso automático se incorporarán a la base de datos los accesos producidos en el día.

5.3 Información Procesada

El sistema de análisis obtiene la información de los *logs* del servidor de vídeo Real Server [15,16]. Este provee información como: dirección IP del cliente, identificador único del reproductor del cliente, instante de realización de la petición, bytes enviados, tiempo de vídeo transmitido, número y tipo de las interacciones del usuario, paquetes enviados, paquetes reenviados, ancho de banda medio consumido, etc.

5.4 Resultados Proporcionados

La herramienta de análisis proporciona numerosos análisis e informes. La información puede ser proporcionada en formato gráfico para su análisis visual o en formato texto para introducirlas en otras herramientas con propósitos más específicos. Dentro de los numerosos informes generados cabe destacar los siguientes:

- **Distribución y evolución de los accesos**. Proporciona información sobre la evolución de los accesos, y la distribución tanto en horarios como en días de la semana. Este análisis es posible realizarlo tanto por secciones como por calidades, incluso por vídeos individuales.

Fig. 3. Análisis individual de una sesión

- **Análisis de la sesión**. Provee información sobre la caracterización de una sesión de un usuario. Si se tiene en cuenta que una sesión se inicia con la petición de reproducción de un vídeo y termina con un *stop* o con la finalización del vídeo, este informe presentará los periodos de *play* o actividad y los periodos de pausa o inactividad. Este análisis es capaz de proveer información de tipo resumido (general o por secciones) e información de cada una de las reproducciones realizadas por los usuarios como la que se muestra en la figura 3.

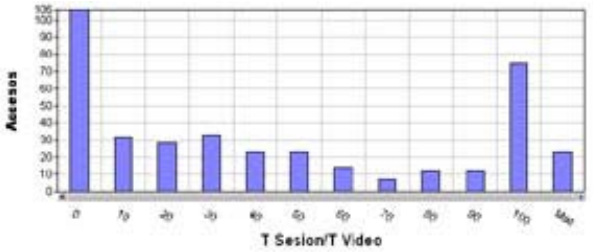


Fig. 4. Histograma de tiempo de vídeo transmitido

- **Análisis del tiempo de vídeo transmitido**. Este informe generará un histograma, que se puede ven la figura 4, con la duración de las reproducciones realizadas por los usuarios en función del tiempo de duración del video. De

este modo será posible conocer si un vídeo es generalmente visualizado hasta el final, si sus usuarios abandonan de forma prematura o si por el contrario ven zonas varias veces (hacen retrocesos en la línea de reproducción). Este análisis está acompañado del de abandonos rápidos que indicará el porcentaje de usuarios que abandonan la reproducción antes de los 10 segundos.

- **Análisis de las interacciones**. Con este análisis se pretende averiguar cuantas interacciones hacen los usuarios y de que tipos son. Además, se pretende situarlas dentro de la línea de reproducción.

- **Análisis de la popularidad**. Este análisis tratará de averiguar si el servicio cumple la ley de Zipf generalizada con diferentes parámetros θ.

- **Análisis de fallos y calidad de la reproducción**. Se estudiarán en este análisis la evolución del número de paquetes perdidos, el número de paquetes de audio retasados y el número de paquetes que ha sido necesario reenviar. Además, se calculará la evolución del número de peticiones donde el tiempo de vídeo transmitido es cero, ya que indicarán que el usuario no disponía de un reproductor adecuado para visualizar el vídeo.

- **Análisis de fidelidad**. Este análisis indicará la fidelidad que los usuarios tienen al servicio. Presentará ordenados los usuarios por el número de accesos realizados. Mediante la observación de la curva presentada será posible saber si el número de usuarios es grande pero realizan pocas visualizaciones, si el número de usuarios es pequeño pero estos son muy fieles, y cuantos usuarios hay de cada uno de esto tipos.

Es importante destacar que la herramienta realiza además informes sobre el consumo de ancho de banda, y los usuarios que acceden al servicio y sus plataformas.

6 Metodología de Análisis para el Proveedor de Servicio

Los elementos en los que un proveedor puede actuar cuando crea un servicio multimedia son fundamentalmente dos: la información qué ofrece y cómo ofrece esa información. Con los análisis del sistema actual debe darse cuenta de qué es lo que funciona bien en el sistema y qué debe de ser mejorado. Para conseguir este objetivo se ha desarrollado una metodología con los tests que es necesario realizar para permitir al proveedor extraer conclusiones acerca del funcionamiento del servicio y proceder a mejorarlo si es necesario. Para que los tests de la metodología se puedan realizar de forma automática, la herramienta de análisis desarrollada se ha adaptado para proveer la información necesaria. Los tests que conforman la metodología son los siguientes:

- **Test de reproducciones fallidas**. Este test calcula el número y el porcentaje de visualizaciones con cero segundos de reproducción. A partir del mismo el proveedor sabrá el número de visualizaciones que han fallado porque el usuario no disponía de un reproductor adecuado para reproducir el vídeo. Para corregir esta situación del proveedor deberá situar algún tipo de aviso en las páginas a través de las cuales se solicita el vídeo, aunque lo más efectivo resulta realizar una comprobación del *plug-in*.

- **Test de abandonos rápidos**. Este test calcula el número y el porcentaje de reproducciones donde el tiempo de vídeo transmitido es menor de 10 segundos. Por debajo de este tiempo se considera que el usuario no tiene interés en el vídeo. Se pueden extraer dos conclusiones fundamentalmente: no interesa la información o la calidad de la reproducción es inaceptable. Para discernir entre las dos opciones estará el test de calidad de la reproducción. En el caso que la información no sea interesante deberá considerarse el abandono de esa temática.

- **Test de calidad de la reproducción**. Este test calculará el número y el porcentaje de reproducciones en las que existen paquetes perdidos, retrasados o fuera de orden. Con los datos obtenidos aquí, será posible discernir si los usuarios abandonan la reproducción por la mala calidad de acceso. De ser así, el problema puede ser solventado tratando con los operadores de red en caso de estar el problema centrado en alguno en concreto o reducir el ancho de banda necesario para transmitir el vídeo.

- **Test de adecuación de la duración**. Este test pretende analizar si la duración elegida para un vídeo es la adecuada. Es difícil detectar si el vídeo se ha quedado corto, es decir, si el usuario esperaba más información de la que se le ha dado. Lo que sí es posible, es detectar si la duración ha sido excesivamente larga. Comparando la duración del vídeo con el tiempo que el usuario lo ha estado reproduciendo será posible analizar esta situación. En caso de observar que los usuarios nunca suelen llegar al final se puede llegar a la conclusión de que la duración es excesivamente larga.

- **Test de máximo interés**. Este test intenta analizar si una información ha resultado de mucho interés para los usuarios. Serán varios los elementos que harán ver este interés. Obviamente el número y la calidad de las de las visitas darán una buena referencia. Sin embargo,

este análisis será completado con dos elementos adicionales. El primero de ellos será si los usuarios realizan saltos hacia atrás en la reproducción, indicando con ellos que cierta parte les interesa. El segundo punto será el número de reproducciones que un mismo usuario realiza sobre un mismo vídeo. La conclusión será la misma que en el caso de los saltos hacia atrás.

- **Test de fidelidad y caracterización del usuario**. Este test trata de comprobar el número de usuarios distintos que tiene el servicio y el número de reproducciones que ha realizado cada uno. Con ello se pretenden saber si los usuarios son fieles al servicio o por el contrario ven un único vídeo y no vuelven a utilizar el servicio. También se intentará determinar si los usuarios ven videos de muchas secciones, con qué frecuencia acceden al servicio, etc.

7 Modelo de Prestaciones

El operador de comunicaciones donde se sitúa el servicio puede actuar sobre algunos elementos para que éste funcione con más calidad. El ancho de banda de salida de los servidores, la capacidad de procesamiento de los mismos, las arquitecturas de servicio y los routings, y el ancho de banda con que los clientes se conectan a la red, son los elementos que se pueden configurar y para ello debe disponerse de información al respecto. Poseer una buena herramienta de análisis permite que se puedan observar el número de accesos simultáneos, el ancho de banda consumido por los mismos y su duración. Sin embargo, lo ideal es disponer de un modelo que permita simular comportamientos en condiciones extraordinarias, y analizar la evolución en el número y el tipo de usuarios. Para desarrollar este tipo de modelos es necesario además de un profundo conocimiento del funcionamiento del servicio y de los protocolos utilizados, disponer de una caracterización de la carga sistema y del usuario que accede al servicio.

7.1 Diseño del Modelo

Este trabajo ha afrontado como parte final de la parte de análisis tocante al operador de red, el desarrollo de un modelo de simulación basado en redes de colas. La herramienta utilizada para su implementación ha sido el entorno de simulación **MODLINE** basado en el lenguaje de modelado **QNAP2**. Una simplificación del modelo desarrollado es el que se presenta en la figura 5.

El modelo caracteriza el comportamiento de un sistema de vídeo *streaming* que utiliza el protocolo RTSP [17]. En él, los clientes realizan peticiones de vídeos que les son transmitidos bajo demanda desde el servidor (o servidores).

El cliente del servicio se ha modelado mediante dos subsistemas, uno capaz de generar peticiones (*play, pause, seek, stop*) al servidor siguiendo el comportamiento de un usuario real. El otro subsistema se encarga de la recepción de los paquetes de vídeo y su presentación en pantalla. El servidor se ha modelado de igual modo mediante dos subsistemas. El primero de ellos está dedicado a procesar las peticiones de los usuarios. En función de estas peticiones, la sesión de los usuarios podrá estar en dos estados, transmitiendo o en pausa. Si el estado es transmitiendo el subsistema de producción transmitirá los *frames* de vídeo a los usuarios.

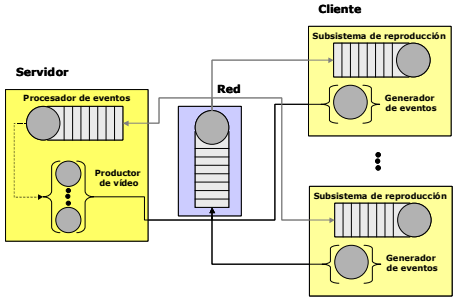


Fig. 5. Modelo desarrollado

La figura 5 se presenta un modelo sencillo para la red de comunicaciones. El modelo final que aquí se desarrolla tiene como misión integrarse en el modelo de red HFC [18], el cual permite a los *managers* de una red de cable probar diferentes configuraciones de forma sencilla.

7.2 Caracterización de la Carga

Fig.6. Menú de configuración de la carga

La carga del servicio puede ser configurada mediante los menús de la herramienta gráfica por el usuario, como se muestra en la figura 6. Los parámetros configurables son los siguientes:

- Ancho de banda de los vídeos. Existen 3 categorías: módem, banda ancha y plus, a las que el usuario puede asignar un valor en Mbps.

- Número de vídeos en las diferentes categorías según su longitud. Los vídeos están divididos en 3 segmentos según su duración: cortos (30'-2:30), medianos (2:30'-5:00'), largos (5:00-1:30:00). Estos vídeos son distribuidos normalmente en cada uno de los intervalos.

7.3 Caracterización del Usuario

El comportamiento de los usuarios se ha definido basándose en el que tienen los usuarios de sistemas reales y que es el que se muestra en la figura 7.

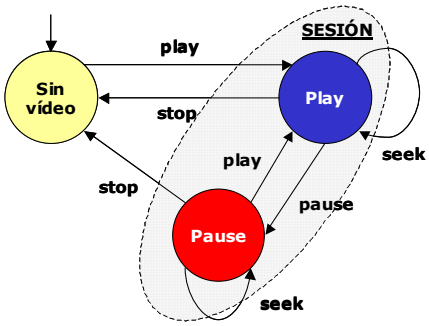


Fig. 7. Comportamiento del usuario

Los parámetros que determinan el comportamiento del usuario han sido configurados a partir de los datos extraídos del sistema real. El primero de ellos ha sido el tiempo entre peticiones. Este se ha modelado mediante una distribución Weibull también observada en [6]. Los otros dos parámetros son los tiempos de play (o tiempo de vídeo transmitido) y de pause. En el caso del tiempo de play es de especial relevancia la forma de la distribución seleccionada para representar este parámetro. A diferencia de resultados extraídos en artículos como [6], en los que se utilizaban distribuciones simples (lognormal, Weibull, gamma) o simples a las que se les incorporaban colas pesadas, los resultados obtenidos a partir del análisis del servicio de La Nueva España sugieren la utilización de dos distribuciones combinadas. Como se puede apreciar en la figura 4, es posible representar el tiempo de vídeo transmitido mediante una primera distribución de tipo Weibull o Gamma para la primera parte, y mediante una distribución Lognormal la segunda. Este comportamiento se atribuye a los dos tipos de usuarios que pueden afrontar la visualización de un vídeo, aquellos que entran a probar y rápidamente pierden el interés, y aquellos a los que el tema les atrae y ven prácticamente el vídeo entero. El peso de las dos distribuciones en la final varía según la longitud del vídeo, tomando mayor protagonismo las de la primera parte (Weibull, Gamma) cuanto más largo es el vídeo. El tiempo de pause, que tiene poca relevancia en las reproducciones analizadas, se ha representado con una distribución exponencial.

7.4 Resultados proporcionados

El modelo actualmente proporciona fundamentalmente 2 resultados, el ancho de banda consumido a la salida del servidor y el ancho de banda consumido por los usuarios.

8 Resultados Obtenidos

8.1 Para el proveedor de servicio

Han sido numerosos los parámetros que proveedor de servicio ha modifica desde la puesta en marcha del sistema de análisis. Algunos de los más relevantes son los siguientes:

- Se ha observado que existía un abundante número de reproducciones con 0 segundos de vídeo transmitido, por lo que se procedió a instalar un detector de *plug-in* en la página de acceso, lo que ha permitido disminuirlas de forma muy importante.

- Se detectó que los usuarios de las calidades banda ancha y plus perdían abundantes paquetes de audio, por lo que se procedió a disminuir el ancho de banda consumido de 110 a 90kbps en el caso de banda ancha y de 400 a 200kbsp en el caso del plus.

- Se observó una gran tasa de abandonos en los vídeos cuya duración superaba los 15 minutos. Para subsanar esta situación se procedió a la disminución de la duración de los nuevos vídeos generados.

Con las modificaciones aplicadas el número de visitas diarias del sistema se ha multiplicado por 10 en los últimos meses. También se ha conseguido aumentar el número de usuarios que han realizado más de 20 reproducciones. En lo que se refiere a la calidad de las visitas se ha logrado disminuir el número de abandonos prematuros y se ha aumentado el porcentaje de visualización de los vídeos.

8.2 Para el operador de red

El servicio multimedia de la Nueva España ha mostrado desde su creación una tendencia ascendente en cuanto al número de visitas. Sin embargo, la evolución más importante desde el punto de vista del operador de red es el aumento de la cantidad de video transmitido en cada una de esas reproducciones. El número de reproducciones fallidas y el de abandonos prematuros se ha disminuido a favor de reproducciones más largas. Mediante la utilización del modelo se ha concluido que el ancho de banda a la salida del servidor debía de ser aumentado. La aplicación prematura de esta medida ha permitido evitar periodos de baja calidad del servicio.

9 Conclusiones

Después de 5 meses de utilización del sistema de análisis sobre el servicio de vide bajo demanda de La Nueva España se puede decir que se ha conseguido una importante mejora de su calidad. Se han orientado los nuevos contenidos para adaptarse mejor a las características demandas por los usuarios, tanto en duración, como temática y ancho de banda

necesario; y se han realizado previsiones del consumo de recursos (ancho de banda) para evitar periodos de calidad deficiente.

10 Trabajos Futuros

A pesar de haber conseguido un gran avance en la calidad del servicio, son numerosos los elementos que deben ser añadidos a este proceso de análisis. Tanto la metodología de análisis y configuración para el proveedor, como la herramienta debe aumentar el número de análisis en el campo de la distribución de peticiones.

El modelo desarrollado actualmente proporciona importantes datos de consumo de ancho de banda en los clientes y en los servidores. Sin embargo, no permite la utilización de arquitecturas de servicio complejas, mediante la utilización de *proxies*. Además, el objetivo último de este modelo es el de ser incorporado al modelo de red desarrollado en proyectos anteriores, lo que permitiría analizar el tráfico generado en los diferentes puntos de la red.

Por último es necesario confeccionar una metodología de análisis para el operador de red. A medida que el número de análisis que se pueden realizar con el modelo aumente será necesario estructurarlos mediante una metodología.

Agradecimientos

Esta investigación ha sido financiada por el operador de comunicaciones **Telecable de Asturias SAU** el periódico **La Nueva España**.

Referencias

[1] C. Chuma, A Bestavros and M. Crovella. "Characteristics of WWW client-based traces". Technical Report BU-CS-95-010, Computer Science Department, Boston University, Abril 1995.

[2] M Arlitt and C. Williamson. " Web Server Workload Characterization: The Search for Invariants". ACM Sigmetrics, Filadelfia, EEUU. Diciembre 1996.

[3] P. Barford, A. Bestavros, A. Bradley and M. Crovella. "Changes in Web Client Access Patterns: Characteristics and Caching Implications". World Wide Web Journal. Diciembre 1998.

[4] C. Griwodz, M. Bär, Lars C. Wolf: Long-term Movie Popularity in Video-on-Demand System, ACM Multimedia. Seattle. 1997.

[5] D. Loguinov, H. Radha: Measurement Study of Low-bitrate Internet Video Streaming, ACM SIGCOMM Internet Measurement Workshop. Noviembre 2001.

[6] Jussara M. Almeida, Jeffrey Krueger, Derek L. Eager, Mary K. Vernon: Analysis of Educational Media Server Workloads, NOSSDAV 2001, Port Jefferson, NY, Junio 2001.

[7] M. Chesire, A. Wolman, G. Voelker, H. Lavy: Measurement and Analysis of a Streaming-Media Workload, USENIX Symposium on Internet Technologies and Systems. Marzo 2001.

[8] Eric W. Wong, V. Lee, K. Ko, K Tang: Multimedia-on-Demand System, ICC2001 IEEE International Conference on Communications. Helsinki. Julio 2001.

[9] E. Veloso, V. Almeida, W Meira, A. Bestavros, S. Jin: A Hierarchical Characterization of a Live Streaming Media Workload, Internet Measurement Workshop. Marsella. Noviembre 2002.

[10] J. R Arias, F. J. Suárez, D. F. García, X. G. Pañeda V. G. García. "Evaluation of Video Server Capacity with Regard to Quality of the Service in Interactive News-On-Demand Systems". Protocols and Systems for Interactive Distributed Multimedia. Coimbra, Portugal, Noviembre 2002.

[11] S. Jin, A. Bestavros: GISMO, A Generator of Internet Streaming Objects and Workloads, ACM SIGMETRICTS. Noviembre 2001.

[12] J. R. Arias, F. J. Suárez, D. F. García, X. G. Pañeda, V. G. García. "A Set of Metrics for Evaluation of Interactive News-on-Demand Systems". ACM International Multimedia Conference. Juan les Pins, Francia. Diciembre 2002.

[13] http://www.analog.cx/

[14] http://www.sane.com

[15] RealNetworks: RealServer 8 Administration Guide. Octubre 2000.

[16] RealNetworks: Helix Universal Server Administration Guide. Julio 2002.

[17] H. Zchulzinne, A. Rao and R. Lanphier. RFC 2326: Real Time Streaming Protocol (RTSP). Abril. 1998.

[18] M. García, X. G. Pañeda, D. F. García, V. G. García, J. R. Arias. "Modeling and Performance Evaluation of an HFC Network Operator". Protocols for Multimedia Systems. Cracovia, Polonia. Octubre 2000.

Modelado y Análisis Transitorio de un Sistema de Distribución de Vídeo bajo Demanda

Isabel Martín[1, 2], Juan José Alins[1], Mónica Aguilar[1], Reinaldo Vallejos[2], Luis J. De la Cruz[1], Jorge Mata[1]

[2]Departamento de Electrónica. Universidad Técnica Federico Santa María.
Avenida España 1680, Valparaíso, Chile.
E-mail: reinaldo@elo.utfsm.cl

[1]Departamento de Ingeniería Telemática. Universitat Politècnica de Catalunya
C/ Jordi Girona 1-3, Mód. C3, Campus Nord, 08034 Barcelona.
E-mail: {isabelm, juanjo, maguilar, ljcruz, jmata}@mat.upc.es

Abstract*. The dramatic growth of Internet and the need of certain Quality of Service (QoS) guarantees of the video-streaming applications make necessary to ensure end-to-end QoS. This paper presents a tool to evaluate the performance of a video-streaming system. This tool provides an estimated QoS given to the user in the connection moment, thus the user can decide to accept or to reject its service demand. Moreover, the system could charge the clients for a tightly service, meanwhile making efficient use of the available bandwidth. In this paper, the analyzed video-streaming architecture is composed of three elements: the MPEG video encoder, the traffic shaper and the supervisor and controller subsystem. It supplies smoothing and control functions to provide a guaranteed QoS over an open broadband network like Internet.*

1 Introducción

El fuerte crecimiento que ha experimentado Internet en los últimos años ha llevado al desarrollo de nuevas aplicaciones con requisitos de Calidad de Servicio (QoS) de diversos grados, según el tipo de aplicación desarrollada y el servicio ofrecido. La actual Internet sólo ofrece un servicio *best-effort*, insatisfactorio como infraestructura comercial en la que se está transformando. Como ejemplo, cada vez son más los usuarios dispuestos a pagar por obtener servicios multimedia bajo demanda y casi bajo demanda, con requisitos de tiempo real y con cierto grado de QoS. Para poder distribuir eficientemente estos flujos multimedia y proveer QoS, durante los últimos años se han desarrollado muchos protocolos, en especial los que proporcionan soporte a los requisitos de tiempo real de las aplicaciones.

Una de las aplicaciones que se han desarrollado con más énfasis, debido a la demanda y buena aceptación entre el conjunto general de usuarios, son las aplicaciones para la transmisión y distribución de flujos multimedia (video y audio) bajo demanda para su reproducción en tiempo real (*video-streaming*).

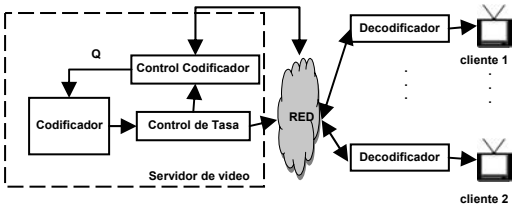


Figura 1. Sistema de distribución de video para su reproducción en tiempo real (*video streaming*).

En la Fig. 1 se presenta el esquema de un sistema *video streaming*. El servidor de video dispone de un conjunto de películas, que pueden ser solicitadas por cualquiera de los clientes. Si la solicitud de conexión es aceptada, se da origen a un flujo de información a través de la red, desde el servidor de video hasta el cliente. Cada fuente de video (película que está siendo transmitida por el servidor de video) genera una gran cantidad de información, lo que hace necesario usar técnicas de compresión, como se representa con el bloque "Codificador" de la Fig. 1. El estándar MPEG-2 es ampliamente reconocido para la codificación de flujos que se transmiten sobre redes de comunicación. Este tipo de flujo presenta una gran variabilidad, producida por factores como el algoritmo de codificación utilizado y por la propia complejidad de las secuencias a codificar. Si la codificación se realiza en modo tasa variable (*Variable Bit Rate*, VBR) se dificulta la gestión de la red, dada la característica de gran variabilidad en la tasa de transmisión del tráfico VBR. En general, una forma de asegurar un cierto grado de calidad de servicio, durante todo el tiempo de conexión, es asignar los recursos necesarios para la transmisión de los instantes de máxima exigencia del tráfico, en el momento de conexión. Para el tráfico VBR esta opción no es eficiente desde el punto de vista de la red, debido a que durante gran parte del tiempo de transmisión la conexión no estará en la condición de máxima exigencia y, por lo tanto, esos recursos estarán siendo subutilizados. Dada la alta exigencia de recursos de los flujos de video, el número de conexiones compartiendo los recursos es pequeño, por lo que la multiplexación estadística pierde su beneficio. Por este motivo, surgen propuestas para controlar la tasa de codificación, y asignar los recursos de red en forma dinámica. Tienen especial interés aquellas aplicaciones capaces de adaptar su tasa de transmisión en función del estado de la red, reduciendo la calidad del flujo emitido en caso de congestión (bloque "Control Codificador" de la Fig. 1).

En la literatura se pueden encontrar diversas propuestas para modelar y evaluar el rendimiento de este tipo de sistemas, tanto en régimen transitorio como estacionario. El objetivo principal es poder predecir la influencia que tiene el tráfico generado por las aplicaciones, en los recursos de la red y la calidad de servicio que se le entrega al usuario. Dos propuestas de especial interés para el desarrollo del presente trabajo, son [1] y [2]. En la primera, se propone un modelo de Proceso de Fluido Modulado por Markov (*Markov Modulated Fluid Process*, MMFP) bidimensional para modelar el tráfico MPEG-VBR. Este modelo es el proceso agregado de dos tipos de minifuentes ON-OFF, que permiten capturar la característica de dependencia a corto y largo plazo del tráfico de video. Y en la segunda, en base a este modelo, se propone un método de asignación dinámica de recursos de la red.

Para evaluar las prestaciones del sistema de distribución de video, se propone utilizar herramientas que provienen del área de investigación denominada Prestabilidad (*Performability*), concepto que integra los aspectos de Prestaciones y Disponibilidad (*Performance* y *Dependability*) [4]. En particular, se propone utilizar un modelo de Cadenas de Markov con Recompensas (CMR) y la técnica de Uniformización (o randomización). Esta metodología se utiliza ampliamente en el área de Prestabilidad; por ejemplo, en [3, 8] se aplica a modelado de tráfico, y en [6] para evaluar un algoritmo de control de admisión de conexión.

Una de las medidas que permite obtener esta metodología es una estimación de la QoS que el usuario percibe durante el tiempo en que se le está ofreciendo el servicio. Una utilidad de esta medida es entregar al usuario una estimación de la QoS que le puede ofrecer el sistema desde ese momento hasta que finalice la transmisión de la película solicitada. Este servicio puede ser muy atractivo para el usuario, dado que puede aceptar o rechazar la conexión según sea la QoS que se le ofrece. Por otro lado, desde el punto de vista del proveedor, ésta medida puede permitir un servicio de tarificación variable al usuario y por tanto ajustada al servicio ofrecido.

Para los sistemas en que está implementado algún método de reserva de recursos, la aceptación de una conexión supone el compromiso, por parte de la red, de mantener una cierta calidad al usuario acotando el valor de algunos parámetros de QoS, es decir de unas medidas objetivas (retardo, pérdidas, etc) que no reflejan todas las métricas de calidad de video (VQM, *Video Quality Metrics*) del flujo transmitido. Debido al mecanismo de renegociación de recursos implementado en el sistema evaluado, en este trabajo es relevante obtener una medida de la calidad de codificación del flujo entregado, dado que el sistema emite flujos de diferentes calidades a lo largo de la transmisión. La valoración final de QoS puede corresponder a una función que relacione diferentes medidas objetivas de VQM con la percepción subjetiva que tiene el usuario (VQM subjetivas).

El resto de este trabajo está organizado de la siguiente manera: La sección 2 presenta el modelo que describe el funcionamiento del sistema a evaluar. En la sección 3 se resuelve el modelo mediante herramientas de evaluación de prestabilidad. A continuación, en la sección 4 se presentan algunos resultados numéricos contrastados con mediciones experimentales, obtenidas en la plataforma del proyecto SSADE [7] implementado por el grupo de investigación Servicios Telemáticos de la Universidad Politécnica de Cataluña. Finalmente, en la sección 5 se presentan las conclusiones.

2 Modelo del sistema

El sistema que se modela y analiza en este trabajo, considera las características del sistema de distribución de vídeo bajo demanda SSADE [7]. El flujo de información se codifica con el algoritmo MPEG-2 en modo VBR (*Variable Bit Rate*). Para reducir la dificultad de la red en transmitir este tipo de tráfico de forma eficiente, se agrega al sistema un mecanismo de suavizado del flujo y un mecanismo de asignación dinámica de recursos de la red.

Se dispone de más de un flujo de video, codificado con diferente valor del parámetro de cuantización (Q) y, por tanto, con diferente calidad. Ello permite variar la tasa de transmisión según los recursos disponibles en la red. Se supone que todos los flujos multimedia están almacenados en el servidor de video, por lo que se tiene un completo conocimiento estadístico de su comportamiento. Esto permite obtener los valores empíricos que se requieren para definir los parámetros del modelo de tráfico de cada flujo. Así mismo, permite obtener una métrica de la calidad del flujo, que en el modelo propuesto se considera constante para cada secuencia VBR. En esta sección se describen los modelos de tráfico y de asignación dinámica de recursos de red utilizados para representar el funcionamiento del sistema global que se ha evaluado.

2.1 Modelo de Tráfico

El modelo de tráfico de una fuente MPEG VBR que se ha considerado en este trabajo es un modelo de fluidos modulado por un proceso de Markov (*Markov Modulated Fluid Process*, MMFP) bidimensional, tal como se propone y analiza en [1]. Este modelo se compone de la agregación de dos tipos de fuentes simples ON-OFF.

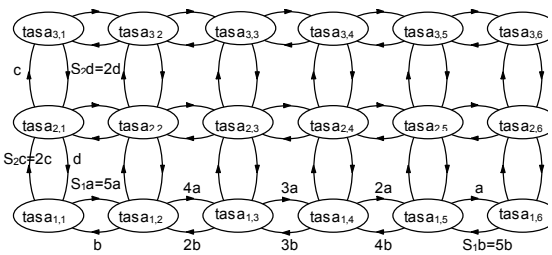


Figura 2a. Modelo de tráfico.

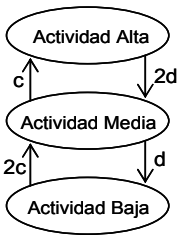


Figura 2b. Modelo del nivel de actividad.

En la Fig. 2a se muestra el diagrama de este modelo, donde cada estado corresponde a un valor de la tasa de transmisión de la secuencia de video. Las transiciones entre estados ocurren cuando se activa o desactiva alguna de las minifuentes ON-OFF. El tiempo entre transiciones de cada minifuente está distribuido exponencialmente. S_1 y S_2 son la cantidad de minifuentes de cada tipo y, como se demuestra en [1], en general se obtiene un buen ajuste de cualquier secuencia de video MPEG VBR con S_1 igual a 5 (o 6) y S_2 igual a 2. En la Fig. 2b se muestran tres niveles de actividad (*Alto*, *Medio*, *Bajo*) para el tráfico de video MPEG VBR (determinados por el estado de las dos minifuentes de tipo 2, S_2). El tiempo de permanencia en estos tres niveles de actividad captura la relación temporal a largo plazo de la generación de tráfico y viene condicionado por las tasas de transición de las minifuentes de tipo 2 [2]. La transición entre estos niveles de actividad modela los posibles cambios significativos de actividad que se producen en una secuencia de video.

Uno de los parámetros de codificación que más influye en la tasa media del flujo, es el parámetro de nivel de cuantización, Q. En [5] se comprueba empíricamente que al comparar las características de una secuencia de video codificada con distintos Q, la forma de los flujos resultantes (subidas y bajadas del nivel de actividad) se mantiene muy semejante entre un flujo y otro, siendo la tasa media del flujo mayor cuanto menor es el valor del parámetro Q. Esto quiere decir que en el modelo MMFP bidimensional de una secuencia de video, el valor de la tasa de transmisión de cada uno de los flujos obtenidos de codificar esa secuencia con distinto Q, sólo se verá modificado por un factor proporcional y las tasas de transición entre estados se considera que son las mismas.

Para garantizar un determinado nivel de QoS en la transmisión de un flujo completo, con una asignación estática de recursos de la red, el sistema debe asignar los recursos que necesite el nivel *Alto* de actividad de dicho flujo. La mayor parte del tiempo el flujo estará en el nivel *Bajo* de actividad, eventualmente saltará al nivel *Medio* y en pocas ocasiones saltará al nivel *Alto* de actividad. Por lo tanto, la identificación de los intervalos de tiempo durante los cuales el flujo está en cada uno de esos tres niveles de actividad, permite llevar a cabo un mecanismo de asignación más eficiente de los recursos de red. Esta asignación es dinámica, según el mecanismo de renegociación de recursos de red propuesto en [2], que se describe a continuación.

2.2 Asignación Dinámica de Recursos de la Red

Con el objetivo de asegurar un determinado nivel de calidad en la transmisión de un flujo de información por la red, se implementan mecanismos o protocolos que permiten hacer una reserva de recursos de la red. Cada aplicación determina cuándo y qué parámetros enviará al protocolo de reserva de recursos para solicitar recursos de red.

El sistema reserva los recursos necesarios para limitar las pérdidas y el retardo sufridos en la transmisión. Debido a la exigencia de mínimo retardo que requieren los servicios de distribución de video en tiempo real (longitudes de memorias pequeñas y fijas) en este estudio se considera que los recursos necesarios de red sólo están condicionados por el ancho de banda (BW) que el servicio de video reserva para la conexión del usuario.

Para aceptar un nuevo usuario, se necesita determinar la cantidad de recursos (BW) requeridos para la nueva conexión y si el sistema es capaz de proporcionárselos. Este BW está limitado por la tasa de la red de acceso a la que está conectado el terminal del usuario y por el BW que facilita la red al sistema de distribución de video. El primero es un dato propio del usuario, que debe comunicar al sistema en el momento en que solicita la conexión. El sistema rechaza la conexión del nuevo usuario si su tasa de conexión es inferior al mínimo BW requerido por la conexión. Si el usuario dispone de suficiente BW en la red de acceso para recibir el servicio de video y tiene contratada reserva de recursos, el sistema invocará al protocolo de reserva de recursos. El bloque "Control de Codificación" de la Fig. 1 controla la petición de recursos de la red y el valor de los parámetros de codificación. El sistema formulará nuevas peticiones a la red para incrementar gradualmente la calidad del flujo transmitido.

Para transmitir una película con reserva de recursos, el sistema acepta la conexión del nuevo usuario si puede asignarle suficiente BW para asegurar la transmisión del flujo menos exigente de la película solicitada. Este mínimo BW requerido es igual a la tasa del nivel más alto de actividad del flujo de menor tasa media de la película solicitada (el flujo de menor calidad de codificación). El mecanismo de reserva de recursos implementado es el propuesto en [2], que en líneas generales sigue las siguientes pautas de funcionamiento:

- El sistema acepta la conexión de un usuario si puede hacer la reserva de recursos requerida para asegurar la transmisión del flujo de video menos exigente (menor tasa media, menor calidad de codificación) disponible de la película solicitada.
- Durante la transmisión, el sistema siempre intenta mejorar la calidad del flujo a transmitir (mayor tasa media, menor Q de codificación). Para esto, realiza una renegociación de reserva

cada cierto intervalo de tiempo (la longitud adecuada de este intervalo depende de las características del flujo) solicitando los recursos necesarios para transmitir el flujo de la siguiente mejor calidad disponible (en el mismo nivel de actividad en que se encuentra).

- Si el flujo de la película en curso cambia a un nivel de actividad más alto, el sistema procede a enviar un flujo de calidad igual o inferior, bajará la calidad hasta que los recursos que tiene asignado en la reserva actual sean suficientes.
- Si el flujo de la película en curso cambia a un nivel de actividad más bajo, puede seguir enviando el flujo con la misma calidad, ya que requiere menos recursos que los asignados actualmente. En caso de que estos recursos asignados también sean suficientes para transmitir el flujo de la siguiente mejor calidad, puede aumentar de calidad. En cualquiera de los dos casos, los recursos excedentes se liberan inmediatamente.

Nótese que este sistema reserva recursos a cada conexión en la medida en que están disponibles, siempre intentando mejorar el servicio, pero también liberando los recursos que el usuario no está utilizando.

2.3 Modelo para una conexión

En esta sección se presenta el modelo analítico para el funcionamiento del sistema de distribución de video descrito, que permite obtener medidas de su rendimiento.

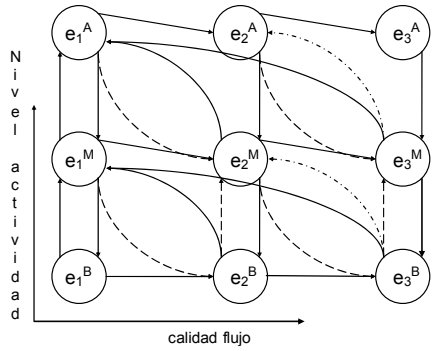


Figura 3. Modelo comportamiento 1 conexión en el sistema.

En la Fig. 3 se presenta el diagrama de estados que modela el comportamiento de una conexión (flujo de video) aceptada en el sistema, y se muestran las transiciones debidas al comportamiento de una determinada conexión.

Se considera que en cada conexión se puede estar transmitiendo un flujo entre 3 posibles calidades de codificación, cada una con 3 posibles niveles de actividad. Sea e_f^a un estado del sistema, que representa la situación en la cual la conexión en observación está transmitiendo un flujo de calidad *f* (1: baja, 2: media, 3: alta) y en nivel de actividad *a* (B: bajo, M: medio, A: alto). Sea $P_{RS}(e)$ un factor que indica la posibilidad de que el sistema asigne a la conexión (en el momento de la renegociación) recursos suficientes para estar en el estado *e*. Sea $P'_{RS}(e_f^a, e_{f'}^{a'})$ un factor que indica si los recursos ya asignados a la conexión (en el estado e_f^a) son suficientes para que la conexión transite al estado $e_{f'}^{a'}$. Por lo tanto, $P'_{RS}(e_f^a, e_{f'}^{a'}) = 1$ si los recursos requeridos para que la conexión transmita el flujo de calidad *f* y nivel de actividad *a* son mayores o iguales a los requeridos para que la conexión transmita el flujo de calidad *f'* y nivel de actividad *a'*; y $P'_{RS}(e_f^a, e_{f'}^{a'}) = 0$ en caso contrario.

En el sistema, el tiempo entre renegociaciones está distribuido exponencialmente con una tasa λ_R. Las tasas de activación y desactivación de las minifuentes ON-OFF que modelan el tráfico, son *c* y *d* respectivamente. Sea $\lambda(e_1, e_2)$ la tasa de transición desde el estado e_1 al estado e_2. A continuación se definen las tasas de transición $\lambda(e_1, e_2)$ de la Fig. 3, para $P_{RS}(e_1) = 1$.

Las transiciones ocurridas en los instantes de renegociación son:

$\lambda(e_f^a, e_{f+1}^a) = \lambda_R \cdot P_{RS}(e_{f+1}^a)$, para $a = A, M, B$ y $f = 1, 2$.

Las transiciones ocurridas por una disminución en el nivel de actividad del flujo son:

$\lambda(e_f^A, e_{f+1}^M) = 2 \cdot d \cdot P'_{RS}(e_f^A, e_{f+1}^M)$ y

$\lambda(e_f^A, e_f^M) = 2 \cdot d \cdot \overline{P'_{RS}}(e_f^A, e_{f+1}^M)$, para $f = 1, 2$.

$\lambda(e_f^M, e_{f+1}^B) = d \cdot P'_{RS}(e_f^M, e_{f+1}^B)$ y

$\lambda(e_f^M, e_f^B) = d \cdot \overline{P'_{RS}}(e_f^M, e_{f+1}^B)$, para $f = 1, 2$.

$\lambda(e_3^A, e_3^M) = 2 \cdot d$ y $\lambda(e_3^M, e_3^B) = d$.

Las transiciones ocurridas por un aumento en el nivel de actividad del flujo son:

$\lambda(e_1^B, e_1^M) = 2 \cdot c$, $\lambda(e_1^M, e_1^A) = c$, $\lambda(e_2^M, e_1^A) = c$,

$\lambda(e_2^B, e_2^M) = 2 \cdot c \cdot P'_{RS}(e_2^B, e_2^M)$,

$\lambda(e_2^B, e_1^M) = 2 \cdot c \cdot \overline{P'_{RS}}(e_2^B, e_2^M)$,

$\lambda(e_3^B, e_3^M) = 2 \cdot c \cdot P'_{RS}(e_3^B, e_3^M)$,

$\lambda(e_3^B, e_2^M) = 2 \cdot c \cdot \overline{P'_{RS}}(e_3^B, e_3^M) \cdot P'_{RS}(e_3^B, e_2^M)$,

$\lambda(e_3^B, e_1^M) = 2 \cdot c \cdot \overline{P'_{RS}}(e_3^B, e_3^M) \cdot \overline{P'_{RS}}(e_3^B, e_2^M)$,

$\lambda(e_3^M, e_2^A) = c \cdot P'_{RS}(e_3^M, e_2^A)$, $\lambda(e_3^M, e_1^A) = c \cdot \overline{P'_{RS}}(e_3^M, e_2^A)$.

Nótese que tanto $P_{RS}(e)$ como $P'_{RS}(e_1, e_2)$ no son probabilidades, sino factores que permiten expresar en forma general el modelo, y que dependen de los recursos totales del sistema y las características particulares de los flujos, respectivamente. Los factores $P_{RS}(e)$, indican la existencia o no del estado *e* en el sistema; es decir, $P_{RS}(e) = 0$ significa que el sistema no puede estar en el estado *e* porque nunca

existen los recursos suficientes y $P_{RS}(e)=1$ significa que el sistema puede transitar al estado *e* (cuando sucede una renegociación) porque existen los recursos suficientes para aceptar la reserva de recursos. Asimismo, en los casos en que una transición ocurre hacia un nivel más bajo de actividad los factores $P'_{RS}(e_1,e_2)$ indican el hecho de que los recursos actualmente asignados son suficientes para transmitir un flujo de mayor calidad y en un nivel de actividad más bajo. Esta situación se puede presentar por dos motivos: Por un lado, los recursos requeridos para transmitir cierto nivel de actividad pueden ser numéricamente mayor o igual a los requeridos para transmitir un nivel más bajo de actividad del flujo en una calidad superior (indicados en la Fig. 3 con líneas segmentadas descendientes); y por otro lado, el sistema reserva (como mínimo) los recursos requeridos para transmitir el flujo de menor calidad en el nivel de máxima actividad. Este último motivo también da lugar a las posibles transiciones hacia niveles más altos de actividad en la misma calidad de flujo. Por último, cuando hay un aumento en el nivel de actividad de un flujo, se pasa a transmitir el flujo de la mejor calidad posible que permitan los recursos asignados actualmente.

2.4 Modelo para N Conexiones

El modelo de la sección anterior (Fig. 3) permite entender el comportamiento del flujo de una sola conexión en el sistema. Ahora, se propone el modelo del sistema para las *N* conexiones aceptadas en el sistema.

Sea $S = \{(n_1^B, n_1^M, n_1^A),\ (n_2^B, n_2^M, n_2^A),\ (n_3^B, n_3^M, n_3^A)\}$ el estado del sistema donde cada componente (n_f^a) es la cantidad de conexiones transmitiendo un flujo de calidad *f* y en nivel de actividad *a*. Por lo tanto, si hay *N* conexiones aceptadas dentro del sistema se tiene que: $\sum_{\forall a}\sum_{\forall f} n_f^a = N$. Sea $S_{+(f,a)-(f',a')}$ un estado del sistema en que se tiene una conexión más respecto del estado *S*, trasmitiendo un flujo de calidad *f* en nivel de actividad *a*; y una conexión menos, trasmitiendo un flujo de calidad *f'* en nivel de actividad *a'*.

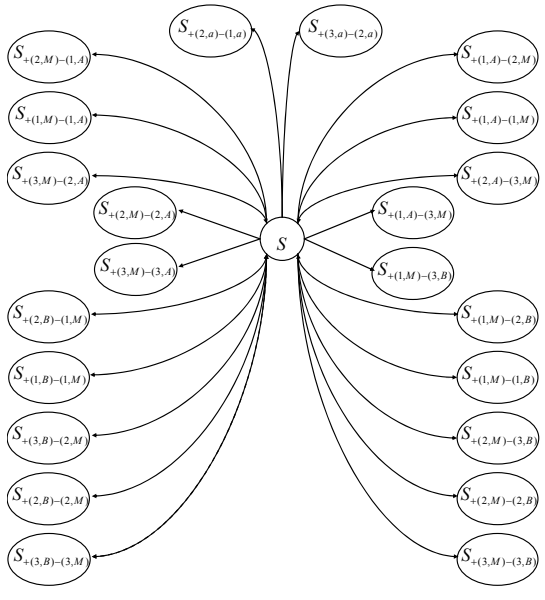


Figura 4. Modelo comportamiento *N* conexiones en el sistema.

Si los recursos totales del sistema son suficientes para que las *N* conexiones estén transmitiendo la máxima calidad en el nivel de actividad máximo, esto es el estado *S* = {(0,0,0), (0,0,0), (0,0,*N*)}, el espacio de estados del sistema es de $(N+8)!/(N!8!)$. En general, la máxima cantidad de estados del sistema es $(N+3m_f-1)!/(N!(3m_f-1)!)$, donde m_f es la cantidad de diferentes calidades de flujo. Dada la complejidad de dibujar el espacio de estados completo del sistema, en la Fig. 4 se representan las transiciones desde y hacia un estado *S* cualquiera del sistema. Los dos estados de salida de *S* ubicados en la zona superior de la Fig. 4, $S_{+(2,a)-(1,a)}$ y $S_{+(3,a)-(2,a)}$ (*a* = *B*, *M*, *A*), representan los estados a los que transitaría el sistema cuando se acepta una renegociación de reserva de recursos. Los estados ubicados en la zona derecha de la Fig. 4 corresponden a los cambios de estado debido a que alguna de las conexiones ha subido de actividad. Y los estados en la zona izquierda, corresponden a los cambios de estado debido a que el tráfico de alguna de las conexiones ha bajado de nivel de actividad.

En las tasas de transición del modelo, que se propone a continuación, se ha considerado que todas las conexiones están transmitiendo tráfico que tiene las mismas tasas de transición de las minifuentes ON-OFF que lo modelan (*c, d*: tasas de activación y desactivación de una minifuente, respectivamente). Entonces, las tasas de transición desde el estado *S* (con $P_{RS}(S)=1$) a los estados $S_{+(f1,a1)-(f2,a2)}$, $\lambda(S, S_{+(f1,a1)-(f2,a2)})$, son las siguientes.

Tasas de transición del estado *S* a los estados posteriores a una renegociación aceptada:

$$\lambda(S, S_{+(2,a)-(1,a)}) = n_1^a \cdot \lambda_R \cdot P_{RS}(S_{+(2,a)-(1,a)}),$$
$$\lambda(S, S_{+(3,a)-(2,a)}) = n_2^a \cdot \lambda_R \cdot P_{RS}(S_{+(3,a)-(2,a)}),\ a = A,M,B.$$

Tasas de transición del estado *S* a los estados en que una de las conexiones ha disminuido la actividad del flujo:

$$\lambda(S, S_{+(2,M)-(1,A)}) = n_1^A \cdot 2 \cdot d \cdot P'_{RS}(e_1^A, e_2^M),$$
$$\lambda(S, S_{+(1,M)-(1,A)}) = n_1^A \cdot 2 \cdot d \cdot \overline{P'_{RS}}(e_1^A, e_2^M),$$
$$\lambda(S, S_{+(3,M)-(2,A)}) = n_2^A \cdot 2 \cdot d \cdot P'_{RS}(e_2^A, e_3^M),$$
$$\lambda(S, S_{+(2,M)-(2,A)}) = n_2^A \cdot 2 \cdot d \cdot \overline{P'_{RS}}(e_2^A, e_3^M),$$
$$\lambda(S, S_{+(3,M)-(3,A)}) = n_3^A \cdot 2 \cdot d,$$
$$\lambda(S, S_{+(2,B)-(1,M)}) = n_1^M \cdot d \cdot P'_{RS}(e_1^M, e_2^B),$$
$$\lambda(S, S_{+(1,B)-(1,M)}) = n_1^M \cdot d \cdot \overline{P'_{RS}}(e_1^M, e_2^B),$$
$$\lambda(S, S_{+(3,B)-(2,M)}) = n_2^M \cdot d \cdot P'_{RS}(e_2^M, e_3^B),$$
$$\lambda(S, S_{+(2,B)-(2,M)}) = n_2^M \cdot d \cdot \overline{P'_{RS}}(e_2^M, e_3^B),$$

$\lambda(S, S_{+(3,B)-(3,M)}) = n_3^M \cdot d$.

Análogamente, tasas de transición del estado *S* a los estados en que una de las conexiones ha aumentado la actividad del flujo:

$\lambda(S, S_{+(1,A)-(2,M)}) = n_2^M \cdot c$, $\lambda(S, S_{+(1,A)-(1,M)}) = n_1^M \cdot c$,

$\lambda(S, S_{+(2,A)-(3,M)}) = n_3^M \cdot c \cdot P'_{RS}(e_3^M, e_2^A)$,

$\lambda(S, S_{+(1,A)-(3,M)}) = n_3^M \cdot c \cdot \overline{P'_{RS}}(e_3^M, e_2^A)$,

$\lambda(S, S_{+(1,M)-(3,B)}) = n_3^M \cdot 2 \cdot c \cdot \overline{P'_{RS}}(e_3^B, e_3^M) \overline{P'_{RS}}(e_3^B, e_2^M)$

$\lambda(S, S_{+(1,M)-(2,B)}) = n_2^B \cdot 2 \cdot c \cdot \overline{P'_{RS}}(e_2^B, e_2^M)$,

$\lambda(S, S_{+(1,M)-(1,B)}) = n_1^B \cdot 2 \cdot c$,

$\lambda(S, S_{+(2,M)-(3,B)}) = n_3^B \cdot 2 \cdot c \cdot \overline{P'_{RS}}(e_3^B, e_3^M) P'_{RS}(e_3^B, e_2^M)$

$\lambda(S, S_{+(2,M)-(2,B)}) = n_2^B \cdot 2 \cdot c \cdot P'_{RS}(e_2^B, e_2^M)$,

$\lambda(S, S_{+(3,M)-(3,B)}) = n_3^B \cdot 2 \cdot c \cdot P'_{RS}(e_3^B, e_3^M)$.

Los factores $P_{RS}(e)$ y $P'_{RS}(e_1, e_2)$ pueden ser expresados de la siguiente forma:

$$P_{RS}(S) = \begin{cases} 1 & , \text{ si } \sum_{\forall f} \sum_{\forall a} n_f^a \cdot RS_f^a < RS_{total} \\ 0 & , \text{ en otro caso.} \end{cases} \quad (1)$$

$$P'_{RS}(e_f^a, e_{f'}^{a'}) = \begin{cases} 1 & , \text{ si } RS_f^a \geq RS_{f'}^{a'} \\ 0 & , \text{ en otro caso.} \end{cases} \quad (2)$$

Donde RS_f^a representa la cantidad de recursos de la red que el sistema reserva a una conexión para transmitir un flujo de calidad *f* a un nivel de actividad *a*; y RS_{total} es la cantidad total de recursos de la red que el servicio dispone para todas las conexiones. El punto restrictivo de estos recursos se encuentra en el nodo de acceso a la red. Si se desprecia el requerimiento de memoria, RS_{total} es igual a la capacidad total de BW del servicio en el nodo de acceso. Debido al requerimiento de asegurar la transmisión del flujo de mínima calidad, $RS_f^a = \max\{RS_1^A, rs_f^a\}$ donde rs_f^a es la mínima cantidad de recursos de red que una conexión requiere para transmitir un flujo de calidad *f* a un nivel de actividad *a*.

3 Solución del Modelo

En la sección 2.4 se ha establecido un modelo estocástico del sistema, específicamente una Cadena de Markov de Tiempo Continuo (CMTC), dado que el tiempo de permanencia en cada estado del sistema tiene distribución exponencial. En esta sección se presenta un método de solución de este modelo con el objetivo de obtener medidas de rendimiento del sistema.

Obtener una solución de esta CMTC mediante un conjunto de ecuaciones diferenciales, es una tarea extremadamente ardua, además del elevado costo en cálculo numérico. Como alternativa, se propone una metodología ampliamente utilizada en el área de investigación de Prestabilidad. Esto es, utilizar un modelo de Cadenas de Markov con Recompensas (CMR) y la técnica de Uniformización. En [4] se encuentra la descripción de esta metodología, que se resume a continuación.

Para modelar el sistema mediante CMR, a cada estado del sistema se le asocia una recompensa. En el caso de una CMTC, esta recompensa es la ganancia de la medida objeto de la evaluación mientras permanece en un cierto estado del sistema. Así, el valor de recompensa asociado a un estado corresponde a la medida de prestación (*performance*) del sistema por cada unidad de tiempo en que el sistema se encuentre en dicho estado. Por lo tanto, la definición de estado debe ser tal que para cada uno de estos estados se pueda asociar un único valor de tasa de recompensa.

Para resolver este modelo de CMR, una técnica frecuentemente utilizada es la uniformización, que consiste en una transformación de un problema presentado en términos de una cadena de Markov en tiempo continuo a una solución en tiempo discreto. Esta transformación facilita los cálculos numéricos para obtener soluciones transitorias. La uniformización se describe formalmente como sigue.

Considere una CMTC $\mathcal{X} = \{X(t): t \geq 0\}$ con espacio de estados finito *S* y generador infinitesimal $\mathbf{Q} = [q_{ij}]$. Sea $\mathcal{Z} = \{Z_n, n = 0, 1, \ldots\}$ una CM de tiempo discreto (CMTD) con espacio de estados finito *S* y matriz de probabilidad de transición $\mathbf{P} = \mathbf{I} + \mathbf{Q}/\Lambda$, donde $\Lambda \geq max_i\{|q_{ii}|\}$, y sea $\mathbf{N} = \{N(t): t \geq 0\}$ un proceso de Poisson de parámetro Λt el cual es independiente de $\mathcal{Z}$. Entonces, se puede interpretar que $X(t) = Z_{N(t)}$ para $t \geq 0$.

Sea $\mathbf{p}(t)$ un vector cuya *i*-ésima entrada es la probabilidad de que la CMTC $\mathcal{X}$ esté en el estado *i* en el tiempo *t* dada una distribución inicial. Sea $\mathbf{v}(n) = \mathbf{v}(0)\mathbf{P}^n$ un vector cuya *j*-ésima entrada es la probabilidad de que la CMTD $\mathcal{Z}$ esté en el estado *j* después de *n* transiciones, donde $\mathbf{v}(0)$ es el vector de probabilidad inicial. Entonces, la ecuación básica del método de uniformización es:

$$\mathbf{p}(t) = \sum_{n=0}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \mathbf{v}(n) \quad (3)$$

Para calcular (3), es necesario truncar el sumatorio infinito en un cierto número *N* de términos. Donde, para cualquier componente de $\mathbf{p}(t)$, el error está dado por $\varepsilon(N) \leq 1 - \sum_{n=0}^{N} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!}$.

Sea $\mathbf{r} = [r_1, r_2, \ldots, r_M]$ el vector de tasas de recompensas asociadas al espacio de estados *S*, y sea $ER(t)$ la recompensa esperada en el tiempo *t*. Utilizando la ecuación (3) (en [4], ec. (3.8)), $ER(t)$ se expresa como:

$$ER(t) = \sum_{n=0}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \{\mathbf{r} \cdot \mathbf{v}(n)\} \quad (4)$$

Donde $\mathbf{r}\cdot\mathbf{v}(n)=\sum_{i=1}^{M} r_i \cdot v_i(n)$. Sean M(t) la recompensa acumulada en el intervalo (0, t) y $EM(t)$ el valor esperado de M(t). A partir de (4) y simples argumentos probabilísticos se puede demostrar que $EM(t)$ es (en [4], ec. (3.10)):

$$EM(t)=t\sum_{n=0}^{\infty} e^{-\Lambda t}\frac{(\Lambda t)^n}{n!}\left[\frac{\sum_{j=0}^{n}\{\mathbf{r}\cdot\mathbf{v}(n)\}}{n+1}\right] \quad (5)$$

Donde $f(n)=\frac{\sum_{j=0}^{n}\{\mathbf{r}\cdot\mathbf{v}(n)\}}{n+1}$, puede ser calculado como $f(n)=\frac{n}{n+1}f(n-1)+\frac{\{\mathbf{r}\cdot\mathbf{v}(n)\}}{n+1}$.

Otra medida de interés es la función de distribución (FDP) de $M(t)$, $\Pr\{M(t)\le m\}$, que en la literatura es la más conocida como medida de *performability*. El cálculo eficiente de esta ecuación no es tan simple como el de (5). En [9] se presenta la solución general que actualmente mejor realiza este cálculo en forma eficiente y numéricamente estable.

En resumen, modelando el sistema con CMR y utilizando la técnica de uniformización, se obtiene un método simple para obtener más de una medida de rendimiento del sistema, simplemente definiendo adecuadamente la estructura de recompensas del sistema. Por ejemplo, a continuación se presenta una de las medidas de interés en el sistema de nuestro análisis, esto es la QoS definida por la calidad de codificación del flujo que se está transmitiendo al usuario.

3.1 Medida de QoS de un usuario en el sistema

No es objetivo de este trabajo definir una función de relación entre medidas objetivas y subjetivas de VQM. Sin embargo, algunos parámetros cuya influencia en la calidad del video que percibe el usuario es conocida, son la pérdida de paquetes y el retardo de los mismos, que en el caso de reserva de recursos son parámetros acotados. El parámetro que principalmente influirá en la degradación producida por la codificación del video, es el parámetro de cuantización Q (el cual influye directamente en la relación señal-ruido, PSNR [5]). Por lo tanto, una buena aproximación de esta QoS es la determinada por la PSNR (o el parámetro de cuantización (Q)) del flujo codificado, o bien una función que relacione más de una VQM objetiva en un valor de VQM subjetiva.

Antes de definir la estructura de recompensas que nos permitirá obtener esta medida de QoS con un modelo de CMR, se hará una integración de los modelos presentados en las secciones 2.3 y 2.4. Sea $\vec{S}=\{u_f^a, S\}$, donde u_f^a describe el estado de la conexión de un usuario en observación, esto es, la conexión está transmitiendo un flujo de calidad f en un nivel de actividad a, y S describe el estado del resto de las conexiones en el sistema (ver definición de S en la sección 2.4). Las tasas de transición de un estado $\{u_f^a, S\}$ a un estado $\{u_f^a, S'\}$ son las definidas en la sección 2.4, y las tasas de transición de un estado $\{u_f^a, S\}$ a un estado $\{u_{f'}^{a'}, S\}$ son las definidas en la sección 2.3. Esta definición permite asociar un único valor de recompensa a cada estado para la medida en particular de QoS de un usuario dado.

Así, a cada estado $\vec{S}=\{u_f^a, S\}$ se le asocia una recompensa $r(\vec{S})$ igual a la valoración de la calidad de codificación del flujo de calidad f, Q_f.

Sea $\mathrm{QoS}_i(T)$ la QoS entregada al usuario, acumulada durante el intervalo T (tiempo duración de película) de duración de la conexión, dado que el sistema le puede asignar los recursos mínimos necesarios y al inicio de su conexión el estado del sistema es i. Entonces se estima el valor medio de QoS que el sistema podría ofrecerle durante el siguiente intervalo de tiempo T, esto es el cálculo del valor esperado de $\mathrm{QoS}_i(T)/T$. Se considera que el número de usuarios total dentro del sistema se mantiene constante durante T. Finalmente, se interpreta el valor numérico de $E\{\mathrm{QoS}_i(T)/T\}$ con una calificación cualitativa (por ejemplo: Platino, Oro, Plata o Bronce) para informar al usuario.

4 Resultados numéricos

En los ejemplos numéricos que se presentan a continuación, se ha evaluado el sistema con 4 calidades de flujo (Q = 17, 10, 6 y 4) de la secuencia completa de la película *Matrix*. Los valores de los parámetros del modelo de tráfico de cada uno de estos flujos han sido obtenidos empíricamente y a priori ya que son propios de la secuencia. Así mismo, a cada flujo se le ha asociado como valor de QoS (recompensa en el modelo CMR) el valor medio de PSNR medido a lo largo de la secuencia. La Fig. 5 muestra la traza de los 4 flujos en un intervalo de la secuencia. La traza resaltada en negrilla es el flujo efectivamente transmitido por el sistema, que se

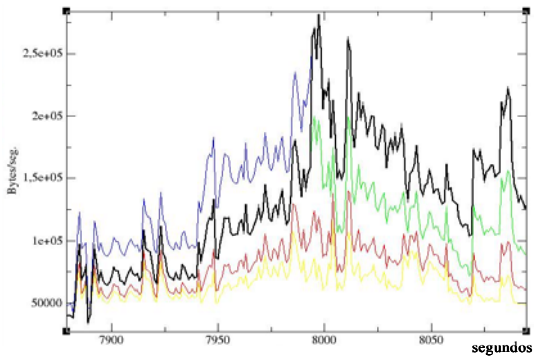


Figura 5. Traza real de película *Matrix* codificada en MPEG2.

corresponde con una u otra calidad de flujo durante distintos intervalos.

La Fig. 6 representa la QoS obtenida en función de la cantidad de recursos disponibles. La λ_R considerada

es igual a 1 renegociación/seg.. El mínimo BW necesario para aceptar la conexión es de 1'732 Mbps. (igual a la tasa máxima del flujo de peor calidad). El máximo BW representado corresponde a la tasa máxima del flujo de mejor calidad. El valor de QoS representado está normalizando respecto de los valores de PSNR máximo y mínimo que ofrece el sistema. Se aprecia como la QoS estimada aumenta a medida que se incrementan los recursos disponibles. Las discontinuidades se producen en aquellos valores de recursos de red que pasan a ser suficientes para un cambio a un flujo de mejor calidad. En concreto, cerca de un valor de 2'5 Mbps de BW disponible hay un cambio significativo en la QoS ofrecida (alcanza el nivel de actividad bajo –donde la secuencia permanece habitualmente- de un flujo de calidad superior).

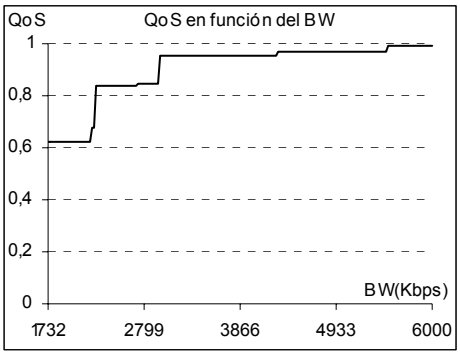


Figura 6. QoS estimada a 1 conexión en función del BW disponible en el servidor de video.

La Fig. 7 representa la QoS obtenida en función de la tasa media de renegociación del algoritmo de asignación dinámica de recursos. El BW total considerado es igual a 3Mbps. Se observa que a partir de un valor de λ_R=0'048 renegociaciones/seg. se consigue una QoS cercana al 90%.

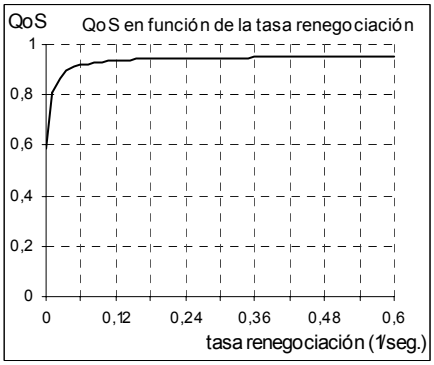


Figura 7. QoS estimada a una conexión en función de la Tasa de Renegociación en el servidor de video.

5 Conclusiones

En este trabajo se ha propuesto un mecanismo de evaluación de las prestaciones de un sistema *video-streaming* que tiene implementado un mecanismo de asignación dinámica de recursos de red. La QoS ofrecida al usuario depende principalmente de la calidad del flujo transmitido (medida por la PSNR de la imagen), degradada por el proceso de codificación en función del parámetro Q. Para obtener medidas de rendimiento del sistema, se ha propuesto un modelo analítico basado en CMR. La solución propuesta ha sido programada en ordenador. En particular se ha obtenido la QoS definida por la calidad de codificación del flujo que se está transmitiendo al usuario. Se ha validado el modelo propuesto contrastando con los valores obtenidos en la plataforma SSADE, obteniéndose valores muy semejantes. Cabe señalar que se encuentra en desarrollo la extensión del modelo para ser incorporada la variación de N durante el tiempo de transmisión (llegada y salida de usuarios del sistema).

Agradecimientos

Este trabajo ha sido financiado por los proyectos de investigación DISQET (CICYT TIC2002-00818), CREDO (CICYT TIC2002-00249), y proyecto FONDECYT (1000055/2000-CONICYT).

Referencias

[1] L. J. De la Cruz, M. Fernández, J. Alins, J. Mata "Bidimensional Fluid Model for VBR MPEG Video Traffic". Broadband Communications: The future of telecommunications, IFIP 1997.

[2] L. J. De la Cruz, J. Mata, "Performanace of Dynamic Resource Allocation with QoS Guarantees for MPEG VBR Video Traffic Transmission over ATM Networks". Proc. Of IEEE GLOBECOM99, Brasil, 1999.

[3] E. de Souza e Silva, R. M. M. Leao, M. Diniz. "Transient Analysis Applied to Traffic Modeling". ACM SIGMETRICS Performance Evaluation Review, Vol. 28, I. 4, Marzo 2001.

[4] B. R. Haverkort, R. Marie, G. Rubino, K. Trivedi (editores), "Performability Modelling. Techniques and Tools", John Wiley & Sons, ISBN: 047149195-0, 2001.

[5] J. Mata. "Contribución a la Gestión Dinámica de Recursos Aplicada al Control de Fuentes de Video de Velocidad Variable en la Red Digital de Servicios Integrados de Banda Ancha". Tesis Ph.D, U. Politécnica de Cataluña, 1996.

[6] J. F. Meyer, "Performability of an Algorithm for Connection Admission Control", IEEE Transac. on Computers, Vol.50 Nº 7: 724-733, 2001.

[7] SSADE, Sistema Seguro de Acceso y Distribución Eficiente de Servicios Multimedia, CYCIT TEL 99-0322, http://ssade.upc.es.

[8] B. Sericola. "Transient Analysis of Stochastic Fluid Models". Publication interne nº 1099, INRIA, Campus de Beaulieu, 35042 Rennes Cédex, France, Avril 1997.

[9] H.Nabli and B. Sericola. "Performability Analysis: A New Algorithm". IEEE Transaction on Computers, vol. 45, Nº 4, April 1996.

Arquitectura Peer-to-Peer escalable para la distribución eficiente de live-streaming en redes IP

Alberto Mozo, Joaquín Salvachúa
Departamento de Arquitectura y Tecnología de Computadores,
Departamento de Ingeniería de Sistemas Telemáticos.
Universidad Politécnica de Madrid
E-mail: amozo@eui.upm.es, jsalvachua@dit.upm.es

Abstract. *Live-streaming distribution presents scalability problems at the source and in the network. Usually, application level multicast solutions construct implicitly or explicitly a spanning tree to distribute the stream, but the necessary live-streaming bandwidth B bps is usually high and the intermediate tree nodes must be capable of forwarding N B bps (where N is the average number of children in the tree). In Peer-to-Peer communities the user is usually connected with xDSL technologies and the upload bandwidth is a fraction of the download bandwidth. Consequently, it is not feasible to have a big number of nodes ready to forward N B bps. We propose a Peer-to-Peer scalable solution in which all the participating nodes in the multicast tree collaborate in the relay streaming process to forward B bps on average. Our solution builds a tree list overlay that allows taking advantage of the small or large fractions of upload bandwidth from all the nodes that participate in the streaming distribution. Likewise, we propose a control network overlay architecture that allows creating interconnected subnetworks generated as the number of peers grows, with the purpose to guarantee the scalability in the construction of the tree lists.*

1. Introducción

La distribución de streaming es un caso concreto de la distribución multicast en la que un emisor envía información a un número variable de receptores. La característica específica de la distribución de streaming como problema de distribución multicast es que solo existe una fuente y que el flujo de información que se envía es constante y está sometido a temporización en la entrega. Actualmente la distribución de streaming en redes IP se realiza utilizando conexiones punto a punto entre el emisor y los receptores en aquellos lugares donde no hay posibilidad de despliegue multicast. Por otro lado en aquellas redes donde si existe capacidad de despliegue multicast, es la red la encargada de reenviar de la forma mas eficiente posible el flujo desde el emisor a los receptores. Lógicamente la primera solución no escala desde el punto de vista tanto del emisor como de la red de transporte. IP Multicast [1] si escala desde el punto de vista del emisor y la red, ya que evita la duplicación de paquetes en la red al mínimo necesario. Sin embargo y pasados una decena de años desde su propuesta inicial, hay al menos tres cuestiones que están frenando la implantación global de esta solución. Por un lado el escalado de los *routers* no es sencillo debido a la información de estado que deben almacenar por cada grupo multicast existente. Por otro, el control de congestión y la fiabilidad no están resueltos de forma satisfactoria aunque existen algunas propuestas al respecto (MTCP, PGMCC, SRM, RMTP). Y finalmente, no existe un modelo claro de tarificación para el tráfico multicast.

Como intento de solución a este problema, en los últimos años se ha desarrollado un esfuerzo de investigación importante trasladando el problema de la distribución multicast desde la capa de red a la capa de aplicación. Es lo que se conoce como *Multicast de nivel de aplicación*. Como el nombre indica, la duplicación de paquetes para alcanzar a varios receptores y el consiguiente reenvío se efectúa desde las aplicaciones situadas en los equipos finales. Este esquema escala perfectamente desde el punto de vista del emisor, pero no llega a ser tan eficiente en cuanto a transmisión de información como IP Multicast. La ventaja fundamental es que el despliegue de multicast a nivel de aplicación no necesita ningún compromiso adicional por parte de la red. El despliegue se realiza desde los equipos finales configurando un *overlay* sobre la red existente.

Actualmente las soluciones existentes para distribución de streaming mediante multicast de nivel de aplicación implementan de forma explícita o implícita un árbol de expansión (*spanning tree*) para distribuir los datos de forma eficiente [2], [3], [4], [5] y [6].

Estos árboles tienen dos problemas fundamentales:

- Los nodos finales (hoja) no contribuyen al reenvío de información, y por lo tanto son exclusivamente nodos consumidores.
- Los nodos intermedios tienen que tener por término medio una capacidad de envío de información de $N\,B$ bps, donde N es el numero medio de hijos de un nodo en el árbol de distribución y B es el ancho de banda del streaming a transmitir.

Estos dos factores condicionan el uso de las soluciones actuales en entornos reales donde los nodos intermedios no tienen porque tener tanta capacidad de envío, y además los nodos hoja si tienen una capaci-

dad de envío que aunque pequeña se está desaprovechando. Si consideramos un árbol de profundidad *K*, anchura *N*, y nodos hoja con una capacidad de envío *b*, *se* está desperdiciando un ancho de banda N^K *b* bits/sg, disponible en las hojas del árbol. Con *N* y *K* suficientemente grandes este valor no es despreciable.

Solución TLP y SON

Proponemos que cada nodo contribuya por término medio con una capacidad de envío de *B*, de tal manera que entre todos los participantes se alcance, al menos, el ancho de banda necesario.

Este tipo de solución encaja perfectamente en un escenario de comunidades Peer-to-Peer (P2P), donde los equipos están conectados mediante accesos xDSL en los que el ancho de banda de subida suele ser una fracción del de bajada y no es factible disponer de un número grande de nodos dispuestos a reenviar esos *N B bps*. La fracción de ancho de banda de subida de los equipos conectados a xDSL se puede utilizar para contribuir al reenvío del flujo global. Aproximadamente, si los nodos hoja son capaces de enviar una fracción φ ([0..1]) de *B*, entonces los nodos intermedios solo necesitan enviar por término medio $[N(1-j)+j]B$ bits/sg.

Nuestro sistema de distribución basado en una lista de *K* árboles le llamaremos TLP (Tree-List P2P). En cada árbol se distribuye una fracción φ del flujo total *B*. Un nodo participante X, asume la responsabilidad de reenviar el flujo *J-esimo* participando como nodo intermedio en el árbol J. Asimismo el nodo *X*, participa como nodo hoja (receptor) en el resto de árboles. De esta forma, fragmentando el flujo *B* en pequeños flujos, la mayor parte de los nodos P2P participantes en la distribución aportan en mayor o menor medida su capacidad de envío para distribuir el flujo.

Existe un compromiso entre el número de subflujos y la información de control que circula para la creación y mantenimiento de los árboles. Cuanto mayor es el número de árboles, menor es el subflujo a transmitir y por lo tanto más nodos pueden colaborar con pequeñas aportaciones a la transmisión del flujo *B*. Cuanto menor es el número de árboles, menor es la información de control necesaria para crear y mantener la estructura.

Colateralmente consideramos la participación de nodos *altruistas*, colaboradores que sin estar interesados en la recepción del flujo *B*, si pueden ayudar en el reenvío de alguno de los subflujos. Por lo tanto se engancharan solamente a uno de los árboles. También es posible conectar nodos *tacaños* que no quieran o no puedan reenviar subflujos debido a la no disponibilidad de ancho de banda de subida. Lógicamente, estos nodos tendrán la menor prioridad posible de cara a engancharse al sistema de distribución.

Para que el proceso de construcción de la lista de árboles sea escalable, definimos una red de control SON (Simple Overlay Network) que permita agrupar y localizar a los distintos nodos participantes de forma escalable. SON estructura la red en pequeñas subredes con capacidad de broadcast escalable. Un nodo *X* cualquiera, utilizando como bootstrap un nodo *A* ya enganchado a la red, busca la red a la que pertenece y se engancha a ella. Una vez dentro de ella localiza mediante broadcast a las raíces de la TLP y se incorpora a la recepción del streaming enganchándose a la lista de árboles.

El método de búsqueda es escalable y la distribución de equipos en subredes se realiza de forma homogénea independientemente del número de nodos existentes en la red. Esto quiere decir que SON funciona con rendimientos similares en modo esparcido o denso. SON tiene incorporado un mecanismo de *broadcast* escalable, que permite implementar protocolos de descubrimiento tipo ARP sin un coste de transmisión excesivo y de forma escalable. Sin embargo SON no está pensado para aplicaciones donde el objetivo sea buscar, lo mas eficientemente posible, valores asociados a una clave, y su rendimiento puede ser inferior al de otros *overlays* como Chord [9], Pastry [8], o CAN [7].

Una característica fundamental de esta red es que los equipos una vez asignados a una subred no necesitan ser recolocados a otra subred según va creciendo la red global y por lo tanto no es necesario mover información de una subred a otra.

El protocolo de funcionamiento de las TLP necesita, como muchos otros protocolos Peer-to-Peer, un mecanismo de descubrimiento de equipos. La posibilidad de mandar broadcast es la forma sencilla de resolver este problema. Lógicamente un broadcast propagado de forma indiscriminada a toda la red no es escalable en lo que el número de peers crece. Proponemos un mecanismo de broadcast progresivo que en base a un campo *Countdown*, (similar al TTL de los datagramas IP), controla su propagación a través de la red, de tal forma que el proceso de descubrimiento de equipos mediante broadcast sea escalable.

El resto del artículo se estructura de la siguiente forma. En la sección 2 describimos las listas de árboles (TLP) con sus procedimientos de unión y abandono de nodos. La sección 3 se dedica al estudio de cómo conseguir escalar el problema cuando el número de nodos es grande. La sección 4 describe la red de control SON necesaria para que las TLP escalen en su construcción. Finalmente presentamos los resultados y conclusiones en la sección 5 junto con el trabajo pendiente de desarrollar.

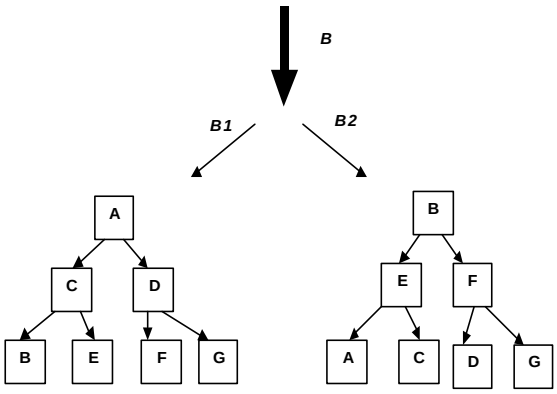


Figura 1: Distribución mediante lista de árboles

2. Listas de árboles de distribución

Las soluciones actuales de streaming mediante redes Peer-to-Peer pasan por la construcción de un overlay network sobre el que se monta un spanning tree para distribuir el flujo de *B* bits/seg. Esta solución no se puede aplicar de forma práctica en entornos heterogéneos P2P debido a que un nodo padre que distribuye a *N* nodos hijos un flujo de *B* bps, necesita como línea de subida, una que tenga un ancho de banda de *NB* bps.

Teniendo en cuenta que los nodos son ordenadores finales que posiblemente tengan limitado el ancho de banda de subida tanto por restricciones físicas (por ejemplo xDSL), como por restricciones lógicas del usuario (que no quiere gastar todo el ancho de banda de subida en distribuir el flujo a otros nodos), este tipo de soluciones no es aplicable de forma práctica en la mayor parte de los casos.

Asimismo, montar un solo árbol de distribución genera que los nodos hoja no contribuyan en el reenvío de información dentro del árbol. En un árbol en el que por término medio cada padre tiene N hijos, nos encontramos con N^K nodos hoja, siendo K la profundidad del árbol de distribución. Suponiendo que los nodos hoja colaboren en el reenvío de información, con una fracción de la capacidad de total jB ($j \in [0..1]$), el resto de nodos soportarán aproximadamente por término medio una capacidad de subida de $[N(1-j)+j]B$, menor que los *NB* de los sistemas actuales.

Definiremos un sistema de distribución basado en una overlay network sobre la que crearemos una de lista de árboles para realizar la distribución del live-streaming. El objetivo es conseguir que todos los equipos que forman el overlay de distribución del flujo, colaboren en mayor o menor medida en esta distribución.

Supongamos un grupo de tamaño *N* de ordenadores a los que hay que distribuirles en tiempo real un flujo de *B* bits/seg. Proponemos como solución una estructura basada en listas de árboles, donde el flujo a transmitir se descompone en *K* flujos, de ancho de banda $B_i = B/K$.

Construiremos *K* árboles encargados de distribuir cada uno de ellos un flujo *Bi*. Supondremos inicialmente que todos y cada uno de los equipos participantes en la distribución deben ser nodos intermedios en alguno de los *K* árboles formados. Llamaremos *nodos* P2P a estos nodos participativos. Posteriormente incorporaremos otros dos tipos de nodos: *altruistas* y *tacaños*. Los nodos *altruistas* participan como nodos intermedios en alguno de los flujos sin necesidad de recibir los *K* flujos (no son receptores). Los nodos *tacaños* participan como nodos hoja en los *K* árboles.

Por ejemplo, suponga 7 equipos a los que se debe distribuir un flujo en tiempo real de capacidad *B*. Formando un árbol de distribución tradicional, con nodos padre alimentando a dos hijos, necesitaríamos un ancho de banda de subida de *2B bits/sg* para todos los nodos intermedios, y *0 bits/sg* para los nodos hojas. Formando una lista de árboles, podríamos definir *K=2*, con flujos *B1* y *B2* de capacidad *B/2 bits/seg*. Los árboles que se formarían serían los de la **Figura 1**. De esta forma, cualquier equipo soportaría como máximo *2Bi bits/seg*, o lo que es lo mismo *B bits/seg* como capacidad de la línea de subida.

El problema radica en como construir de forma óptima los *K* árboles: asignar a un equipo a un determinado árbol y donde colocarlo. Esto influye en los *buffers* de los nodos receptores debido a que los flujos llegarán en instantes de tiempo diferentes debido a la latencia total de la transmisión por los distintos nodos de la red. La realización de este proceso debe realizarse de forma que sea escalable cuando el número de equipos *N* crezca de forma exponencial.

Para que los mecanismos de unión y abandono del árbol funcionen de forma escalable supondremos que los equipos están agrupados en un *overlay network* con capacidad de broadcast. Mas tarde desarrollaremos la arquitectura de la red de control SON (Simple Overlay Network) que permite construir las listas de árboles de forma escalable.

2.1. Parámetros de construcción de la TLP

Un nodo *X* ofrece un ancho de banda de subida *Qi*, y uno de bajada de *B bps* para poder recibir correctamente el flujo. La ecuación fundamental que da factibilidad al problema de asignación es

(Ec. 1) $$\sum_i Qi \geq (N-1)B$$

Por lo tanto por término medio los nodos deben ofrecer aproximadamente un ancho de banda de subida de

(Ec. 2) $$B_{up} = \frac{N-1}{N} B\,bps$$

El valor de K nos lo da la siguiente relación

(Ec. 3) $$K = \frac{B}{Min(Qi)}$$

La filosofía de esta fórmula es intentar por un lado no fragmentar excesivamente el flujo de partida B en un número grande de árboles que necesitan una dinámica de control y gestión de su topología, y por el otro conseguir que la mayor parte de los nodos puedan participar en el reenvío del flujo Bi a otros nodos o equipos terminales. Supondremos inicialmente que el valor de K lo prefijamos. El valor elegido tiene pros y contras, y estamos trabajando sobre ello.

Los nodos tendrán una posición distinta en cada uno de los K árboles a los que pertenecen. En K-*1*árboles serán hojas y estarán en el fondo del árbol, y en uno de los árboles estarán en una posición que puede variar entre la raíz y el nodo padre de una hoja. Por lo tanto, si la altura del árbol es excesiva un nodo puede recibir al menos uno de sus flujos retrasado en el tiempo con respecto a los demás. Este desfase se compensa con *buffering*, y lógicamente cuanto más desfase más buffer. Aun estando situado en alturas similares en dos árboles, un nodo podría recibir los dos flujos desfasados debido a distintas latencias en la transmisión. Hemos descartado este factor en una primera fase para abordarlo mas adelante.

El objetivo es construir árboles que impliquen el menor *buffering* posible en los nodos. Por lo tanto, construiremos árboles con la menor altura posible, con el objeto de que en el peor de los casos (un nodo situado como raíz en un árbol y hoja en el resto) la diferencia de tiempos en la recepción de los flujos sea mínima, obviando la variación de los tiempos de transmisión entre distintos nodos, tal y como comentamos anteriormente. Para construir árboles de altura mínima, situaremos a los nodos con mayor capacidad de transmisión en las zonas altas del árbol.

2.2. Unión con la lista de árboles

Un nodo X que quiere unirse como nodo intermedio a un árbol de distribución ofrece una capacidad de subida b, de tal forma que puede reenviar información a n nodos. Siendo

(Ec. 4) $$n = b \big/ B/K$$

Para cada uno de los n enlaces, el nodo almacena en las variables (C_n , O_n) el número máximo de nodos que puede tener debajo de él (C_n) y los nodos que actualmente tiene (O_n). Inicialmente, cuando un nodo va a unirse al árbol todos los pares (C_n , O_n) tienen el valor (1/0) para los n enlaces.

Esta información de estado sirve para conseguir un cierto equilibrio en los árboles formados, pero no es crítico que esté actualizada en cada momento.

Por lo tanto las actualizaciones se realizarán de tal forma que no consuman excesivo ancho de banda realizándose dentro de paquetes *Keep-alive*.

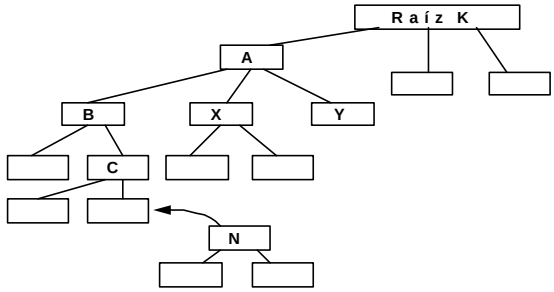


Figura 2. Escenario de unión de un nodo intermedio N.

El valor C_n sirve para equilibrar todas las ramas de un árbol. Se utiliza tanto en la unión nodos intermedios, como de nodos hoja. El nodo a unir se encamina al enlace con el menor C_n. La resta $L_n = C_n - O_n$ nos da el número de huecos libres y permite conocer si por debajo del enlace n hay hueco para enganchar a un nodo hoja ($L > 0$) .

El proceso básico de un nodo X para unirse a la lista de árboles consiste en determinar en cual de los flujos va a participar como *nodo intermedio*. Para ello transmite un paquete *hello_ALL* (broadcast) dentro de la red y las raíces de los K flujos le contestan con los valores de su capacidad total y ocupada actual ($\boldsymbol{C}, \boldsymbol{O}$) junto con su dirección.

$$\left(\boldsymbol{C} = \sum_n C_n \quad , \quad \boldsymbol{O} = \sum_n O_n \right)$$

Para dotar al sistema de cierta tolerancia a fallos, todas las raíces tienen la información de las otras raíces, y cuando contestan mandan su información junto con la del resto de raíces.

- Si no recibe contestación de ninguna raíz, supone que es el primer equipo en la red y decide ser raíz. Como no hay mas equipos, se convierte en raíz de todos los K flujos.
  Asigna toda su capacidad a uno de los árboles:
  $$\forall n \ (C_n = 1 \ , \ O_n = 0) \quad , \quad (\boldsymbol{C} = n, \boldsymbol{O} = 0)$$
  Se sitúa como raíz del resto de árboles pero con capacidad cero:
  $$\forall n \ (C_n = 0 \ , \ O_n = 0) \quad , \quad (\boldsymbol{C} = 0, \boldsymbol{O} = 0)$$
- Si recibe información de alguna de las raíces, decide unirse como nodo intermedio a la raíz que tiene un menor valor de L, siendo $\boldsymbol{L} = \boldsymbol{C} - \boldsymbol{O}$. El objetivo es tener los K árboles equilibrados en cuanto a su capacidad de absorber nuevos miembros. Al resto de los árboles se une como hoja. En el caso de que el nodo X tenga más capacidad que la raíz elegida, X manda a la raíz un paquete *ChangeRoot* y se convierte en la nueva raíz de este flujo, pasando la raíz actual a ser su hijo.

Para conectarse ya sea como nodo intermedio u hoja, X envía un paquete *Connect a* la raíz y esta va reenviando a su vez el paquete a través del árbol hasta llegar a la posición correcta. Su posición en el árbol destino vendrá determinada por su capacidad de reenvío (n), de tal manera que nunca haya un nodo con

capacidad inferior por encima de él. Existe un paquete *Status* que permite comunicar al nodo padre el valor (***C , O***) de un nodo. Este paquete se utiliza cuando un nodo detecta un cambio en la topología de alguno de los subárboles que depende de él, o cuando su padre se lo requiere explícitamente.

2.3. Abandono de la lista de árboles

El abandono de un nodo u hoja de la lista de árboles se puede producir por las siguientes causas:

1. Abandono explícito del miembro.
2. Un hijo no recibe información del padre.
3. Abandono sin aviso de un hijo.

   El padre no recibe información de control de congestión y da al hijo como desconectado.
4. Fallo de una raíz.

Caso 1:
Si un nodo abandona de forma explícita, avisa mediante el paquete *Leave* a sus padres en los *K* árboles.

Caso 2:
Si un nodo *C* no recibe información de su nodo padre *B*, se da por desconectado, avisa a la raíz del problema y pide a ésta una nueva conexión al árbol. La raíz comprueba que el padre *B* no está activo mediante la transmisión de un paquete *Ping*. Si el padre *B* no contesta, la raíz lo desconecta del árbol mandando un paquete broadcast *Disconnect* que es recibido por el A padre del nodo B. Si la raíz detecta que el padre B si que está conectado, mete la queja del nodo hijo C en una lista negra y cuando el número de quejas supera un umbral, se desconecta al nodo problemático B mediante un broadcast *Disconnect* como en el caso anterior.

Para evitar que cuando un nodo deja de recibir tráfico de su padre, sus hijos al no recibir tráfico se empiecen a desconectar también, éste les envía paquetes *Void-Traffic* de forma continua para que éstos no piensen que el nodo está caído, sino que está intentando reconectarse.

Caso 3:
Si un hijo abandona sin avisar al padre, este lo detecta porque no recibe información de control de congestión durante un lapso de tiempo prudencial. La medida a tomar es desconectar del enlace al hijo y avisar del cambio de status de los enlaces a los nodos jerárquicamente superiores al padre.

Caso 4:
La caída de una raíz *K* puede ser detectada por los hijos de ésta o por un equipo *X* que pida un *Connect*. El equipo que detecta el problema pide a otra raíz que compruebe la conectividad de la raíz problemática *K*. Si la otra raíz consigue contactar, el problema puede residir en el equipo *X*. Si la otra raíz no consigue contactar con *K* , se inicia un algoritmo de consenso distribuido para elegir a la raíz sustituta de *K*, dando prioridad a los hijos de *K*.

3. Escalado del problema

Cuando el número *N* de equipos crece la solución inicialmente propuesta carece de escalabilidad. Para evitar este problema formaremos los árboles con un número N_K de equipos (N_K << N), que no genere problemas de escalabilidad a las raíces de los árboles, y definiremos una operación de unión de árboles. Con estas dos operaciones, construiremos de forma distribuida y escalable árboles de tamaño *N*. Para desarrollar el protocolo de formación de los árboles supondremos que disponemos de:

- Una red de control (SON) que permite la de formación de subredes que agrupen un máximo de *N* equipos dentro de ellas. Estas subredes se agruparan formando una especie de árbol de expansión. SON proporcionará una primitiva tipo *attach* para engancharse a una subred concreta .

- Un procedimiento de envío de broadcast escalable de dos tipos:
  - Broadcast plano dentro de la subred (*BP*)

    El broadcast es recibido exclusivamente por los equipos que pertenecen a la subred donde se ha generado.
  - Broadcast externo que sale fuera de la subred (BE)

    Este broadcast sale fuera de la subred y se propaga de forma controlada a otras subredes, de tal manera que se recibe de forma gradual por los equipos de otras subredes.

El paquete broadcast lleva asociado un campo numérico de tipo entero llamado *Countdown*, que se decrementa cada vez que atravesamos una nueva subred.

Solo cuando atravesamos una subred y el campo llega a 0, el broadcast se transmite a los equipos de la subred. En caso contrario el broadcast se reenvía hacia todas las subredes conectadas con la subred actual. El equipo emisor solo tiene que ir mandando sucesivos broadcast con valores de *Countdown* incrementales (1,2,3 ...) para ir alcanzando equipos de otras subredes de forma gradual y escalable.

Tanto el mecanismo de formación de grupos en subredes como el broadcast (plano BP y externo BE) son claves para garantizar la escalabilidad del sistema. Además, estos mecanismos deben contemplar que el funcionamiento del sistema sea óptimo teniendo en cuenta que el número de equipos inicialmente puede ser poco denso (*sparse-mode*) para posteriormente pasar a ser muy denso (*dense-mode*). Estas características son proporcionadas por SON y las describiremos mas adelante.

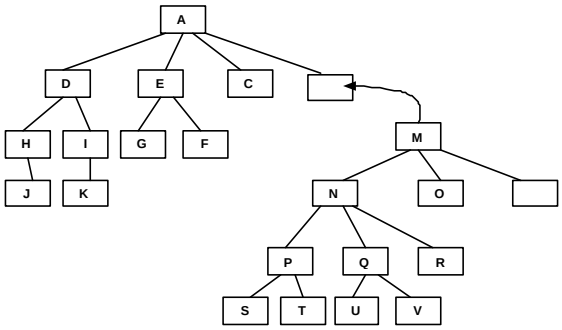


Figura 3. Unión de árboles.

3.1. Unión de árboles

Toda raíz de un árbol dejará al menos uno de sus enlaces sin ocupar para poder enganchar otros árboles (**Figura 3**). Si el número de enlaces es *n*, *j* es el número de enlaces destinados a enganchar otros árboles y *p* es el número de enlaces destinados a distribuir el *stream* entre los equipos de la subred (*n=p+j).*

Cuando un árbol necesita engancharse con otro se elige entre las raíces un *negociador*, que utilizará el broadcast externo (BE) para preguntar por negociadores de otras redes. Mediante el incremento del campo *Countdown* iremos accediendo al resto de subredes. Cuando un negociador externo se ponga en contacto con nuestro negociador, se intercambiarán información de sus respectivos árboles y elegirán quien engancha a quien.

Estamos en fase de estudio para determinar un algoritmo óptimo que conduzca a la unión de árboles con la menor altura posible. Asimismo también estamos analizando las diferencias de rendimiento variando la relación *p* y *j*.

4. Red de control: SON (Simple Overlay Network)

Para conseguir soportar un número creciente exponencialmente de equipos necesitamos agruparlos en subredes para distribuir el algoritmo. Definiremos una arquitectura de red *overlay* de control que llamaremos SON (Simple Overlay Network) que permita agregar, localizar equipos en subredes y mandar broadcast dentro de ellas con el objetivo de que los mecanismos de conexión y localización entre los equipos sean escalables.

El mecanismo de funcionamiento de SON tiene algunas semejanzas con Pastry [8], CAN [7] y Plaxton [10], pero se diferencia de éstas en que aunque se puede utilizar en los mismos entornos, su diseño está pensado teniendo en mente las TLP. Aunque los rendimientos en búsquedas son inferiores a los que se pueden conseguir con Pastry, CAN o Chord [9], la información de control transmitida y el tamaño de las tablas de los nodos es menor (CAN y Pastry). SON no necesita mover claves de un nodo a otro cuando entran nuevos nodos en la red como hace Chord o CAN. Ni Chord, ni Pastry ni CAN estructuran de forma homogénea la red en subredes independientemente del número de nodos existentes, ni implementan sobre ella un mecanismo de broadcast escalable, que permita descubrir equipos en la red.

Está fuera del ámbito de este artículo una descripción exhaustiva de los mecanismos de unión y abandono de la red. En [11] se describen detalladamente los procesos mencionados.

Sea *M* el número de equipos que se pueden conectar a la red de control. Identificaremos un equipo como raíz de la red de control. La red de control estará estructurada en subredes. Dentro de cada subred estarán localizados hasta *N* equipos, siendo *N* un parámetro de funcionamiento de SON. Las subredes las formarán los *N* equipos que pertenecen a ellas. Uno de ellos dentro de cada subred toma el papel de concentrador o representante de la red. La estructura de la red de control es un árbol de subredes.

La red estará formada en todo momento por un número variable de subredes que dependerá del número de equipos que se quieran conectar. Si suponemos una función *hash* de *D* dígitos en base *R* (supongamos decimal, R=10) cada posible valor representará una subred. La función *hash* debe generar colisiones para que exista la probabilidad de que varios equipos estén en la misma subred.

Para permitir una transición desde el modo *esparcido* al modo *denso* de funcionamiento no solo existirán las 10^D subredes, sino que además inicialmente se crearán las subredes intermedias "---", "0---", "1---", "2---",...., "00-", "01-", "02-", "10-","11-", "12-", ..., "111-", "112-", y así sucesivamente hasta llegar a las subredes finales (suponiendo *D*=4): "0000", "0001", ..., "1000", "1001", etc. En total se pueden llegar a crear: $10 + 10^2 + ... + 10^D$ subredes con *N* equipos cada una de ellas.

En la Figura 4 podemos ver una red de control parcialmente desarrollada según se han ido uniendo los miembros. Los parámetros de la red son *R=3 y D=3*. Suponiendo la existencia inicial del equipo *A0* que hace de concentrador de la subred inicial "---", los primeros *N* equipos (*A1, ..AN*) que piden unirse a la red, obtienen como resultado la pertenencia a la red "---". El equipo *C0* pide conectarse a la red. Como la subred "---" está ya llena, esta petición conducirá a la creación de una nueva subred. Dependiendo del hash que haya obtenido el nuevo equipo, se creará la "0—", "1—" o "2—".

Supongamos que el hash obtenido es "122". Este valor equipara con "1—" y por lo tanto se creará esta subred. *C0* hace de concentrador de esta nueva subred, que admitirá los N primeros equipos que tengan un hash equiparable con "1—".

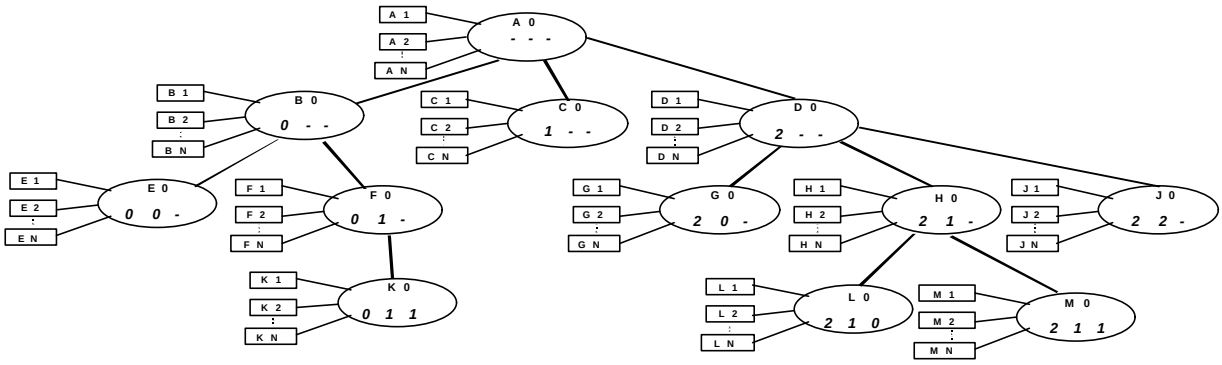


Figura 4. Ejemplo de Red de control.

Suponiendo que *B0* (con hash 010) crea la subred "0—", cuando el equipo *N+1* con hash equiparable a "0—" (*E0* con hash 002) realiza su petición, se crea la nueva subred "00-" con *E0* como concentrador.

De esta forma se llega a crear las subredes terminales, como por ejemplo "001" con *K0* como concentrador (el hash de K0 lógicamente debe ser 011). El problema que presenta esta red es que para realizar una búsqueda de un equipo hay que efectuar en el peor de los casos *D* búsquedas, pero como *D = O(LogM)*, siendo *M* el número total de nodos, el sistema escala con *M*.

4.1. Búsquedas en SON y enlaces dimensionales

La red permite a un equipo localizar la dirección del concentrador de una subred cualquiera (siempre que ésta exista). Este proceso de búsqueda en la red de control debe ser escalable. Para ello la búsqueda se podrá iniciar desde cualquier concentrador de la red.

Esto nos permite buscar cualquier equipo. Para ello, buscaremos, de forma lineal y empezando por la red terminal, el concentrador de la red cuyo hash coincide con él del equipo a buscar. Una vez que tengamos la dirección del nodo/concentrador, le mandaremos un paquete de búsqueda explicito al nodo para que nos busque el equipo que queremos localizar. Si no está en la subred terminal, buscaremos de forma lineal en las subredes que están por encima de la que acabamos de probar.

4.1.1 Búsqueda de subredes

Inicialmente hemos supuesto que los concentradores tenían enlaces con las subredes padre e hijos en la jerarquía del árbol existente. Supondremos además que todo concentrador tiene *r* enlaces ($r < R$) con otros concentradores. Llamaremos a estos enlaces *dimensionales.* Concretamente en la Figura 4, *L0* (concentrador de 210) tiene enlaces a **0**10 (siguiente de **2**10), 2**2**0 (siguiente de 2**1**0), y 21**1** (siguiente de 21**0**). Si alguna de éstas redes no existe, el enlace está vacío. En la Figura 5 se pueden ver claramente estos enlaces dimensionales.

El proceso de búsqueda se intenta realizar desde y entre los nodos (concentradores) terminales. Si el proceso de búsqueda se interrumpe debido a que falta o falla un enlace a un nodo de la misma altura (dentro del árbol), el proceso se realiza subiendo un nivel dentro del árbol. Una vez solventado ese problema tenemos dos alternativas, volver a bajar lo antes posible a un concentrador de nivel inferior y seguir realizando la búsqueda, o seguir en el nivel actual.

Una solución sencilla y que no genera ciclos de encaminamiento es no bajar hasta llegar al nodo adecuado. En otras palabras, no realizar descensos hasta llegar al nodo intermedio definitivo (aquel que es ascendente de la red que se está buscando). En esta situación no hay posibilidad de ciclos porque la búsqueda es directa.

Este es el algoritmo que inicialmente implementaremos y que puede llegar a producir congestión si un número grande de consultas empiezan a subir hacia la raíz del árbol. Si los enlaces de los nodos terminales están correctamente activados, es lógico pensar que éstas subidas deberían ser minoritarias.

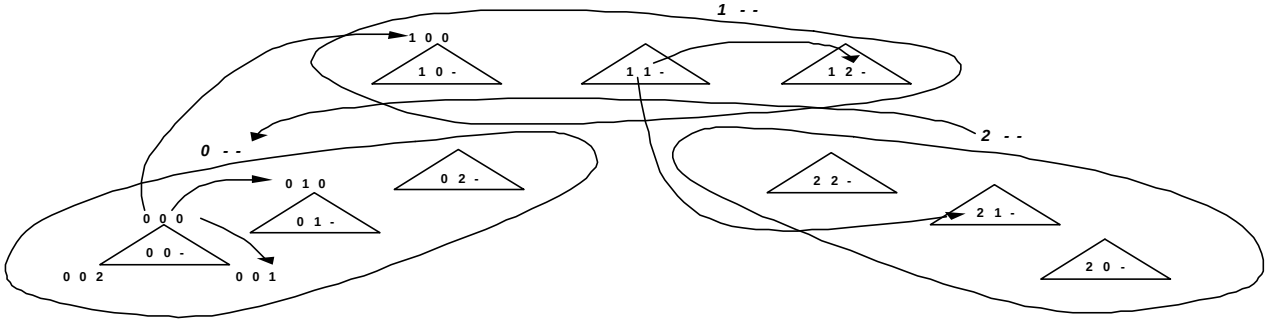


Figura 5. Enlaces dimensionales.

5. Conclusiones

Aunque nuestro trabajo esta todavía en fase de desarrollo, podemos extraer las siguientes conclusiones. Esperamos que estas podrán ser enriquecidos con posteriores experiencias.

Con los sistemas tradicionales de construcción de *overlays* para distribución de streaming desplegamos árboles de distribución que consiguen escalar el problema desde el punto de vista del servidor tradicional de streaming, pero que imponen unos requisitos de partida no implementables en escenarios reales (gran número de nodos intermedios con capacidad de reenvío de *n B bits/seg*). Con las listas de árboles TLP somos capaces de reducir estas restricciones:

- Los nodos por término medio solo necesitan reenviar *B bits/seg* frente a los *n B bits/seg* de las soluciones P2P tradicionales.
- Aprovechamos la capacidad de reenvío de todos los nodos. Aunque esta capacidad sea inferior al flujo *B*, el receptor contribuye con lo que puede al flujo total *(N-1)B* bits/seg. Esta solución no es incompatible con las existentes debido a que somos capaces de incorporar nodos *tacaños* cuya contribución al reenvío es *0* bits/seg.
- Podemos incorporar nodos *colaboradores* que sin estar interesados en recibir el flujo total, pueden aportar una fracción de su ancho de banda de subida al overlay de distribución. El que no estén obligados a un uso estresante de sus líneas de subida puede animar a que la comunidad P2P adopte una filosofía de *"hoy por ti mañana por mi"*.

Nuestros escenarios iniciales de equipos que quieren colaborar para distribuir el *stream* son máquinas de usuarios conectadas a la red utilizando tecnologías xDSL en las que el enlace de subida suele estar infrautilizado. Aunque este escenario puede llegar a reducir drásticamente el tráfico que se cursa desde el servidor de streaming haciendo escalable el proceso, el tráfico se concentra en el mejor de los casos (respetando que nodos cercanos en el árbol tengan cercanía en la red) en los extremos de la red. No obstante si pensamos en un futuro a medio plazo donde los usuarios tengan además de su acceso xDSL un acceso inalámbrico de alta capacidad con coste cero (redes WI-FI), incluso el tráfico que ahora se concentraría en los extremos de la red (ISP), pasaría a desviarse a través de las redes WI-FI desplegadas.

La red SON tiene algunas semejanzas estructurales con Pastry, Chord o CAN, pero se diferencia de ellas debido a su capacidad natural de agrupar equipos en subredes pasando de modo esparcido a denso sin ninguna merma de rendimiento. Pero creemos que la solución presentada es más flexible y escalable. De hecho estamos investigando la generalización de los resultados a otras aplicaciones P2P.

Referencias

[1] S. Deering and D. Cheriton. Multicast Routing in Datagram Internetworks and Extended LANs. *ACM Transactions on Computer Systems, Mayo 1990.*

[2] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable application layer multicast. *Proceedings of ACM Sigcomm, Agosto 2002.*

[3] P. Francis. Yoid: Extending the Multicast Internet Architecture, 1999. *White paper http://www.aciri.org/yoid.*

[4] A. Rowstron, A.-M. Kermarrec, M. Castro, and P. Druschel.. Scribe: The design of a large-scale event notification infrastructure. *Proceedings of 3rd International Workshop on Networked Group Communication, Nov. 2001.*

[5] Y.-H. Chu, S. G. Rao, S. Seshan, and H. Zhang. Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture. *Proceedings of ACM SIGCOMM, Agosto 2001.*

[6] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Application-level multicast using content-addressable networks. *Proceedings of 3rd International Workshop on Networked Group Communication, Nov. 2001.*

[7] S. Ratnasamy, P. Francis, et al. A Scalable Content-Addresable Network. *Proceedings of ACM Sigcomm, Agosto 2001.*

[8] A. Rowstron, P. Druschel. Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. *18th IFIP/ACM Conference on Distributed Systems Platforms (Middleware 2001), Nov. 2001.*

[9] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. *ACM SIGCOMM, Agosto 2001.*

[10] C.G. Plaxton, R. Rajaraman, and W. Richa, Accesing nearby copies of replicated objects in a distributed environment. *ACM Symposium on Parallel Algorithms and Architectures, Junio 1997*

[11] A. Mozo, J. Salvachúa. Efficient Distribution of Live-Streaming in Heterogeneous Peer-To-Peer Environments. *Enviado a ACM SIGMM, Nov. 2003.*

Implementación de un Servidor de Calificaciones Mediante Técnicas de Transmisión Multimedia sobre Redes de Paquetes

Fco. Javier Muñoz Calle, Juan Manuel Vozmediano Torres
Área de Ingeniería Telemática. Universidad de Sevilla.
Camino de los descubrimientos, s/n. 41092 - Sevilla.
Teléfono: 954 48 73 84 Fax: 954 48 73 85
E-mail: {fjmc, jvt}@trajano.us.es

Abstract Since the Internet revolution, the systems based on packet-switched networks for efficient transport of voice and video have become more and more popular. This paper presents the design and implementation of a service based on techniques of speech transport according to ITU-T Recommendation H.323. The created application offers a useful and flexible service, at the same time it benefits from the considerable number of existing kits. This shows the capabilities of this technology as well as its good performance and relevance for new developments.

1 Introducción

Las técnicas de transmisión multimedia sobre redes datagramas han alcanzado en los últimos años una importancia cada vez más notable, tanto por el desarrollo de estándares como por el fuerte crecimiento que experimentaron las redes IP (fenómeno Internet). Esto ha llevado a la aparición de numerosos productos basados en tecnología IP (Internet Protocol) así como a la integración de dichas técnicas en los grandes sistemas de telecomunicación, como sucede por ejemplo en el sistema de telefonía de 3ª generación UMTS (Universal Mobile Telecommunications System).

Se ha evolucionado de la tecnología de transmisión de voz tradicional hacia tecnologías de transmisión de Voz sobre IP (VoIP) basadas en el despliegue de una única red de paquetes integradora de todos los servicios. Esta integración de servicios en una única red de paquetes permite el desarrollo de nuevos servicios que pueden asociar aspectos que anteriormente estaban segmentados por motivos de la tecnología de red sobre la que se basaban. Ejemplos de estos servicios son: servicios de localización, grupos de salto, transmisión de vídeo, integración de buzón de voz y correo electrónico, fax, autenticación, control de acceso y seguridad.

Una vez consolidadas las técnicas para la transmisión de medios (voz, vídeo y datos) sobre Internet, se está en disposición de crear servicios accesibles a cualquier usuario de modo directo, a pesar de lo cual no existe en el mercado un elevado número de estas aplicaciones.

Teniendo en cuenta todo lo anterior, este artículo se enmarca dentro de un conjunto de estudios realizados sobre la transmisión de VoIP (Voice over IP) a través de redes de paquetes. En concreto, se ha desarrollado una aplicación servidora de eminente utilidad práctica, basada en la pila de protocolos ITU-T H.323.

La finalidad buscada con la realización de este proyecto era, haciendo uso de técnicas de transmisión de voz sobre redes IP, ofrecer a los alumnos servicios de gran utilidad práctica. En concreto, se optó por la realización de una utilidad que permitiese la consulta a distancia de las calificaciones (y fecha de revisión) de los exámenes actualmente realizados, tanto a través de Internet como desde un teléfono convencional.

Dicho servidor, con el objeto de dotarlo de la máxima portabilidad posible, fue desarrollado usando la librería PWLib, lo que permite que con el mismo código la aplicación funcione en dos de las plataformas más extendidas actualmente, Unix y Windows. Por otro lado, aunque el programa se ha realizado inicialmente con esa finalidad, su funcionamiento es totalmente genérico, lo que permite su uso con cualquier otro fín sin la necesidad de realizar ninguna modificación.

Las principales aportaciones del trabajo descrito son:

- Implementación de un servicio de gran versatilidad accesible de forma permanente a cualquier usuario, tanto a través de Internet como desde la red telefónica convencional.
- Incorporación de un elevado número de codecs (codificadores-decodificadores) que permiten la integración y comunicación de la herramienta con la práctica totalidad de herramientas ya existentes en el mercado para la transmisión de voz sobre redes de paquetes.
- Posibilidad de comprobar de manera práctica la viabilidad y más que aceptable eficiencia de las técnicas de transmisión multimedia sobre redes IP.
- Estudio y uso de las distintas librerías, aplicaciones y demás herramientas necesarias para poner en funcionamiento una utilidad de transmisión VoIP.

- Integración de las redes telefónicas con las redes IP mediante el empleo de pasarelas.

El resultado es un sistema servidor, basado en técnicas VoIP, capaz de ejecutarse tanto en la plataformas Windows como Unix. Dicho sistema aporta la posibilidad de ofrecer de manera inmediata un servicio accesible a cualquier usuario, de forma telefónica o a través de Internet, y adaptable a las características concretas del problema que se pretenda resolver.

El resto del texto está organizado del siguiente modo. En el epígrafe 2 se describen brevemente los sistemas VoIP. A continuación, el apartado 3 muestra las características del sistema implementado. Las pruebas realizadas y resultados se destacan en el punto 4 y, finalmente, el apartado 5 expone las conclusiones.

2 Transmisión multimedia sobre redes de paquetes

Para cursar tráfico de voz a través de redes de "paquetes" (unidades de datos que se transportan por la red) se requiere la “paquetización” de las muestras de voz previamente digitalizadas. También, y dada las limitaciones de recursos de estas redes, es necesario reducir el consumo de ancho de banda mediante compresión de datos. En el extremo de destino se realizan estas funciones a la inversa, y se introduce una función de compensación de variación de retardo.

La telefonía IP (o VoIP) es una denominación abreviada de las comunicaciones multimedia sobre redes de datagramas, implicando la presencia de cierto número de “medios” (voz, datos, vídeo, etc.) que comparten los enlaces de comunicación, lo que exige una conveniente codificación de dichos medios para obtener una mejor compartición de recursos con calidad aceptable. Los escenarios de aplicación de VoIP permiten la comunicación de usuarios de tres modos distintos en función del terminal utilizado, con claras diferencias de complejidad, coste y equipamiento necesario:

- PC-PC: utilización de terminales tipo PC, o equivalentes, interconectados mediante una red de datos. Es la técnica más fácil y barata, necesitando sólo una tarjeta de sonido, un micrófono y un altavoz.

- Teléfono-Teléfono: se basa en el uso de terminales tradicionales, interconectados mediante un backbone IP (línea de transmisión de información que conecta, dentro de una red IP, puntos diferenciados geométricamente, con un elevado ancho de banda) y gateways (permiten interconectar las centralitas tradicionales con la red IP). Presenta una mayor complejidad y coste al necesitar un gateway y la harmonización de direcciones IP-E164.

- Teléfono-PC: interconexión de usuarios conectados a redes de datos y redes telefónicas tradicionales. Se trata de la opción más compleja, requiriendo software compatible y un gateway entre ambas redes.

Existen 3 soluciones desarrolladas para la señalización del servicio de VoIP: modelo H.323 **[1]** solución ITU-T, International Telecommunication Union-Telecommunication, 1996), modelo SIP **[2]** solución IETF, Internet Engineering Task Force, 1999) y modelo MeGaCo **[3]** solución conjunta IETF-ITU). Dada su mayor difusión, la tecnología elegida para el desarrollo de la herramienta aquí presentada es la descrita en la serie H de ITU-T.

La Recomendación H.323 de ITU-T (versiones 1, 2, 3 y 4) define los componentes, procedimientos y protocolos para ofrecer comunicaciones multimedia en redes de paquetes sin QoS garantizada, describiendo tanto los terminales, equipos y servicios, como la señalización necesaria para comunicaciones multimedia sobre redes IP (también puede soportarse sobre redes IPX /SPX o redes ATM). La tecnología de transmisión de VoIP basada en este protocolo fue la primera en aparecer (1996), presentando por ello un mayor número de productos en el mercado, siendo el estándar más utilizado (empresas como Lucent Technologies, Cisco, Teldat, NetSpeak ó NetPhone han introducido productos VoIP basados en dicho estándar).

La Tabla 1 resume las principales Recomendaciones ITU-T sobre transmisión VoIP mediante H.323.

Tabla 1: Principales normas ITU-T del modelo H.323

G	Rec.	Título
Señalización y control de llamada	H.323	Marco y protocolo de redes alámbricas para transporte de la señalización de llamadas multiplexadas
	H.225.0	Paquetización, sincronización, y señalización
	H.245	Control de canal
	Q.931	Especificación de capa 3 de interfaz usuario-red de RDSI para control de llamada básica
	T.120	Conferencias de datos en tiempo real

G		Rec.	Título
Otras		H.450.x	Servicios suplementarios. Señalización/ procedimientos para servicios similares a telefónicos
		H.235	Requerimientos de seguridad para proveer autentificación y cifrado en sistemas H.323
		H.332	Conferencias a gran escala basadas en H.323
Codecs	Audio	G.7xx	G.711, G.722, G.723 . 1, G.72 8, G.729
	Vídeo	H.26x	H.261 y H.263

3 Descripción del sistema

El sistema implementado se compone de una aplicación servidora (en lenguaje C++), denominada SNAIT (Servidor de Notas del Área de Ingeniería Telemática) y que, ejecutándose en una máquina a la que se puede acceder desde el exterior vía telefónica o mediante Internet, permanece constantemente a la espera de las llamadas entrantes de los alumnos. De esa forma, cada alumno podrá conocer sus calificaciones sin la necesidad de personarse en el lugar físico en el que éstas son publicadas. Para ello, el alumno deberá realizar la llamada al servidor y, tras identificarse con su DNI, tendrá acceso a toda la información disponible en ese momento sobre sus recientes exámenes.

Aunque el programa se ha implementado pensando en el propósito expuesto, su funcionamiento se ha desarrollado de manera totalmente genérica, lo que permite su aplicación en cualquier otra tarea sin la necesidad de realizar ninguna modificación.

3.1 Herramientas empleadas

Inicialmente se presentan los distintos elementos de partida empleados para el desarrollo de SNAIT, todos ellos enfocados al uso del lenguaje C++:

- Entorno de desarrollo: con el objeto de implementar un código útil para que la utilidad SNAIT pueda ejecutarse tanto en sistemas Unix (Linux, Solaris, FreeB SD, BeOS, ...) como Win32 (Windows 9x/ME/NT/2000/XP), se hizo uso de la librería PWLib (Portable Windows Library) **[4]**.

- Soporte H.323: como base para el desarrollo de aplicaciones que deseen usar la pila de protocolos H.323 para comunicaciones multimedia sobre redes basadas en paquetes se empleó la librería OpenH323 (Open source H.323) **[5]**, desarrollada en C/C++ y fuertemente basada en PWLib. Se compone de una amplio conjunto de clases que implementan las distintas partes de la norma H.323 y demás elementos relacionados. Así, incluye clases que definen los distintos mensajes definidos en las recomendaciones H.225.0, H.245 y H.235, además de proporcionar el soporte necesario para la transmisión de audio y vídeo (codecs, RTP, ...).

- Conjunto de aplicaciones: OpenAM (Open source Answer Machine) como punto de partida; clientes de telefonía H.323 tales como OhPhone (Open source H.323 Phone), OpenPhone (Open source Phone) o NetMeeting **[6]**; y diversas herramientas de audio como Sox (SOund eXchange) **[7]**, Tools G.723.1 o grabadores/reproductores de audios.

3.2 Estructura general del servidor

SNAIT se basa en la transmisión de VoIP mediante la aplicación de la norma H.323, estando pensado para comunicaciones PC-PC o Teléfono-PC. La Fig. 1 muestra un posible esquema de la interconexión de SNAIT con la red telefónica PSTN (Public Switched Telephone Network) y la red de paquetes TCP/IP (Tranfer Control Protocol/Internet Protocol).

Las características más notables del sistema son:

- Aplicación servidor implementada en lenguaje C++, basada en la librería PWLib, funcional bajo los entornos Unix y Windows. Implementa la torre de protocolos H.323 para la transmisión de los datos, haciendo uso para ello de la librería OpenH323.

- Siguiendo el modelo cliente-servidor, el servidor SNAIT permanece constantemente a la espera de llamadas entrantes, usando para ello un listener H.323 en escucha permanente, pudiendo aceptar múltiples conexiones simultáneas.

- Tras llegar una solicitud de conexión y ser ésta establecida, se reproduce un mensaje de salida OGM (OutGoing Message) que indica al alumno que debe identificarse con su DNI. Cuando sea validado, podrá acceder a toda la información disponible en ese momento sobre sus recientes exámenes, todo gestionado por un menú, configurable, integrado por una serie de mensajes de voz (previamente almacenados) controlados mediante tonos DTMF (Dual Tone Multi Frequency).

- No requiere soporte hardware para audio (en la máquina del servidor no se reproducen los ficheros, sino en la máquina del cliente). Soporta los siguientes codecs: G.723.1, G.711-μ, G.711-A, GSM, MS-GSM, implementados mediante software.

- Permite ficheros de audio "Wav" y "Raw" para los mensajes de salida, admitiéndose los formatos "PCM, mono (1 canal), 8000 Hz, 16-bits" en codecs G.711-μ, G.711-A, GSM, MS-GSM. y "G.723.1, mono (1 canal), 8000 Hz, 6400 bps" para el codec G.723.1.

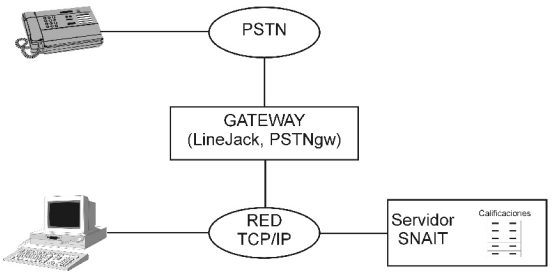


Figure 1: Interconexión de SNAIT con Internet y la red telefónica

- Genera un registro de cada llamada recibida. Permite la creación de una traza, con un nivel de detalle configurable, de todas las operaciones que se van realizando durante cada llamada recibida.

3.3 Funcionamiento interno

Dada la fuerte dependencia del código con las librerías PWLib y OpenH323, se deriva que la mayoría de clases implementadas en SNAIT descienden de clases pertenecientes a dichas librerías, como se puede observar en la Fig. 2.

La clase "PObject", que proporciona una serie de funciones básicas, es la clase padre para todas las demás. Las clases propias de SNAIT son:

- "snait": Clase principal del programa que contiene, básicamente, el hilo principal de la aplicación y el tratamiento de los ficheros de configuración.

- "MyH323EndPoint": gestión de las distintas conexiones de entrada/salida del endpoint (punto final) H323 correspondiente a la máquina servidora, activo durante todo el tiempo que el servidor esté en funcionamiento.

- "MyH323Connection": representa una conexión particular entre dos endpoints, asociada a un ejemplar MyH323EndPoint que la gestiona. Al menos hay dos canales (control y señalización); los canales de datos serán creados mediante el canal de control. Así, cada vez que llega una llamada entrante, estableciéndose una nueva conexión, se crea un ejemplar de esta clase que controlará dicha conexión, siendo destruida al cierre de la misma.

- "PCM_OGMChannel" y "G7321_OGMChannel": gestionan el envío al llamante de los mensajes de audio haciendo uso de los codecs GSM/G.711 y G.723.1.

- "G7231_File_Capability" y "G7231_File_Codec": se encargan de la gestión de capacidades así como de la lectura de los ficheros de audio cuando se usa el codec G.723.1.

- "PConfigArguments": empleada para el tratamiento de los argumentos indicados en la línea de comandos cuando se realiza la ejecución del servidor.

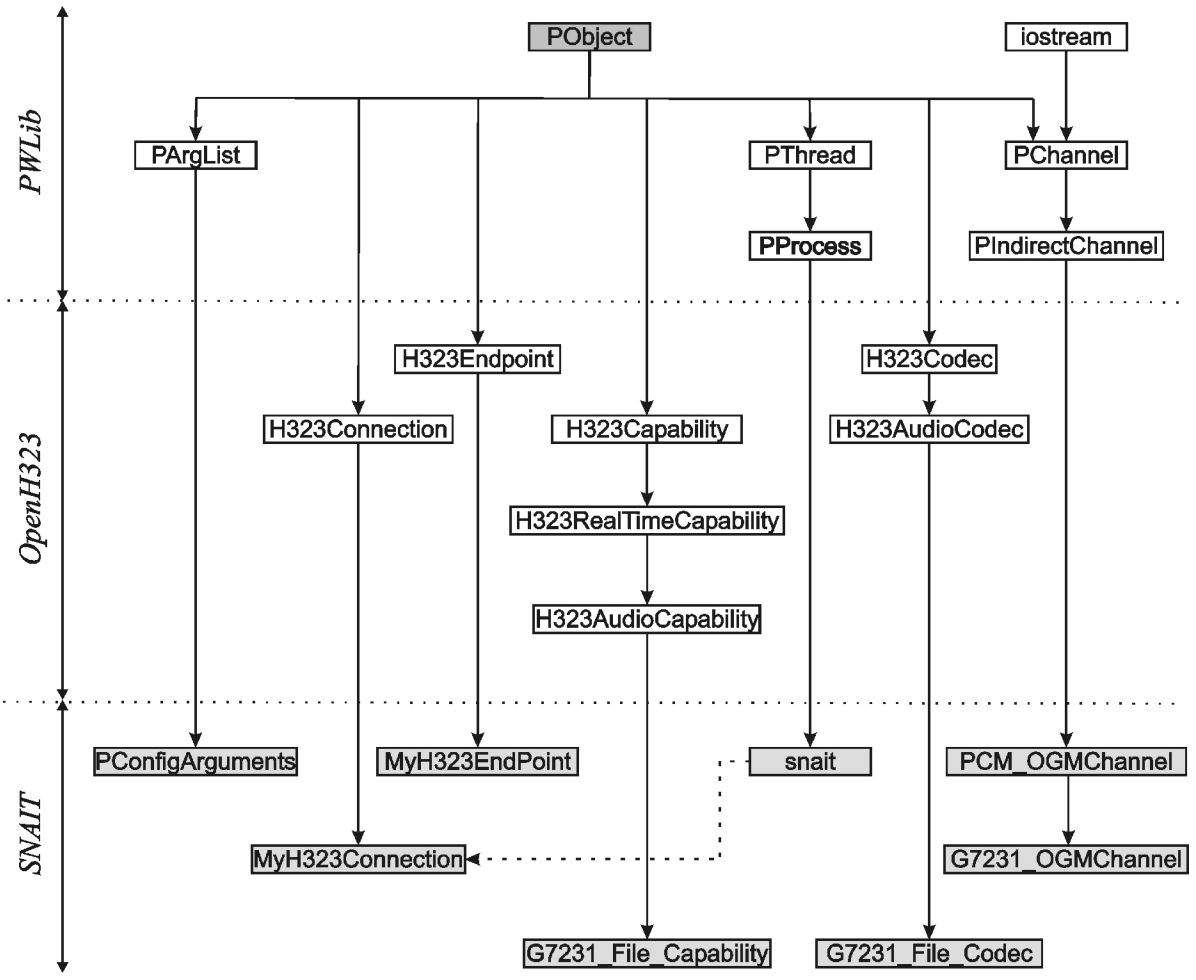


Figure 2: Estructura de clases de SNAIT

El proceso llevado a cabo en el hilo principal dl programa es el siguiente:

1º Lectura de los argumentos pasados al programa y ficheros de configuración. Según estos valores se realizan unas u otras operaciones como la habilitación del chequeo de memoria o la preparación de los ficheros de traza.

2º Creación de un ejemplar "H323ListenerTCP" con todos los parámetros (puerto y dirección IP) necesarios para realizar la escucha.

3º Creación del ejemplar "MyH323EndPoint" que se encargará de gestionar el endpoint correspondiente a SNAIT, incluyendo los datos de usuario, capacidades de los distintos codecs, posible uso de Gatekeeper y otros elementos (tiempo de inactividad, uso de ficheros Wav o Raw, registro de la llamada, ...).

4º Puesta a la escucha del ejemplar "MyH323EndPoint" en la interfaz dada por el ejemplar de "H323ListenerTCP". El servidor queda a la espera de llamadas entrantes, que serán gestionadas por ese ejemplar "MyH323EndPoint" creado, la cual se mantendrá durante toda la ejecución actual del programa.

El funcionamiento de la aplicación se basa en el uso de un conjunto de archivos imprescindibles para su correcto funcionamiento: configuración (en formato texto, controlan el funcionamiento general del servidor y la localización del resto de archivos), calificaciones (conjunto de archivos de texto que contienen la información servida por la aplicación; en el caso mostrado, las calificaciones de los alumnos para una asignatura determinada), traza (escritos por la aplicación con la monitorización de las llamadas realizadas al servidor así como de las distintas operaciones realizadas en su ejecución) y de audio (contienen los distintos mensajes reproducidos al usuario llamante).

3.4 Interconexión con la red telefónica

El servidor se encuentra instalado en una máquina con acceso a Internet, transmitiendo la voz mediante el protocolo H.323. De acuerdo con ello, para permitir el acceso al servicio desde la red telefónica resulta necesario el empleo de un gateway PSTN-H323 (Fig. 3). El proceso que tiene lugar al realizar una llamada desde un teléfono convencional es: la llamada, a través de la red telefónica, llega al gateway, el cual la reencamina hacia el servidor, a través de Internet, mediante el protocolo H.323; al recibir la llamada, el servidor la gestionará y emitirá la respuesta, que llegará al usuario a través, de nuevo, del gateway.

La implementación de esa pasarela está constituida por una parte hardware (tarjetas físicas a la que se conectan ambas redes) y software (gestiona la configuración de la conexión, siendo el gateway propiamente), dependientes de si la conexión se realiza con la red:

- RTC (Red Telefónica Conmutada): se requiere el uso de tarjeta tales como la LineJack **[8]**, con soporte H.323, y aplicaciones como el programa SwitchBoard o la utilidad abierta PSTNgw para gestionarla.

- RDSI (Red Digital de Servicios Integrados): además de la correspondiente tarjeta RDSI, se requiere un gateway software RDSI-H323, tales como Openisdngw **[9]**, isdn2h323 **[10]** ó isdngw **[11]**.

3.5 Consideraciones sobre clientes

Dado el sistema descrito, el usuario dispone de dos métodos de acceso al servidor, en función de la red desde la que realice la llamada:

- Llamada desde la red telefónica: basado en el uso de un teléfono, comprende el método más directo, reduciéndose a una simple llamada al número telefónico que se haya asignado al servidor.

- Llamada a través de Internet: requiere tanto del acceso a Internet como de la adecuada configuración de audio en la máquina de usuario. Además, resulta imprescindible tener instalado un cliente H.323 capaz de comunicarse con SNAIT, el cual deberá acceder por el puerto de escucha por defecto ("1720" según el estándar, modificable en la configuración del servidor) y soportar algunos de los codecs ofrecidos (GSM 06.10, MS-GSM, G.723.1, G.711 ley-u ó G.711 ley-A, en orden de calidad de audio ofrecida). Una vez cumplidos estos requisitos, bastará realizar una llamada desde el cliente a la dirección de Internet en la que se encuentra el servidor a la escucha.

Figure 3: Esquema de conexión a la red telefónica

Una vez realizada la llamada al servidor por parte del usuario, éste comenzará a recibir mensajes de voz indicándole los datos que debe facilitar en cada momento para acceder a la información deseada. Dicha información debe ser introducida con el teclado de la máquina cliente, siendo dígitos normalmente (luego el teclado numérico será suficiente).

La información introducida por el usuario es transmitida mediante el uso de señalización DTMF. Aunque el sistema de transmisión estandarizado por la ITU-T para la señalización DTMF es el envío de tonos dentro de banda, la mayoría de las aplicaciones de VoIP del mercado usan señalización Q.931 fuera de banda. Por ello, en SNAIT se ha implementado soporte DTMF dentro y fuera de banda, lo que representa una importante característica diferenciadora respecto a las otras aplicaciones H.323 existentes en el mercado.

4 Plan de pruebas

Tras solventar los múltiples problemas que acompañan al desarrollo de una utilidad de las características de SNAIT (complejidad de las librerías PWLib y OpenH323, codecs, tonos DTMF fuera de banda, ...) se procedió a su instalación y prueba.

El sistema se encuentra instalado en un entorno gestionado por una centralita RDSI, encontrándose instalado el servidor en una máquina UNIX. En adición a SNAIT se ha desarrollado un gateway RDSI-H323, gwRDSIH323, basado en los existentes en el mercado pero ampliado con ciertas facilidades adicionales que permiten dotar al sistema de una elevada flexibilidad y robustez.

El sistema descrito ha soportado múltiples pruebas, tales como el sometimiento del servidor a la recepción de múltiples llamadas simultáneas provenientes tanto de la red telefónica como de la parte IP, habiéndose obtenido los resultados esperados con una adecuada QoS.

5 Conclusiones

Resultado del desarrollo realizado se desprenden una serie de aspectos de destacable importancia:

- La aplicación desarrollada representa la posibilidad de ofrecer un servicio autónomo de gran utilidad práctica en múltiples aplicaciones, al mismo tiempo que puede adaptarse de manera casi inmediata para su uso en diferentes entornos de trabajo.

- La librería OpenH323ofrece una implementación abierta de la pila de protocolos H.323 que posibilita la realización, de forma relativamente sencilla, de aplicaciones que manejen tráfico multimedia sobre redes de paquetes, lo que de otro modo no sería fácilmente viable dada la complejidad de la torre H.323. Además, dado ese carácter abierto, pone a disponibilidad de cualquier desarrollador la implementación de estas funcionalidades que anteriormente sólo estaban al alcance de grandes grupos de trabajo.

- Resaltar la portabilidad y funcionalidad ofrecida por la librería PWLib, esencial en el desarrollo de aplicaciones complejas que deban ejecutarse en múltiples plataformas.

- La utilidad implementada representa un adecuado punto de partida para la creación de otros servicios de similar enfoque práctico, como podría ser una centralita encargada de gestionar y redirigir de forma automática las llamadas entrantes. De este modo, la herramienta presentada sirve de marco integrador para el desarrollo de soluciones de transmisión multimedia sobre redes IP.

Referencias

[1] ITU-T H.323: “Marco y protocolo de redes alámbricas para el transporte de la señalización de llamadas multiplexadas”, noviembre 2000.

[2] M. Handley et al; “SIP: Session Initiation Protocol”; IETF 2543; marzo 1999, http://www.ietf.org/rfc/rfc2543.txt

[3] F. Cuervo, N. Greene, A. Rayhan et al; “Megaco Protocol Version 1.0”; IETF 3015; noviembre 2000, http://www.ietf.org/rfc/rfc3015.txt.

[4] Librería PWLib; http://www.openh323.org /docs/PWLib/ index.html

[5] Proyecto OpenH323; http://www.openh323.org

[6] Brent Baccala, Martin Schiffers; "Linux NETMEETING HOWTO"; http://www.freesoft.org /software/NetMeeting

[7] SoX-Sound eXchange;http://sox.sourceforge.net

[8] Quicknet technologies;http://www.quicknet.-net

[9] Carlos Sevilla: “ISDN-H323 Gateway Open ISDNGateway”; http://www.openisdngw.org

[10] Linux H.323 - ISDN Gateway, http:// www.telos.de/linux/H323/

[11] ISDN Gateway; http://www.virtual-net.fr /h323 /isdngw

Experiencias sobre una Implementación Libre y Abierta del Estándar MHP para TV Digital Interactiva

Alberto Gil Solla, José J. Pazos Arias, Martín López Nores, Yolanda Blanco Fernández
Departamento de Ingeniería Telemática. Universidad de Vigo
ETSE Telecomunicación, Campus Universitario s/n
36200 Vigo
Teléfono: 986 812186 Fax: 986 812116
E-mail: jose@det.uvigo.es

Abstract. *The quick expansion of interactive digital TV is paving the way for a promising convergence of media, telecommunications and information technology, offering viewers increasingly exciting and interactive programming. The need of standardization led DVB to define standards for digital video broadcasting in all transmission networks, and recently for a generic common interface for interactive services, the Multimedia Home Platform (MHP). This last standard will enable interoperable applications to be downloaded from broadcast networks and executed on receivers with specific hardware and software implementations from any manufacturer. The software architecture of a MHP receiver consists of a multitask real-time OS that gives support to the middleware below the MHP API. The first implementations of MHP receivers make use of a proprietary OS. In this paper, we present our experience implementing a prototype based on an open platform over RT-Linux.*

1 Introducción

Una sociedad digitalmente interconectada no puede concebirse sin la existencia de una gran variedad de puntos de acceso a los sistemas empleados para intercambiar información. Resulta evidente hoy en día que existe una gran diversidad en el perfil de la gente que quiere acceder a esa interconexión global, los contextos en los que esta comunicación puede tener lugar y las herramientas usadas para implementar los sistemas de información. Más allá de la minoría de ciudadanos que se sienten cómodos usando un ordenador, es necesario desarrollar nuevos mecanismos de intercomunicación para la gente que no quiere usar un ordenador, no sabe cómo usar un ordenador, no puede usar un ordenador o, la mayoría de la población fuera del primer mundo, no tiene acceso a un ordenador.

El ritmo vertiginoso al que avanza la tecnología hoy en día está provocando que, incluso para la gente que puede comprar un ordenador, no siempre es fácil comenzar a utilizar una herramienta tan compleja, con la consiguiente aversión que se genera en muchos casos. La mayoría de las personas no se sienten cómodas cambiando sus hábitos, lo que dificulta la creación de una sociedad completamente interconectada a través del ordenador como único canal. Es necesario aprovechar los hábitos y costumbres profundamente establecidos en la sociedad para encontrar y promover nuevos usos para los canales de comunicación tradicionales que nos rodean. Desde este punto de vista, el futuro de la televisión como punto de intercambio general de información es incuestionable.

No hay duda de que la televisión es todavía el medio de comunicación de masas más relevante de nuestra sociedad, y ese papel se está viendo fortalecido por la llegada de la televisión digital. Esta transición tecnológica no sólo permite la recepción de muchos más canales (con mayor calidad de imagen y sonido) que la televisión analógica tradicional, sino que sus posibilidades a la hora de desarrollar y emitir contenidos interactivos hacen que la programación sea más variada y atractiva.

Por otra parte, Internet está creciendo de forma exponencial y ya se ha convertido en un medio de comunicación de masas de gran relevancia. El desarrollo frenético que ha sufrido el hardware y el software en la última década nos ha colocado a las puertas de la transmisión generalizada de audio y vídeo en tiempo real a través de la red. Sin embargo, la mayor parte de los operadores de televisión piensan que esto no supondrá una dura competencia o un peligro, sino una gran oportunidad que debe conducir a la integración de estos dos medios.

Esta opinión se ve reforzada por el hecho de que el desarrollo de la televisión digital y la creación de contenidos apropiados son indudablemente tareas de un gran coste económico. Para reducir el riesgo de estas inversiones es necesario alcanzar un número de usuarios que permita la viabilidad financiera en un periodo razonable. Y, por supuesto, si queremos que estos usuarios paguen por ver la televisión, es necesario proporcionarles servicios de valor añadido. En este escenario, donde los operadores están buscando desesperadamente aplicaciones novedosas, el acceso a Internet (y a los servicios existentes alrededor de la red) supone una opción irrenunciable.

Por tanto, el crecimiento de la televisión digital interactiva anuncia una convergencia inexorable de los medios de comunicación y la sociedad de la

información, proporcionando a los teleespectadores una programación interactiva cada vez más excitante.

El principal problema al que se enfrenta la televisión digital hoy en día se deriva de la incapacidad de los receptores de televisión actuales para procesar y mostrar la señal digital. La aproximación más extendida es el empleo de un receptor digital o SetTop box (STB) que proporciona el servicio de conversión de señales y da soporte a los procesos de interacción con el usuario.

Hasta el momento, la mayoría de las redes de distribución de televisión digital muestran una integración vertical, en la que un único proveedor controla toda la cadena de difusión: la cabecera, el sistema de acceso condicional, los equipos transmisores, el hardware y el software del STB. Para el desarrollo, difusión y ejecución de aplicaciones interactivas, estas redes emplean APIs propietarias, por ejemplo, MediaHighway (Canal Satélite Digital), OpenTV (Vía Digital), d-box Network, etc.

Sin embargo, para conseguir la integración de estos nuevos servicios en el mercado de la televisión, es necesario normalizar las tecnologías empleadas a lo largo de la cadena de distribución, porque sólo un mercado horizontal y sólidas garantías de compatibilidad permitirán la reducción de costes y alcanzar al mayor número de usuarios posible.

Hoy por hoy, la principal organización reguladora en este campo es el consorcio DVB (*Digital Video Broadcasting*). Desde su creación en 1993, el DVB ha definido un conjunto de estándares abiertos para la difusión de señales de vídeo y servicios interactivos sobre las distintas redes de transmisión, incluyendo el satélite, el cable o la difusión terrena. Los servicios y redes basados en este conjunto de normas están siendo empleados profusamente en todo el mundo. Recientemente, los objetivos del proyecto DVB se han extendido para trabajar en una API genérica que permita el desarrollo y difusión de aplicaciones interoperables y su ejecución en receptores con hardware y software específico, desarrollado por cualquier fabricante. Este nuevo estándar, conocido como MHP (*Multimedia Home Platform*) [1], define un contexto de ejecución para las aplicaciones y un interfaz software (la API MHP) para que éstas puedan acceder a los recursos hardware de cualquier tipo de receptor, desde STBs a televisores digitales o PCs multimedia.

En esta comunicación, se describe la experiencia de los autores en el diseño e implementación de un prototipo de receptor MHP basado en una plataforma abierta sobre RT-Linux. El resto de la comunicación se organiza de la siguiente forma: la sección 2 proporciona una introducción a la arquitectura software de un receptor; la sección 3 hace especial hincapié en la elección del sistema operativo; las secciones 4, 5 y 6 se centran en el diseño de la API MHP y las entidades que gestionan la ejecución de las aplicaciones y el flujo de los eventos de usuario; finalmente, la sección 7 expone algunas conclusiones.

2 El estándar MHP

Para conseguir aplicaciones interoperables, que se puedan ejecutar en múltiples receptores, tres son los principales problemas que se deben resolver:

- La información enviada en el flujo de transporte debe abstraerse del código nativo del microprocesador del receptor. Para ofrecer un entorno de ejecución abstracto a las aplicaciones asociadas a los contenidos audiovisuales, el STB se convierte en una máquina virtual desde la perspectiva de las aplicaciones. Es decir, las aplicaciones no serán desarrolladas en el código nativo del microprocesador, sino que serán clases Java independientes del hardware. Por tanto, una aplicación MHP (conocida como aplicación DVB-J o Xlet) va a ser el bytecode de un programa, escrito en Java, que será interpretado por una máquina virtual Java que debe existir en el STB.
- La información enviada en el flujo de transporte debe minimizarse en lo posible. Para implementar su funcionalidad, una aplicación hará uso de las librerías y recursos existentes en el receptor. Para lograr la compatibilidad, el interfaz de acceso a esos recursos debe ser estándar, y ese ha sido el segundo objetivo de MHP: la definición de una API que deben implementar los receptores para que las aplicaciones accedan de forma normalizada a los recursos del STB.
- Las aplicaciones son objetos externos, cuya ejecución debe ser coordinada y controlada por el sistema operativo del receptor. Para ello, debe existir un mecanismo predefinido de comunicación entre las aplicaciones y el software del sistema. MHP especifica un conjunto de señales que cualquier aplicación debe estar preparada para recibir y ante las cuales debe comportarse de una determinada forma, según un ciclo de vida predefinido. Asimismo, la norma especifica un conjunto de llamadas que las aplicaciones deben realizar para comunicar la evolución de su ejecución al software del sistema.

Por tanto, el núcleo de la norma MHP está basado en una plataforma conocida como DVB-J, que incluye la máquina virtual Java (según la especificación de Sun Microsystems). Una entidad del software de sistema, denominada Gestor de Aplicaciones, será la encargada de coordinar la ejecución de las aplicaciones y las comunicaciones con su entorno. Un conjunto de paquetes Java proveen los interfaces entre las aplicaciones, las funciones de un receptor DVB y las redes de comunicación a las que está conectado. Igualmente, también se define en la norma los formatos de los contenidos que debe poder gestionar el receptor, la torre de protocolos que debe implementar y la señalización adecuada para coordinar el correcto funcionamiento del conjunto.

La arquitectura software de un receptor MHP (Fig. 1) consiste en un sistema operativo que da soporte al *middleware* colocado bajo la API MHP: el Gestor de

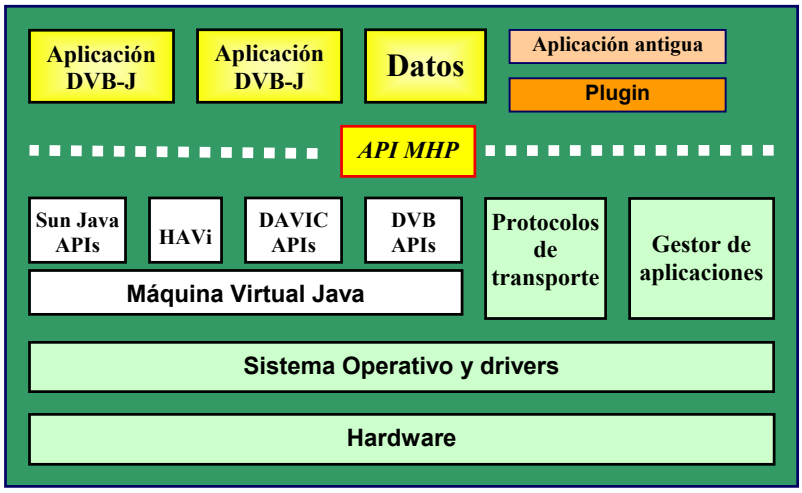


Fig. 1. La arquitectura MHP

Aplicaciones, la torre de protocolos de comunicaciones y la máquina virtual Java. Adicionalmente, entre esta última y la API MHP, debe existir un conjunto de APIs que implementan el acceso a los recursos: parte del núcleo fundamental definido en la plataforma Java, la API Sun Java TV [2], la API HAVi [3], la API DAVIC [4] y algunas APIs específicamente definidas por el DVB (por ejemplo, *org.dvb.lang* y *org.dvb.event*). Como se puede ver, para definir la norma MHP, el DVB ha adoptado (con las adaptaciones pertinentes) mucho trabajo ya existente, desarrollado por compañías privadas y organizaciones reguladoras en proyectos anteriores de características similares.

Para poder reutilizar aplicaciones antiguas (anteriores a MHP) y poder ejecutarlas en receptores MHP, es necesario contar con alguna clase de *plugin*, que sirva como interfaz entre el software de la aplicación y el *middleware* del sistema.

Una última entidad no mostrada en la figura es el denominado *Home Navigator*, un interfaz gráfico que debe implementar el STB para que el usuario pueda interaccionar con el sistema, cambiando el canal visualizado, ejecutando y parando aplicaciones, etc.

Además de publicar la norma, el DVB está jugando un papel activo en la promoción de esfuerzos de desarrollo que plasmen la arquitectura descrita en productos concretos que demuestren su viabilidad. En especial, cabe destacar el desarrollo de una implementación de referencia (actualmente sólo disponible para la plataforma Windows) que está llevando a cabo a través del IRT [12], un instituto de investigación y desarrollo de los operadores de televisión pública de Alemania, Austria y Suiza.

3 El Software de Sistema

3.1 El Sistema Operativo

Las soluciones habituales en la primera generación de receptores están basadas en sistemas operativos de tiempo real empotrados. Estos sistemas operativos han sido tradicionalmente diseñados de acuerdo a dos principios básicos: el cumplimiento de los requisitos impuestos por tareas de tiempo real (porque son empleados normalmente en ese tipo de contextos) y adecuación a un tipo específico de aplicaciones [5].

Sin embargo, esta clase de sistemas operativos no son apropiados para dispositivos multifunción, como nuestro prototipo, porque, aparte del cumplimiento de los requisitos de tiempo real, también son necesarios ciertos servicios típicamente sólo disponibles en sistemas operativos de propósito general (sistema de ficheros, gestión de memoria, máquina virtual Java, soporte para comunicaciones, etc.). Hasta ahora, no había sistemas operativos que cumplieran de forma razonable con todos estos requisitos.

Hoy en día, sí creemos que una aproximación basada en Linux puede ser válida. Linux no fue originalmente desarrollado para ser un sistema operativo de tiempo real, pues su intención era servir de soporte para estaciones de trabajo y servidores; por tanto, Linux evolucionó como un sistema operativo de propósito general.

Sin embargo, la popularidad, flexibilidad y potencia de Linux ha conducido a los desarrolladores de sistemas a intentar emplearlo en un número de aplicaciones cada vez más extenso, incluyendo sistemas de tiempo real y empotrados. Debido a la naturaleza abierta de su código fuente, Linux responde con rapidez a las exigencias planteadas en este tipo de sistemas, tanto en términos de limitación del tamaño como en la necesidad de optimizar el rendimiento de las tareas de tiempo real.

En este contexto han aparecido algunas soluciones, como RT-Linux [6], que proporciona tiempos de respuesta máximos acotados mediante la combinación de un núcleo simple de tiempo real sobre el que corre el Linux convencional empleando el tiempo sobrante para ejecutar los servicios típicos de un sistema operativo de propósito general. Con esta solución, el sistema puede responder a eventos en tiempo real y proporcionar simultáneamente a las aplicaciones el conjunto de servicios del Linux convencional.

Estos cambios hacen posible el uso de este sistema operativo en un número creciente de dispositivos. Actualmente, Linux se está convirtiendo en una solución apropiada como sistema operativo empotrado para STBs, equipos multimedia o dispositivos móviles. Muchos fabricantes de equipos electrónicos de consumo ya experimentan con él. Dentro del campo de la televisión digital interactiva, es importante destacar la TV Linux Alliance [7], creada en el año 2001 por 24 firmas líderes en el sector. Su objetivo principal es definir un entorno Linux estándar para el mercado de los STBs. Esta especificación debe, principalmente, proporcionar una plataforma estable, puesto que la evolución del núcleo convencional de Linux es demasiado rápida para la industria de electrónica de consumo.

La utilización de Linux como sistema operativo en nuestro prototipo nos proporciona las siguientes ventajas:

- Es un sistema operativo robusto, estable y fiable, con una extensa comunidad de desarrolladores.
- La disponibilidad de código abierto aporta la ventaja de una gran flexibilidad de diseño. Adicionalmente, la extensa comunidad de desarrolladores implica una rápida disponibilidad de *drivers* para los nuevos periféricos y dispositivos hardware, al igual que una gran facilidad para portar nuevas aplicaciones.
- La disponibilidad de diferentes versiones de componentes *middleware* (por ejemplo, diferentes implementaciones de la máquina virtual Java) permite una comparación fácil de rendimientos y funcionalidades.

Resumiendo, hemos seleccionado el sistema operativo RT-Linux porque proporciona un conjunto de funcionalidades robusto y dispone de drivers y extensiones que permiten una programación avanzada, manteniendo una implementación reducida. Además, la plataforma abierta de RT-Linux nos permite crear, portar y probar *middleware* y aplicaciones con una gran facilidad.

RT-Linux es un núcleo relativamente simple que gestiona y comunica tareas de tiempo real, donde el Linux convencional es una tarea de baja prioridad. Para la implementación de un prototipo de estas características es suficiente usar una versión simplificada de Linux que contenga solamente los elementos necesarios (incluyendo la torre de protocolos de comunicaciones y la máquina virtual Java) e implementar el gestor de aplicaciones como una tarea de tiempo de real de alta prioridad, la cual se comunica con los servicios del Linux convencional a través del núcleo.

De esta forma, el diseño e implementación del receptor MHP se limita al conjunto de APIs sobre la máquina virtual Java, el Gestor de Aplicaciones y el *Home Navigator*.

En la figura 2 podemos ver un diseño global de este receptor, conteniendo los elementos del *middleware* (incluyendo algunas APIs que describiremos más adelante), las fuentes de información y su interacción.

3.2 El Subsistema Gráfico

Una de las características principales de esta nueva televisión es el frecuente uso de pantalla que realizan las aplicaciones que implementan los servicios interactivos. Las aplicaciones deben compartir la pantalla con los contenidos audiovisuales tradicionales, solapándose con estos o mezclándose según el grado de transparencia definido.

Es necesario, por tanto, proporcionar a las aplicaciones un entorno gráfico y unos recursos para que puedan mostrar su funcionalidad sin necesidad de transportar demasiada información desde el proveedor de contenidos al STB.

En el caso de una plataforma Java sobre Linux, las aplicaciones disponen del AWT sobre el entorno de ventanas X11. Sin embargo, esta solución supone una carga excesiva para un STB ya que la mayor parte de las funcionalidades que nos aporta no son necesarias

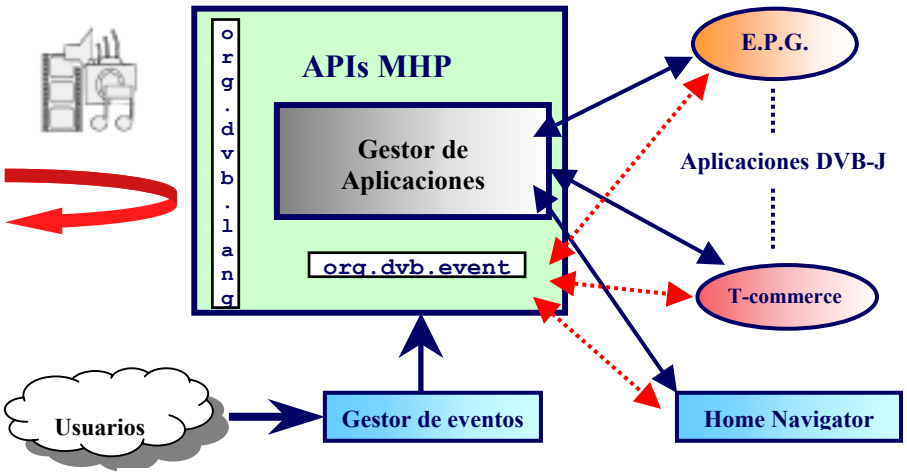


Figura 2. Diseño global del receptor MHP

en el contexto de la televisión. Por contra, los recursos de memoria y CPU sí que son un bien escaso en un STB, por lo que sería deseable una solución más ligera y flexible.

La solución que hemos adoptado en nuestro prototipo ha sido el desarrollo de un entorno gráfico propio, implementando una versión propietaria del AWT. Esta solución nos permite optimizar el consumo de recursos del STB, al tiempo que nos aporta un alto grado de flexibilidad.

Para reducir el esfuerzo de implementación y potenciar al máximo el carácter abierto de nuestro prototipo, nos hemos decantado por un diseño jerárquico, en el que cada capa se implementa mediante alguna librería de dominio público. En la figura 3 se puede observar la estructura de capas definida, en la que podemos observar:

- En la base del entorno, la más cercana al hardware de gráficos, se emplea el dispositivo FrameBuffer de Linux, que ofrece una abstracción de bajo nivel de la tarjeta gráfica empleada, si bien no está disponible para todas las tarjetas existentes en el mercado. Este nivel permite un acceso muy básico a la memoria de vídeo de la tarjeta (aunque de una forma estándar) para seleccionar el valor de cada píxel de pantalla, cambiar la resolución, el número de colores, etc.
- Sobre el dispositivo FrameBuffer hemos colocado una librería de dominio público llamada DirectFB [8] que ofrece un interfaz para dibujar puntos, líneas, formas básicas, representación de los formatos gráficos más habituales (GIF, PNG, JPEG), reproducción de vídeo, gestión de transparencias, etc.
- La librería de dominio público GTK [9] aprovecha las facilidades que aporta la librería DirectFB para ofrecer un amplio conjunto de componentes tales como botones, ventanas, menús, etc. A pesar de estar escrita en C, esta librería realiza una aproximación al paradigma de programación orientada a objetos que facilita en gran medida la implementación sobre ella del AWT.
- Sobre esta última librería se sitúan las clases Java (desarrolladas haciendo uso del interfaz JNI para código nativo) que implementan el AWT que esperan encontrar las aplicaciones MHP.

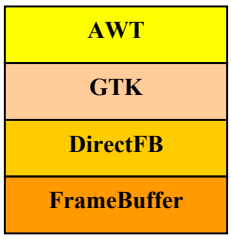


Fig. 3. Estructura en capas del entorno gráfico

3.2 El Acceso al Flujo de Transporte

El principal canal de recepción de información del receptor será el flujo de transporte recibido a través del interfaz de red correspondiente, una tarjeta de recepción de TV digital DVB, ya sea por satélite, cable o difusión terrena.

Sin duda, este es uno de los elementos que presentan más problemas de portabilidad del software, por las características singulares de cada tarjeta y la escasa disponibilidad de software libre (o incluso propietario) para Linux.

Por este motivo, nos hemos decantado por la Linux DVB API de la iniciativa LinuxTV [10]. Se trata de una API de software libre para Linux que, aunque de muy bajo nivel, está disponible para varias de las tarjetas más populares del mercado (incluyendo la Hauppauge WinTV Nexus-s con la que trabajamos en el prototipo), por lo que ya se ha aprobado su próxima inclusión en el kernel de Linux.

Esta API nos aporta la funcionalidad de filtrar flujos elementales de audio, vídeo y secciones privadas con la simple indicación del PID de los paquetes del flujo de transporte, para posteriormente acceder a su contenido. De esta forma, desde un nivel significativamente bajo, se construyen todas las tablas de señalización definidas por el DVB y se reconstruyen los ficheros que contienen las aplicaciones MHP y sus datos asociados.

4 La API MHP

La norma MHP cubre un amplio rango de aspectos relacionados con las tareas de un STB, desde la sintonización del flujo de transporte a través del interfaz de red hasta la adecuada representación de los contenidos enviados por el operador. Por esta razón, y especialmente por compatibilidad, el interfaz con las aplicaciones debe ser necesariamente amplio. Por tanto, la especificación incluye APIs con varios orígenes, entre las cuales cabe destacar:

- La norma HAVi, desarrollada por el consorcio HAVi. Esta norma define las características más apropiadas que deben tener los elementos empleados para desarrollar el interfaz gráfico de las aplicaciones en un contexto como la televisión.
- Algunos apéndices del estándar DAVIC, desarrollados por el consorcio DAVIC. En estos apéndices se tratan temas relacionados con funciones de bajo nivel del STB, como pueden ser la sintonización del flujo de transporte, el acceso condicional, o el filtrado de secciones privadas MPEG-2 para acceder a datos o señalización privada insertados en el flujo de transporte.
- La API JavaTV de Sun Microsystems. Esta API describe un interfaz de alto nivel, independiente del protocolo de transporte, para acceder a la información de servicio del flujo de transporte, los detalles de la programación de los servicios (canales) transmitidos o la localización del código

y datos de las aplicaciones. Además, también define los detalles del ciclo de vida de las aplicaciones y su comunicación con el Gestor de Aplicaciones a través del interfaz Xlet.

Además de estas APIs, extraídas de trabajos anteriores, el DVB ha definido nuevos interfaces para gestionar las nuevas funcionalidades de este tipo de receptores. Se trata del paquete *org.dvb*, entre cuyos interfaces podemos destacar:

- Una API de listado y lanzamiento de aplicaciones (*org.dvb.application*), que proporciona al Gestor de Aplicaciones y a otras aplicaciones la capacidad de descubrir las aplicaciones disponibles en cada servicio de televisión y solicitar su lanzamiento.
- Una API (*org.dvb.*si) para el acceso completo a toda la información de servicio (señalización) definida en las normas DVB. Por tanto, es una API de acceso a información de servicio dependiente de protocolo, sobre la que se construirá la API independiente de protocolo antes mencionada.
- El paquete *org.dvb.lang* nos permite obtener las clases Java que implementan las aplicaciones remotas. Para ello, define las características de los cargadores de clases que debe implementar la máquina virtual Java de la plataforma, especificando un sistema de delegación entre distintos cargadores de clases similar al existente en la plataforma Java 2 de Sun.

  Las clases de las aplicaciones se recibirán a través de un carrusel de objetos integrado en el flujo de transporte (Fig. 2). Este carrusel sigue la norma DSM-CC de MPEG-2 [11]. Esta funcionalidad es esencial, puesto que permite al STB ser algo más que un simple reproductor de audio y vídeo.

  Sin embargo, la naturaleza cíclica del carrusel deriva generalmente en problemas de latencia. Los objetos no están disponibles en todo momento, sino que solamente pueden ser leídos en instantes específicos, lo que ralentiza su proceso de adquisición. A causa de esto, las aplicaciones deben ser necesariamente simples. Sin embargo, hemos comprobado que el orden de los objetos enviados en el carrusel afecta en gran medida a esta latencia: la mayor parte de las veces, una elección apropiada del orden de empaquetamiento puede reducir significativamente el tiempo de espera.

  Además, hemos conseguido importantes mejoras en la reducción de la latencia mediante la implementación de un mecanismo de caché. Aquí, un caché convencional puede acelerar la ejecución de las aplicaciones tras la primera vez. Para acelerar la primera ejecución, los mejores resultados se consiguen mediante una estrategia de carga anticipada (*prefetching*). Sin embargo, para mantener un caché reducido y simultáneamente trabajar con aplicaciones grandes, sería útil implementar un mecanismo de coordinación. Nuestros mejores resultados en la reducción de la latencia pasan por adjuntar a las aplicaciones cierta información sobre el orden más probable en el que se van a necesitar los objetos. En ese caso, cuando se selecciona un canal, el Gestor de Aplicaciones puede comunicar ese orden al caché. Desafortunadamente, esta es una solución propietaria y ningún mecanismo similar se ha definido todavía en el estándar.
- Pero, quizás, el paquete más importante es el *org.dvb.event*, puesto que permite a las aplicaciones y a resto de los elementos del sistema recibir eventos de los usuarios, proporcionando los mecanismos de interactividad que caracterizan a esta nueva televisión (Fig. 2).

  Este paquete define una entidad fundamental, el Gestor de Eventos (Fig. 2), que es el encargado de recibir todos los eventos de usuario y distribuirlos entre las aplicaciones que los hayan solicitado. En caso de recibir un evento no solicitado por ninguna aplicación, éste será enviado al *Home Navigator*. El objetivo de esta API es proporcionar compatibilidad con el mecanismo estándar de eventos de *java.awt* y definir un nuevo mecanismo de acceso a los eventos de usuario para las aplicaciones que no usen el AWT.

  La API también define dos formas en las que las aplicaciones pueden acceder a los eventos: acceso exclusivo (cada evento será entregado a una única aplicación) o acceso compartido (cada evento puede ser entregado a múltiples aplicaciones).

  El Gestor de Eventos, ante la generación de un evento, estudiará las características de las aplicaciones que lo hayan solicitado. Si a una de ellas se le ha concedido en exclusiva, se le comunicará su ocurrencia, ya sea por el mecanismo de eventos de Java (si es una aplicación que use el AWT) o por el definido en *org.dvb.event* (si no usa el AWT). Si varias lo han solicitado, se le entregará a todas las que no usen el AWT o a aquella que tenga el foco de entre las que empleen el AWT. Este comportamiento, con más detalle, se puede ver en la Fig. 4.

  Adicionalmente, esta API también proporciona mecanismos para informar a las aplicaciones de cuándo han ganado o perdido acceso a los eventos que se pudieran generar.

  Como veremos en la siguiente sección, otra importante funcionalidad de la API *org.dvb.event* es la de permitir al usuario modificar el ciclo de vida de las aplicaciones, es decir, participar en los procesos de carga, ejecución y terminación de las aplicaciones.

5 El Gestor de Aplicaciones

El gestor de aplicaciones es otro nuevo elemento definido ex profeso para el estándar MHP. Esta entidad coordina la correcta ejecución de las aplicaciones en el sistema, gestionando su ciclo de vida y los canales de comunicación que pueden

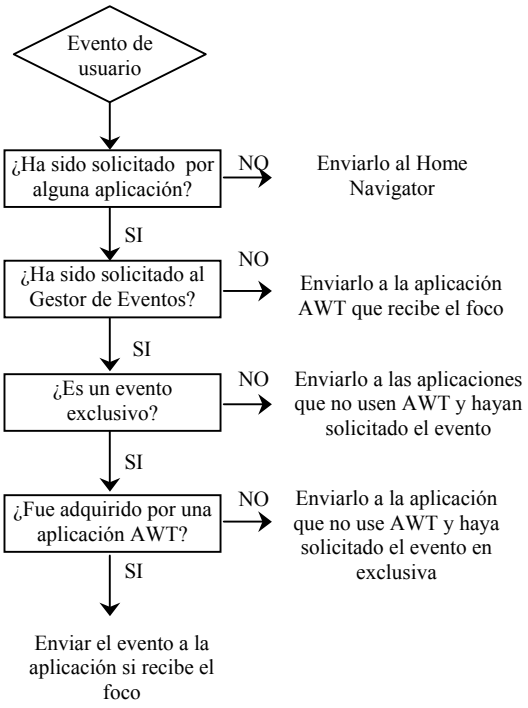


Fig.4. Algoritmo de entrega de eventos

influir en él. Es el responsable de inicializar las aplicaciones, ejecutarlas, detenerlas y destruirlas.

El ciclo de vida de las aplicaciones introduce el elemento fundamental que acerca la televisión al mundo de los ordenadores, suavizando el camino de la integración. Este ciclo de vida (Fig. 5) es definido en la API JavaTV de Sun, de aquí su similitud con el de un *applet*, con pequeños cambios en la definición de los estados que lo acercan al entorno de la televisión, donde serán ejecutados.

El ciclo de vida de una aplicación MHP consiste en un conjunto de estados en los que puede encontrarse la aplicación a lo largo de su ejecución y las señales que provocan las transiciones entre estados. Cada estado determina la actividad de una aplicación en él, los recursos de los que dispone y sus posibles evoluciones. Una aplicación conforme a MHP debe informar al gestor de aplicaciones de cada cambio de estado realizado voluntariamente (fruto de sus tareas) e implementar los procedimientos que se definen a tal efecto en la norma para recibir mensajes del gestor de aplicaciones ordenando un cambio de estado.

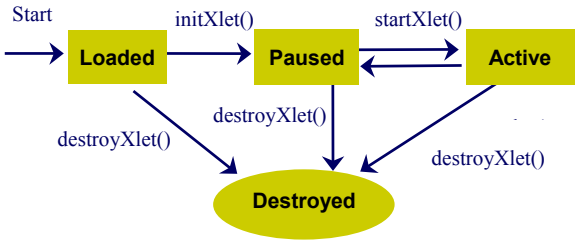


Fig. 5. El ciclo de vida de una aplicación MHP

El gestor hace uso de la API *Xlet* (parte de la norma JavaTV) para comunicarse con las aplicaciones. El interfaz Xlet (que debe ser implementado por todas las aplicaciones) es empleado por el gestor para comunicar las órdenes, mientras que el interfaz XletContext (también definido en JavaTV) es implementado por el gestor para que las aplicaciones lo utilicen para comunicarle los cambios internos.

Por otra parte, este ciclo favorece la adaptación al típico entorno de ejecución en el STB, donde varios elementos pueden actuar sobre el ciclo de vida de las aplicaciones. El gestor de aplicaciones es siempre la entidad responsable de encauzar las órdenes que actuarán sobre las aplicaciones. Los cambios en el estado de las aplicaciones pueden estar provocados por su funcionamiento interno (por ejemplo, finalizar tras hacer su trabajo), provenir de fuentes externas, como puedan ser el proveedor de contenidos, el usuario u otra aplicación, o ser generados directamente por el Gestor de Aplicaciones ante necesidades eventuales del sistema (Fig. 6).

El proveedor de contenidos puede emplear los mecanismos de señalización existentes en el flujo de transporte para modificar el ciclo de vida de una determinada aplicación, añadir nuevas aplicaciones o destruir otras. El gestor debe monitorizar el flujo de transporte para detectar estos cambios en la señalización. Al respecto, observamos que una implementación más eficiente consistiría en usar un interfaz asíncrono asociado al sistema responsable de decodificar y gestionar el flujo de transporte.

En nuestra opinión, una posible mejora sería que el operador enviase una señal cuando una aplicación deje de estar disponible en el sistema.. Esto evitaría que el gestor tuviese que tener una copia de las señales previas y comprobar qué aplicaciones están presentes en el sistema cada vez que una nueva señal es recibida.

Otra aplicación existente en el sistema puede también actuar sobre una determinada aplicación por medio del Gestor de Aplicaciones, haciendo uso de la API *org.dvb.application* (de Listado y Lanzamiento de Aplicaciones), pero sólo si tiene los correspondientes permisos.

Finalmente, como mencionamos en la anterior sección, el usuario también puede modificar el ciclo de vida si la aplicación ha definido eventos de usuario. Sin embargo, en este caso, creemos que la decisión más acertada es que la comunicación entre el usuario y el gestor de aplicaciones tenga lugar a través del gestor de eventos y, este a su vez, a través del Home Navigator, empleando la API *org.dvb.application*. Así, el Navigator oculta los detalles del Gestor de Aplicaciones al usuario.

6 El Home Navigator

El *Home Navigator* es una aplicación especial residente del sistema (es decir, no es una aplicación MHP transportada en el carrusel de objetos) que permite la interacción entre el usuario y el receptor.

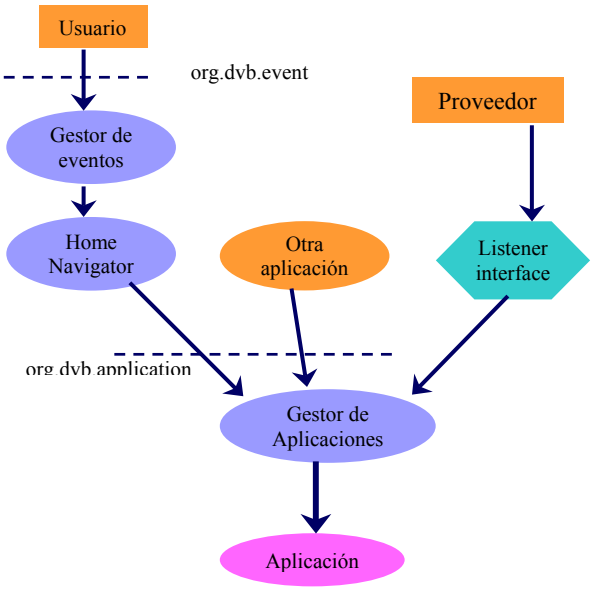


Fig 6. Control del ciclo de vida

El navigator es un interfaz gráfico que, básicamente, permite al usuario seleccionar servicios (canales), ejecutar las aplicaciones disponibles en cada servicio e interaccionar con ellas.

Otras interesantes funcionalidades del *Home Navigator* definidas en el estándar incluyen el inicio del acceso a Internet a través de un navegador, menús de ayuda , gestión de perfiles de usuario, etc.

No hay restricciones acerca de la forma de implementar el *Home Navigator*. A este respecto, el estándar sólo describe brevemente sus funciones. De hecho, debemos notar que el *Home Navigator* no es una aplicación DVB-J y, por tanto, no tiene que seguir el ciclo de vida antes mencionado.

Sin embargo, debemos considerar que el *Home Navigator* debe incluir una parte DVB-J, puesto que tendrá acceso a elementos del sistema definidos como DVB-J en el estándar, como la sintonización de servicios o el control del ciclo de vida de las aplicaciones. Por estas razones, es recomendable una implementación basada en Java que, además, permitiría el uso de las librerías gráficas del JDK.

7 Conclusiones

El desarrollo de una sociedad digitalmente interconectada requiere la disponibilidad de múltiples puntos de acceso a los sistemas de comunicación, con características tan variadas como la gente que debe usar estos. La televisión es un dispositivo muy familiar a todos los sectores de la población, está presente en casi todos los hogares y, ahora, potenciado por la revolución digital, se encuentra en una situación privilegiada para convertirse en un canal de acceso a la emergente sociedad de la información. El estándar MHP, desarrollado por el consorcio DVB, es el primer intento de normalizar el marco digital que está entrando en nuestras casas a través del salón.

La arquitectura software de un receptor MHP consiste en un sistema operativo multitarea y de tiempo real que da soporte a los elementos *middleware* colocados sobre él y debajo de la API MHP: el gestor de aplicaciones, la torre de protocolos de comunicaciones la máquina virtual Java y varias APIs. Las implementaciones convencionales del receptor hacen uso de un sistema operativo propietario, cuyas desventajas residen en la necesidad de desarrollar todo el *middleware* para él.

En esta comunicación hemos presentado nuestra experiencia en el diseño e implementación de un prototipo de receptor MHP basado en una plataforma abierta sobre RT-Linux. Esta decisión nos permite reducir los esfuerzos de desarrollo puesto que una importante parte del *middleware* (torre de protocolos de comunicaciones y máquina virtual Java) ya está disponible. Con el incremento de la capacidad computacional de los procesadores y el descenso de su precio (y de la memoria), es viable integrar los servicios convencionales ofrecidos por Linux como una de las tareas de tiempo real de RT-Linux. El Gestor de Aplicaciones, el elemento clave de la arquitectura MHP, se implementa como otra tarea de tiempo real de mayor prioridad.

El carácter abierto de Linux nos aporta una gran cantidad de librerías de dominio público que pueden ser empleadas (con mayores o menores adaptaciones) para implementar grandes apartados de la funcionalidad del prototipo.

Referencias

[1] DVB Consortium. Multimedia Home Platform (MHP) 1.1, (2001).

[2] Sun Microsystems. Java TV API specification, versión 1.0, (2000).

[3] HAVi. HAVi (Home Audio/Video interoperability architecture) user interface level 2, versión 1.0, (2000).

[4] DAVIC. DAVIC specification 1.4.1 part 9, complete DAVIC specifications, (1999).

[5] Arnold S. Berger. *Embedded Systems Design: An Introduction to Processes, Tools and Techniques*. Osborne McGraw-Hill, (2001).

[6] Michael Barabanov. *A linux-based real-time operating system*. M.S. thesis, New Mexico Institute of Mining and Technology, Socorro, New Mexico, USA, (1997).

[7] The TV-linux Alliance: http://www.tvlinuxalliance.org

[8] DirectFB. http://www.directfb.org

[9] The GIMP Toolkit. `http://www.gtk.org`

[10] LinuxTV. http://www.linuxtv.org

[11] ISO. ISO/IEC 13818 – 6 “Information Technology – Generic Coding of Moving Pictures and Associated Audio Information: Extensions for Digital Storage Media Command and Control”, 1996

[12] IRT. http://www.irt.de/IRT/mhp/mhp-e.htm

La radio por Internet y el *streaming* audio en directo. Análisis y comparativa de herramientas de *streaming* audio

L.M Regueras, M.J. Verdú & R. Mompó
Departamento de Teoría de la Señal, Comunicaciones e Ingeniería Telemática.
Campus Universitario Miguel Delibes. Universidad de Valladolid
Camino Viejo del Cementerio, s/n 47011 Valladolid
Teléfono: 983 42 39 82 Fax: 983 42 36 97
E-mail: {lregueras, mverdu, rmompo}@grupoinnovacion.com

***Abstract**. The rising popularity of Internet and the arrival of broadband networks have increased use of streaming media applications, with a traffic that is quite different from Web traffic, dominant until now. It is therefore important to understand and characterize the traffic associated with these applications in terms of their behavior characteristics and impact on the network, in order to support properly these new services. This paper studies the traffic associated with a specific audio streaming on-live service, the radio through the Internet, and presents analysis and comparisons of different streaming multimedia products, RealPlayer, Windows Media, QuickTime and WinAmp. In addition, in this study we take into account the user point of view; since it is ultimately the user (and his/her degree of satisfaction) who will decide whether a service goes to be or not successful.*

1 Introducción

Hoy en día, las redes de comunicaciones soportan una amplia variedad de aplicaciones. Además, dado que éstas son cada vez más rápidas y sofisticadas, también están permitiendo el desarrollo de otras nuevas, como es el caso de las aplicaciones de ***streaming media***. Pero desgraciadamente, el rápido crecimiento y desarrollo de estas nuevas aplicaciones, con unos requerimientos mucho más estrictos y ambiciosos, no ha venido acompañado de una completa comprensión y caracterización de su comportamiento. De hecho, cuando realmente se ha visto esta necesidad ha sido en los últimos años, con el fin de asegurar la calidad de servicio adecuada y propia a los diferentes tipos de aplicaciones.

Las aplicaciones de ***streaming media*** se están convirtiendo en servicios cada vez más demandados sobre la nueva generación de redes de banda ancha [27]; cada vez son más los usuarios que utilizan este tipo de servicios, en la forma de difusión de radio o incluso de televisión a través de Internet [30]. Además, según [11], en Europa se quiere impulsar la transmisión de radio y televisión por Internet, con el objetivo de llegar a una mayor y/o más dispersa audiencia [9] Asimismo, la empresa de medición de audiencias en Internet, Nielsen NetRatings, ha estimado que, durante el primer trimestre de 2002, un 23% de la población española mayor de 16 años había utilizado la radio por Internet en los últimos seis meses [22].

Con todo esto, se puede ver la importancia de caracterizar un servicio de *streaming audio* en directo, como es la radio por Internet. En concreto, en este artículo se va a estudiar y analizar cuál es el comportamiento de la radio por Internet y cuál es su estado de desarrollo. Es decir, se analizará cuales son las características más relevantes del tráfico de red generado por este servicio, lo cual permitirá examinar cuáles son los efectos y los parámetros más relevantes (perfiles de tráfico, longitud y tipo de paquetes, protocolos empleados, tasa de bit, etc.) para los diferentes canales de radio y sus correspondientes sistemas de *streaming media*. De esta forma, analizando las trazas capturadas de estas aplicaciones se podrá comparar el comportamiento presentado por cada una de las configuraciones empleadas, y así, mostrar una comparativa general del comportamiento del *streaming media* y de las diferentes herramientas.

1.1 Trabajos relacionados

En el área del análisis de red, son muchos los estudios que tratan sobre el tráfico IP en general [14] [24] y más concretamente sobre el tráfico Web [8] [10] [12], dada la fuerte importancia y el gran crecimiento de este tipo de tráfico en la pasada década [1] [15] [29]. Sin embargo, en el campo del ***streaming media*** no han sido demasiados los estudios empíricos dirigidos a analizar las características del tráfico [7] [19] [20], y su impacto en la red; a pesar de que, tal y como se ha comentado, en los últimos años las tendencias están cambiando y está aumentando el tráfico debido a estas aplicaciones [18].

En [19], a partir de trazas recogidas de varios servidores, se realiza un estudio del tráfico de *streaming* audio. El foco de este trabajo, como también ocurre con el presentado en [20], está basado en las características del tráfico a nivel de red, (longitud de los paquetes y del tiempo de llegada de paquetes); sin embargo, en [7], al analizar el comportamiento del *streaming media* en general (y no sólo del audio), y comparar y contrastar estas

características con el tráfico Web *no-streaming*, se centra en características del nivel de aplicación (duración y tamaño de las sesiones, popularidad de servidores y objetos...). Por otra parte, en [16] se lleva a cabo un análisis de las propiedades estructurales del tráfico RealAudio y se desarrolla un modelo de aplicación y en [17] se presenta un estudio empírico, tanto de Real Player como de Media Player, donde se muestra como éstos tienen un comportamiento claramente diferente y se expone el impacto que este tipo de tráfico tiene en la red.

La principal diferencia entre todos estos trabajos y el aquí presentado, se centra en no ver el *streaming* audio de una forma general, sino en considerar un servicio concreto, como es la transmisión de radio a través de Internet. Por una parte, esto va a permitir analizar y comparar el comportamiento de diferentes tecnologías de *streaming*, y no sólo de RealAudio. Además, en este estudio, se tendrá en cuenta el punto de vista del usuario, algo que no se ha hecho en ninguno de los otros trabajos.

1.2 Background

El *streaming media* es la transmisión de audio y vídeo desde un ordenador a otro [5]. En este tipo de aplicaciones, los datos son transmitidos desde un servidor a un programa cliente o reproductor (*player*), el cual los decodifica y los muestra al usuario a través del sistema de sonido/vídeo del host. La fuente de contenido puede ser un fichero digitalizado en el servidor o la salida digital de un *códec* de audio/vídeo (es decir, puede ir desde oír un CD, ver una película de vídeo a escuchar un programa de radio o ver un canal de televisión). Estos servicios pueden tener cientos de usuarios simultáneos y normalmente, y no hay ningún problema si el flujo es almacenado en *buffers* en recepción, para compensar la congestión de la red y el jitter.

A la hora de distribuir aplicaciones de *streaming* en la Web existen dos métodos: el uso de un servidor Web estándar o el uso de un servidor de *streaming* separado y especializado. El *streaming* en tiempo real requiere el empleo de servidores específicos; ya que éstos permiten un mayor nivel de control en la entrega de contenido (son más eficientes y flexibles, además de proporcionar una mejor calidad de audio y vídeo), y si bien también suelen ser más complicados de configurar y mantener [9] y más caros, su uso acaba importando beneficios a las inversiones realizadas [4]. Así, para poder escuchar un canal de radio por Internet, además de disponer de la tarjeta de audio, convenientemente instalada, el navegador debe tener instalado el "*plug-in*" correspondiente al sistema de *streaming media* empleado en la emisión.

Dos de los programas de *streaming media* más extendidos son RealSystem de RealNetworks y Windows Media de Microsoft, éste último de más reciente aparición pero que ha irrumpido con gran fuerza [6]. Real Player es uno de los sistemas más ampliamente distribuido y está disponible en la mayoría de sistemas operativos [13]. De hecho, RealNetworks tiene el 70% del mercado de *streaming* a través de Internet y su cliente está instalado en casi un 90% de los PCs de los hogares.

2 Metodología

Varias trazas de audio han sido capturadas, durante los meses de septiembre a diciembre de 2002, en un segmento de red de un departamento de la E.T.S.I.T. de la Universidad de Valladolid. A este segmento están conectados varios equipos (principalmente, PCs Pentium II a 350 MHz que corren bajo Windows). En estos PCs es donde se han instalado las oportunas herramientas de medida, encargadas de capturar y analizar el tráfico de red, y los clientes o programas correspondientes a los reproductores de *streaming*, lo que permite capturar todo el tráfico originado por o recibido hacia dicho usuario y llevar a cabo una adecuada caracterización del mismo. De esta forma, las trazas capturadas proporcionan una visión del comportamiento de la aplicación desde el lado del cliente y no del servidor, como hacen la mayoría de estudios que tratan estos temas [31]. Es interesante señalar que este estudio también podría haberse llevado a cabo a través de los ficheros de *log* del cliente o del servidor, dos métodos bastante utilizados para la caracterización Web. Sin embargo, se ha elegido el uso de un esquema basado en trazas dado que a partir del tráfico capturado de la red, y teniendo en cuenta los protocolos de los niveles superiores, se puede establecer el comportamiento de los usuarios y registrar la actividad de cualquier cliente. A su vez, el uso de trazas es una alternativa mucho más rica y de la que se ha visto presenta grandes ventajas [28].

El análisis y la captura de las trazas se ha realizado de varias formas diferentes con el fin de tener una visión lo más completa y general posible. Por una parte, las trazas han sido capturadas, y filtradas, mediante el uso de WinDump y analizadas posteriormente gracias a tcptrace y xplot, junto a algunos scripts escritos para ello. Y por otra, con el objetivo de obtener un mayor nivel de detalle sobre el contenido de los paquetes, también se ha hecho uso de otras herramientas más potentes, como es el caso de Protocol Inspector de Agilent.

Asimismo, los aproximadamente 200 "clips" o estaciones de radio seleccionadas (de carácter nacional e internacional), desde donde obtener las diferentes trazas, se han elegido de forma que permitan caracterizar, de la manera más amplia posible, el tráfico generado por las aplicaciones de *streaming* audio en directo y por tanto también de las diferentes herramientas de *streaming*. Además, para que los resultados no estén marcados, o al menos en la menor medida posible, por ningún condicionante temporal y/o geográfico, de cada "clip" se han capturado datos a lo largo de diferentes sesiones (dispersas en el tiempo) y desde diversos servidores (cuando esto ha sido posible).

Finalmente, y antes de pasar a analizar los resultados obtenidos, es importante comentar que, si bien es verdad que se ha analizado el tráfico a nivel de un único usuario, se han recogido datos bajo diferentes condiciones (y también cuando son varios los usuarios del mismo segmento de red que escuchan la radio a través de Internet), y se ha visto como a nivel del tráfico transmitido y recibido por un usuario éste no varía.

3 Análisis de los resultados

A partir de las trazas del tráfico recibido desde varias canales de radio *on-line* (nacionales e internacionales), en esta sección presentamos algunas propiedades estructurales del tráfico generado por las aplicaciones *streaming* audio en directo y del uso de las diferentes herramientas de *streaming*.

En primer lugar, se ha visto como la mayoría de las emisiones de radio en directo a través de Internet hacen uso del sistema Real Player, seguido por Windows Media Player, y ya en un menor porcentaje (en torno al 9% y 1,5%) aparece el nombre de otros dos reproductores de *streaming*, como son WinAmp de Nullsoft y QuickTime Player de Apple, respectivamente. En realidad, el porcentaje de penetración de WinAmp es mucho mayor; sin embargo, a la hora de considerar estos datos sólo se han tenido en cuenta aquellas estaciones de radio *on-line* con una cierta entidad; ya que SHOUTcast, (el sistema de *streaming* audio basado en WinAmp), es de carácter gratuito y permite que cualquier persona que lo desee (con el único requisito de que disponga de un enlace de alta velocidad) pueda crear su propia estación de radio a través de Internet y de hecho, se ha podido comprobar como eso es lo que está ocurriendo.

3.1 Protocolos y números de puerto

Al iniciar una conexión con un servidor de *streaming*, el cliente y el servidor determinan automáticamente la elección del protocolo de transporte de datos que van a emplear. Esta auto-configuración de los protocolos puede ser anulada por el usuario, pero es el escenario recomendado y usado por defecto en casi todos los sistemas de *streaming*.

La mayoría de los servidores de *streaming*, soportan una amplia variedad de protocolos de transporte (TCP, UDP unicast y multicast, e incluso HTTP sobre TCP); sin embargo, en este caso resulta interesante comprobar como en todas las situaciones analizadas el protocolo (a nivel de transporte) empleado es TCP. Este resultado no se ajusta al obtenido en [16], donde se observa como la mayoría del tráfico generado por este tipo de aplicaciones es UDP; sin embargo, si se acomoda más a los datos presentados en [21], donde se comenta como la existencia de cortafuegos puede ser un factor determinante del uso extendido de TCP.

Por otra parte, el número de puerto utilizado está directamente relacionado con la tecnología y el protocolo empleado; así, por ejemplo, RealServer escucha en tres puertos TCP diferentes desde donde inicia la conversación, autentica al servidor, e intercambia los mensajes de control antes y durante la conexión. Además, en el caso de usar TCP como protocolo de transporte de datos, estos puertos son utilizados también para, una vez realizado el proceso de autenticación, transmitir el tráfico de datos y de control desde/hacia el servidor [30].

Analizando la distribución de los números de puerto desde donde los servidores de *streaming*, (para los diferentes canales de radio *on-line*), emiten los datos de audio, se ha observado como predominan dos valores, 1755 y 554 (con un porcentaje del 53% y 35%, respectivamente). El número de puerto 1755 está relacionado con el uso de Windows Media Player como reproductor de *streaming*; mientras que el 554, junto con el 7070, son los empleados tanto por Real Player como por QuickTime Player; ya que éstos están asociados con el uso del protocolo RTSP (*Real Time Streaming Protocol*). Por otra parte, WinAmp Player se conecta a un servidor SHOUTCast, cuyo puerto de acceso establecido por defecto es el 8000, pero con la posibilidad de especificar cualquier otro. Y de ahí, que se haya observado también “otros” números de puertos. Finalmente, la presencia de HTTP (en cerca del 5% de los casos analizados) no debe ligarse al uso particular de ninguna herramienta de *streaming*; sino que es algo que se ha negociado entre el cliente y el servidor, quienes deciden hacer uso de esta opción cuando el resto de posibilidades han fallado (por ejemplo, debido a la existencia de proxys).

3.2 Longitud de los paquetes

Además de enviar datos a tasas específicas, el tráfico suele estar dominado por tamaños de paquetes específicos. Para todos los flujos de audio asociados a los diferentes sistemas de *streaming* analizados, el tamaño de los paquetes de Media Player es el que muestra una distribución más definida. Por regla general, en un clip de baja tasa de datos, casi el 100% de los paquetes recibidos son de un único tamaño; mientras que para altas tasas de datos, presenta dos distribuciones, una próxima al valor del máximo tamaño de segmento y otra a valores de menor longitud (con el resto del paquete proveniente de la capa de aplicación).

Real Player, en vez de transmitir paquetes de una longitud fija, combina el uso de paquetes (normalmente pequeños) de más de un tamaño. Por su parte, el tamaño de los paquetes de WinAmp se caracteriza por un elevado uso de paquetes de gran tamaño, combinados con otros paquetes de menor longitud (que no siempre se corresponden con el último fragmento de una misma unidad de datos proveniente del nivel superior).

Finalmente, con QuickTime se comprueba como, para cada flujo, se transmiten paquetes de tamaños muy diferentes, pero situados todos ellos en torno a un mismo rango de valores.

En la Fig. 1 se muestra una gráfica donde se resume (para todos los reproductores de *streaming*) cual es la distribución o PDF (*Probability Density Function*) del tamaño de los paquetes de todas las pruebas, normalizando para ello, el tamaño de los paquetes por el tamaño medio visto en cada prueba. En este gráfico puede verse como los tamaños de los paquetes de Windows Media y QuickTime están concentrados en torno a la media, normalizada a 1 (lo cual está en completa concordancia con lo visto anteriormente). El tamaño de los paquetes de WinAmp se concentra entre 1 y 1,2 veces la media (superiores al valor medio); mientras que con RealPlayer se observa como el valor medio no se corresponde al tamaño real de los paquetes.

3.3 Tiempo entre-llegadas

En esta sección se examina, para los diferentes sistemas de *streaming*, cuál es el comportamiento del tiempo entre-llegadas o diferencia de tiempo en que dos paquetes consecutivos llegan al cliente desde el servidor. Para ello, si los paquetes son recibidos en los instantes t_0, t_1 y t_2, se calcula $\delta_1 = t_1 - t_0$ y $\delta_2 = t_2 - t_1$ y son éstos los valores que se representan de forma gráfica.

Como también se hizo en el apartado anterior, en la Fig. 2 se muestra un resumen de cual es la distribución o CPDF (*Cumulative Probability Density Function*) del tiempo entre-llegada de los paquetes de todas las pruebas, normalizando el tiempo entre-llegada por el valor medio visto en cada prueba. En esta figura se observa como Windows Media Player presenta una fuerte concentración en torno a la media, cosa que no ocurre con el resto de los sistemas. Por otra parte, el hecho de que tanto en RealPlayer como en WinAmp la mayor concentración de valores se produzca por debajo de la media, pone de manifiesto la existencia de huecos en la transmisión de los paquetes.

Para entender mucho mejor que es lo que está ocurriendo realmente, en la Fig. 3 se muestra el número de secuencia de datos recibido con relación al tiempo de llegada de los paquetes. En esta gráfica puede verse como el flujo Windows Media se ajusta al proceder de un flujo a tasa de bit constante; transmisión de paquetes de longitud fija a una tasa de llegada constante. Sin embargo, con WinAmp y Real Player, el tráfico no es un flujo basado en una tasa ideal constante. En ambos casos, en vez de enviar los datos suavemente, envían ráfagas cortas de paquetes separados por huecos, lo que indica que hay algún patrón más complejo en el protocolo de transporte subyacente. Además, en el caso de WinAmp, el flujo de datos viene marcado por el tamaño de ventana

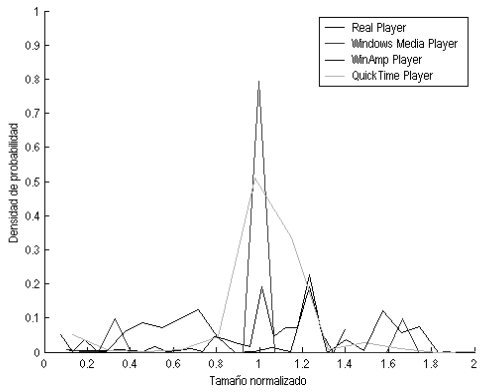


Fig. 1. PDF tamaño paquetes normalizado.

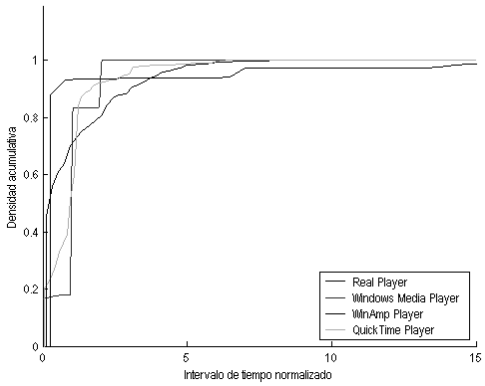


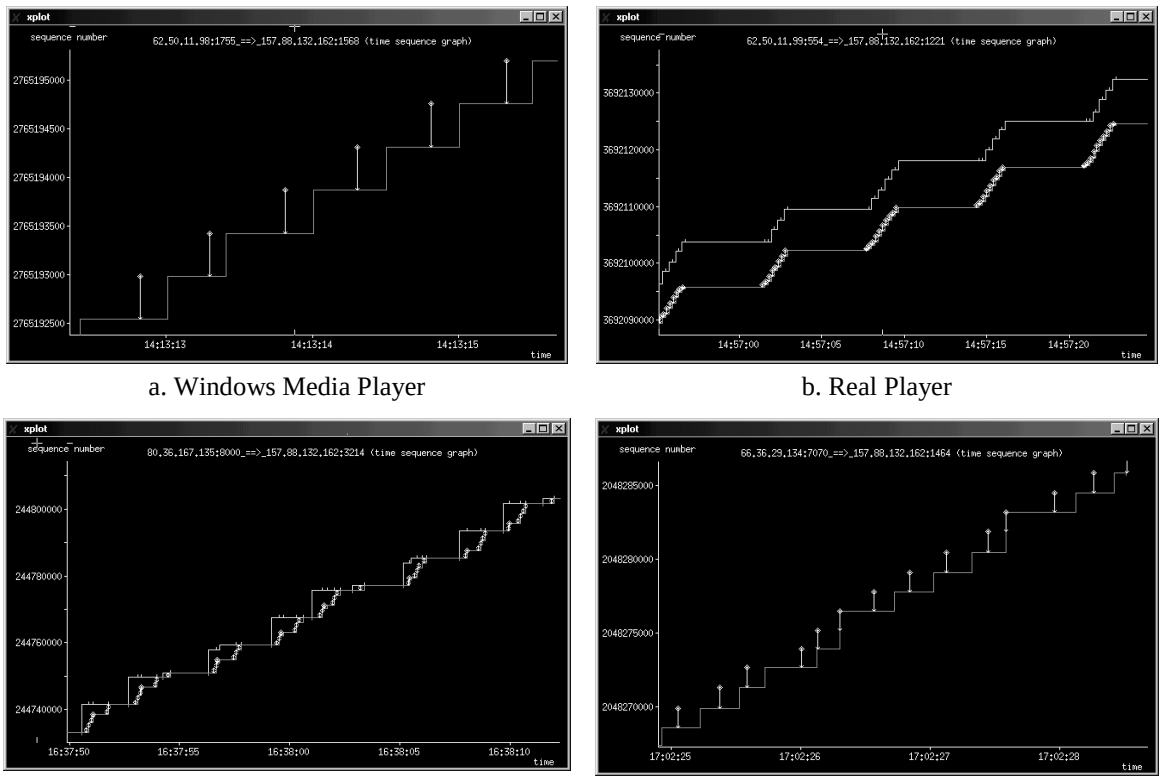
Fig. 2. CDF tiempo entre-llegadas normalizado.

anunciado por el receptor. Por su parte, QuickTime sigue un patrón que, si bien no se ajusta exactamente al proceder de Windows Media aún lo hace mucho menos al seguido por RealPlayer. Finalmente, QuickTime se basa en la transmisión de paquetes de longitud variable (en torno a un determinado rango de valores) espaciados también en intervalos de tiempo variable, consiguiendo así una tasa de transmisión constante de acuerdo a la tasa de codificación anunciada.

3.4 Ancho de banda y tasa de datos codificada

Analizando las tasas de codificación más empleadas por las estaciones de radio *on-line*, se ha comprobado como se centran en torno a 16-32 kbps. Sin embargo, una cosa es la tasa a la que el codificador en el servidor de *streaming* codifica el flujo de audio y otra muy diferente, es el consumo real de ancho de banda empleado en la transmisión. Este último valor estará relacionado con la tasa de datos codificada; pero también con cómo se transmite esa información.

Observando el throughput descendente generado por los diferentes sistemas de *streaming*, se ha visto como realmente Windows Media sigue el patrón típico de un flujo a tasa de bit constante, donde el valor del throughput en cada instante se ajusta al



a. Windows Media Player

b. Real Player

c. WinAmp Player

d. QuickTime Player

Fig. 3. Número de secuencia de datos

throughput medio. Por otra parte, Windows Media, desde la fase de establecimiento propia del protocolo TCP hasta el comienzo de la transmisión de datos (entre el servidor y el reproductor de audio), transcurren unos 5 segundos; algo que no ocurra con el resto de reproductores. Un comportamiento completamente diferente es el que se observa con el resto de reproductores: Real, WinAmp y QuickTime Player. En este caso, ya no se aprecia el proceder propio de una transmisión a tasa de bit constante; el throughput instantáneo se aleja del throughput medio y la tasa de llegada de paquetes varía considerablemente de unos instantes a otros.

Otra importante conclusión se obtiene al analizar el flujo de datos al comienzo del proceso de *streaming*. Tanto WinAmp como Real Player transmiten a una tasa mayor que la tasa de transmisión hasta que se llena el *buffer*, en cuyo momento empiezan a transmitir a la tasa de datos (o tasa de *playback*); mientras que con Windows Media y QuickTime Player se observa justo lo contrario. Windows Media siempre guarda información en *buffer* a la tasa de *playback* lo que justifica que su tasa de datos no sea a ráfagas. Esto puede explicar el porqué de muchas de las cosas comentadas a lo largo de esta sección.

3.5 Asimetría en el volumen de tráfico

A la hora de estudiar el volumen de tráfico, un aspecto importante es conocer cual es la asimetría existente entre el volumen de datos ascendente y descendente.

Analizando la cantidad de paquetes, y bytes, ascendentes y descendentes capturados en cada una de las trazas, provenientes de los diferentes flujos, se obtienen unas proporciones que van del 9:1 al 19:1. Este carácter fuertemente asimétrico del tráfico trasmitido entre el servidor y el reproductor de audio, era algo que se podía esperar, dado que el tráfico ascendente está ligado únicamente al envío de datos de control generados por el usuario y al envío de asentimientos, debido al uso de TCP como protocolo de transporte. Precisamente por este motivo, si bien es verdad que a nivel de datos existe una fuerte asimetría, a nivel de paquetes las cosas cambian, y se observa una mayor simetría.

4 Punto de vista del usuario

Una vez caracterizado el servicio de la radio por Internet desde el punto de vista de la red y del tráfico generado, es aconsejable dar un paso más e incluir también el punto de vista del usuario; ya que el éxito o no de un servicio dependerá del grado de satisfacción percibido por el usuario y es un factor que siempre debería tenerse en cuenta.

El objetivo de esta sección es comparar la calidad percibida por los usuarios, (cuando éstos escuchan la radio en directo a través de Internet), debido al uso de diferentes tasas de codificación y herramientas de *streaming*. Para ello, se han grabado diferentes fragmentos de audio, de unos 15 segundos de duración, de varias estaciones de radio *on-line*. A partir de estos fragmentos, se han seleccionado unas

20 personas de diferentes edades y sexo para que realicen un "test de degradación" (cuyo cuestionario es análogo al empleado en [26] para medir la calidad de la Voz sobre IP), y se han analizado los resultados obtenidos.

Un primer punto de estudio ha sido comprobar cómo afecta (en la calidad percibida por el usuario y para un mismo sistema), una mínima variación en la tasa de codificación, o lo que es lo mismo, qué diferencia observa un usuario cuando escucha un mismo programa de radio a dos tasas de codificación diferente (por ejemplo, 20 y 16 kbps). En este caso se ha visto como la mayoría de las personas encuestadas no notó apenas diferencia; e incluso, varios de los entrevistados llegaron a señalar el fragmento codificado a 16 kbps como de mayor calidad.

En segundo lugar, se analizó qué es lo que ocurre cuando una misma estación de radio emite, a una misma tasa de codificación, a través de dos sistemas de *streaming* diferentes (Windows Media y Real Player, por ejemplo). En este contexto, siguiendo un esquema análogo al comentado en el párrafo anterior, se ha visto como casi el 100% de los encuestados responden que las dos conversaciones (grabadas bajo condiciones de estabilidad) son idénticas, o prácticamente indistinguibles; aunque eso sí, todas las personas que han apuntado alguna diferencia entre ambas, señalan al fragmento reproducido a través de Real Player como de mayor calidad (este mismo resultado se obtiene en el estudio presentado en [23]). Esta diferencia es mayor en condiciones de inestabilidad, lo cual está en concordancia con el uso del *buffer* hecho por cada una de las dos plataformas. No se han podido realizar comparaciones con QuickTime porque no se ha encontrado ninguna estación de radio que emita en español a través de este sistema, y lógicamente realizar este tipo de pruebas en otro idioma resulta mucho más complicado. Además, ya a simple "vista" se ha podido comprobar como, para una misma tasa de codificación, tanto QuickTime como WinAmp proporcionan "escuchas" de peor calidad.

Para finalizar, se comprobó si un aumento general en la tasa de codificación y por tanto, un aumento del uso de recursos de red, supone un incremento "proporcional" en la calidad percibida por los usuarios. Para ello, los encuestados realizaron un "test de satisfacción" con fragmentos de radio convencional codificados a diferentes tasas (8, 16, 20 y 32 kbps), y a partir del cual se calcula su MOS (*Mean Opinion Score*), como una media de las puntuaciones dadas por todos los receptores [3] [25]. Los resultados obtenidos en este caso pusieron de evidencia como muchas veces un mayor consumo de recursos no se ve reflejado en un aumento proporcional del nivel o grado de satisfacción percibido por los usuarios finales. Además, tal y como se comenta en [2], es presumible pensar que esta diferencia sea aún menor cuando el usuario no se centra en escuchar la radio, sino que la escucha mientras realiza otras tareas, como navegar por la Web. Éste es un resultado importante que todo proveedor de contenidos, especialmente en aplicaciones de este tipo, que suponen un importante consumo de recursos, (y además de forma constante y durante largos periodos de tiempo), debe tener en cuenta. La gran mayoría de proveedores de contenido se centran en codificar los datos a la mayor tasa posible, en función del ancho de banda disponible por el usuario final, y no tienen en cuenta que estos datos deben compartir (y competir por) ancho de banda con otras muchas aplicaciones y que además, no por hacer uso de más recursos, el usuario va a estar más satisfecho; sino que de hecho, puede ocurrir todo lo contrario porque disminuya la calidad con que estaba realizando el resto de tareas. Asimismo, es interesante señalar como, por regla general, son las emisoras musicales las que emplean mayores tasas de codificación.

5 Conclusiones

A partir del estudio de un servicio concreto de *streaming* audio en directo, como es la radio *on-line*, se han podido sacar varias conclusiones generales sobre el comportamiento de los diferentes sistemas de *streaming*, el tráfico por ellos generado y su impacto en la red. Realmente, el empleo de una herramienta u otra es un factor importante, dado que va a marcar el perfil general del tráfico generado ya que cada uno presenta sus propias características, tal y como se resume en la Tabla 1. Asimismo, se pueden enumerar otras conclusiones obtenidas en esta misma línea:

- Para una misma herramienta de *streaming*, las características del tráfico son independientes de si se ha empleado el puerto propio del protocolo de *streaming* o a través del puerto HTTP. Esto es así dado que aquí, las acciones de control llevadas a cabo por los usuarios son casi nulas.
- Las herramientas, y no tanto el esquema de codificación, son los que marcan el perfil del tráfico generado. La transmisión de audio reproducido por dos sistemas de *streaming* diferentes y con un mismo esquema de codificación sigue un patrón diferente, a la hora de empaquetar los datos, y no sólo en cuanto a longitud de paquetes se refiere. Asimismo, esquemas de codificación diferentes (como es el caso del uso de técnicas CBR o VBR para la transmisión) apenas varían los resultados obtenidos para un sistema de *streaming* dado.

Las herramientas, y no el uso en sí de un mismo protocolo de transporte, son las que marcan la diferencia en el patrón del tráfico de red. Este es el caso de Real y QuickTime, ambos utilizan RTSP, como protocolo de transporte, sin embargo sus estadísticos son completamente diferentes. Esto, va a permitir identificar tráfico de diferentes aplicaciones, que comparten el mismo número de puerto.

Tabla 1. Herramientas *streaming*: Resumen características

Característica	Windows Media Player	Real Player	WinAmp Player	QuickTime Player
Patrón paquetes	Longitud fija	Mezcla paq. 2/3 longitudes	Patrón fijo paq. varios tamaños	Distribución en torno media
Tamaño general paquetes	Variable	Variable	Próximo MSS	Variable
Tasa llegada	Constante	A ráfagas	A ráfagas	Variable
Tiempo medio entre-llegadas	Alto	Bajo	Bajo	Medio
Comienzo *streaming*	< tasa *playback*	> tasa *playback*	> tasa *playback*	< tasa *playback*
Tamaño *buffer* por defecto[1]	5 segundos	30 segundos	24 segundos	10 segundos
Puertos TCP	1755	7070, 544	8000	7070, 544
Uso HTTP	Sí	Sí	Sí	Sí
Calidad	Muy buena	Muy buena	Regular	Buena

Finalmente, al analizar la relación entre la calidad de un servicio, desde el punto de vista del usuario, y el consumo de ancho de banda se ha podido comprobar como realmente, en muchos casos, aunque el usuario pueda apreciar diferencia entre una y otra tasa de codificación, ésta no compensa el incremento de ancho de banda, y mucho menos el decremento del número de usuarios que pueden estar activos en un momento dado en la red. Además, cuanto mayor es la tasa de codificación, más probabilidad hay de corte momentáneo del flujo de audio por congestión. Finalmente, ya se ha comentado como el servicio de radio en directo presenta unas características muy particulares. La gente que se acostumbra a hacer uso de este servicio, establece conexiones, por regla general, de larga de duración y además, como ocurre con la radio tradicional, la escucha mientras realiza otras tareas, por lo que sus requerimientos de calidad disminuyen.

Referencias

[1] M. Álvarez-Campana, A. Azcorra, J. Berrocal, A.B. García y J.R. Pérez, "MEHARI: An IP/ATM Traffic Analysis Platform based on Configurable Patterns", Proc. of the 6th HPOVUA'99, Bolonia, Italia, Junio 1999.

[2] A.H. Anderson, L. Smallwood, R. MacDonald, J. Mullin. y A. Fleming, "Video data and video links in mediated communication: What do users value?", International Journal of Human Computer Studies, 1999/2000.

[3] J. Anderson, "Methods for Measuring Perceptual Speech Quality", White Paper, Agilent Technologies, USA., Octubre 2001.

[4] "Streaming Media. A Comparative Cost Analysis Microsoft Windows Media Technologies and RealNetwork RealSystem G2", White Paper, Approach, Inc., Enero 1999.

[5] J. Cattaneo, "Entendiendo el Streaming Media", Artículos ICTnet, No. 40, Julio 2002. URL: http://www.ictnet.es/novedades/articulos/112.htm

[6] R. Chamorro e I. Penedo, "Internet: Estado del arte", Monográfico sobre Internet, ASTIC, 2000. URL: http://www.astic.es/estarte.htm

[7] M. Chesire, A. Wolman, G.M. Voelker y H.M. Levy, "Measurement and Analysis of a Streaming Media Workload", Proc. of the 3rd USENIX Symposium on Internet Technologies and Systems, San Francisco, California, USA, Marzo 2001.

[8] L. Cherkasova y M. Karlsson, "Dynamics and Evolution of Web Sites: Analysis, Metrics and Design Issues", HP Laboratories, HPL-2001-1R1, 2001. URL: http://www.hpl.hp.com/techreports/2001/HPL-2001-1R1.html

[9] D. Cunningham y N.J. Francis, "An Introduction to Streaming Video", Cultivate

[1]Valores obtenidos para una tasa de codificación de 21 kbps.

Interactive, No. 4, Mayo 2002. URL: http://www.cultivate-int.org/issue4/video/

[10] A. Feldmann, "BLT: Bi-Layer Tracing of HTTP and TCP/IP", Proc. of the WWW-9/Computer Networks, Vol. 33, No. 1-6, pp. 321-335, Mayo 2000.

[11] A. de las Fuentes, "Europa quiere impulsar las radios y teles por Internet", Ariadn@, El Mundo, No. 111, 27 Octubre 2002. URL: http://www.elmundo.es/ariadna/2002/111/1035633796.html

[12] F. Hernández-Campos, F. Donelson, K. Jeffay y D. Ott, "What TCP/IP Headers Can Tell Us About the Web", Proc. of the ACM SIGMETRICS/Performance, Boston, MA, USA, Junio 2001.

[13] "Users of Media Player Applications Increased 33 Percent Since Last Year", Jupiter Media Metrix, Press Release, Abril 2001. URL: http://www.jup.com/company/pressrelease-.jsp?doc=pr010403

[14] J. Kilpi, "TCP-Dump trace from a commercial ISP", Technical Report 27, Proyecto COST-257, 2000. URL: http://nero.informatik.uni-wuerzburg.de/cost/TDs/257td0027.pdf

[15] R. Koga y S. McCreary, "Traffic workload overview", CAIDA, Junio 1999. URL: http://www.caida.org/Learn/Flow/tcpudp.html

[16] K-C. Lan y J. Heidemann, "Structural Modeling of RealAudio Traffic", Technical Report 544, USC Information Sciences Institute (ISI), Marina del Rey, CA, USA, Julio 2001

[17] M. Li, M. Claypool y R. Kinicki, "MediaPlayer™ versus RealPlayer™ - A Comparison of Network Turbulence", Proc. of the 2nd ACM SIGCOMM Internet Measurement Workshop, Marsella, Francia, Noviembre 2002.

[18] S. McCreary y K.C. Claffy, "Trends in Wide Area IP Traffic Patterns - A View from the Ames Internet Exchange", Proc. of the 13th ITC Specialist Seminar on IP Traffic Modeling, Measurement and Management, Monterey, CA, pp. 1–11, Septiembre 2000.

[19] A. Mena y J. Heidemann, "An Empirical Study of Real Audio Traffic", Proc. of the IEEE Infocom 2000, pp. 101-110, Tel-Aviv, Israel, Marzo 2000.

[20] J.v.d. Merwe, R. Cáceres, Y. Chu y C. Sreenan, "mmdump: A Tool for Monitoring Internet Multimedia Traffic", ACM Computer Communication Review (CCR), Vol. 30, No. 5, Octubre 2000.

[21] J.v.d. Merwe, S. Sen y C. Kalmanek, "Streaming Video Traffic: Characterization and Network Impact", Proc. of the 7th International Workshop on Web Content Caching and Distribution (WCW), Boulder, USA, Agosto 2002.

[22] "Estudio Nielsen NetRating sobre usos de Internet y números de Usuarios", AUI (Asociación Española de Usuarios de Internet), 2002. URL: http://www.aui.es

[23] "Comparison Testing of RealAudio 8 and Windows Media Audio", Test Report, NSTL, Noviembre 2000. URL: http://www.nstl.com/Downloads/real-vs-windows-media.pdf

[24] V. Paxson, "Measurements and Analysis of End-to-End Internet Dynamics", Ph.D. Thesis Computer Science Division, Universidad de California, Berkeley, USA, Abril 1997.

[25] S. Pracht y D. Hardman, "Voice Quality in Converging Telephony and IP Networks", White Paper, Agilent Technologies, Octubre 2001.

[26] I. Qazzaz, "Verification of Voice over IP (In Network Integration Solution area)", Master Thesis Report, Chalmers University of Technology, Abril 1999.

[27] H. Shojania y B. Li, "Experiences with MPEG-4 Multimedia Streaming", Proc. of the 9th ACM Multimedia Conference, pp. 492-494, Ottawa, Canadá, Sept-Octubre 2001.

[28] D. Staehle, K. Leibnitz y P. Tran-Gia, "Source Traffic Modeling of Wireless Applications", Technical Report No. 261, Universidad de Würzburg, Alemania, Junio 2000. URL: http://nero.informatik.uni-wuerzburg.de/~leibnitz/work/TR/tr261.pdf

[29] K. Thompson, G. Miller y R. Wilder, "Wide Area Internet Traffic Patterns and Characteristics", IEEE Network Magazine, Vol. 11, No. 6, pp. 10-23, Nov-Diciembre 1997.

[30] R.S. Turnbull, A.G. Davis y M.D. Walker, "Robust Audio Streaming over IP", Proc. of the 9th Annual Conference of the Internet Society INET'99, San José, USA, Junio 1999.

[31] Y. Wang, M. Claypool y Z. Zuo, "An Empirical Study of RealVideo Performance Across the Internet", Proc. of the ACM SIGCOMM Internet Measurement Workshop 2001, San Francisco, USA, Noviembre 2001

Sesión 6A

Modelado y control de tráfico (II)

Dimensionado Eficiente de la Red de Acceso UMTS en Presencia de Múltiples Clases de Tráfico

A. B. García, M. Alvarez-Campana, E. Vázquez, J. Berrocal, J. Vinyes
Departamento de Ingeniería de Sistemas Telemáticos. Universidad Politécnica de Madrid
ETSI Telecomunicación. Ciudad Universitaria s/n. 28040 Madrid
E-mail: abgarcia@dit.upm.es

Abstract*. This article focuses on the dimensioning of the UMTS (Universal Mobile Telecommunications System) access network infrastructure. The current UMTS specifications define a protocol architecture for the access network terrestrial interfaces based on ATM (Asynchronous Transfer Mode). In this context, the main objective of our work is to determine the optimum transmission capacities necessary to satisfy the QoS (Quality of Service) requirements of each UMTS traffic class, specifically at the interface between each base station and its controller. By using a suitable simulation model, we obtain dimensioning rules for some significant UMTS applications. Also, some basic ATM traffic multiplexing strategies are examined in order to investigate their possible benefits regarding QoS parameters improvement or transmission capacity savings.*

1 Introducción

Dentro de los sistemas de comunicaciones móviles de tercera generación (3G), el sistema UMTS (Universal Mobile Telecommunications System) es uno de los más destacados. Planteado como evolución de los sistemas GSM (Global System for Mobile Communications) y GPRS (General Packet Radio Service), su especificación corre a cargo del foro 3GPP (3rd Generation Partnership Project).

Uno de los principales objetivos de UMTS es facilitar el acceso a una mayor gama de servicios que sus predecesores, incluyendo el soporte de aplicaciones multimedia y, en general, de aquéllas para las que la capacidad de las actuales redes GSM/GPRS resulta insuficiente. Para la consecución de este objetivo, resulta imprescindible la adopción de tecnologías adecuadas a la naturaleza multiservicio de UMTS, capaces de satisfacer los requisitos de QoS (Quality of Service) de cada aplicación y, al mismo tiempo, garantizar el uso eficiente de los recursos de red. Estos aspectos resultan especialmente críticos en la red de acceso, por la escasez de recursos que habitualmente caracteriza a este tramo en todo sistema de comunicaciones móviles celulares.

Las especificaciones de la red de acceso radio terrestre UMTS (UTRAN) establecen el empleo de WCDMA (Wideband Code Division Multiple Acess) en la interfaz radio y de ATM[1] (Asynchronous Transfer Mode) en la infraestructura de transmisión fija. La combinación de estas tecnologías sienta las bases para el despliegue de una red de acceso multiservicio con garantías de QoS. Sin embargo, a la dificultad intrínseca que plantea la planificación de este tipo de redes, en el caso de la UTRAN viene a añadirse el criterio de minimización de costes.

En la composición del coste de la UTRAN no sólo interviene el del equipamiento de red (estaciones base, controladores, etc), sino también el de la infraestructura de transmisión necesaria para su interconexión. El elevado coste que puede suponer esta infraestructura, tanto si es propia como alquilada, plantea la necesidad de disponer de una metodología que permita su dimensionado óptimo, siendo éste precisamente el objetivo del presente trabajo.

De manera más concreta, este artículo se centra en el dimensionado de la interfaz Iub, que conecta cada estación base (Nodo B) con su controlador (RNC, Radio Network Controller). La elección se debe a que es ésta la interfaz más numerosa en la red de acceso y, por tanto, la que con mayor peso incide en los costes globales de transmisión. En este contexto, se plantea una metodología para la determinación de la capacidad mínima requerida en la interfaz Iub en función de la mezcla de tráfico a soportar.

En primer lugar, se selecciona un conjunto representativo de aplicaciones correspondientes a las cuatro clases de tráfico que, a efectos de QoS, se definen en las especificaciones del 3GPP. Mediante el desarrollo de modelos de tráfico adecuados y la determinación de los requisitos de QoS a satisfacer para cada aplicación en la interfaz Iub, se aborda el proceso de dimensionado. La complejidad del problema conduce a la realización de un simulador y a la ejecución de diversos experimentos. El análisis de los resultados permite extraer una serie de conclusiones y pautas básicas de dimensionado para la interfaz Iub. Dentro del estudio, se presta especial atención al ahorro de capacidad que puede obtenerse mediante el efecto de la multiplexión estadística de tráfico.

[1] En el futuro, está prevista la utilización de IP (Internet Protocol) como alternativa a ATM.

2 Escenario de estudio

2.1 Protocolos en la interfaz Iub

Como se ha indicado, las especificaciones actuales del 3GPP establecen el empleo de ATM como tecnología de transporte en la UTRAN. En la Fig. 1 se muestra la arquitectura de protocolos de la interfaz Iub para el plano de usuario.

Las tramas radio de los canales de transporte dedicados se encapsulan sobre FP (Framing Protocol) para su transporte entre Nodo B y RNC. Se asume un canal de transporte dedicado por sesión de usuario (como previsiblemente ocurrirá en las primeras implantaciones de UMTS). Por motivos de eficiencia, cada canal dedicado se soporta sobre una conexión AAL2 (ATM Adaptation Layer 2), lo que permite multiplexar hasta 248 comunicaciones sobre un mismo canal virtual ATM.

Nótese que esta arquitectura de protocolos puede soportarse sobre interfaces físicas punto a punto o sobre una red ATM. Por ello, y con objeto de que los resultados sean aplicables a distintos escenarios de red, este estudio se centra en la determinación de las capacidades requeridas a nivel ATM.

2.2 Diferenciación de tráfico AAL2/ATM

Al establecer la correspondencia entre conexiones AAL2 y canales virtuales ATM, es necesario considerar los requisitos de QoS de la interfaz Iub. La especificación TS 23.107 [1] del 3GPP define cuatro clases de tráfico según su tolerancia al retardo: conversacional (la menos tolerante a retardo), *streaming*, interactiva y *background* (la más tolerante al retardo). El hecho de que cada clase de tráfico tenga sus propios requisitos de QoS (ver apartado 2.3), conduce a la necesidad de utilizar algún mecanismo de diferenciación de tráfico en la interfaz Iub. A pesar de su importancia, este aspecto no se aborda con suficiente detalle en las especificaciones del 3GPP, dejando libertad a fabricantes y operadores. No obstante, a la vista de la arquitectura de protocolos definida para la interfaz Iub, cabe plantearse la posibilidad de efectuar la diferenciación del tráfico a nivel AAL2 o a nivel ATM.

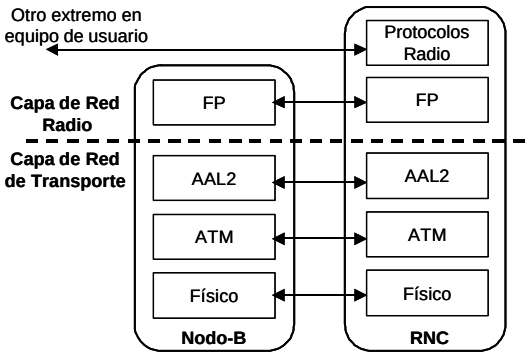


Fig. 1: Plano de usuario en la interfaz Iub

A lo largo de la literatura pueden hallarse propuestas de diferenciación tanto a nivel AAL2 [2][3], como a nivel ATM [4][5]. Al comparar ambos métodos se ha constatado [2][6] la superioridad del segundo enfoque, especialmente en presencia de altas cargas. Por este motivo, en este trabajo se ha optado por el método de diferenciación de tráfico a nivel ATM.

De manera más precisa, se ha decidido el empleo de un multiplexor AAL2 dedicado por clase tráfico. La diferenciación de tráfico se aplica a nivel de células ATM, mediante el empleo de conexiones virtuales separadas o compartidas. El primer criterio conduce a un procedimiento básico de dimensionado para la interfaz Iub que asume la segregación de capacidades ATM por clase de tráfico. El segundo enfoque permitirá investigar la posibilidad de refinar el método de dimensionado mediante la compartición estadística de una capacidad común entre flujos de tráfico de distintas clases.

2.3 Caracterización de tráfico y de QoS

Un aspecto clave de la metodología de dimensionado desarrollada es la elección de un modelo de tráfico válido para las cuatro clases definidas en UMTS y adecuado a las particularidades de la interfaz Iub. Este último aspecto se tiene en cuenta mediante la inclusión de los efectos de los protocolos radio y el protocolo FP. Siguiendo un enfoque similar al de otros estudios [7][8], se ha optado por un modelo estructurado en niveles.

Tal como se muestra en la Fig. 2, el modelo consta de dos niveles: ráfagas y paquetes. A nivel de ráfagas se distinguen dos estados, Alto y Bajo, lo que permite representar dos tasas binarias distintas. La duración de cada estado sigue una distribución exponencial. El comportamiento en cada estado a nivel de paquetes (tramas FP) se define mediante los parámetros tiempo entre paquetes y tamaño de paquete, para los que se asume una distribución constante.

El modelo asume fuentes activas (por ejemplo, una conversación telefónica o sesión de navegación web en curso), motivo por el cual no se considera un tercer nivel de sesiones por encima del nivel de ráfagas. Por otro lado, también se supone que el sentido de la comunicación que se modela es el descendente (de RNC a Nodo B), por ser el que habitualmente presenta mayor volumen de tráfico.

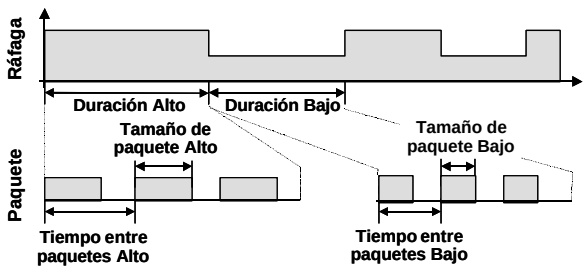


Fig. 2: Modelo de tráfico

Mediante su oportuna parametrización, el modelo es aplicable a las cuatro clases de tráfico UMTS. Así, para llevar a cabo el estudio se ha seleccionado un conjunto de aplicaciones representativo de cada clase: voz (clase conversacional), vídeo streaming (clase *streaming*), navegación web (clase interactiva) y correo electrónico (clase *background*). Los valores resultantes de la particularización del modelo para cada aplicación se muestran en la Tabla 1.

En la Tabla 1 se muestran también los requisitos de QoS a satisfacer sobre la interfaz Iub para cada aplicación. Los parámetros seleccionados son los dos que más directamente influyen en el dimensionado de la interfaz Iub: el ratio de pérdidas de trama (FLR – Frame Loss Ratio) y el máximo retardo de trama (para el 99 percentil o el 95 percentil de las tramas, según se trate de una clase de tiempo real o no).

Los valores de la Tabla 1 se han obtenido a partir de datos de diversos documentos entre los que se incluyen especificaciones del 3GPP y de otros organismos. Se remite al lector a [9][10] para más detalles.

3 Modelo de simulación

A fin de capturar con suficiente detalle las características esenciales de la interfaz Iub se ha optado por el desarrollo de un modelo de simulación. El simulador permite representar diversos escenarios de tráfico y configuraciones (físicas o lógicas) de red. Su arquitectura básica se muestra en la Fig. 3.

El modelo permite establecer uno o más canales virtuales (o trayectos virtuales) ATM de tipo CBR entre RNC y Nodo B. Un mismo trayecto virtual puede, a su vez, transportar flujos de tráfico de una o varias aplicaciones (clases de tráfico), posibilitando de este modo el análisis de diversas estrategias de diferenciación de tráfico. Si bien es cierto que el tráfico que genera cada usuario aislado puede presentar un perfil temporal a ráfagas, el tráfico agregado de un conjunto significativo de comunicaciones que comparten la misma conexión ATM tendrá un carácter menos variable, motivo por el cual se ha considerado adecuada la elección de la categoría CBR en lugar de VBR-rt.

Tabla 1: Parámetros de tráfico y QoS

	Clase / Aplicación			
	Conversacional	Streaming	Interactiva	Background
	Voz	Vídeo	Web	Correo
Parámetros de tráfico				
Duración media Alto/Bajo (s)	3 / 3	5 / 5	1,65 / 60	6,25 / 120
Tamaño de paquete Alto/Bajo (bytes)	40 / 13	171 / 91	331 / 0	171 / 0
Tiempo entre paquetes (ms)	20	20	40	40
Tasa a nivel ATM Alto/Bajo (kbit/s)	19,4 / 7,2	82,5 / 45,1	80,1 / 0	41,3 / 0
Requisitos de QoS en la red de transporte de Iub				
Ratio errores tramas	9,50E-05	2,18E-04	4,00E-05	3,00E-05
Retardo de trama (ms)	10 (99-p)	10 (99-p)	15 (95-p)	15 (95-p)

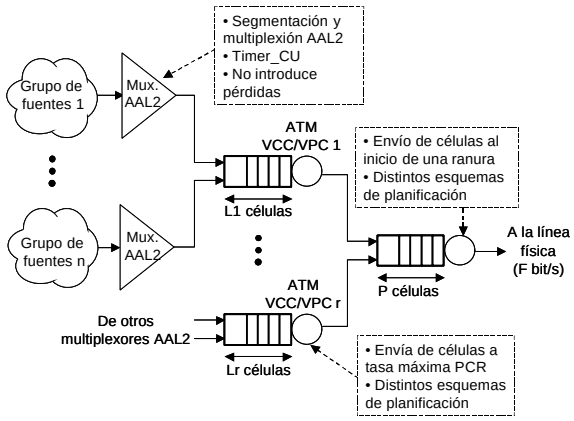


Fig. 3: Arquitectura del simulador

Más específicamente, el simulador permite definir grupos de usuarios (de fuentes de tráfico) basadas en el modelo de tráfico descrito anteriormente. A cada grupo de fuentes se le asocia un multiplexor AAL2, que modela con detalle las funciones de este protocolo [11] (segmentación, reensamblado, multiplexión y manejo del temporizador Timer_CU).

La salida de cada multiplexor AAL2 se dirige a un canal o trayecto virtual ATM, dedicado o compartido por otros multiplexores AAL2 (en cuyo caso, se pueden establecer prioridades). El flujo de células resultante se redirige a un multiplexor ATM, lo que permite modelar la multiplexión de varios canales virtuales (con posibilidad de establecer distintas prioridades entre ellos) sobre un mismo enlace físico.

El simulador permite la configuración de otros parámetros: tasa de bit de línea, tasa de pico (PCR, (Peak Cell Rate) de las conexiones ATM, tamaño de colas ATM, valor de temporizador Timer_CU de AAL2, ... En cuanto a los resultados generados por el simulador, se incluyen los parámetros de QoS más relevantes (pérdidas, retardo, jitter), así como medidas de tráfico (tasas de bit , utilización, ...). Una descripción más detallada del simulador se puede encontrar en [9][12].

4 Resultados

4.1 Dimensionado para clases aisladas

Este primer bloque de experimentos asume la existencia de un canal virtual ATM independiente para cada clase de usuarios. Bajo este supuesto, el objetivo que se plantea es la determinación de la capacidad mínima necesaria para satisfacer a un conjunto de usuarios de una misma clase de QoS. Para ello, se ha procedido a la realización de baterías de experimentos separados para cada clase de tráfico.

Así, para cada una de las cuatro aplicaciones seleccionadas se han realizado simulaciones para distintas poblaciones de usuarios (entre 25 y 150). Para cada número de usuarios, se procede a ir aumentando la capacidad del canal virtual en

incrementos iguales a la tasa de pico de una fuente (correspondiente al valor de tasa Alto en la Tabla 1).

En la Tabla 2 se resumen los parámetros de simulación utilizados en los experimentos. Cabe mencionar la moderada capacidad de las colas empleadas, así como el valor de 1 ms del temporizador Timer_CU. Ello se debe a los estrictos requisitos de retardo a satisfacer en la interfaz Iub.

En cuanto al medio físico, se considera en todos los casos el empleo de una o varias interfaces E1 (en cuyo caso se asume el empleo de multiplexado inverso ATM, IMA).

A partir de los resultados de los experimentos se han obtenido diversas familias de curvas que muestran la influencia del número de usuarios y la capacidad del canal virtual en los dos parámetros de QoS seleccionados. En la Fig. 4 se muestra un ejemplo de los resultados del ratio de pérdida de paquetes (FLR) para la aplicación de vídeo streaming. Junto a los resultados del simulador se muestran los obtenidos con una aproximación analítica basada en un modelo de fluidos [13].

En función del número máximo de usuarios a soportar, la gráfica permite determinar la capacidad mínima necesaria para satisfacer un cierto objetivo de pérdidas. La capacidad se expresa de manera normalizada, tomando como unidad la tasa de pico de un usuario.

Tabla 2: Parámetros de simulación para escenarios de clases aisladas

	Voz	Vídeo	Web	Correo
Capacidad nivel físico	E1	7xE1	10xE1	E1
Cola nivel físico (células)	10			
Cola nivel ATM (células)	nº usuarios	2 x nº usuarios	nº usuarios	nº usuarios
Timer_CU (ms)	1			

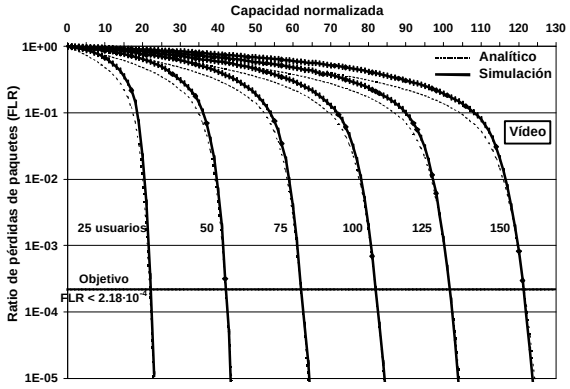


Fig. 4: Ratio de pérdidas de paquetes de vídeo en función de la capacidad disponible normalizada

El análisis de las curvas permite constatar el efecto de la ganancia por multiplexión estadística, en virtud del cual la capacidad total requerida es menor que la suma de las tasas de pico de los usuarios que comparten el canal virtual. Este efecto se analizará con más detalle posteriormente. Las curvas muestran también la existencia de un codo a partir del cual el FLR disminuye bruscamente al incrementar la capacidad. Esto permite concluir la existencia de una capacidad crítica a partir de la cual es posible reducir de manera significativa el FLR.

También es posible observar cómo, a efectos de dimensionado, la estimación analítica proporciona una buena aproximación en la zona de interés (FLR ≤ $2.18 \cdot 10^{-4}$). Esto sucede también en el caso del tráfico de voz. En el caso de web y correo electrónico, la aproximación de fluidos no es tan buena debido a que el tráfico exhibe una mayor intermitencia (comportamiento a ráfagas). En cualquier caso, habida cuenta que el error cometido es siempre menor que una unidad de capacidad normalizada, puede considerarse una aproximación razonable.

En la Fig. 5 se muestra un ejemplo de las curvas de retardo medio y 99-percentil para las tramas de voz. La gráfica permite determinar la capacidad mínima normalizada (con respecto a la tasa de pico de un usuario) necesaria para satisfacer unos ciertos objetivos de retardo.

El análisis de los resultados de retardo permite determinar la existencia de un rango de capacidades relativamente estrecho que, de ser superado, da lugar a una importante disminución en el retardo (medio y 99-percentil). A partir de esta zona, la pendiente se suaviza, indicando una menor sensibilidad del retardo frente a la capacidad. Asimismo, se puede apreciar cómo el codo en las curvas se produce precisamente en la zona donde el 99-percentil para el retardo se encuentra alrededor del requisito impuesto para la voz en Iub. A raíz de estas observaciones, se puede formular como criterio de dimensionado la elección de un punto que se encuentre en la zona de poca sensibilidad (de menor pendiente), pero no muy alejado del codo, siempre que en este punto ya se

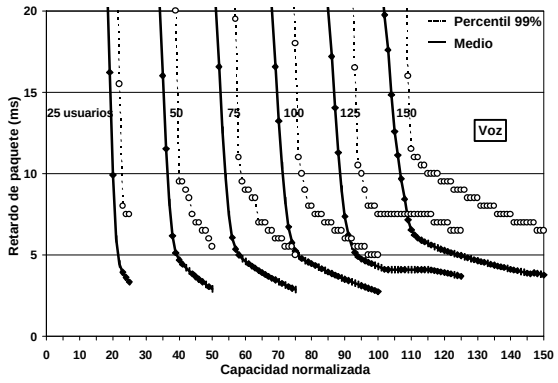


Fig. 5: Retardo de paquetes de voz en función de la capacidad disponible normalizada

cumpla el objetivo máximo para el retardo. De este modo se garantiza el cumplimiento de los objetivos de retardo, manteniéndose éste dentro de una zona estable.

Las curvas de retardo para vídeo streaming, dentro de la correspondiente zona de interés, tienen un aspecto similar a las de la voz. Por otro lado, en los casos de web y de correo electrónico se observa un cambio de pendiente menos brusco.

A partir de las curvas de pérdidas y retardos obtenidas para cada aplicación, y en base a los objetivos de QoS establecidos en la Tabla 1, se pueden derivar unas pautas de dimensionado para la interfaz Iub. El resultado se representa en la Fig. 6.

Las curvas de la Fig. 6 indican, para cada tipo de aplicación y en función del número máximo de usuarios, la capacidad mínima necesaria en kbit/s (a nivel ATM) que satisface simultáneamente los dos requisitos de QoS. En otras palabras, los valores representados corresponden a la mayor de las capacidades requeridas para cumplir el criterio de retardo o el de pérdidas. En cualquier caso, cabe señalar que ambos criterios arrojan capacidades mínimas similares debido a que los dos objetivos de QoS son bastante estrictos.

A partir de las curvas de dimensionado, resulta interesante analizar el efecto de la ganancia por multiplexión estadística. Ésta se puede expresar como el cociente entre el número de usuarios a soportar y la capacidad mínima normalizada. El resultado se muestra en la Fig. 7.

Lógicamente, cuanto mayor es el número de usuarios, mayor es la ganancia por multiplexión estadística. Al comparar las dos gráficas, se observa que la ganancia para la voz y el vídeo es menor (hasta un orden de magnitud) que para el web y el correo electrónico. La explicación está en el marcado comportamiento a ráfagas del tráfico generado por estas dos últimas aplicaciones. Los largos periodos de silencio hacen bastante improbable la coincidencia en el tiempo de una proporción elevada de fuentes en estado Alto. En consecuencia, la capacidad requerida es mucho menor que la suma de tasas de pico de cada fuente.

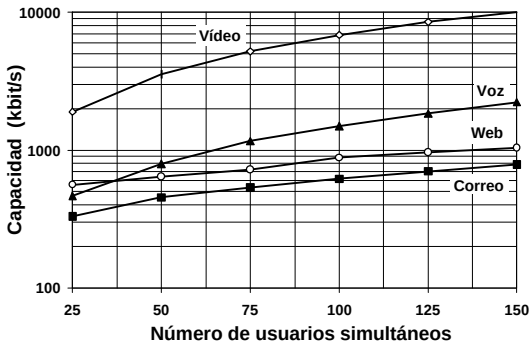


Fig. 6: Capacidad mínima para cumplir requisitos de QoS

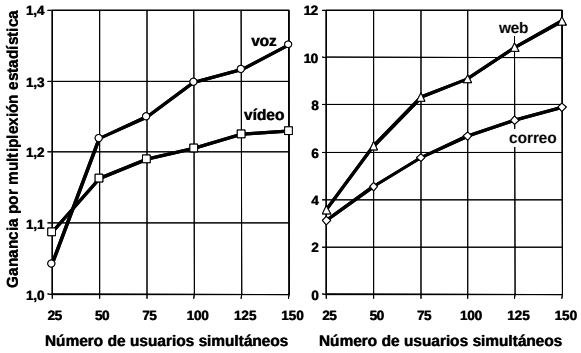


Fig. 7: Ganancia por multiplexión estadística

Nótese que una mayor ganancia por multiplexión estadística no implica un mejor aprovechamiento de capacidad, ya que el factor de utilización también es proporcional al cociente entre las tasas medias y de pico de las fuentes. En el caso de web y de correo electrónico (las aplicaciones con mayor ganancia), el cociente es bastante bajo, por lo que la utilización es más pobre que para las aplicaciones de voz y vídeo.

El diferente grado de aprovechamiento de los recursos de transmisión observado para cada tipo de aplicación conduce a plantear el segundo escenario de simulación, al que se dedica el siguiente apartado.

4.2 Dimensionado para mezcla de clases

En este escenario de simulación se considera la multiplexión de varias clases de tráfico sobre un mismo trayecto virtual. El objetivo es investigar el posible ahorro de recursos de transmisión que puede obtenerse mediante la compartición de una misma capacidad entre usuarios de varias clases. Para ello, se lleva a cabo un conjunto de experimentos contemplando el empleo de un mismo trayecto virtual ATM para usuarios de voz y de web.

En todas las simulaciones el número total de usuarios se mantiene constante (100 usuarios), realizándose experimentos para diversas proporciones de usuarios de web y de voz. Concretamente, se consideran tres casos: a) 25 usuarios de voz y 75 de web, b) 50 de voz y 50 de web y c) 75 de voz y 25 de web. En lo que sigue, utilizaremos la notación 25/75, 50/50 y 75/25, respectivamente, para referirnos a cada caso. Los parámetros de simulación utilizados en los experimentos se resumen en la Tabla 3.

A la hora de multiplexar los flujos de células ATM de cada grupo de usuarios, y teniendo en cuenta los diferentes requisitos de QoS de cada aplicación, se plantea la posibilidad de utilizar distintas estrategias de planificación. Esto nos lleva a considerar dos estrategias simples: FIFO (First In First Out, esto es, sin distinción) y PRIO (prioridad absoluta de la voz frente al web).

Tabla 3: Parámetros de simulación para escenarios de mezcla de clases

Capacidad nivel físico	2xE1
Cola nivel físico (células)	10
Cola nivel ATM (células)	100
Timer_CU (ms)	1

Como primera aproximación al problema, se decide partir de los resultados obtenidos en el escenario de clases aisladas (al que nos referiremos como estrategia SEP). Así, en los primeros experimentos se fija la capacidad del trayecto virtual a un valor igual a la suma de capacidades individuales requeridas por cada grupo de usuarios. Al analizar los resultados, se comprueba una notable disminución de las pérdidas, siendo en ocasiones nulas durante el período simulado.

La explicación de este fenómeno está en la utilización de una cola compartida (de tamaño igual a la suma de las empleadas en la estrategia SEP). Esto permite que una clase de tráfico disponga de más buffer durante los intervalos en los que la otra genera menos tráfico, de manera que se reducen las situaciones de desbordamiento.

En cuanto al retardo, el comportamiento varía según la estrategia de planificación y la proporción de usuarios de voz y de web. En la Tabla 4 se muestran los retardos máximos de trama para los distintos escenarios considerados.

En el caso de FIFO, se aprecia una reducción del retardo con respecto a la estrategia SEP salvo para la voz en la mezcla 25/75. El retardo en este caso resulta algo mayor, aunque sin rebasar el valor máximo objetivo. La estrategia PRIO logra en todos los casos reducir el retardo para la voz (99-percentil), siendo el resultado mejor cuanto menor es la proporción de usuarios de este tipo. En contrapartida, el retardo para el web (95-percentil) se ve perjudicado conforme aumenta la proporción de usuarios de voz.

Los resultados del retardo se entienden mejor al analizar las distribuciones de retardo de los paquetes. Así, en el caso de la voz se ha podido constatar con ambas estrategias una disminución significativa del "jitter" (variación de retardo), algo muy deseable cuando se trata de voz empaquetada. En el caso del tráfico web, el método PRIO provoca una mayor dispersión de retardos, mientras que FIFO la reduce.

Tabla 4: Retardo máximo de trama (ms)

	Mezcla	**SEP**	**FIFO**	**PRIO**
Retardo para voz 99%-percentil (ms)	**25/75**	7,5	10	1,5
	50/50	9,5	8,5	2
	75/25	9	6,5	3,5
Retardo para web 95%-percentil (ms)	**25/75**	15	10,5	13
	50/50	12,5	9,5	13
	75/25	12	8,5	14

El buen comportamiento de FIFO en la mayoría de los casos analizados se explica por la predominancia del tráfico de voz frente al de web en todos los escenarios de mezcla de usuarios considerados. Esto ocurre incluso en el caso 25/75, ya que 25 usuarios de voz generan más tráfico que 75 de web. Esta predominancia de tráfico de voz resta parte de su eficacia a la estrategia PRIO, puesto que da prioridad precisamente al grupo de usuarios que más tráfico genera.

En definitiva, puede concluirse que, en muchas situaciones prácticas, la compartición de capacidad junto con una simple estrategia FIFO constituye una política de gestión de tráfico eficaz para la interfaz Iub. Los resultados prueban que este tipo de soluciones puede redundar en una mejora de prestaciones frente al enfoque de capacidades segregadas. Esto da pie a considerar la posibilidad de, manteniendo los objetivos de pérdida y de retardo, utilizar una capacidad compartida menor que la suma de las requeridas en el escenario de clases separadas. Para acometer este nuevo estudio, se efectúa una tanda de simulaciones donde la capacidad de partida (a la que nos referiremos como 100%) se va disminuyendo de manera gradual.

Al analizar el comportamiento del FLR al disminuir la capacidad, se observan idénticos resultados para los métodos FIFO y PRIO. Al reducir la capacidad, el FLR aumenta, siendo el tráfico web el más afectado y, por tanto, el que limita el máximo ahorro. La Fig. 8 muestra la evolución del FLR para los paquetes de web en función de la capacidad. Dependiendo de la mezcla de voz y web considerada, se confirma la posibilidad de reducir la capacidad entre un 8% y un 16% sin comprometer el objetivo de FLR.

El comportamiento del retardo con la capacidad varía según la estrategia de multiplexión considerada. Así, en el caso de FIFO, es el retardo de la voz el que limita el máximo ahorro de capacidad permisible. Como se indica en la Fig. 9, éste oscila entre un 0% y un 19%, según la mezcla de tráfico considerada.

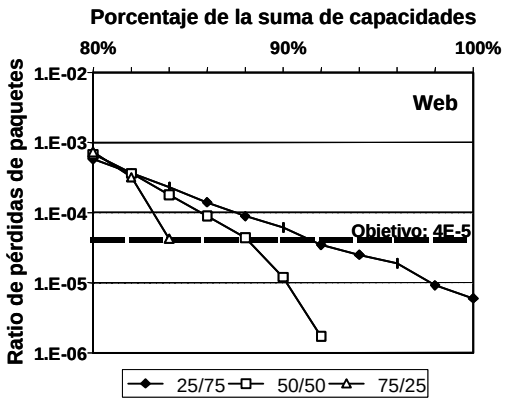


Fig. 8: Evolución del ratio de pérdidas de paquetes de web con la capacidad

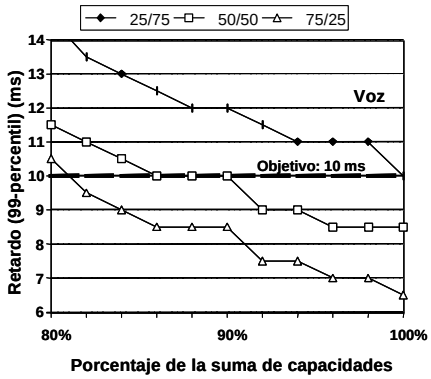


Fig. 9: Evolución del retardo de paquetes de voz con la capacidad (FIFO)

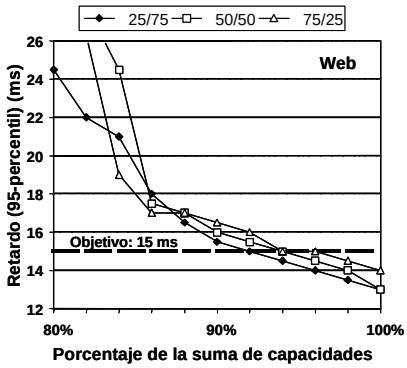


Fig. 10: Evolución del retardo de paquetes web en función de la capacidad (PRIO)

En la estrategia PRIO, por el contrario, el criterio más restrictivo lo impone el retardo máximo tolerable para los paquetes de web. En este caso, el ahorro máximo varía entre un 4% y un 8%, tal como se observa en la Fig. 10.

En la Tabla 5 se muestra un cuadro resumen con los ahorros máximos de capacidad que, para los distintos casos simulados, permiten la satisfacción simultánea de los objetivos de pérdidas y de retardos.

Tabla 5: Porcentajes máximos de ahorro de capacidad para los escenarios simulados

		Mezcla		
		25/75	**50/50**	**75/25**
FIFO	*Límite por pérdidas (web)*	*8%*	*12%*	*16%*
	Límite por retardo (voz)	*0%*	*10%*	*19%*
	Ahorro máximo posible	**0%**	**10%**	**16%**
PRIO	*Límite por pérdidas (web)*	*8%*	*12%*	*16%*
	Límite por retardo (web)	*8%*	*6%*	*4%*
	Ahorro máximo posible	**8%**	**6%**	**4%**

A la vista de los resultados, podría deducirse que la estrategia de compartición de capacidad entre clases de tráfico no permite un ahorro significativo de recursos de transmisión en la interfaz Iub. Esta afirmación puede ser válida para topologías de interconexión en estrella entre Nodos B y RNCs realizadas mediante E1s. En la práctica, sin embargo, cabe la posibilidad de considerar topologías con concentración de tráfico (por ejemplo, interconexión de Nodos B en cadena o a través de un backbone ATM), donde una ligera disminución de capacidad en cada Nodo B puede suponer un ahorro significativo de recursos de transmisión en alguno de los tramos de la red de acceso.

Por ejemplo, supóngase un conjunto de cuatro Nodos B conectados a un RNC mediante una topología en cadena, cada uno de ellos soportando una mezcla de tráfico de 75 usuarios de voz y 25 de web (esto es, el caso 75/25 en los experimentos). Bajo el escenario de clases aisladas, cada Nodo B requeriría una capacidad de transmisión de 1725 kbit/s, mientras que con una estrategia FIFO cabría considerar una capacidad compartida un 16% menor (esto es, 1449 kbit/s). En el primer caso, el tramo inicial desde el RNC al primer Nodo B requeriría un grupo IMA con cuatro E1 (para acomodar 4 x 1725 = 6900 kbit/s). Sin embargo, en el segundo caso bastaría con tres E1 (4 x 1449 = 5796 kbit/s, que pueden transportarse sobre un grupo IMA de 3 E1).

Teniendo en cuenta que en un escenario real de despliegue de una red de acceso UMTS el número de Nodos B puede ser muy elevado, cabe esperar que puedan plantearse con frecuencia situaciones similares a la descrita.

A modo de ejemplo, en la Fig. 11 se muestra una topología física de interconexión entre Nodos B y sus controladores en la que intervienen conmutadores ATM y enlaces de larga distancia que concentran tráfico de varias células. En estas circunstancias, la estrategia de compartición de capacidad puede suponer un considerable ahorro de costes para el operador.

5 Conclusiones y trabajos futuros

En este artículo se ha abordado el problema del dimensionado de recursos de transmisión en la red de acceso terrestre de UMTS. Mediante el empleo de técnicas de simulación, se han analizado diversos escenarios de mezclas de tráfico correspondientes a las cuatro clases de QoS definidas por el 3GPP.

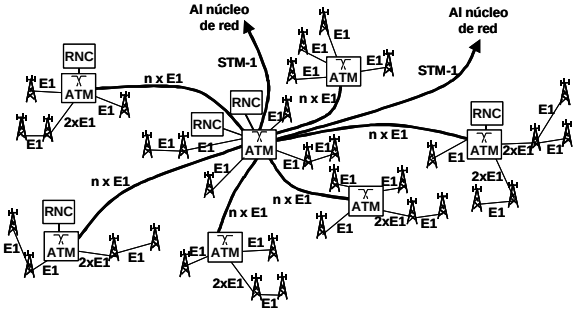


Fig. 11: Ejemplo de topología de red de acceso UMTS mediante backbone ATM

Establecidos los requisitos de QoS a satisfacer para cada clase de tráfico en la interfaz Iub, se ha estimado la capacidad mínima necesaria según el número de usuarios a soportar. En una primera aproximación, se ha considerado un escenario de segregación de capacidades por clase de tráfico. Posteriormente, se han estudiado algunas estrategias básicas de compartición de capacidad entre clases de tráfico. Los resultados han demostrado que este segundo enfoque puede contribuir a una ligera mejora de prestaciones o, alternativamente, posibilitar el ahorro de recursos de transmisión en la interfaz Iub. Si bien el ahorro no parece significativo al considerar los Nodos B de manera aislada, en topologías con concentración de tráfico puede suponer importantes ventajas económicas.

Dentro de las líneas de continuación de este trabajo se contempla la ampliación del estudio a topologías de interconexión entre Nodos B y RNC más complejas, especialmente en escenarios con concentración de tráfico. Adicionalmente, se pretende extender el análisis al resto de las interfaces de la UTRAN (Iu e Iur). Por último, y en línea con los trabajos que actualmente se desarrollan en el 3GPP, se plantea también la adaptación del simulador para su aplicación a redes de acceso UMTS basadas en tecnología IP.

Referencias

[1] 3GPP. “Quality of Service (QoS) concept and architecture”. 3GPP TS 23.107.

[2] O. Isnard, J.M. Calmel, A.L. Beylot, G. Pujolle. “Handling Traffic Classes at AAL2 / ATM layer over the Logical Interfaces of the UMTS Terrestrial Access Network”. 11th IEEE International Symposium on Personal, Indoor and Mobile Radio Communication, PIMRC. London (Inglaterra), Septiembre 2000. ISBN: 0780364635.

[3] M. Menth, M. Schmid, H. Heiß, R. Reim. “MEDF - A Simple Scheduling Algorithm for Two Real-Time Transport Service Classes with Application in the UTRAN”. Technical Report No. 294, University of Würzburg, Febrero 2002.

[4] J.H. Chung, Y.H. Kwon, K.H. Cho, D.K. Sung, O.H. Jang. “Performance Evaluation of an AAL2 Link Transmission Scheme for Voice and Data Packets in BS-BSC Links”. IEEE 52nd Vehicular Technology Conference 2000, VTC 2000 Fall, Vol. 4, 2000. pp. 1610-1614. ISBN: 0780365070.

[5] S. Yoo, H. Park. “Quality-of-Service Provisioning for Mobile Voice and Data Services over ATM Network using AAL2”. 3rd International Conference on Advanced Communication Technology, 3rd ICACT, Muju (Corea), Febrero 2001.

[6] G. Eneroth, G. Fodor, G. Leijonhufvud, A. Rácz, I. Szabó. “Applying ATM/AAL2 as a Switching Technology in Third-Generation Mobile Access Networks”. IEEE Communications Magazine, Vol. 37, No. 6, Junio 1999. pp. 112-122. ISSN: 0163-6804.

[7] ETSI. “Universal Mobile Telecommunications System (UMTS); Selection procedures for the choice of radio transmission technologies of the UMTS”. TR 101 112 V3.2.0, Abril 1998.

[8] A. Klemm, C. Lindemann, M. Lohmann. “Traffic Modeling and Characterization for UMTS Networks”. IEEE Global Telecommunications Conference 2001, GLOBECOM’2001, Internet Performance Symposium, San Antonio, TX, Noviembre 2001. ISBN: 0780372069.

[9] A.B. García, E. García, M. Álvarez-Campana, J. Berrocal, E. Vázquez. “A Simulation Tool for Dimensioning and Performance Evaluation of the UMTS Terrestrial Radio Access Network”. Lecture Notes in Computer Science (“Protocols and Systems for Interactive Distributed Multimedia”), Volume 2515, Noviembre 2002, pp. 49-60. Ed. Springer-Verlag. ISSN: 0302-9743, ISBN: 3-540-00169-7.

[10] A.B. García. “Técnicas de dimensionado y soporte de calidad de servicio para redes de acceso de sistemas de comunicaciones móviles UMTS”. Tesis Doctoral, Universidad Politécnica de Madrid, E.T.S.I. Telecomunicación, 2002.

[11] ITU-T. “B-ISDN ATM Adaptation layer specification: Type 2 AAL”. ITU-T Rec. I.363.2.

[12] A.B. García, M. Alvarez-Campana, E. Vázquez, J. Berrocal. “Simulation of Quality of Service Mechanisms in the UMTS Terrestrial Radio Access Network”. Fourth IEEE Conference on Mobile and Wireless Communications Networks, MWCN 2002, Stockholm (Suecia), Septiembre 2002. ISBN: 0-7803-7606-4.

[13] O. Hersent, D. Gurle, J.P. Petit. “IP Telephony: Packet-based multimedia communications systems”, Addison-Wesley (2000). ISBN: 0-201-61910-5.

Análisis y Modelado de la Red de Datos de un Operador de Cable

M. García, V.G. García, X.G. Pañeda, R. Bonis
Departamento de Informática. Universidad de Oviedo
Campus Universitario de Viesques. Edificio Departamental Oeste
33204 Gijón, Asturias
Teléfono: 985 18 25 19 Fax: 985 18 19 86
E-mail: manuel@atc.uniovi.es, (victor, xabiel)@correo.uniovi.es, rbonis@telecable.es

Abstract. *This paper describes a study done into the network of a cable operator which provides data services using hybrid fiber-coax (HFC) technology. The objective of the study is to develop a network model able to predict traffic evolution on the network channels, related to the number of subscribers assigned by the network operator to each channel.*

The process followed in the development of the network model begins with the analysis of the traffic measurements taken on the network channels; then, based on the results of the analysis, a traffic model is proposed. Finally, the traffic model is combined with a physical representation of the network to produce the final network model. The results obtained throughout the process have been compared with the real measurements showing a valid statistical adjustment.

1 Introducción

En la última década hemos asistido al desarrollo de las llamadas tecnologías de la información, motivado fundamentalmente por el acceso a *Internet*. Inicialmente el medio de acceso estaba restringido casi exclusivamente a líneas telefónicas convencionales de baja velocidad. La expansión del fenómeno de *Internet* hizo que se plantearan nuevas alternativas de acceso, una de ellas fueron las redes de televisión por cable, más conocidas como redes de cable.

Las redes de cable permitían ofertar un ancho de banda muy superior a las líneas telefónicas convencionales, si bien previamente debían superar una limitación: la necesidad de establecer un canal de retorno. Así, mientras la difusión de la señal de televisión es unidireccional, en cambio, la transmisión de datos ha de ser bidireccional. La implementación del canal de retorno ha supuesto el principal reto de las redes de cable. En las redes de cable antiguas, basadas exclusivamente en cables coaxiales, la implementación de un canal de retorno resulta un proceso costoso y complejo, por lo que su uso para la transmisión de datos es reducido. En cambio, existe un grupo de redes modernas basadas en la combinación de cable coaxial y fibra óptica, donde resulta sencillo la implementación del canal de retorno; estas redes reciben el nombre de redes híbridas de fibra y coaxial (HFC) y allí donde están presentes han alcanzado un elevado nivel de aceptación como medio de acceso.

Las redes HFC, sin embargo, no han sido estudiadas en demasiada profundidad a nivel de prestaciones para la transmisión de datos. La mayoría de los estudios existentes, están relacionados con el proceso de estandarización de un protocolo de control de acceso al medio (MAC) para el canal de retorno, y en menor medida con las prestaciones del protocolo TCP sobre las redes de cable. El trabajo presentado en [1] recoge las diferentes propuestas existentes para los protocolos MAC del canal de retorno, si bien, dos son las más importantes: la propuesta inconclusa del IEEE 802.14 y el estándar *de facto* DOCSIS (Data Over Cable Service Interface Specification); aunque ninguna de las dos propuestas es claramente superior a la otra, como se indica en [2].

Aparte de estos estudios, sólo un reducido número de trabajos aborda el problema de las prestaciones de las redes de cable para la transmisión de datos. Entre los trabajos más destacados se pueden citar: [3], donde se analiza la capacidad de una red HFC para soportar transmisiones multimedia utilizando ATM, e implementada sobre una red experimental, y [4], donde se evalúan, mediante modelado analítico y simulación, las prestaciones que percibe el abonado accediendo a páginas *web* mediante una red de cable.

En este trabajo se propone un modelo de simulación de la red HFC del operador TELECABLE, S.A.. El modelo se ha obtenido a partir del análisis de las medidas de tráfico tomadas en la red de datos del operador. El modelo de simulación propuesto permite establecer la necesidad de ancho de banda en cada canal en función de dos parámetros, el número de abonados que han sido asignados por el operador a cada canal y el momento del día considerado. A partir del modelo de simulación se pueden obtener otras medidas de prestaciones, tales como la utilización o ancho de banda requerido en el acceso a *Internet*, o la productividad de los distintos canales de la red.

En la sección 2 se describen las particularidades de

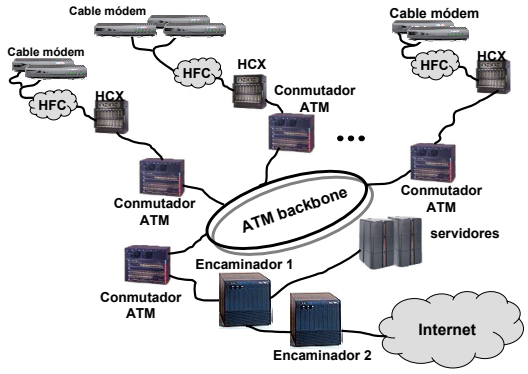


Figura 1: Arquitectura de la red del operador.

la red del operador de cable, así como las medidas disponibles. Seguidamente, en la sección 3 se resumen los resultados más relevantes obtenidos del análisis de las medidas, utilizados posteriormente en el modelo. El desarrollo del modelo se divide en dos partes, el modelo de tráfico, al que se dedica la sección 4 y el modelo global descrito en la sección 5. Los resultados obtenidos y su comparación con los valores medidos se muestran en la sección 6. Finalmente, la sección 7 presenta las conclusiones obtenidas con este trabajo.

2 Descripción de la red

En este artículo se considera la red de datos del operador TELECABLE, S.A.. Este operador, creado en 1995, inicialmente suministraba servicios de televisión por cable. Con el desarrollo de las tecnologías de la información ha adaptado su red para convertirse en un operador global, suministrando servicios de televisión, voz y acceso a *Internet*. Su red de datos está estructurada en un conjunto de redes HFC paralelas, llamadas ramas HFC, e interconectadas por un anillo ATM central. Utilizando esta red, el operador cubre un área geográfica formada por tres ciudades y más de medio millón de habitantes. En la Fig. 1 se puede ver una representación de la arquitectura de red del operador.

Cada una de las ramas HFC en las que se estructura la red, está formada por un canal de bajada, y hasta 6 canales de retorno. El canal de bajada es el canal por el cual los abonados de la rama reciben los datos, tiene una capacidad de aproximadamente 30 Mbps, y es único para todos los abonados. Los canales de retorno son los utilizados por los abonados para enviar sus peticiones, tienen una capacidad más reducida, en torno a 1.9 Mbps y el operador asigna a cada abonado un canal de retorno. La gestión de los canales la realiza el elemento denominado controlador (representado como HCX en la Fig. 1). Este elemento realiza además labores de filtrado del tráfico con origen y destino en la propia rama.

Las condiciones de servicio de los abonados son las siguientes: tarifa plana de conexión durante las 24 horas del día, una capacidad máxima de 128Kb en el canal de bajada y 64Kb en el canal de retorno, la calidad de servicio es del tipo "*best effort*", es decir, el ancho de banda disponible se reparte entre todos los abonados conectados.

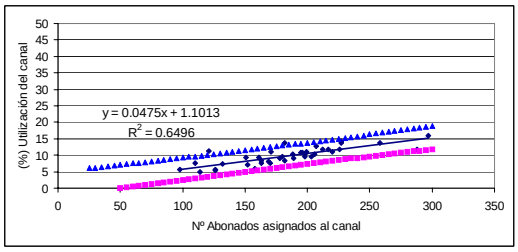
Sobre esta red y en estas condiciones de funcionamiento, se han tomado las medidas de tráfico en los controladores de cada rama HFC, tanto para los canales de bajada como de retorno, así como en el encaminador de acceso a *Internet*. Las medidas de tráfico se han tomado con la herramienta MRTG (Multi-Router Traffic Graph), que permite registrar el tráfico en cada canal y generar un fichero de traza de tamaño fijo. Su forma de trabajar es la siguiente: la herramienta toma medidas constantemente con una resolución de 5 minutos, al cabo de 600 medidas, las seis medidas más antiguas se promedian para obtener una medida de 30 minutos de resolución; el proceso se repite para otras resoluciones. El fichero de traza final registra 600 + 600 + 600 + 730 medidas de 5 minutos, 30 minutos, 2 horas y 24 horas de resolución. De esta forma, el fichero de traza es capaz de registrar un periodo ligeramente superior a dos años de la evolución de tráfico, manteniendo constante el tamaño del fichero de traza.

Se dispone de medidas en dos fechas distintas, con aproximadamente un año de diferencia: 8 de enero de 2001 y 16 de enero de 2002. Entre esas fechas, la situación de la red de datos ha evolucionado significativamente, pasando de 8.331 abonados y 10 ramas HFC a 17.369 abonados y 18 ramas HFC. A pesar del espectacular crecimiento experimentado, el número de abonados por canal de retorno se ha mantenido aproximadamente estable.

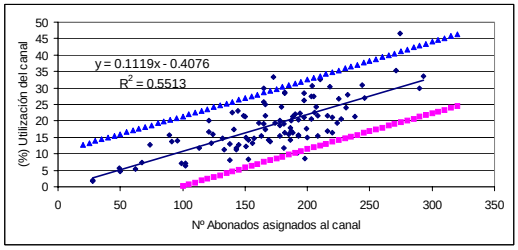
Aparte de las medidas de tráfico en los canales, la información disponible sobre la red de cable del operador, se completa con el número de abonados asignados por el operador a cada uno de los canales de retorno en ambas fechas.

3 Análisis del tráfico

Las medidas de tráfico se han analizado para obtener información sobre las características del tráfico en la red. El grupo de medidas comprende 7 canales de bajada y 39 de retorno para las medidas de 2001, y 18 canales de bajada y 102 de retorno para las medidas de 2002. El análisis realizado contempla aspectos generales del tráfico: la evolución del tráfico a lo largo del tiempo y su periodicidad; el análisis de las propiedades estadísticas del tráfico: valor medio, valor de pico, relación entre tráfico de pico y tráfico medio; así como otras propiedades estadísticas más específicas como la autocorrelación y la autosemejanza ó "self-similarity". En esta

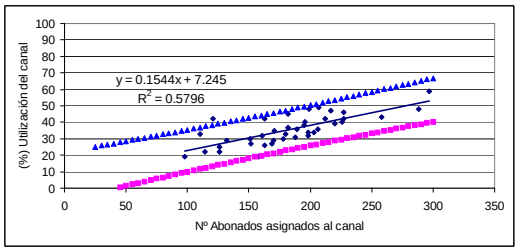


(a) Canales de retorno, medidas de 2001

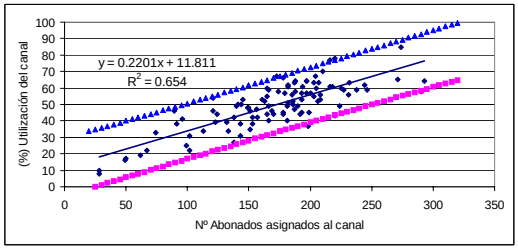


(b) Canales de retorno, medidas de 2002

Figura 2: Relación tráfico medio y nº abonados.



(a) Canales de retorno, medidas de 2001



(b) Canales de retorno, medidas de 2002

Figura 3: Relación tráfico de pico y nº abonados.

sección se recogen, de forma resumida, los resultados más importantes obtenidos en el análisis, y que posteriormente, se utilizarán en la construcción del modelo de la red. Una versión más detallada del análisis realizado puede encontrarse en [5].

Independientemente de los cambios ocurridos en la red en el año transcurrido entre las dos medidas, el número de abonados por canal de retorno se ha mantenido aproximadamente constante, en cambio, los valores de tráfico registrados en cada canal se han incrementado considerablemente. Esta variación del tráfico se produce en todos los canales, pero es más acusada en los canales de retorno. En los canales de retorno se observa un acusado incremento de los valores máximos de tráfico y en mayor medida de los valores mínimos, así mientras en el año 2001 existían periodos de actividad muy baja o nula en los canales, en el 2002 estos periodos han desaparecido, tendiendo el tráfico a ser más uniforme a lo largo del tiempo. La principal explicación de este comportamiento se encuentra en el elevado uso de las aplicaciones punto a punto ("peer to peer" ó P2P), que permiten el intercambio continuado de información de forma automática.

Otro resultado del análisis es que independientemente de la tarifa plana, el tráfico en los canales de la red sigue un patrón de comportamiento humano de aproximadamente 24 horas; si bien, en las medidas de 2002 comienzan a aparecer frecuencias secundarias que indican periodos de utilización más continuados.

Uno de los objetivos del análisis fue establecer relaciones entre los valores del tráfico y el número de abonados. Se estudió primeramente la evolución del valor medio de tráfico en los canales, respecto al número de abonados asignados por el operador a cada canal. El análisis gráfico de los puntos obtenidos mostró que se podían aproximar mediante un modelo de regresión lineal. En la Fig. 2 se muestran los gráficos obtenidos para los canales de retorno, para los dos grupos de medidas considerados. En cada gráfico los puntos se aproximan mediante una línea de tendencia, cuya ecuación y coeficiente de determinación se muestran en el gráfico. Los gráficos se completan con los intervalos de confianza para el modelo de regresión lineal, para un 95% de nivel de confianza.

Del mismo modo, se estableció la relación entre los valores de pico del tráfico y el número de abonados asignados a cada canal. La Fig. 3 muestra el análisis gráfico realizado en los canales de retorno para los dos grupos de medidas disponibles.

Las conclusiones obtenidas de este análisis, es que el uso de los canales es aproximadamente proporcional al número de abonados que tienen asignados, quiere esto decir que la proporción de usuarios conectados y el uso que éstos hacen del canal, es similar en todos los canales. Esta relación se mantiene a lo largo del tiempo, si bien varían los parámetros que relacionan el tráfico y el número de abonados. En los gráficos se observa como de un año al siguiente se incrementan los valores de la pendiente de las líneas de regresión, indicando un mayor volumen de tráfico.

Otros resultados importantes obtenidos están relacionados con la propiedad estadística de autosemejanza. Esta propiedad, que representa la invarianza del tráfico respecto a la escala de observación, fue detectada en las redes de área local por primera vez en el trabajo [6]. Desde entonces ha sido observada en otros muchos tipos de tráfico, llegando a considerarse como una propiedad inherente al tráfico de los nuevos sistemas de

comunicación. La propiedad de autosemejanza se explica por la existencia de dependencias de efecto prolongado (LRD), frente a las dependencias a corto plazo (SRD), que eran las únicas consideradas hasta entonces.

La existencia de la propiedad de autosemejanza en el tráfico se detecta por la forma de la función de autocorrelación. Una función de autocorrelación que decae rápidamente indica relaciones del tipo SRD, mientras que una función de autocorrelación que decae lentamente es indicativa de la existencia de relaciones del tipo LRD y por tanto de la propiedad de autosemejanza.

En la Fig. 4 se muestra la forma de la función de autocorrelación para el tráfico en uno de los canales de bajada de la red HFC. Se observa que la función de autocorrelación decae lentamente, lo que indica la existencia de la propiedad de autosemejanza. Esta forma de la función de autocorrelación ha sido obtenida, salvo una excepción, en los 162 canales analizados.

El nivel de autosemejanza se mide utilizando el coeficiente de *Hurst*, H. El tráfico cumple la condición de autosemejanza cuando el valor de H está comprendido en el intervalo $[0.5 \leq H < 1]$. Cuanto más próximo a 1 esté el valor de H, mayor es el grado de autosemejanza. Para calcular el valor de H existen varios métodos, (en [7] se presenta una recopilación de los métodos existentes) la mayoría de ellos basados en técnicas heurísticas. El método empleado para las medidas disponibles se conoce como método de la varianza agregada. Este método se basa en representar la variación del logaritmo de la varianza de diferentes agrupaciones de datos, respecto al logaritmo del tamaño de la agrupación; a partir de la pendiente de la línea de regresión se obtiene el valor del parámetro H. Utilizando el método de la varianza agregada se han obtenido valores del parámetro H superiores a 0.5 en todos los casos, lo que confirma la autosemejanza del tráfico en los canales de la red de cable. Para las medidas de 2001 los valores están en el intervalo $(0.785, 0.990)$ y en el intervalo $(0.8, 0.98)$ para las medidas de 2002.

4 Modelo de tráfico

El objetivo del análisis fue determinar las principales propiedades del tráfico en la red y establecer relaciones útiles, para posteriormente construir un modelo de red. El modelo de red permitirá evaluar el comportamiento de la red de cable ante nuevas condiciones de funcionamiento.

En el desarrollo del modelo de red, ha de distinguirse entre el modelo de tráfico y el modelo físico de la red. El modelo de tráfico representa el uso que los abonados hacen de la red, mientras que el modelo físico incorpora el modelo de tráfico sobre

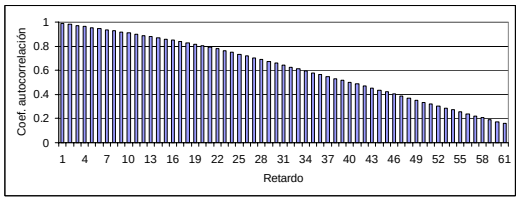


Figura 4: Función de autocorrelación del tráfico.

la arquitectura de la red existente. En esta sección se desarrolla el modelo de tráfico, dejando para la sección siguiente el modelo completo de la red.

El desarrollo de modelos de tráfico ha sido paralelo a la evolución de los sistemas de comunicación, aunque pueden distinguirse dos etapas marcadas por la comprobación de la propiedad de autosemejanza. La primera etapa, correspondiente a los modelos de tráfico tradicionales, basados fundamentalmente en procesos de Poisson y Markov. En [8] se resumen los modelos de tráfico utilizados hasta ese momento. La segunda etapa, viene marcada por la propiedad de autosemejanza, una vez que se comprobó que los modelos tradicionales fallaban en sus predicciones al no contemplar las relaciones de tipo LRD asociadas a la autosemejanza. Por tanto, los nuevos modelos de tráfico han de considerar la naturaleza autosemejante del tráfico.

Varias son las alternativas existentes para generar tráfico de naturaleza autosemejante, algunas de ellas son los modelos: $PT \bigotimes \lambda$, *M/Pareto* y *N-Burst*. Estos modelos se basan en la utilización de distribuciones con un sesgo muy grande, conocidas en la literatura como "*heavy-tailed*", de las cuales la distribución de Pareto es el ejemplo más conocido. Otra alternativa la constituye el uso de modelos denominados ON/OFF. En estos modelos una fuente distingue entre un periodo de emisión, ON, y un periodo de silencio, OFF; eligiendo la distribución de uno de los periodos del tipo Pareto se obtiene tráfico de naturaleza autosemejante.

En este artículo se propone un modelo de tráfico capaz de reproducir el tráfico en la red y su naturaleza autosemejante. Este modelo de tráfico supera las limitaciones de los modelos de tráfico existentes:

- Mientras que en la mayoría de modelos de tráfico existentes están limitados a una clase de tráfico en particular, en este modelo se considera el tráfico agregado en la red, sin realizar ningún tipo de distinción.
- Más importante si cabe, es que en los modelos existentes el tráfico se genera a partir de un conjunto de parámetros, pero no existe o no es sencillo determinar la relación de estos parámetros con el sistema real. En

cambio, los parámetros del modelo de tráfico desarrollado se obtienen directamente de la información disponible de la red de cable: el número de abonados asignados a cada canal y el instante de tiempo considerado.

El modelo de tráfico debe representar adecuadamente el tráfico en los canales de retorno, que representan la interacción de los abonados con la red, estando el tráfico en los canales de bajada influenciado por el tráfico en los canales de retorno. Para reproducir los valores del tráfico en los canales de retorno han de considerarse tres elementos: las características del canal de retorno, el protocolo de control de acceso al medio (MAC) y el perfil de usuario.

4.1 El canal de retorno

Cada canal de retorno se representa como un medio compartido con una tasa de transmisión efectiva de 1.5525 Mbps. Se puede considerar el canal de esta forma puesto que en el momento de realizar las medidas, no se soportaban calidades de servicio y todos los abonados se reparten el canal.

4.2 Control de acceso al medio

El protocolo empleado en los canales de retorno es un protocolo propietario similar al DOCSIS. El acceso al canal se organiza en tramas, enviándose una trama cada 102.4 milisegundos. La trama está compuesta de 512 celdas ATM, de las cuales sólo 414 son efectivas. Del total, cada cable módem puede utilizar un máximo de 63 celdas por trama, aunque el tamaño máximo está controlado por el operador en función del ancho de banda asignado; para un ancho de banda de 64Kb el máximo permitido es de 16 celdas por trama.

La forma de trabajo del protocolo MAC hace que sobre el canal se produzca una superposición de fuentes de tipo ON/OFF, generando ráfagas, que son las causantes de la autosemejanza del tráfico. En la Fig. 5 se muestra este comportamiento; cada cable módem puede transmitir hasta 16 celdas en una trama; su periodo ON; teniendo que esperar hasta la siguiente trama para continuar transmitiendo; periodo OFF. Como existen múltiples cable módem en cada canal, el resultado final sería la superposición de periodos de comunicación con periodos de silencio intercalados.

4.3 Perfil de usuario

Este es el aspecto más importante en la definición del modelo de tráfico, pues representa la forma en la que los abonados interactúan con la red. Deben considerarse dos aspectos: cuándo y cómo usan los abonados la red.

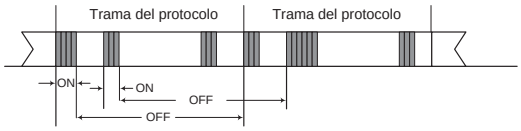


Figura 5: Efecto ON/OFF en el canal de retorno.

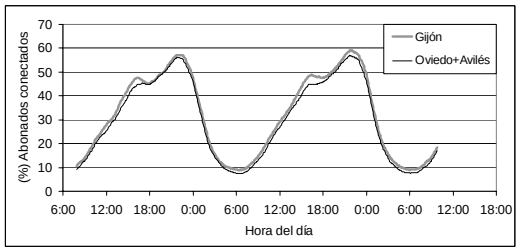


Figura 6: Patrón de conexión de los abonados.

4.3.1 Cuándo se usa la red

Independientemente de la tarifa plana existente, el uso de la red sigue un patrón de 24 horas. Con las medidas disponibles del servidor DHCP, no es posible conocer el número de abonados conectados a la red en cada canal, aunque sí que es posible conocer el número total de abonados conectados en cada uno de los dos grupos de ciudades en los que se organiza la red. La Fig. 6 muestra el patrón de conexiones en cada uno de los grupos de ciudades para las medidas de 2002.

Las medidas aportadas por el servidor DHCP, permiten considerar que el porcentaje de conexiones en cada canal de retorno es similar, aunque pueden existir ligeras variaciones.

Para calcular el número de abonados conectados en cada momento se procede de la siguiente forma: Se expresa de forma porcentual el número de abonados conectados y se calcula la transformada de Fourier para la serie de datos. Posteriormente se utiliza la transformada inversa, pero truncada al 90% de energía para obtener una función compacta del porcentaje de uso en cada periodo de muestreo. Multiplicando el porcentaje obtenido por el número de abonados asignados al canal, se obtiene el número de abonados conectados. Para incluir un cierto margen de variabilidad, el número de abonados conectados se considera distribuido normalmente, con media el valor obtenido y desviación un 5% del valor medio. El nivel de la desviación se obtiene a partir de la variabilidad observada en las medidas del servidor DHCP.

4.3.2 Cómo se usa la red

El segundo paso en la determinación del perfil de usuario es definir cómo los abonados demandan servicios a través de la red de cable. En este punto

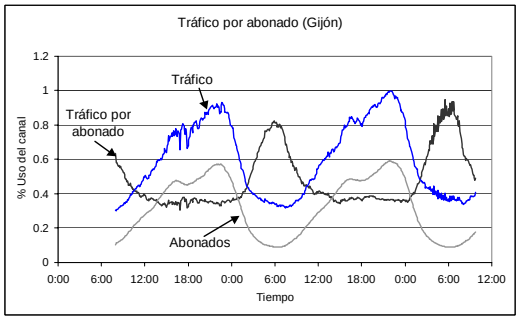


Figura 7: Evolución del tráfico por abonado.

se hace la suposición de que el tráfico total en la red se puede dividir en dos tipos: *interactivo* y *no interactivo*. El tráfico interactivo estará ligado al número de abonados conectados, mientras que el tráfico no interactivo no depende del número de abonados conectados. El tráfico interactivo estaría ligado a protocolos del tipo HTTP, mientras que el tráfico no interactivo estaría relacionado con las aplicaciones de tipo P2P.

Esta suposición se apoya en el análisis de las medidas de tráfico disponibles. Por un lado, medidas del tráfico en el router de acceso a *Internet* disgregadas por servicios, muestran que la aplicación de tipo P2P, *Edonkey* constituye casi la mitad del tráfico total y muestra un patrón constante en el tiempo. Por otro lado los servicios basados en HTTP, que constituyen aproximadamente el 15% del total, muestran una evolución cíclica. Por otro lado, si en cada grupo de ciudades se considera el tráfico de retorno total y se divide por el número de abonados conectados, se obtendría como resultado el tráfico por usuario. En la figura 7 puede verse la forma de la gráfica resultante: un zona lineal de tráfico estable con el número de abonados, y una zona independiente del número de abonados conectados.

La obtención de las proporciones de tráfico de tipo interactivo y del tráfico no interactivo ó de tipo P2P, se realiza a partir de las medidas de tráfico en cada canal de retorno. Para el tráfico P2P o no interactivo, se toman los valores de tráfico a altas horas de la noche, donde el número de abonados interactivos es mínimo. Posteriormente, se estableció la dependencia de estos valores con la relación tráfico de pico a tráfico medio mediante un modelo potencial. De esta forma se determina el tráfico de tipo P2P presente en cada canal de retorno. Para el tráfico de tipo interactivo se obtiene el porcentaje de tráfico por abonado. Al tráfico medio se le resta el valor del tráfico P2P, dando lugar al tráfico interactivo total. Este valor se divide entre el número medio de abonados efectivos, es decir, los abonados que realmente están físicamente realizando peticiones; este valor vendrá dado por el valor medio de abonados conectados menos el valor mínimo, que serán los que

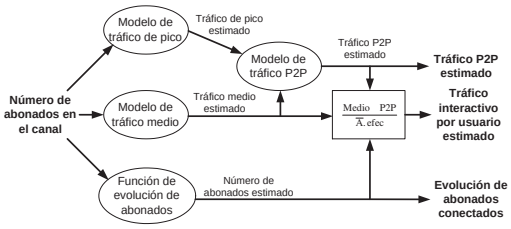


Figura 8: Proceso de estimación del tráfico.

soporten las aplicaciones de tipo P2P. En la Fig. 8 se resume el proceso de obtención de los valores del modelo de tráfico a partir del número de abonados asignados al canal.

El modelo de tráfico ha sido implementado usando el lenguaje de modelado y simulación QNAP2[1]. Los resultados obtenidos han sido validados por comparación con los valores reales y se muestran en la sección 6.

5 Modelo global de la red

El modelo de la red se obtiene incorporando el modelo de tráfico al modelo que representa el comportamiento físico de la red. El proceso de construcción de la red se ha realizado en dos pasos. En un primer paso se construyó un modelo genérico de una rama HFC y se validó dicho modelo. Finalmente, se construye el modelo global de la red haciendo interactuar tantos modelos de rama como sean necesarios y los elementos que representan la cabecera de la red.

El lenguaje de modelado QNAP2 permite la definición de objetos que proporcionan un patrón de comportamiento. Utilizando esta posibilidad se ha creado un objeto que define el modelo de tráfico, particularizado para cada canal según el número de abonados asignados. De igual forma se define un objeto que encapsula una rama HFC completa. El modelo global está formado por los elementos que definen la cabecera de red y tantos objetos de tipo rama HFC como ramas existan. Por tanto, el modelo construido es escalable pues se adapta al crecimiento de la red sin más que definir nuevos objetos de tipo rama HFC.

La estructura final del modelo de la red de cable se ajusta a la arquitectura de red mostrada en la Fig. 1, donde cada rama está representada por un conjunto de colas, y el anillo central está representado por una cola que permite intercambiar el tráfico entre las distintas ramas y la cabecera.

En este modelo es necesario determinar: las proporciones de tráfico hacia los posibles destinos (*Internet*, servidores, entre ramas, intra rama), y el tamaño del tráfico demandado por los abonados.

[1]QNAP2 fue desarrollado por el INRIA y es una marca registrada por SIMULOG

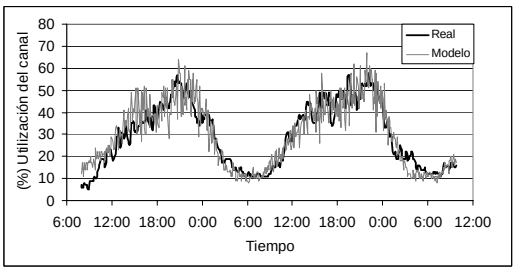


Figura 9: Canal GI01CC02-UP7, 247 abonados.

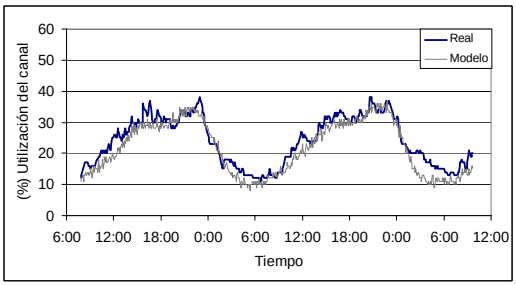


Figura 10: Canal OV01CC01, 977 abonados.

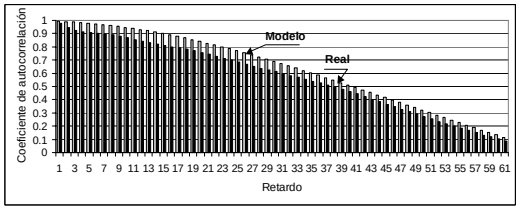


Figura 11: Comparativa de la función de autocorrelación, peor ajuste.

Las medidas disponibles solamente permiten calcular algunos de las proporciones de tráfico, siendo poco significativas las que no se pueden cuantificar. Comparando la suma de todos los tráficos de retorno y el tráfico dirigido hacia *Internet*, se obtiene que más del 95% del tráfico de la red HFC se dirige hacia *Internet*. El tráfico hacia los servidores representa un porcentaje muy bajo respecto al total, y mayoritariamente debido a peticiones realizadas desde *Internet*. El tráfico entre ramas se obtiene como diferencia estando entre el 0.5% y el 2% del total.

El tamaño del tráfico demandado por los abonados se expresa como una razón respecto al tráfico enviado por el canal de retorno. Se distingue entre el tráfico de tipo P2P o no interactivo y el tráfico interactivo. Los valores de las razones se obtienen a partir de los datos de tráfico en el router de acceso a *Internet*, considerando la parte continua de tráfico como tipo no interactivo y la parte restante como tipo interactivo. Los tamaños de tráfico en los canales de bajada se obtendrán multiplicando el tráfico en el canal de retorno por un valor obtenido según una distribución normal, con los valores medios y desviaciones previamente calculados.

6 Resultados

Los resultados obtenidos con los diferentes modelos desarrollados han sido validados, por comparación con las medidas reales obtenidas del sistema. La comparación se efectúa de tres formas diferentes: por comparación gráfica de los perfiles de tráfico obtenidos, por comparación mediante intervalos de confianza y por comparación de propiedades estadísticas representativas, eligiéndose como tales el valor de la función de autocorrelación y la propiedad de autosemejanza.

El primer modelo utilizado fue el modelo de tráfico, encargado de generar el tráfico en los canales de retorno de la red. En la Fig. 9 se muestra la comparativa del perfil de tráfico en uno de los canales de retorno.

La técnica de validación por intervalos de confianza, se basa en obtener la serie diferencia, entre las series a comparar y obtener el intervalo de confianza; si el intervalo de confianza incluye el valor 0, ambas series son estadísticamente equivalentes para el nivel de confianza considerado. Aplicando este método al conjunto de los canales de retorno, se ha obtenido un intervalo de confianza para el 95% de nivel confianza de $[-11.33, 13.55]$.

En el caso del modelo de tráfico las funciones de autocorrelación obtenidas oscilan desde el ajuste perfecto, hasta ligeras diferencias en la parte final de la función. En el caso de los valores del parámetro H que mide la autosemejanza, el error está por debajo del 10% excepto en dos casos, siendo el error medio del 2.77%.

Los resultados del modelo de rama se omiten por ser un paso intermedio hacia el modelo global de la red. En el caso del modelo global de la red, se comparan los valores de tráfico obtenidos en todos los canales de bajada, así como el tráfico registrado en el router de acceso a *Internet*, tanto saliente como entrante.

En la Fig. 10 se comparan los perfiles de tráfico real y generado por el modelo en uno de los canales de bajada. Aplicando el método de validación por intervalos al tráfico en los canales de bajada, se obtiene para un 95% de nivel de confianza un intervalo de $[-6.328, 14.159]$, el intervalo incluye el cero, lo que confirma la equivalencia estadística de los perfiles de tráfico. En la Fig. 11 se muestra el peor caso obtenido en la comparativa de las funciones de autocorrelación, como se puede observar, la diferencia es baja. Finalmente, el parámetro H de autosemejanza presenta un error medio del 2.59% y el valor extremo del 10.59%.

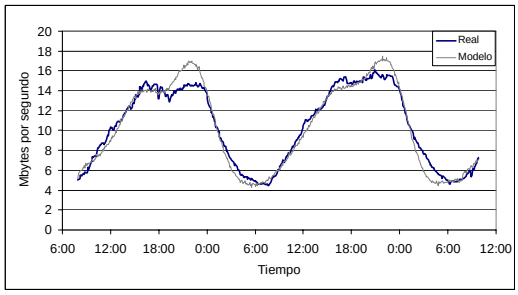


Figura 12: Comparativa del tráfico de *Internet*.

Con el modelo global también se obtienen los valores de tráfico a través del router de acceso a *Internet*; como muestra, en la Fig. 12 se compara el perfil del tráfico entrante con dirección a la red de cable. Se puede observar que existe un buen ajuste de los perfiles de tráfico. El ajuste se confirma por los intervalo de confianza, que incluyen al valor cero. Las funciones de autocorrelación obtenidas son prácticamente idénticas a las reales y el error relativo en el parámetro H es del 0.34% para el tráfico dirigido a *Internet* y del 1.10% del tráfico con dirección a la red de cable.

7 Conclusiones

Este artículo presenta de forma condensada el trabajo desarrollado en [9]. A lo largo del artículo se ha desarrollado un modelo capaz de representar el comportamiento de una red de cable. Las principales características del modelo construido son:

- El modelo está basado en medidas de tráfico, obtenidas de una red de cable real. Esta característica permite la validación del modelo por comparación con resultados reales.
- El modelo de tráfico desarrollado no requiere hacer distinción entre tipos de tráfico y relaciona los parámetros del modelo directamente con características de la red: número de abonados asignados a cada canal.
- El modelo se ha implementado usando objetos de simulación que encapsulan la funcionalidad de una rama HFC completa, lo que le confiere la capacidad de ser un modelo escalable y adaptarse a la evolución de la red.
- Los resultados obtenidos, tanto para el modelo de tráfico, como para el modelo global de la red de cable, han sido validados confirmando su equivalencia estadística.

Este modelo se plantea como una herramienta destinada a evaluar el comportamiento de la red de cable, facilitando la toma de decisiones de planificación: el modelo permite obtener la evolución del tráfico en los canales, en función de la hora y el número de abonados asignados a cada canal.

Agradecimientos

Agradecemos la colaboración de TELECABLE, S.A., por compartir los datos de su red de cable y su disposición para resolver cualquier duda.

Referencias

[1] I. Polevoy. "On Performance of HFC MAC Layer Algorithms". Master Thesis Institute of Computer Science, University of Jerusalen. Septiembre (2000).

[2] N. Golmie, F. Mouveaux, D. Su. "A Comparison of MAC Protocols for Hybrid Fiber/Coax Networks: IEEE 802.14 vs. MCNS". Proceedings of the 16th International Conference on Communications. Vancouver, Canada, pp. 266-272, Junio (1999).

[3] I. Borges, F. Fontes, J. Bastos, J. Loureiro. "Interactive Services over Hybrid Fibre-Coax Networks". Proceedings of international conference on ATM, ICATM99. Colmar, Francia. Junio (1999).

[4] N.K. Shankaranarayanan, Z. Jiang, P. Mishra. "User-Perceived Performance of Web-browsing and Interactive Data in HFC Cable Access Networks". Proceedings of the IEEE International Conference on Communications. Helsinki, Finlandia. Junio (2001).

[5] M. García, X.G. Pañeda, J.R. Arias, V.G. García, F.J. Suárez. "Análisis del tráfico en la red HFC de un operador de cable". Segundo congreso iberoamericano de telemática. Mérida, Venezuela. Septiembre (2002).

[6] W. Leland, M. Taqqu, W. Willinger, D. Wilson. "On the self-similar nature of Ethernet traffic (Extended version)". IEEE/ACM Trans. on Networking. Vol 2, nº 1, pp 1-15. Febrero (1994).

[7] M. Taqqu, V. Teverovsky, W. Willinger. "Estimators for long-range dependence: an empirical study". Fractals. Vol 3, nº 4, pp 785-798. (1995).

[8] V.S. Frost, B. Melamed. "Traffic Modeling for telecommunications Networks". IEEE Communications Magazine, pp 70-81. (1994).

[9] M. García. "Modelado de prestaciones de redes de área metropolitana de transmisión de datos, basadas en tecnología híbrida fibra-coaxial". Tesis Doctoral, Dpto. de Informática, Universidad de Oviedo, Junio (2003).

Modelado de tráfico WAP en redes IP

F.J. González Cañete, E. Casilari, F. Sandoval
Departamento de Tecnología Electrónica. Universidad de Málaga
Campus Universitario de Teatinos. Complejo Tecnológico. E.T.S.I. de Telecomunicación
29071 Málaga
Teléfono: 952 13 13 52 Fax: 952 13 14 47
E-mail: equinoxe@dte.uma.es, ecasilari@uma.es, sandoval@dte.uma.es

***Abstract**. In this work the WAP protocol architecture has been studied because the WAP 2.0 architecture is going to be used in the next third generation wireless networks, and it will allow these mobile terminals to access the Internet content, so new models for this kind of traffic are needed. A statistical model study of the TCP connection level and the deck level is done using the traces obtained with a HTTP WAP browser simulator, and these WAP traffic models are compared with the equivalent models of the Web traffic.*

1 Introducción

Sin duda alguna, el desarrollo de Internet en los últimos años ha superado con creces las expectativas más optimistas, y este mismo éxito puede ser extrapolable al mundo de las comunicaciones móviles. No hace tanto tiempo en el que prácticamente los únicos usuarios de los teléfonos móviles eran los altos ejecutivos de las empresas que necesitaban estar localizados en cualquier momento y, sin embargo, en el año 2002 se han superado ampliamente los 900 millones de usuarios de teléfonos móviles en todo el mundo, y 29 millones en España [1]. El siguiente paso en la evolución de ambos mundos, Internet y telefonía inalámbrica, es la posibilidad de acceso desde los teléfonos móviles a la *Web*, hecho que fue consumado con la creación del WAP (*Wireless Application Protocol*).

Si bien es cierto que WAP no ha resultado tener, hasta ahora, todo el auge que esperaban las compañías que pretendían prestar los servicios de acceso, bien es cierto que la tecnología ha ido mejorando de manera que se resolvieran los principales problemas que hacían que los usuarios no se decantaran por usar estos servicios: la velocidad y el elevado coste. Cuando aparecieron los primeros terminales con navegadores WAP, el servicio portador era GSM (el nombre deriva del comité *Groupe Speciale* Mobile) o Sistema Global para Comunicaciones Móviles, que es un servicio orientado a conexión, por lo que la tarificación es por tiempo, y de muy baja velocidad de transferencia, ya que sólo alcanza los 9200 bps. En los últimos años, y a la espera de la tercera generación de móviles que usen UMTS (*Universal Mobile Telecommunication System*), ha aparecido una extensión de GSM para comunicaciones de datos en modo paquete, el GPRS (*General Packet Radio Service*), cuya tarificación es por cantidad de datos transmitidos y cuya tasa de transmisión es mucho más alta que con GSM (hasta 171,2 kb/s).

En definitiva, WAP consiste en una serie de protocolos y arquitecturas que permiten el acceso desde un teléfono inalámbrico a contenidos situados en servidores *Web* de forma independiente al servicio portador.

1.1 Arquitectura WAP 1.x

WAP está basado en una arquitectura cliente-servidor [2] en la que se amplía la arquitectura típica de la *Web* para que se tenga acceso desde una red inalámbrica. En la arquitectura *Web*, el cliente es un ordenador con un navegador que realiza peticiones HTTP al servidor, que es el encargado de devolver las páginas solicitadas (Fig 1).

La arquitectura WAP 1.x extiende la estructura *Web* introduciendo un elemento intermedio, el *Proxy WAP*, cuyas funciones más importantes incluyen la de traducción (*Gateway*) entre los protocolos usados en Internet y el medio radio, además de poder ser utilizado como caché de páginas visitadas para que el rendimiento del sistema sea óptimo (Fig. 2).

En este esquema, el cliente es un terminal móvil con capacidad de navegación WAP que hace las peticiones de las páginas a un *Proxy* WAP, que es el encargado de traducirlas a peticiones HTTP, las cuales son enviadas al servidor *Web* que contiene la página solicitada. El servidor devuelve la página al Proxy y éste la recodifica para ser enviada al cliente de una manera eficiente a través del medio radio.

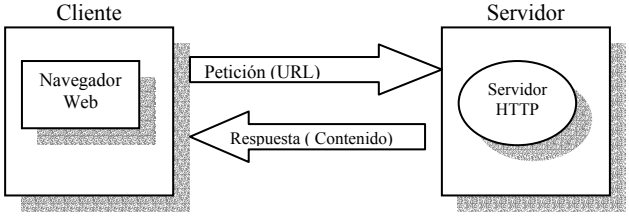


Fig. 1. Arquitectura *Web*

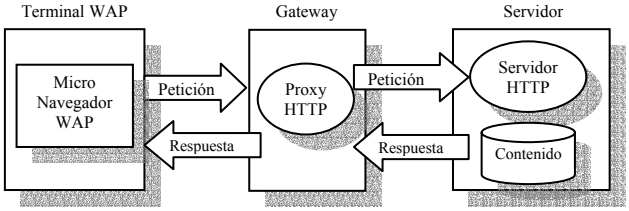


Fig. 2. Arquitectura WAP

La razón de que se hayan desarrollado diferentes protocolos para el medio radio, y por tanto, que sea necesaria la utilización de un *Proxy* WAP, es debido a que es un medio en el que la cantidad de datos a enviar debe ser lo menor posible, ya que el ancho de banda es reducido, y la probabilidad de pérdidas es elevada. La torre de protocolos para la versión WAP 1.x, puede verse en la Fig. 3. En la parte *Web* se distinguen los protocolos clásicos del mismo, como son HTTP, SSL, TCP e IP, aunque se añade una capa más, el WAE (*Wireless Application Environment* – Entorno de aplicación inalámbrica), ya que las páginas WAP que vayan a ubicarse en un servidor tienen que tener un formato específico basado en un lenguaje especialmente desarrollado para WAP, el WML (*Wireless Markup Language*) que es un lenguaje de marcas similar al HTML, y su lenguaje de *script* asociado, el WMLScript. En la parte radio de la arquitectura, y en la zona más alta de la torre de protocolos, nos encontramos también el WAE, ya que el terminal WAP debe incorporar un micronavegador que sea capaz de interpretar los comandos recibidos en WML y WMLScript. Seguidamente nos encontramos el protocolo WSP (*Wireless Session Protocol* – Protocolo Inalámbrico de Sesión) que posee las funcionalidades del HTTP, pero mediante una codificación más compacta. WSP es el encargado del mantenimiento de las sesiones. WTP (*Wireless Transaction Protocol* – Protocolo Inalámbrico de Transacciones) proporciona varias clases de servicio de transacciones, así como seguridad usuario-usuario. Existe una capa orientada a la seguridad, WTLS (*Wireless Transport Layer Security* – Capa de Seguridad de Transporte Inalámbrico) que proporciona integridad y privacidad de los datos, así como servicios de autentificación. En un nivel inferior nos encontramos WDP (*Wireless Datagram Protocol* – Protocolo Inalámbrico de Datagramas), que sirve de capa de transporte y como intermediario entre los protocolos portadores y los de las capas superiores. En la parte más baja de la torre de protocolos nos encontramos los posibles portadores, entre los que se encuentran, además de los mencionados GSM y GPRS, otros como DAMPS, CDMA, PHS, etc., hecho que demuestra que WAP es independiente de la tecnología usada como portadora del servicio.

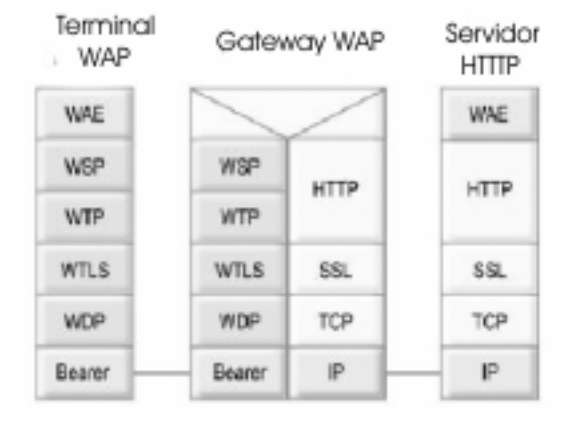


Fig. 3. Torre de protocolos WAP 1.x

Como puede advertirse en la Fig. 3, prácticamente existe una relación uno a uno entre los protocolos de Internet y los del medio radio, de forma que sea relativamente sencillo realizar la traducción entre ambas zonas.

1.2 Arquitectura WAP 2.0

En la versión del protocolo WAP 2.0 se extiende la arquitectura de protocolos de forma que se asemeje a la arquitectura *Web* (Fig. 4), pero con ciertas mejoras, como es el hecho de que sea el propio servidor el que pueda iniciar el envío de información sin que el cliente haya realizado ninguna petición (tecnología *Push*). Una posible utilidad de este servicio es que el servidor avise al cliente cuando se den una serie de condiciones, como puede ser una noticia importante o que le ha llegado un correo electrónico.

La arquitectura del WAP 1.x está pensada para optimizar los escasos recursos radio debido al escaso ancho de banda que éste ofrece. Esto sucede con GSM y también con GPRS, aunque la evolución de las tecnologías está consiguiendo que el ancho de banda se esté incrementando notablemente, hasta el punto que con UMTS se puede obtener un ancho de banda de hasta 2 Mbps. UMTS es precisamente el escenario idóneo para aplicar la arquitectura WAP 2.0 y, por lo tanto, conseguir la confluencia hacia IP.

Cuando la tercera generación de móviles sea una realidad, estos terminales tendrán la capacidad de acceder a Internet usando IP directamente, con lo que

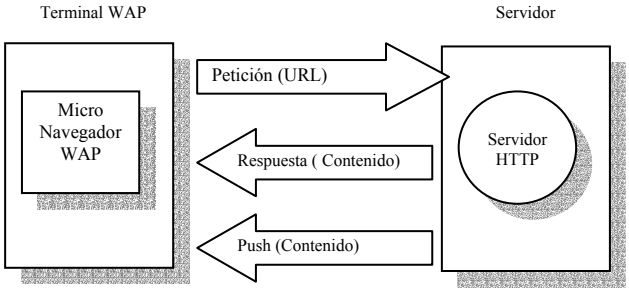


Fig. 4. Arquitectura WAP 2.0

la versión 2.0 del WAP es la ideal para este tipo de escenarios. Se hace necesario por tanto el estudio del tráfico WAP en las redes IP. El presente trabajo pretende estudiar el tráfico WAP sobre esta arquitectura, en la que el terminal móvil es capaz de acceder a los contenidos usando directamente IP.

2 Banco de pruebas y toma de muestras de tráfico

La arquitectura que se ha implementado para obtener las muestras de tráfico que se han analizado en el presente artículo se muestra en la Fig. 5. Se ha utilizado como emulador de cliente de terminal el OpenWave SDK 6.1 en su versión de HTTP [2]. Este software es capaz de realizar peticiones a servidores WAP y mostrar los contenidos en el emulador de terminal. El ordenador donde fue instalado el emulador se encuentra en una red local Ethernet a 100 Mbps conectada a RedIris a través de la red de la Universidad de Málaga. Para obtener las muestras, primero se hizo una lista con un total de 547 sitios WAP que fueron obtenidos de la página *Web* [4]. Una vez obtenida la lista de direcciones, se desarrolló un programa que iba obligando al emulador a visitar cada uno de estos sitios de manera consecutiva y automática, a razón de una visita por minuto. Las capturas de tráfico han sido realizadas con Ethereal 0.9.8 [5], y procesadas con una versión del software tcptrace [6] modificada en el Departamento de Tecnología Electrónica de la Universidad de Málaga. Este programa permite analizar y extraer estadísticos de las muestras a nivel de conexión.

El punto de captura de las muestras se situó directamente en el ordenador en el que estaba instalado el emulador de terminal WAP. Este punto de captura se asemeja mucho al usado en [7], que realiza las capturas HTTP entre el servidor *Web* y el *Gateway*. En el caso particular que nos afecta, se puede considerar que el emulador funcionaría como *Gateway* y como cliente a la vez, por lo que se está eliminando el medio radio. No se ha modelado el comportamiento del usuario del servicio, por lo que no se ha hecho una navegación dentro de cada sitio visitado, sino que simplemente se ha consultado la página principal del sitio. De este modo se pretende estudiar las características inherentes de los contenidos de las páginas WAP.

3 Resultados

Del análisis de las muestras de tráfico, se han obtenido los datos de las 1601 conexiones que han sido realizadas para visitar los 547 sitios WAP. De ellas, el 98.2 % fueron conexiones completas (finalizadas mediante paquetes de FIN), mientras que la traza a partir de la que se han obtenido los datos es

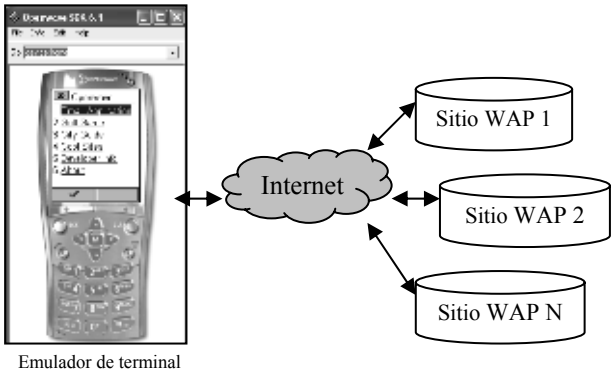


Fig. 5. Arquitectura de las pruebas

correcta para dicho fin. Al mismo tiempo, la baja proporción de las conexiones reseteadas prueba que los servidores WAP no emplean conexiones persistentes.

3.1. Estudio a nivel de conexión

El primer estudio que puede realizarse es con respecto a las conexiones TCP establecidas con los servidores WAP. La media del tiempo de conexión obtenida es de 8.7 segundos, con una desviación típica de 15.7 segundos, lo que nos demuestra que la toma de datos es correcta, ya que tanto la media como la desviación típica son bastante inferiores a los sesenta segundos que se han tomado como intervalo entre la visita de un sitio WAP y el siguiente, por lo que se tiene tiempo suficiente para completar las peticiones y éstas no se solapan en el tiempo.

En la Tabla 1 pueden observarse los datos estadísticos obtenidos de los bytes por conexión tanto para el sentido ascendente (*uplink*) de la conexión, es decir, del terminal hacia el servidor, como en el sentido descendente (downlink), del servidor hacia el terminal. En la Tabla 2 se muestran los datos de los paquetes por conexión. Si comparamos estos valores con los equivalentes para el tráfico *Web*, nos encontramos que la cantidad de datos (bytes) a transmitir es menor para el caso de WAP, debido fundamentalmente a que el tamaño de los contenidos es mucho más pequeño, es decir, menos texto e imágenes de menor tamaño [8]. Este hecho puede apreciarse claramente si comparamos los resultados obtenidos en trabajos como [9] en el que se propone, para modelar el tamaño de las conexiones *Web*, una composición de la distribución de Pareto y Logaritmo-Normal con medias 7.2 y 14.8 KBytes, o en [10] con una distribución *heavy-tailed* de media entre 8 y 10 KBytes, o en [11] en el que se propone una distribución Logaritmo-Normal con media entre 7.7 y 10.7 KBytes. En [12] se hace el mismo estudio pero a nivel de paquete, proponiéndose un modelo log-normal con media 8.3 paquetes. En todos los casos, la media es mayor para el caso *Web* que para el WAP, cuyas conexiones poseen un tamaño medio de 2 KBytes.

Tabla 1. Bytes por conexión

	Uplink	Downlink	Dl/Ul
Bytes totales	μ = 1584 σ = 317 Med = 1539	μ = 2159 σ = 5251 Med = 811	1.36
Bytes útiles	μ = 1534 σ = 142 Med = 1539	μ = 2154 σ = 5243 Med = 811	1.4

(μ: Media, σ:Desviación típica, Med: Mediana)

Tabla 2. Paquetes por conexión

	Uplink	Downlink	Dl/Ul
Paquetes totales	μ = 7.6 σ = 4 Med = 7	μ = 6.5 σ = 5 Med = 6	0.85
Paquetes de datos	μ = 2 σ = 0.3 Med = 2	μ = 2.2 σ = 5 Med = 1	1.1

Estos datos también ponen de manifiesto que el tráfico descendente es mayor al ascendente, así como su variabilidad. Esto es debido a que en el sentido ascendente solo se realizan peticiones de objetos, mientras que en el descendente lo que se manda son los objetos propiamente dichos, cuyos tamaños varían

notablemente, aunque no tanto como en el mundo *Web*. El hecho de que el cociente entre la desviación típica y la media de la cantidad de datos en bytes por conexión sea superior a uno, nos hace pensar un posible comportamiento *heavy-tailed* en el sentido descendente de la comunicación.

Si representamos la función de distribución complementaria del número de bytes no retransmitidos por conexión enviados por el servidor (sentido descendente), se observa que puede aproximarse mediante una función hiperbólica (Fig. 6).

En esta figura se ha representado la distribución de probabilidad complementaria (CDF):

$$G(x) = 1 - F(x) = \text{Prob}(X > x)$$

donde la cola de G(x) se ha aproximado por una expresión hiperbólica:

$$f_{x\to 10^3}(x) = A\frac{1}{x^{\alpha}}$$

siendo los parámetros A = 3617 bytes y α = 1.28, los cuales fueron obtenidos mediante una regresión de mínimos cuadrados en escala logarítmica.

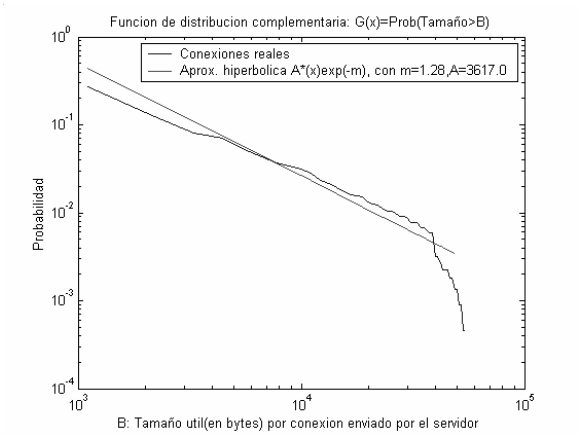


Fig. 6. Distribución del tamaño útil por conexiones enviado por el servidor

Esta figura nos demuestra claramente el comportamiento *heavy-tailed* entre los 10^3 bytes y los $4*10^4$ bytes del tráfico que genera un servidor con contenidos WAP, con lo que se sigue el mismo comportamiento que el tráfico *Web* [13] y demuestra que la principal diferencia entre ambos tipos de tráfico es a nivel de escala, esto es, con respecto al tamaño absoluto de los objetos que se transfieren.

Analizando el tráfico ascendente, se observa estudiando su histograma normalizado (Fig. 7) que existe un rango de tamaños característicos en las conexiones que el cliente envía hacia el servidor. Este rango se encuentra entre los 1514 bytes y los 1700 aproximadamente, y se corresponden a los paquetes de petición GET de HTTP que el cliente envía al servidor para solicitar cada uno de los objetos. Sin embargo, en trabajos como [14] donde se hacen las peticiones GET hacia el servidor a través de un *Proxy* WAP, la media de los tamaños de las peticiones GET tienen valores entre los 82.1 bytes y los 112 bytes en función del tipo de modelo de tráfico usado. Este tamaño reducido se debe a que las peticiones deben ser optimizadas para el medio radio. La gráfica muestra claramente que la mayoría del tráfico que se genera en el terminal es debida a las peticiones de objetos hacia el servidor. Cabría esperar, sin embargo, que estas peticiones fueran mucho más pequeñas en tamaño de lo que son realmente. Sin embargo, en la petición GET no solo se envía la URL de la página que se está solicitando, sino que también se mandan otros datos como la lista de tipos MIME (*Multipurpose Internet Mail Extension*) que acepta el cliente, las cabeceras de sesión o la identificación del cliente, lo que hace que el tamaño aumente considerablemente.

3.2. Estudio a nivel de *deck*

En la terminología WAP, un *deck* se corresponde con un fichero escrito en el lenguaje de marcas WML y que puede o no contener código WMLScript. Cada *deck* está dividido en uno o más *cards*, que se

corresponden con cada una de las pantallas por las que puede ir navegando el cliente sin tener que realizar la petición de otro *deck* al servidor. Si se busca la analogía con el mundo de Internet, se encuentra que un *deck* se correspondería con un conjunto de páginas *Web* que son cargadas a la vez en la misma petición. Sin embargo, y dado que el elemento máximo que provoca una petición a un servidor WAP es el *deck*, se va a realizar el estudio comparándolo con la máxima cantidad de información que se trata en Internet, que es la página *Web*.

El número total de *decks* WAP visitados en el experimento ha sido de 686, que es ligeramente superior al número de sitios visitados, ya que puede darse el caso de que el *deck* principal del sitio llame a otra URL, con lo que se visitan dos o más *decks* de un mismo sitio. Para identificar los *decks* en las muestras se observaron las conexiones consecutivas a un mismo servidor (caracterizado por su dirección IP), de forma que cada grupo de conexiones consecutivas iniciadas en un intervalo menor de 60 segundos a una misma dirección IP fueron identificadas como un *deck*.

En la tabla 3 se observan los datos estadísticos obtenidos el estimar el tamaño en bytes de los *decks*, el número de conexiones TCP con el servidor que realiza cada *deck* y el tiempo entre conexiones consecutivas dentro de un mismo *deck*. Debido a que los terminales WAP poseen un escaso espacio de representación de la información y, además, el ancho de banda del medio radio para el que fue inicialmente diseñado es escaso, el protocolo WAP fue desarrollado de forma que se envíen pequeñas cantidades de información hacia el terminal.

Como se puede observar, el tamaño medio de un *deck* es bastante más pequeño que el equivalente estudiado para tamaños de páginas *Web*, que oscila entre los 9 KBytes obtenidos en trabajos como [15] y los 54 KBytes de [16]. En la tabla 4 se ha representado la distribución porcentual del tamaño de los *decks* pudiéndose observar que el 75 % de los *decks* son de menos de 3000 bytes. Este hecho también afecta al número de conexiones que se realizan dentro de un *deck*, ya que la cantidad de objetos que es necesario obtener del servidor es pequeño debido a que poseen pocas o ninguna imagen y además, la cantidad de texto es escaso. Por el contrario, en la literatura se han obtenido medias de conexiones por página *Web* que oscilan entre los 1.9 modelados con una distribución Gamma de [12] y las obtenidas en [17] modeladas también con una distribución Gamma de media 5.5, con lo que se observa un mayor número de

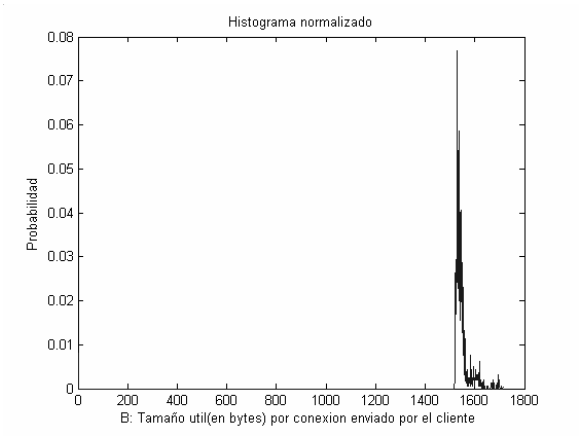


Fig. 7. Histograma del tamaño útil por conexión enviado por el cliente

conexiones por página en la *Web* que en el WAP. También la variabilidad del tamaño de las conexiones es mayor en la *Web*, donde se observan desviaciones típicas que oscilan entre los 6.6 y los 12.74 obtenidos en [18], bastante mayores que los 2.8 del número de conexiones por *deck* del WAP.

De las muestras obtenidas se ha intentado aproximar la distribución de probabilidad del número de conexiones por página con una distribución de Poisson (λ=1.33), una distribución geométrica (p=0.571) y una distribución binomial negativa (R=0.2578, P=0.1619), que pueden verse en la Fig. 6. La mejor aproximación se ha obtenido con la distribución geométrica.

En la Tabla 5 puede verse que los *decks* que realizan entre una y tres conexiones al servidor WAP constituyen el 88 % del total con lo que queda evidenciada la pequeña cantidad de conexiones que realiza WAP en comparación con la *Web* a nivel de página.

El tiempo entre conexiones dentro de un *deck* se espera que sea del mismo orden que entre páginas *Web*, hecho que se constata si comparamos los datos obtenidos en el presente experimento y los de [18], donde se analizaron siete muestras diferentes con unas medias de tiempo entre conexiones de la misma página que oscilan entre los 2.29 y 2.99 segundos, según la muestra analizada, y con desviaciones típicas de entre los 4.48 y los 5.63 segundos, pero con medianas de entre 0.3 y 0.66 segundos.

Tabla 3. Estadísticas de los *deck* (*downlink*)

	μ	σ	Mediana
Tamaño *deck* (bytes)	5028	13686	1772
Conexiones/*deck*	2.3	2.8	2
Tiempo entre conexiones de un mismo *deck* (seg)	1.82	3.4	0.96

Tabla 4. Distribución del tamaño (bytes) de los *decks*

	Porcentaje
< 1000 bytes	24.19 %
1000 – 2000 bytes	31.63 %
2000 – 3000 bytes	21.42 %
3000 – 4000 bytes	8.45 %
4000 – 5000 bytes	2.76 %
5000 – 6000 bytes	0.43 %
> 6000 bytes	11 %

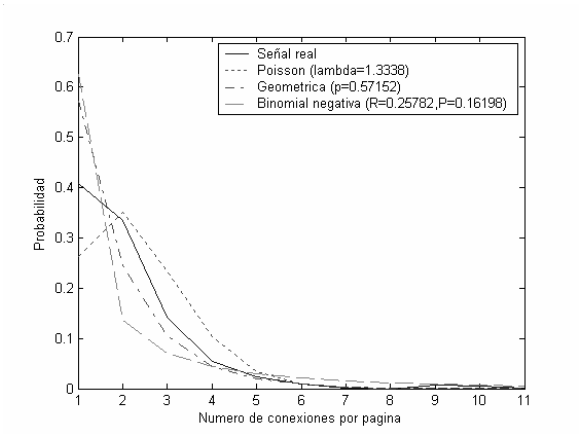


Fig. 6. Aproximación distribución del número de conexiones por página

Tabla 5. Distribución del número de conexiones por *deck*

	Porcentaje
1 conexión	40.8 %
2 conexiones	33.3 %
3 conexiones	14.2 %
4 conexiones	5.4 %
Más de 4 conexiones	6.1 %

4 Conclusiones

En el presente artículo se ha comentado la arquitectura del protocolo WAP en sus versiones 1.x y 2.0. Con la versión 2.0 del protocolo y, previendo su aplicabilidad en redes inalámbricas de tercera generación, se ha realizado un experimento enfocado a obtener datos estadísticos y modelos del tráfico generado entre el terminal con capacidades WAP y el servidor de contenidos.

Del estudio de las muestras a nivel de conexión TCP se concluye que la cantidad de datos que se envían en el sentido descendente de la comunicación es mayor que en el sentido ascendente, ya que en el sentido ascendente únicamente se hacen peticiones de objetos y la respuesta del servidor es devolver el objeto solicitado. No obstante, esta asimetría es mucho menor que en la *Web* por la simplicidad de los contenidos WAP. Por esta misma razón, la variabilidad del tráfico en el sentido descendente también es mayor, en tanto que los objetos a devolver son de un tamaño muy variable. También se han comparado estos resultados con los obtenidos en la *Web*, con lo que se ha demostrado que el tráfico WAP es menor y de menor variabilidad que el *Web*.

Se ha modelado el tamaño útil de las conexiones enviadas por el servidor y se ha comprobado que posee un comportamiento *heavy-tailed* entre los 10^3 bytes y los $4*10^4$ bytes, con lo que se sigue el mismo comportamiento ya registrado en el tráfico *Web*. También se ha constatado que el tamaño útil por conexión enviado por el cliente está en un rango de valores entre 1514 y 1700 bytes, correspondientes fundamentalmente a las peticiones GET de HTTP que realiza el cliente.

De los datos estadísticos de los *decks* WAP, se concluye que los tamaños de los mismos son más pequeños que el tamaño de las páginas *Web* debido al menor número de objetos que tiene un *deck* y a la reducida cantidad de información que posee. Es más, el 75 % de los *decks* estudiados tenían un tamaño inferior a 3 Kbytes. También se demuestra que, debido al menor número de objetos que posee un *deck*, la cantidad de conexiones que se tienen que realizar para obtenerlos es también mucho más pequeña que en el *Web*. La distribución del número de conexiones por *deck* ha intentado ser aproximada mediante varias distribuciones de probabilidad, obteniéndose el mejor ajuste con una distribución Geométrica. Por último, se ha demostrado que los tiempos entre conexiones de un mismo *deck* sigue el mismo comportamiento que en el *Web*.

Como líneas futuras a este trabajo se podrían sugerir el modelado del comportamiento de cliente, así como realizar una navegación dentro de cada sitio WAP, de esta forma se obtendrían estadísticas más reales del tráfico WAP. Otra posible línea de investigación a seguir sería modelar el medio radio para la arquitectura WAP 1.x.

Agradecimientos

Este trabajo ha sido financiado en parte por el Ministerio de Ciencia y Tecnología (MCYT), Proyecto Nº TEL 99-0755.

Referencias

[1] J. M. Huidobro Moya, Comunicaciones Móviles, Thomson Paraninfo 2002.

[2] OpenWave Systems Inc: http://www.openwave.com/

[3] Open Mobile Alliance: http://www.wapforum.com

[4] Directorio WAP: http://www.pyweb.com/es/sites

[5] Tcptrace, analisis software available at: http://irg.cs.ohiou.edu/software/tcptrace/tcptrace.html

[6] The Ethereal Network Analyzer: http://www.ethereal.com

[7] Irene C. Y. Ma, James Irvine. "Characteristics of WAP traffic". Proceedings on the European Wireless 2002: next generation wireless networks. Florencia (Italia) , Febrero 2002.

[8] S. Buchholz, S. Jaensch, S. Alexander. "Flexible Web traffic modeling for new application domains". Proceedings of the IASTED International Conference on Applied Modeling and Simulation (AMS 2002). Cambridge (USA), Noviembre 2002.

[9] P. Barford, M. Covella. "Generating representative web workloads for network and server performance evaluation". Technical report BU-CS-97-006, Computer Science Department, Boston University, 1997.

[10] B.A. Math. "En empirical model of HTTP network traffic". Proceedings of the IEEE INFOCOM'97, vol 2, Kobe (Japón), pp. 592-600, Abril 1997.

[11] A. Reyes Lecuona, E. González, E. Casilari, J.C. Casasola, A. Díaz-Estrella. "A page oriented WWW traffic model for wireless systems simulations". Proceedings of International Teletraffic Congress (ITC-16), Vol 3.b, pp. 817-826, Edimburgo (UK), Junio 1999.

[12] S. Khaunte, J.O. Limb. "Statistical characterization of a WWW browsing session". Technical report GIT-CC-97-17, Georgia Tech. College of Computing, Junio 1997.

[13] M. E. Crovella, A. Bestavros. "Self-Similarity in World Wide Web. Evidence and possible causes", IEEE/ACM Transactions on Networking, Vol. 5, Nº. 6, pp. 835-846, Diciembre 1997.

[14] P. Stuckmann, H. Finck, T. Bahls. "A WAP traffic model and its appliance for the performance analysis of WAP over GPRS", Proceedings of the IEEE International Conference on the Third Generation Wireless and Beyond (3GWireless'01), San Francisco (USA), Junio 2001.

[15] B. A. Math. "An empirical model of HTTP network traffic", IEEE INFOCOM'97, Kobe (Japón), pp. 592-600, Abril 1997.

[16] N. Vicari. "Measurement and modeling of WWW-sessions", Tech. Rep. 184, I_nstitute of Computer Science. University of Würzburg, 1997.

[17] H. Choi, J. Limb, "A behavioral model of web traffic", International Conference of Networking Protocol'99 (ICNP 99), Septiembre 1999.

[18] A. Reyes Lecuona, Modelado de Tráfico de Clientes WWW, Tesis doctoral, Universidad de Málaga, Julio 2001.

Método heurístico de generación de tráfico sintético de juegos en red multicast

Juan Hernández-Serrano, Josep Pegueroles, Miquel Soriano
Departamento de Ingeniería Telemática. Universitat Politècnica de Catalunya
Jordi Girona 1 y 3. Campus Nord, Módulo C3, UPC.
08034 Barcelona
Teléfono: 93 401 78 09 Fax: 93 401 59 81
E-mail: jserrano@entel.upc.es

Abstract. *Key to the utility of secure multicast is the efficient operation of re-keying protocols. User behaviour knowledge is necessary for effectively design and test of these protocols. In this paper we propose a model for player arrivals in a multiplayer game multicast scenario. The model is based on the hypothesis that interarrival times between players fit a heavy-tailed distribution. Our work uses a Gamma-Pareto function as heavy-tailed distribution. We define a non derivable Gamma-Pareto distribution that we find ideal for heuristic traffic generation. The MATLAB function used to generate synthetic traffic is also shown. This function will allow us to validate our model with real traffic traces in future works. Finally, figures of interarrival time and histogram are presented in order to show the proper behaviour of the heuristic model.*

1 Introducción

Actualmente existe un continuo desarrollo de aplicaciones y servicios a través de Internet basados en multicast. Servicios, como el WebTV, videoconferencia, o juegos en red; se ofrecen de forma simultánea a una comunidad de usuarios generalmente extensa.

Estos servicios requieren comunicaciones entre grupos de usuarios, es decir, “unos cuantos-a-uno” o “unos cuantos-a-unos cuantos” en vez de conexiones unicast “uno-a-uno” o broadcast “uno-a-todos”. Es evidente que el modelo multicast, que minimiza el flujo de información entre grupos de usuarios, permite un mayor aprovechamiento del medio y por tanto se convierte en una necesidad. Al igual que en unicast y broadcast, se ha de poder garantizar un entorno seguro para multicast.

La seguridad en multicast se centra tanto en la seguridad base (gestión de claves, secreto y autenticación de datos) como en la seguridad de la infraestructura (protocolos de enrutamiento). Si bien el segundo punto ha sido ampliamente estudiado, el primero es un tema aún en desarrollo.

La seguridad base debe garantizar, entre otras cosas, la confidencialidad, la autenticación de todos los participantes y la integridad de los contenidos. Por lo tanto ha de basarse en el intercambio de información cifrada entre los miembros de un grupo.

Al igual que en unicast, la autenticación se consigue mediante técnicas propias de la PKI. En cambio, la confidencialidad e integridad de los contenidos requieren únicamente de una única clave de sesión conocida por todos los miembros del grupo multicast.

En los grupos multicast es habitual que los miembros puedan darse de alta o de baja en cualquier instante. Esto hace necesario el uso de algoritmos de renegociación de claves que garanticen que sólo los usuarios autorizados en un determinado momento puedan interpretar la información que se intercambia en el grupo en ese momento.

Para la correcta propuesta y validación de estos algoritmos es necesario el uso de herramientas de simulación, tanto de la propia red multicast como de los diferentes algoritmos (LKH [1], OFT [2], Batch-LKH [3]). Sólo mediante el análisis de los resultados de las simulaciones se podrá, a posteriori, implantarlos en dicha red.

En este documento se propone un método heurístico para generación de tráfico sintético según el comportamiento de usuarios en entornos de juegos en red multi-jugador. El método propuesto se basa en el modelo descrito por Henderson en [4]. El objetivo es generar trazas de peticiones de altas y bajas de usuarios que permita, en trabajos posteriores, llevar a cabo la simulación de algoritmos de re-negociación de claves.

El resto del artículo está organizado como sigue. En la siguiente sección se presenta una breve descripción del escenario a estudio, justificando el modelo propuesto por Henderson en [4]. Posteriormente, en la sección 3 se describe el modelo matemático utilizado. En la sección 4 se explica la generación heurística de tráfico sintético haciendo uso de las herramientas MAPLE y MATLAB. También se muestran las estadísticas más representativas de los patrones de tráfico generados. Finalmente, en el último apartado, se exponen las conclusiones y líneas futuras derivadas de este trabajo.

2 Comportamiento de usuarios en juegos en red

Los juegos en red para múltiples jugadores representan actualmente una de las formas más populares de comunicación de grupos en Internet. Tristan Henderson en [4] realizó un modelo de comportamiento de usuarios multicast en este tipo de escenarios. Este modelo fue obtenido a partir de las trazas generadas por usuarios reales en un servidor del juego Half-Life, muy popular en Internet.

En [5] y [6] se muestra como el tiempo entre llegadas de usuarios para aplicaciones de un único usuario sigue una distribución de Poisson. En cambio, para aplicaciones multi-usuario, y especialmente en aquellas en las que hay interacción entre los usuarios, como en los juegos multi-jugador, se ha llegado a la conclusión [7, 4] de que los tiempos entre llegadas de usuarios están muy correlados entre sí, lo que típicamente se modela mediante funciones "heavy-tailed". Esto es debido, en nuestro caso, a que a medida que aumenta el número de usuarios, los tiempos entre llegadas disminuyen; lo que corrobora la hipótesis [4] de que el número de jugadores es determinante en la decisión de otros para unirse al juego.

La gráfica Log-Log de una función "heavy-tailed" tiende a una recta con pendiente del orden de a. Para obtener este parámetro a partir de trazas reales Henderson hizo uso del estimador de Hill [8], que dice que una distribución aleatoria X es "heavy-tailed" si cumple (1)

$$P[X > x] \sim x^{-a}, \text{cuando } x \to \infty, 0 < a < 2. \qquad (1)$$

estimándose *a* como en (2)

$$\hat{a}_n = \left(\frac{1}{k} \sum_{i=0}^{i=k-1} (\log X_{n-i} - \log X_{n-k}) \right)^{-1} \qquad (2)$$

donde *n* es el número de observaciones, y *k* indica cuántas de las observaciones más largas se han usado para calcular a_n. En concreto, Henderson obtuvo una pendiente *a=1,15*.

3 Modelo matemático

Para conseguir una mayor claridad y sencillez en la exposición se ha prescindido de un excesivo rigor matemático sin que ello conlleve la pérdida de generalidad y validez de los resultados. Debido a la complejidad de muchos de los cálculos y a las limitaciones de la herramienta MAPLE que hemos utilizado, para muchas demostraciones ha sido necesario basarse en métodos gráficos y no en resultados analíticos.

Para modelar el comportamiento "heavy-tailed", se ha optado por utilizar una función híbrida Gamma-Pareto. La función Gamma nos sirve para modelar las dependencias a corto plazo o SRDs, mientras que la función Pareto, que nos define una caída de cola lenta, sirve para modelar matemáticamente las dependencias a largo plazo o LRDs.

Veamos a continuación cómo se definen estas funciones.

3.1 Función Gamma

La distribución de la v.a. X, recibe el nombre de distribución Gamma de parámetros *a > 0* y *p >0 (X ® G(a,p))* si su función densidad es como en (3).

$$\begin{aligned} f(x) &= \frac{a}{\Gamma(p)}(ax)^{p-1}e^{-ax} \qquad para \quad x > 0 \\ f(x) &= 0 \qquad para \quad x = 0 \end{aligned} \qquad (3)$$

donde G(*p*) se define como en (4):

$$\Gamma(p) = \int_0^\infty x^{p-1}e^{-x}dx \qquad para \quad p > 0 \qquad (4)$$

función que sigue la expresión definida por (5):

$$\Gamma(p) = (p-1)\Gamma(p-1) = (p-1)! \quad \text{con} \quad \Gamma(1) = 1 \qquad (5)$$

de la que se deduce que (3) puede expresarse como en (6):

$$f(x) = \frac{a}{(p-1)!}(ax)^{p-1}e^{-ax} \qquad para \quad x > 0 \qquad (6)$$

donde los parámetros *a* y *p* determinan la escala y la forma de la función respectivamente.

La Fig. 1 muestra gráficas ejemplo de la función Gamma para distintos valores de *a* y *p*.

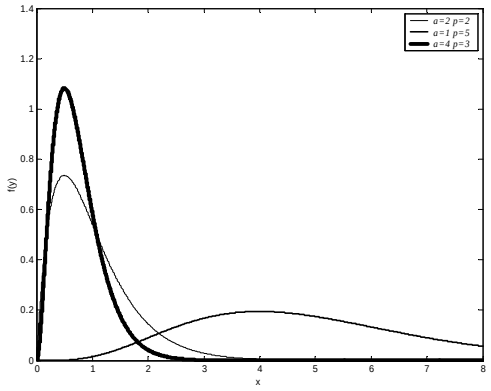


Figura 1. Función densidad de probabilidad Gamma

3.2 Función Pareto

La distribución de la v.a. *X*, recibe el nombre de distribución de Pareto de parámetros $c > 0$ y $a > 0$ si $1 - F_X(x)$ $(P[X > x])$, decrece de forma potencial. Por ello se suele definir asociada a su función de distribución (7).

$$F_X(x) = \begin{cases} 0 & si \quad x < c \\ 1 - \left(\frac{c}{x}\right)^a & si \quad x \geq c \end{cases} \tag{7}$$

La función densidad sigue la expresión definida en (8).

$$f_X(x) = \begin{cases} 0 & si \quad x < c \\ \frac{ac^a}{x^{a+1}} & si \quad x \geq c \end{cases} \tag{8}$$

El parámetro *c* es el mínimo valor permitido de *x*, y *a* determina la pendiente de la cola en el gráfico Log-Log (obsérvese que en ese caso es una recta).

La Fig. 2 muestra gráficas ejemplo de la función Pareto para distintos valores de *a* y *c*.

3.3 Función híbrida Gamma-Pareto

La función Gamma-Pareto es, como su propio nombre indica, una híbrida entre las funciones Gamma y Pareto. Simplemente consiste en sustituir la parte de cola Gamma por una cola Pareto de caída más lenta y unir las dos funciones en un punto umbral tal y como se define en 9.

$$f_{G-P}(x) = \begin{cases} \frac{a}{(p-1)!}(ax)^{p-1}e^{-ax} & 0 < x \leq x_{umbral} \\ \frac{ac^a}{x^{a+1}} & x_{umbral} < x < \infty \end{cases} \tag{9}$$

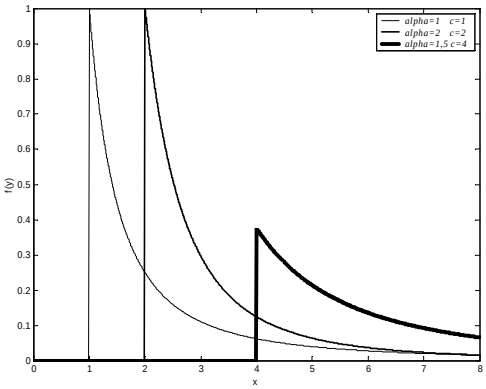


Figura 2. Función densidad de probabilidad Pareto

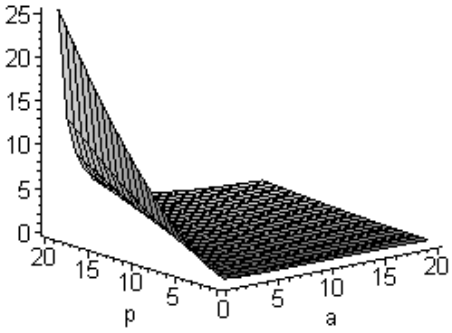


Figura 3. $c_1(a,p)$ - $c_2(a,p)$ **con** $a=1,15$

Para la obtención de este punto umbral fijamos tres condiciones: la función ha de ser continua, derivable y de área 1 (se trata de una función densidad de probabilidad). Luego matemáticamente ha de cumplir (10), (11) y (12):

$$f_{Gamma}(x_{umbral}) = f_{Pareto}(x_{umbral}) \tag{10}$$

$$f'_{Gamma}(x_{umbral}) = f'_{Pareto}(x_{umbral}) \tag{11}$$

$$\int_{-\infty}^{+\infty} f_{Gamma-Pareto}(x)dx = 1 \tag{12}$$

Si consideramos un parámetro conocido tenemos un sistema de 3 ecuaciones y 3 incógnitas. Teniendo en cuenta que mediante el análisis de tráfico real mediante el estimador de Hill [8] se obtiene el parámetro *a* (pendiente de la cola Pareto), de ahora en adelante lo supondremos conocido.

Como se demuestra más adelante, éste es un sistema de ecuaciones sin solución para valores de *a* entre 1 y 2, rango que nos interesa según Hill [8].

Teniendo en cuenta que la función logaritmo es continua entre 0 y 8, para resolver (10) y (11) podemos trabajar en ejes logarítmicos, simplificando enormemente los cálculos, ya que la función Pareto se convierte en una recta.

De (11) es inmediato obtener (13)

$$x_{umbral} = \frac{p+a}{a} \tag{13}$$

Sustituyendo este valor en (10) podemos obtener un valor de *c* en función de *a* y *p*, que denominamos $c_1(a,p)$. De igual forma, sustituyendo en (12) se obtiene otro valor de *c* en función de *a* y *p*, que denominamos $c_2(a,p)$. La Fig. 3 muestra la función diferencia de estas dos funciones, que como se observa es una superficie que nunca toma el valor 0, lo que significa que el sistema no tiene solución.

Es por lo tanto es necesario plantearse un modelo que no cumpla las tres condiciones. El hecho de que la función híbrida Gamma-Pareto tenga que ser una

función densidad de probabilidad impone las condiciones (10) y (12) de continuidad y área 1, luego todo hace pensar que debemos tomar un modelo que no cumpla la condición (11) de derivabilidad. Como se verá en el apartado 4.2 esta elección, bien acotada, no desvía sustancialmente el comportamiento deseado del modelo.

3.3 Estimación Gamma-Pareto no derivable

De acuerdo a lo discutido en el apartado anterior podemos introducir un ligero error en el cálculo del punto umbral de tal forma que se cumplan las condiciones (10) y (12), pero minimizando la discontinuidad en la derivada para aproximarnos lo máximo a ese punto umbral que nos garantiza derivabilidad, condición (11).

Observemos el comportamiento de los parámetros si introducimos una ligera discontinuidad en la derivada en el punto x_{umbral}. Fijamos un *error_diff* como en (14)

$$f'_{Gamma}(x_{umbral}) - f'_{Pareto}(x_{umbral}) = error\_diff \qquad (14)$$

y substituimos el nuevo valor de x_{umbral} obtenido tanto en (10) como en (12). De cada substitución podemos despejar el valor de a en función de c y p, obteniendo $a_1(c,p)$ y $a_2(c,p)$. Si representamos la superficie $a_1(c,p)$ - $a_2(c,p)$ y buscamos su intersección con el plano 0, es fácil observar que es una recta con un valor de p fijo (véase Fig. 4). Es decir, el valor de p obtenido no depende del valor de c, con lo que fijada una discontinuidad tenemos un único valor de p válido (véase Fig. 5). De esta forma a la hora de realizar un modelo solamente debemos fijar la discontinuidad máxima que podemos aceptar, limitando nuestra decisión sobre el parámetro p. De seguido, o bien fijamos el parámetro a de la parte Gamma, o bien el parámetro c de la Pareto. La elección de este último nos permite controlar de forma más intuitiva cuánta parte de función queremos Gamma y cuánta Pareto.

En la Fig. 6 se muestra una función Gamma-Pareto obtenida con el modelo no derivable y tomando como parámetros iniciales c=5, a=1,15 y una discontinuidad en la derivada=1,5.

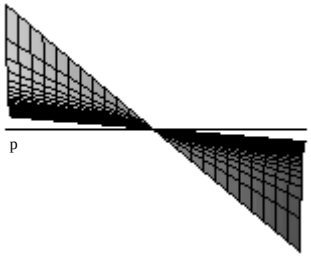


Figura 4. Intersección de $a_{21}(c,p)$-$a_{23}(c,p)$ con el plano 0

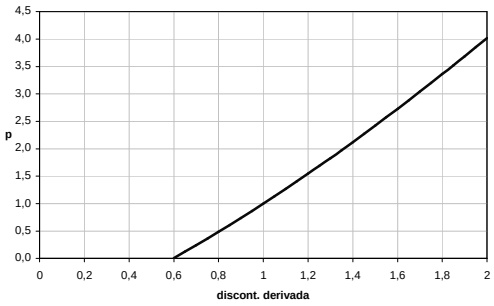


Figura 5. Relación p con la disccontinuidad en la derivada

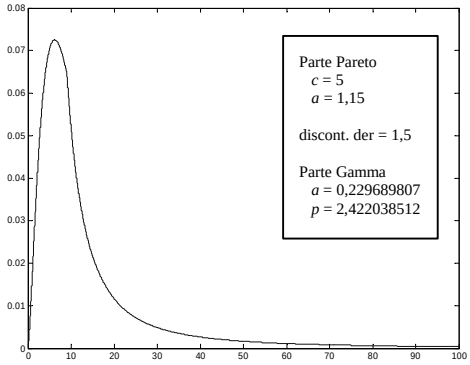


Figura 6. Función densidad de probabilidad Gamma-Pareto

4 Generación de tráfico sintético

A fin de obtener una simulación lo más real posible de los algoritmos de re-negociación de claves en entornos multicast seguro, se requiere generar patrones de tráfico que contemplen tanto llegadas como salidas en el sistema.

Si se considera el tiempo de permanencia total de un usuario en el servidor, la existencia de otros usuarios no afecta a la decisión individual de abandonar el sistema. En este caso, el tiempo de sesión por jugador se podría considerar exponencial [9].

Si interesa tener en cuenta únicamente el tiempo en que un jugador permanece en una misma partida (tiempo de juego) es posible que siga una distribución "heavy-tailed". El hecho de que afecte tanto la habilidad del jugador como el número de contrincantes que pueden echarle de la partida refuerza la hipótesis de fuerte correlación entre número de usuarios y tiempo de juego.

En cualquiera de los dos casos expuestos, el patrón final de tráfico a generar se puede obtener a partir de las peticiones de llegadas de usuarios. Éste se modela mediante un vector en el que cada posición es una unidad temporal lo suficientemente pequeña como para que no se produzcan dos o más peticiones mientras transcurre. Por convenio se representa una llegada como un +1 en la posición del vector

correspondiente al momento en que se producen. El resto de posiciones se representan con un 0. En el caso de que se quisiera modelar también las peticiones de salida se representarían con un -1.

El vector de llegadas puede obtenerse fácilmente a partir de un vector de tiempo entre llegadas, que tal y como se explica en el apartado 2 sigue una distribución "heavy-tailed". Este vector ha de contener en cada posición el tiempo que transcurre entre llegadas consecutivas de usuarios. Dicho tiempo lo modelamos mediante la función híbrida Gamma-Pareto definida en el apartado 3.

4.1 Generación de trazas con distribución Gamma-Pareto

Para obtener muestras que sigan una distribución Gamma-Pareto definimos dos zonas, la *Zona*1 o Gamma (15) y la *Zona*2 o Pareto (17). De esta forma generamos muestras con una distribución *Gamma*(*a*,*p*) con probabilidad *P*(*Zona*1) y muestras con una distribución *Pareto*(*c*,*a*) con probabilidad *P*(*Zona*2). Además despreciamos aquellos valores que devueltos por la distribución Gamma sean mayores que x_{umbral} y aquellos que devueltos por la Pareto sean menores que x_{umbral}. *P*(*Zona*1) y *P*(*Zona*2) se definen como en (16) y (18).

- $Zona1 = 0 < x < x_{umbral}$ (15)

$$P(Zona1) = \int_0^{x_{umbral}} f_{G-P}(x)dx \quad (16)$$

- $Zona2 = x_{umbral} < x < 8$ (17)

$$P(Zona2) = \int_{x_{umbral}}^{\infty} f_{G-P}(x)dx = 1 - P(Zona1) \quad (18)$$

La generación de muestras que sigan una distribución *Gamma*(*a*,*p*) está implementada en MATLAB como `gamrnd(p,1/a)`. Sin embargo, no hay una implementación que siga una distribución *Pareto*(*c*,*a*). A continuación se muestra cómo hemos resuelto el problema de generar trazas que sigan una distribución *Pareto*(*c*,*a*).

Es sabido que es computacionalmente sencilla la generación de números (pseudo) aleatorios siguiendo una distribución Uniforme entre dos números dados, que por simplificación son en la mayoría de casos 0 y 1. Por lo tanto, el problema se puede abordar como el paso de una distribución Uniforme a una Pareto. La solución se muestra en (19).

$$\text{Si } X \text{ es } U(0,1) \Rightarrow Y = cX^{-\frac{1}{a}} \text{ es } Pareto(c,a) \quad (19)$$

Es decir, que si *X* sigue una función de distribución Uniforme entre 0 y 1, $Y=cX^{-1/a}$ sigue una distribución Pareto de parámetros *c*, *a*.

Demostración

$F_Y(y) = P(Y \le y) = 1 - \left(\frac{c}{y}\right)^a = \int_{\left(\frac{c}{y}\right)^a}^{1} 1dx$, que es la $P\left(X \ge \left(\frac{c}{y}\right)^a\right)$ si *X* es Uniforme entre 0 y 1. Como $P\left(X \ge \left(\frac{c}{y}\right)^a\right) = P(cX^{-\frac{1}{a}} \le y) = F_Y(y) = P(Y \le y)$, se llega a la conclusión final, teniendo en cuenta solamente soluciones reales, de que $Y=cX^{-1/a}$.

4.2 Implementación en MATLAB

Para la generación de trazas con distribución Gamma-Pareto (vector del tiempo entre llegadas) hemos usado la herramienta MATLAB. MATLAB provee un lenguaje de programación cómodo y potente con funciones de generación de números aleatorios ya implementadas. Por otra parte, esta elección nos permite integrar fácilmente el generador de tráfico con el simulador de algoritmos de re-negociación de claves ya desarrollado con esta herramienta [10].

La siguiente función de MATLAB nos permite generar muestras aleatorias con distribución Gamma-Pareto una vez conocidos todos los parámetros:

```
function rand_number=rndgampareto(pzona1,
  a,p,c,alpha,xumbral)

  pzona=rand(1);
  if pzona<=pzona1
    rand_number=gamrnd(p,1/a);
    while rand_number > xumbral
        rand_number=gamrnd(p,1/a);
    end
  else
    rand_number= c/(rand(1))^(1/alpha);
    while rand_number <= xumbral
        rand_number=c/(rand(1))^(1/alpha);
    end
  end
```

La función `rndgampareto` nos devuelve un número aleatorio que sigue una distribución Gamma-Pareto como la definida en el apartado 3.3. El funcionamiento de la función es exactamente el explicado en el apartado 4.1. Primero miramos en que zona estamos, si `pzona=rand(1)` es menor que `pzona1` nos encontramos en la zona Gamma, en caso contrario nos encontramos en la zona Pareto. Si estamos en la zona Gamma hemos de generar un número aleatorio menor que x_{umbral} que siga una distribución Gamma. En caso contrario generamos un número aleatorio mayor que x_{umbral} que siga una distribución Pareto.

En el apartado 4.3 demostramos gráficamente que este algoritmo efectivamente genera números aleatorios que siguen una función densidad de probabilidad Gamma-Pareto.

4.3 Resultados obtenidos

A continuación se detalla la aplicación concreta del modelo de generación de tráfico para un tiempo entre llegadas con la siguiente distribución Gamma-Pareto. Suponemos conocida la cola Pareto, con parámetros c=1 y a=1,15. Establecemos una discontinuidad en la derivada en el punto umbral de 1,5, lo que nos fija un valor de p=2.422038507 (véase Fig. 5), a=1.148449032 y x_{umbral}=1.804205889.

Vector del tiempo entre llegadas

Mediante el programa MATLAB realizamos una simulación de 50.000 muestras. El tiempo entre llegadas (Fig. 7) presenta tal y como deseamos el fenómeno de LRD o dependencias a largo plazo que se reflejan en valores muy altos en algunas muestras. Evidentemente con probabilidad muy baja ya que se corresponden con valores del eje de ordenadas muy elevados.

Tomando el tráfico más al detalle (véase Fig. 8) se observa como la mayoría de valores se agrupan en tiempos entre llegadas muy pequeños (0-10 unidades de tiempo), que se corresponderían a la parte Gamma de la función.

Histograma del tiempo entre llegadas

La Fig. 9 muestra el histograma del tiempo entre llegadas. Para poder tener una mayor resolución hemos despreciado, a la hora de realizarlo, las muestras con un valor mayor que 10.000 lo que nos ha permitido incrementar la resolución a costa de un error despreciable. Se observa como el histograma refleja perfectamente el comportamiento Gamma-Pareto modelado (véase Fig. 10).

Recordemos que el modelo utilizado comporta un error o discontinuidad en la derivada en x_{umbral}. Este error se ha acotado por diseño tal y como hemos explicado en el apartado 3.3. En el histograma de la Fig. 9 hemos utilizado una discontinuidad en la derivada de 1,5, y puede observarse que la propia resolución del histograma impide apreciar la no derivabilidad de la función en x_{umbral}.

Vector de llegadas

Como se explica al principio de este apartado, a partir del tiempo entre llegadas es inmediato obtener un vector de llegadas al sistema. La Fig. 11 muestra un detalle de este tráfico. El tiempo está dado en unidades de tiempo, a las que daremos una duración u otra en función del análisis de tráfico real a simular, como por ejemplo el presentado en [4]. El vector de llegadas sirve como datos de entrada para el simulador de algoritmos de re-negociación de claves comentado en 4.2, lo cual debe permitirnos obtener resultados adecuados de dichos algoritmos sobre escenarios reales.

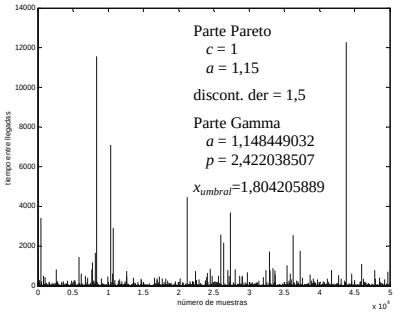


Figura 7. Tiempo entre llegadas

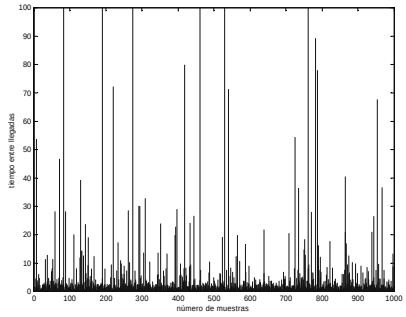


Figura 8. Detalle del tiempo entre llegadas

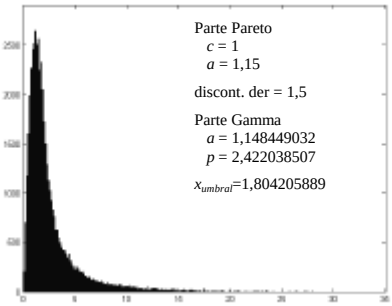


Figura 9. Histograma del tiempo entre llegadas

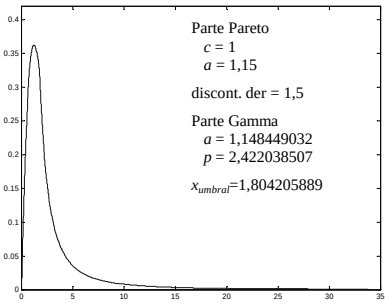


Figura 10. Función Gamma-Pareto modelada

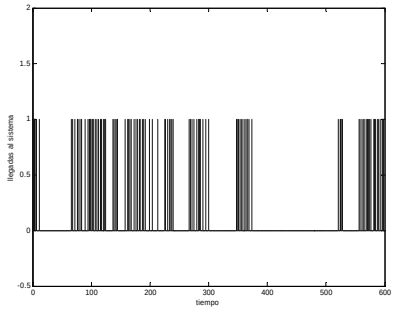


Figura 11. Detalle de tráfico sintético generado

5 Líneas Futuras

Hasta el momento todos los parámetros utilizados en el método heurístico de generación de tráfico se basan en el estudio de campo realizado por Henderson en [4], de todas formas sería deseable obtener estos parámetros a partir de nuestro propio entorno de pruebas. De hecho, del análisis de Henderson en [4] sólo obtenemos una estimación del parámetro *a*, lo cual nos otorga todavía dos grados de libertad en el modelo. Un análisis propio de tráfico real nos debe permitir fijar más parámetros y, por lo tanto, reducir los grados de libertad del modelo propuesto.

Actualmente no existe una red multicast suficientemente extendida. La red MBone (Multicast Backbone) no tiene la aceptación que se le creía hace unos años, y parece que poco a poco va siendo desplazada, al menos en Europa, por la red GEANT. El hecho de que el número de sesiones y usuarios multicast sea muy reducido, nos lleva a la conclusión de que son más representativos los patrones de comportamiento de usuarios de juegos en red sobre Internet que los patrones que podamos obtener de una sesión multicast.

Como principal línea futura se propone la implementación de un servidor de juegos sobre Internet con el fin de obtener estadísticas propias del comportamiento de usuarios. Esto permitirá contrastar y validar el tráfico real para llegadas de usuarios con el tráfico obtenido mediante el método propuesto en este artículo. Asimismo se podrán estudiar las hipótesis realizadas en cuanto a tiempo de sesión y tiempo de juego de los usuarios, parámetros que fijarán el comportamiento de las salidas.

6 Conclusiones

En este artículo se ha presentado un método heurístico de generación de tráfico sintético siguiendo una distribución Gamma-Pareto. Este tipo de tráfico modela el comportamiento de usuarios en juegos en red y facilitará la simulación de algoritmos de renegociación en entornos multicast. Se ha demostrado que la unión de una función Gamma y otra Pareto no puede garantizar las tres condiciones deseadas de continuidad, derivabilidad y área 1 para $1 < a < 2$. Como consecuencia, se ha desarrollado un modelo que sólo garantice la primera y tercera condición, acotando la discontinuidad en la derivada. Se ha desarrollado este modelo de generación mediante el software MATLAB, esto permite integrarlo fácilmente en el simulador de algoritmos de renegociación de claves multicast disponible. Por último los resultados obtenidos demuestran que los patrones generados se ajustan perfectamente a la teoría matemática aplicada.

Agradecimientos

Este trabajo ha sido soportado por el proyecto DISQET [CICYT TIC2002-00249], dentro del Plan Nacional de I+D.

Referencias

[1] Harney, Harder. “Logical Key Hierarchy Protocol (LKH)”. I-D Harney-sparta-lkhp-sec-00. Marzo 99. Work in progress.

[2] Balenson, McGrew, Sherman. “Key Management for large Dynamic Groups: One-Way Function Trees and Amortized Initialization”. I-D irtf-smug-groupkeymgmt-oft-00. Agosto 2000. Work in progress

[3] Yang Li, Lam Gouda. “Batch Rekeying for Secure Group Communications”. ACM SIGCOMM 2001, San Diego, Agosto 2001

[4] T. Henderson, S. Bhatti. “Modelling user behaviour in networked games”. Junio 2001.

[5] A. Feldmann, A. C. Gilbert, W. Willinger, T. G. Kurtz. “The changing nature of network traffic: Scaling phenomena”. Computer Communication Review, 28(2):5--29, Abril 1998.

[6] V. Paxson, S. Floyd. “Wide-area traffic: The failure of Poisson modelling”. IEEE/ACM Transactions on Networking, 3(3):226--244, Junio 1995

[7] M. S. Borella. “Source models of network game traffic”. Computer Communications, 23(4):403--410, Febrero 2000.

[8] B. M. Hill. “A simple general approach to inference about the tail of a distribution”. The Annals of Statistics, 3(5):1163–1174, Septiembre 1975.

[9] V. A. Bolotin. “Modeling call holding time distributions for CCS network design and performance analysis”. IEEE Journal of Selected Areas In Communications, 12(3):433–438, Abril 1994.

[10] J. Pegueroles, F. Rico-Novella. “Balanced Batch LKH: New Proposal, Implementation and Performance Evaluation”. IEEE Symposium on Computers and Communications - ISCC'2003, 2003.

Contribución a la optimización de sistemas de localización en redes celulares móviles: Smart Layer *

Israel Martín-Escalona, Francisco Barceló

Dept. d'Enginyeria Telemàtica de la Universitat Politècnica de Catalunya

Av. Canal Olímpic s/n, Barcelona 08860, Spain.

imartin@entel.upc.es ; barcelo@mat.upc.es

***Abstract**. This paper proposes a new approach to reduce the costs (e.g. signalling, latency, power consumption, etc.) of providing location services on cellular networks, especially on those that use a combination of location techniques in a hybrid fashion. For this purpose, an application layer is designed for the Serving Mobile Location Centre (SMLC). This smart layer is intended to manage location requests according to their Quality of Service (QoS) requirements (mainly accuracy and delay), and to decide about the most convenient standalone or hybridized technique, i.e. the one that minimizes the costs involved in the mobile-station position achievement. In addition, this layer takes advantage of possible concurrency and of the history of previous requests to improve the overall location system performance.*

1 Introducción

1.1 Servicios Basados en Localización

En esta última década, los sistemas de telefonía móvil públicos han experimentado un crecimiento muy superior al que la más optimista de las predicciones pudiera haber vaticinado. Los servicios de voz ofrecidos en este tipo de sistemas han crecido en demanda de una forma similar a como lo han hecho los servicios de datos en internet. Sin embargo, tras de esta fase de clara expansión, la mayoría de los expertos parecen coincidir en alertar de un cierto nivel de saturación a corto plazo para este tipo de servicios convencionales basados en voz. Este hecho ha motivado que los principales operadores del sector se hayan iniciado la búsqueda nuevos tipos de servicios, más atractivos que los actuales. Pese a la existencia de una gran variedad de servicios de datos en la red fija, la inmensa mayoría de ellos requieren de altos anchos de banda y bajas latencias, requisitos que no pueden ser satisfechos por los sistemas celulares actuales, al menos hasta que el despliegue de tecnologías como UMTS (Universal Mobile Telephone System) o WLAN (Wireless Local Area Network) se haga efectivo. A pesar de todas estas limitaciones, el futuro a corto plazo parece dibujar un escenario en el que múltiples aplicaciones inyecten tráfico sobre redes de telefonía móvil [1].

En este nuevo escenario, la localización de las estaciones móviles se muestra como un servicio clave [2]. Además de serlo por si mismo -un usuario podría desear conocer cual es su situación exacta-, la información de localización podría ser utilizado por otros servicios de valor añadido. En este último tipo de servicios, el usuario no desea conocer su posición de manera explícita, aunque ésta es utilizada por una aplicación de nivel superior para llevar a cabo un determinado servicio. Claros ejemplos de este tipo de aplicaciones son los servicios de emergencia, en los que las instituciones implicadas requieren de la información de localización del usuario siniestrado. En este marco debe notarse que, este tipo de servicios basados en localización (LCS) pueden ser empleados por los operadores para mejorar gestión de sus propios recursos y de esta forma optimizar las prestaciones y el rendimiento del sistema de comunicaciones que los suministra [3, 4].

1.2 Necesidad de hibridación

El nivel de precisión requerido por los distintos LCS depende del servicio específico a ofrecer, pudiendo variar desde el centenar hasta tan sólo unos pocos metros. Hoy en día existen múltiples tecnologías capaces de proporcionar distintos nivel de precisión y latencia [5, 6, 7, 8, 9, 10, 11, 12]. Los métodos basados en identificador de celda (Cell-ID) son los que disponen de una disponibilidad absoluta, si bien gozan de una precisión altamente variable, la cual

* Este artículo ha sido financiado por la Comisión Europea a través del proyecto EMILY IST-2000-26040 y el gobierno español a través del proyecto CICYT TIC2000-1041-C03-01.

depende del tamaño de la celda o sector en cuestión. De esta manera, estos métodos presentan niveles de precisión que parten del centenar de metros (Ej. microceldas) y que pueden llegar a la decena de kilómetros (Ej. macroceldas en entornos rurales). Los métodos basados en la potencia recibida (NMR) aparecen como un complemento a los anteriores, ofreciendo baja latencia a costa de un nivel de precisión también bajo, debido principalmente a la variabilidad del medio radio. Otros métodos como el Timing Advance (TA) y Round Trip Time (RTT) se encuentran disponibles en las principales redes de telefonía móvil celular, por lo que su uso no plantea ningún problema de disponibilidad. Sin embargo, el tiempo hasta obtener las cifras de RTT/TA es altamente variable ya que este proceso puede requerir de la intervención del terminal móvil. Además, la precisión ofrecida por este tipo de técnicas no puede compararse a la alcanzada mediante métodos basados en triangulación. Normalmente, con el fin mejorar estas cifras de precisión, se emplean técnicas basadas en el ángulo de llegada (AoA). Sin embargo, esta serie de técnicas resultan muy costosas y adolecen de múltiples limitaciones si se las compara con las basadas en triangulación.

El presente estudio se centra en la hibridación de métodos basados en triangulación, los cuales en general ofrecen mejores resultados en cuanto a precisión que los no basados en este principio. Sin embargo, este tipo de técnicas adolece de una menor disponibilidad: nótese que las tecnologías basadas en Cell-ID, TA/RTT o NMR ofrecen una cobertura a nivel de localización total, ya que la información de localización asociada a una estación móvil está disponible siempre que dicha estación se encuentre conectada a la red. Por el contrario, los métodos basados en triangulación requieren de un mínimo de tres emisores de señal (ej. satélites, estaciones base, etc.) para poder cursar una petición de localización en 2 dimensiones. Este es el principio sobre el que operan las técnicas de triangulación terrestre como pueden ser el Enhanced Observed Time Difference (E-OTD) en sistemas GSM (Global System for Mobile Communications) y GPRS (General Packet Radio Service), Observed Time Difference of Arrival (OTDOA) funcionando sobre sistemas UTMS o el sistema de navegación basado en satélites Global Positioning System (GPS). Assisted GPS (A-GPS) es una técnica similar a GPS que emplea una red terrestre para enviar información de asistencia, como puede ser el almanac, efemérides, etc., a la estación móvil. Esto se traduce en una ganancia aproximada de 20 dB en la sensibilidad del receptor con respecto a la ofrecida por sistemas GPS, lo cual se traduce en una mejora considerable los tiempos hasta la primera conexión (TTFF). A pesar de ello, debe tenerse en cuenta que los receptores A-GPS operan en peores condiciones que los GPS. Esto es debido a que en general, los terminales móviles A-GPS son transportados en bolsillos, carteras, bolsas que viaje, etc., mientras que los utilizados en GPS se emplean normalmente en sistemas de navegación sobre vehículos en los que la recepción de señal es continua. Además, el la técnica de localización A-GPS está disponible únicamente en determinados sistemas, encareciendo los costes de producción de dichos terminales. Además esta tecnología proporciona excelentes prestaciones en entornos al aire libre, junto a un rendimiento pobre en escenarios de interior.

Debe notarse que E-OTD ofrece una baja disponibilidad en entornos rurales, donde es poco probable la presencia de tres estaciones base de forma concurrente. Por otro lado, A-GPS presenta una baja precisión en escenarios de interior, donde la potencia procedente de los satélites es mínima y la desmodulación de la misma, menos probable. Tal y como puede observarse, estos dos sistemas parecen ser complementarios: E-OTD en entornos en los que no se goza de visibilidad directa con los satélites y A-GPS en escenarios al aire libre. Este hecho sugiere que ambos sistemas podrían combinarse y de esta forma, minimizar las limitaciones del sistema resultante: tan sólo los escenarios rurales e interiores presentarían problemas de cobertura en términos de localización [13].

1.3 Objetivos y estructura

Este paper propone la implementación de una capa de aplicación en el SMLC: la Smart Layer. Esta nueva capa tiene por objetivo minimizar el coste asociado a los servicios de localización proporcionados en el sistema. Dicho coste está regido por diversos factores, los cuales dan lugar a una función de coste. El presente documento define esta función de coste de manera genérica y propone algunos factores de coste a modo de referencia. Además de la función de coste, la Smart Layer propone vías alternativas para llevar a cabo la minimización de coste: gestión de procesos LCS concurrentes, datos históricos e identificación dinámica de entorno.

El resto de este documento se estructura de la siguiente forma. La sección 2 define la Smart Layer en profundidad. Para ello, dicha sección se divide en tres secciones. La sección 2.1 define la arquitectura que da soporte a la Smart Layer, mostrando los elementos que deben ser modificados para implantar esta arquitectura. La sección 2.2 por su parte, presenta los factores de coste propuestos como base de la función de coste multi-objetivo. Por último, la sección 2.3 propone el uso de datos históricos con el fin de optimizar el rendimiento global del sistema. En la sección 3 se propone un algoritmo de alto nivel como posible implementación de la Smart Layer. En la sección 4, se evalúa el rendimiento de dicha implementación desde el punto de vista de concurrencia de procesos de localización. Por último, en la sección 5 se incluyen las conclusiones a las que han llegado los autores.

2 Smart Layer

2.1 Arquitectura del sistema

La Smart Layer es una capa de aplicación que tiene por objetivo mejorar las capacidades tanto de los nuevos sistemas de localización, como de los ya implantados. De esta forma, la Smart Layer opera eligiendo la técnica de localización óptima para cada proceso de localización, al tiempo que maximiza el rendimiento global del sistema. La Smart Layer entiende como técnica de localización óptima aquella que, además de cumplir los requerimientos de QoS, minimiza una función de coste multi-objetivo previamente definida. La QoS en un LCS se define como la combinación de precisión mínima y latencia máxima requerida por el LCS en cuestión. Debe notarse que la técnica elegida como óptima puede ser híbrida o no, ya que la Smart Layer no impone restricciones en este sentido.

La Fig. 1 muestra la arquitectura de un sistema de localización híbrido integrando la Smart Layer. Tal y como se muestra en dicha figura, todas la funcionalidades de la Smart Layer se concentran en un elemento de red llamado SMLC. Este elemento puede ser único, dando lugar a una plataforma de localización centralizada, o puede haber múltiples instancias del mismo, dando lugar en este caso a un sistema de localización distribuido. En cualquier caso, el SMLC se comporta como el núcleo gestor, local o global, de las peticiones de localización recibidas por el sistema. La razón de implementar la Smart Layer en el SMLC responde al hecho de que esta capa de aplicación pretende ser transparente al sistema de localización. Es por ello que, la Smart Layer utiliza las propias facilidades del sistema de localización, concentradas en cada SMLC, en lugar de añadir nuevas funcionalidades al mismo. Este diseño permite por lo tanto, minimizar el coste tanto de implantación en sistemas ya desplegados como de implementación en sistemas futuros.

La Fig. 2 presenta el diagrama de bloques asociado a la implementación de la Smart Layer en el SMLC. En dicho diagrama se especifican las entradas necesarias para que la Smart Layer pueda llevar a cabo sus funciones. Algunas de estas entradas, tales como la

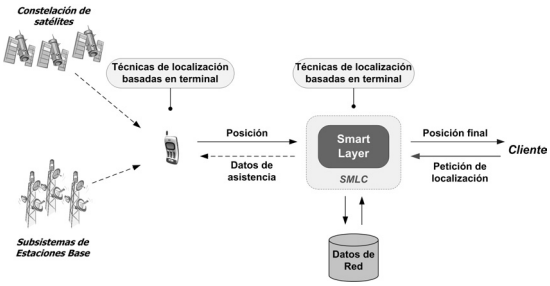


Figura 1: Arquitectura de un sistema de localización híbrido incluyendo la Smart Layer

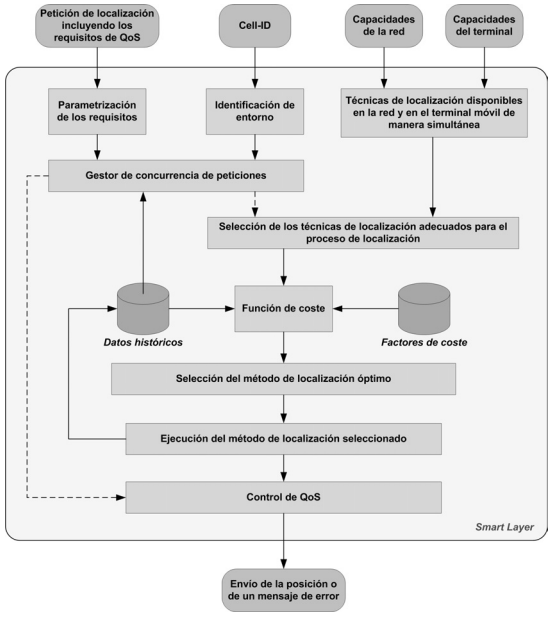


Figura 2: Diagrama de bloques de la Smart Layer

información y particularidades referentes al LCS (Ej. el identificador de cliente LCS, petición de localización 2D/3D, etc.), los requerimientos de QoS (es decir, la precisión y el retardo) o el Cell-ID, están asociadas directamente con cada petición de localización. De esta forma, estos parámetros deben ser proporcionados a el SMLC sea cual sea el sistema de localización que se considere. Sin embargo, existen otra serie de parámetros que son requeridos de forma explícita por la Smart Layer, como son las capacidades en términos de localización tanto de la red como de la estación móvil, es decir el conjunto de técnicas de localización disponibles tanto en el terminal como en la red. Por consiguiente, toda implementación de la Smart Layer deberá especificar el procedimiento que se seguirá para acceder a esta información.

Tal y como se muestra en la Fig. 2, la Smart Layer se divide en cuatro procesos principales:

- *Descartar técnicas de localización.* Este es el primero de los cuatro procesos a ejecutar en la Smart Layer. Su propósito es descartar toda técnica de localización que no sea adecuada para el proceso de localización que se está solicitando. La decisión de marcar una técnica de localización como no válida depende del tipo de LCS (más concretamente de la QoS requerida por el LCS), el tipo de entorno en el que se enmarca la MS, así como de las técnicas de localización permitidas de forma simultánea tanto en la red como en el terminal móvil.

- *Escoger el método de localización óptimo.* Este proceso tiene por objetivo elegir, de

entre las técnicas de localización disponibles tras ejecutar el proceso anterior, la más adecuada para el proceso de localización a realizar. Este método de localización puede coincidir con una sola técnica de localización o incluir varias de ellas dando lugar a una técnica híbrida.

- *Gestión de los procesos de localización.* Este proceso verifica que la posición obtenida tras ejecutar la técnica de localización escogida en el proceso anterior, cumpla con los requisitos de QoS especificados en la petición de localización.

- *Mejora del rendimiento.* Este proceso mantiene un repositorio de datos históricos con el objetivo de mejorar las prestaciones globales del sistema de localización. De esta forma, tras obtener la información de posicionamiento, sea o no satisfactoria, se procederá a actualizar esta base de datos histórica. El objetivo final de este proceso es que un mismo sistema de localización sea capaz de cursar más tráfico.

Las siguientes secciones definen de forma más precisa, el modo de funcionamiento de las principales capacidades de la Smart Layer.

2.2 Función de coste y factores de coste

Una de las principales mejoras introducidas por la Smart Layer sobre los sistemas de localización actuales es la capacidad para minimizar el coste de los procesos de localización. Este coste, tal y como se especifica en la Smart Layer, no está predeterminado sino que se define mediante una función de coste multi-objetivo. Dicha función de coste puede observarse en la Ecuación (1), donde *CF* es el nombre de la función de coste, LT_i indica la técnica de localización número *i*, la cual está siendo evaluada, donde $1 \le i \le N_{LT}$; α_k indica el peso del factor de coste *k* y N_{LT} es el número de técnicas de localización consideradas como válidas para el proceso de localización.

$$CF(LT_i) = \sum_k \alpha_k Cost\ Factor_k(LT_i). \quad (1)$$

Tal y como muestra la Ecuación (1), la función de coste utiliza diversos factores, cada uno de ellos con un determinada peso dentro de dicha función, para cuantificar el coste asociado al uso de una determinada técnica de localización. De esta forma, se procederá a escoger el método de localización óptimo, definido como el que proporciona coste mínimo.

Debe notarse que la evaluación del coste mediante la función definida en la Ecuación (1) produce valores adimensionales, puesto que dicha función integra múltiples factores heterogéneos. En cuanto al número de factores de coste permitidos en dicha función o el número máximo de técnicas de localización a utilizar, la Smart Layer no incluye ninguna restricción. Es tarea del operador especificar estos parámetros, así como los pesos asociados a cada uno de los factores de coste que integran la función. De esta forma, la Smart Layer se presenta como una solución altamente particularizable, ya que en cualquier momento se pueden añadir a su definición tanto factores de coste como técnicas de localización.

Los siguientes párrafos definen algunos factores de coste que pueden ser usados como marco de referencia para una implementación tipo de la Smart Layer.

2.2.1 Volumen de señalización (VS)

Este factor de coste tiene por objetivo penalizar el uso de aquellas técnicas de localización que generan un gran volumen de tráfico. La Ecuación (2) presenta la expresión propuesta para estimar este factor, donde LT_i indica la técnica de localización que está siendo evaluada y N_P es el número de paquetes utilizados por dicha técnica.

$$VS(LT_i) = \sum_{k=1}^{N_P} Length(Packet_k). \quad (2)$$

2.2.2 Tiempo Total de Transmisión (TTT)

Este factor de coste se incluye para considerar muchas de las limitaciones asociadas al envío de información de localización a través de las distintas interfaces de red: el tamaño de la información, el número de interfaces a atravesar y las limitaciones propias de cada canal utilizado. De esta forma, el *TTT* pretende favorecer aquellas técnicas de localización que requieren del envío de menos información y utilizan los canales de mayor throughput para llevar a cabo la transmisión de dicha información.

El valor del *TTT* se obtiene sumando los Tiempos Parciales de Transmisión (*TPT*) asociados a la petición de localización, tal y como se muestra en la Ecuación (4). La Ecuación (3) define el TPT_k como la cantidad de bytes enviados (*Info*) divididos por el throughput en bytes/μs del canal *k* por el que se está transmitiendo dicha información. El valor resultante se redondea entonces al entero inmediatamente superior. Este redondeo es realizado para favorecer el rendimiento computacional de la Smart Layer en cuanto a aritmética se refiere.

$$TPT_k = \left\lceil \frac{Info\ (bytes)}{Throughput_k\ \left(bytes/\mu s\right)} \right\rceil\ [\mu s], \quad (3)$$

$$TTT = \sum_{k=0}^{N_C} TPT_k \quad [\mu s]. \tag{4}$$

2.2.3 Consumo de potencia

El consumo de potencia juega un papel vital en toda aplicación destinada a terminales móviles. Por consiguiente, esta variable debería ser minimizada en cualquier sistema de localización. La cuantificación de este factor de coste resulta complicada, ya que el consumo de potencia asociado a una técnica de localización en concreto, depende directamente del rendimiento del propio terminal móvil. Esta información no se encuentra disponible en la red, y actualmente no existe ningún protocolo definido que permita una comunicación con el MS para conocer este dato. Por consiguiente, debido a estas limitaciones, los autores han decido proponer el uso de una tabla de cuantificación preestablecida, basada en la experiencia del propio operador.

2.2.4 Balanceo de Carga (BC)

Este factor pretende favorecer el uso de una determinada técnica de localización sobre el resto. Para poder hacer esto, el operador debe establecer β_P y β_{NP}, que son los factores asociados a las situaciones de penalización y no penalización respectivamente. El rango de valores permitidos para estos dos factores es cualquiera, siempre que se garantice que $\beta_{NP} < \beta_P$. Además de esto, el operador es libre de establecer el porcentaje de carga deseado para cada una de las técnicas de localización. De esta forma, cuando una petición LCS llega, se genera un número aleatorio uniformemente distribuido entre 0 y 1 (ambos valores incluidos). Este número se utiliza de forma conjunto con estos porcentajes para decidir cuál de las técnicas de localización no será penalizada. Finalmente, se utiliza las Ecuaciones (5) y (6) para cuantificar el Coste de BC (*CBC*)

$$CBC_{No\ Penalizado} = \beta_{NP}\,, \tag{5}$$

$$CBC(LT_i)_{Penalizado} = \frac{\beta_P}{Porcentaje(LT_i)}\,. \tag{6}$$

Un ejemplo del procedimiento descrito anteriormente podría ser el siguiente. Se desea cuantificar tres técnicas de localización mediante este factor de coste. Los porcentajes de carga deseados para cada una de las tres técnicas son del 0.1, 0.25 y 0.65 respectivamente. Tal y como establece el procedimiento descrito anteriormente, se genera un número aleatorio. Para este ejemplo, se considera que este número es 0.67. De esta forma, las dos primeras técnicas serían penalizadas mientras que la tercera no lo sería.

2.2.5 Probabilidad de Indoor

El propósito de este factor de coste es penalizar aquellas técnicas de localización que no sean adecuadas para su ejecución en entornos considerados interiores. La Ecuación (7) presenta una función cuyo objetivo es evaluar si una técnica de localización es adecuada para su uso en escenarios de interior o no. En dicha ecuación, LT_i representa la técnica de localización que está siendo evaluada. La Ecuación (8) por su parte, es la encargada de cuantificar el coste asociado a este factor (*IPC*). En esta ecuación, P_{indoor} indica la probabilidad media de estar situado en un entorno de interior, dentro del área de cobertura considerada.

$$Indoor(LT_i) = \begin{cases} 1, \text{ si } LT_i \text{ es una técnica indoor} \\ 0, \text{ caso contrario} \end{cases}, \tag{7}$$

$$IPC(LT_i) = Indoor(LT_i)[1 - 2P_{indoor}] + P_{indoor}\,. \tag{8}$$

2.3 Datos históricos

2.3.1 Identificación de entorno

La identificación de entorno proporciona a la Smart Layer información acerca de las condiciones reales del canal radio, en el escenario en el que se encuentra el MS. Esta caracterización de escenario se realiza capturando datos representativos durante la ejecución del servicio LCS, para posteriormente actualizar una base de datos histórica. La especificación de la Smart Layer no limita la cantidad de variables consideradas a la hora de generar esta base de datos histórica. De esta forma, será el operador el que, según sus intereses, defina los parámetros oportunos que van a ser considerados al llevar a cabo la identificación de entorno. Como referencia, los autores proponen una caracterización de escenario basada en la probabilidad de indoor. Esta caracterización puede mejorarse de diversas formas. Una de ellas, la conocida como sub-celling, consiste en dividir el área de cobertura definida por la celda en regiones de menor tamaño llamadas sub-celdas. El tamaño de estas sub-celdas queda a elección del operador. De esta forma, el mismo procedimiento seguido para el caso de las celdas puede ser aplicado a las sub-celdas, si bien el menor tamaño de estas últimas proporciona mayor precisión al sistema en términos de identificación de entorno.

2.3.2 Minimización del retardo

Uno de los factores que más puede limitar el funcionamiento de la Smart Layer es el retardo introducido desde que la petición es recibida, hasta que el método de localización óptimo es seleccionado. Para minimizar este retardo, la Smart Layer ha definido dos tipos de datos históricos. Un primer grupo tiene por objetivo proporcionar

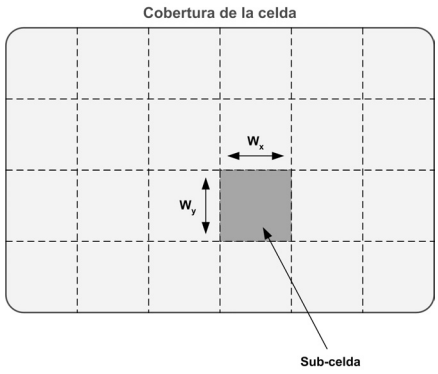


Figura 3: Identificación mediante sub-celling

información sobre el último proceso de localización cursado con éxito. De esta forma, este grupo está compuesto por variables asociadas a un único MS. El segundo grupo incluye información histórica asociada a un LCS. Este grupo pretende evitar cálculos innecesarios dentro de la Smart Layer, especialmente los derivados de la ejecución de la función de coste. Este objetivo puede alcanzarse gracias a la periodicidad asociada a los LCS: normalmente los procesos LCS no generan una única petición sino que envían varias según una cierta tasa. De esta forma, las nuevas peticiones de localización generadas dentro de un mismo proceso LCS pueden sacar provecho de los datos históricos de la primera petición y por consiguiente, evitar algunos procedimientos de la Smart Layer.

3. Implementación

Gran parte de la efectividad de la Smart Layer pasa por la manera en que ésta se implemente. Debido a esto, el algoritmo propuesto en este documento ha sido generado con el propósito de minimizar el retardo introducido por la Smart Layer y optimizar la concurrencia de peticiones de localización.

La Fig. 4 ilustra el algoritmo seguido para la implementación de la Smart Layer. El funcionamiento de la Smart Layer puede resumirse en las siguientes líneas: Cuando una nueva petición de localización es recibida, el sistema verifica si hay otra petición de localización cursándose, asociada al mismo MS. En caso de que sea así y que la precisión y retardo requeridos para la petición en curso sean más restrictivos que los demandados para la nueva petición de localización, la Smart Layer aplaza la ejecución esta última petición para así reaprovechar el resultado del proceso de localización que está en curso. En caso contrario, es decir, que no haya ninguna petición de localización siendo cursada, la Smart Layer verifica si dispone de datos históricos para el MS en cuestión. Si el sistema dispone de datos históricos asociados al MS y el servicio de localización, la Smart Layer pasa a caracterizar el entorno mediante dicha información. Si las condiciones no han variado (mismo CI y mismo LCS), el algoritmo ejecuta el mimo método de localización que fue escogido para la última petición de localización. Si por el contrario no se dispone de datos históricos para un MS y LCS específicos, o las condiciones en las que se ha recibido esta nueva petición de localización son distintas a las referenciadas en el histórico, se requiere de la intervención de la función de coste. De esta forma, el método de localización óptimo es escogido y ejecutado, con el objetivo de obtener la información de posicionamiento de la MS. Una vez el SMLC ha recibido dicha información y con independencia del resultado, la Smart Layer procede a actualizar la base de datos histórica. Tras esto, la Smart Layer verifica si se ha recibido un error o una posible posición. En caso de recibir información de posicionamiento, la Smart Layer pasa a comprobar si los requisitos de QoS han sido alcanzados. En caso de que no sea así o que se haya recibido un error, la Smart Layer intenta repetir todo el proceso, siempre y cuando la estimación del tiempo requerido para ello no supere los requerimientos de retardo especificados para el LCS. Si no pudiera repetirse el procedimiento, la Smart Layer reportaría la última posición conocida o un error en función del tipo de LCS y de su propia implementación. Finalmente, si se ha podido obtener una posición que cumpla con los requerimientos de QoS especificados, la Smart Layer envía dicha información al cliente LCS que realizó la petición, con lo que el algoritmo finalizaría.

4. Evaluación de la Smart Layer

Esta sección pretende proporcionar una evaluación preliminar del rendimiento de la Smart Layer. De esta

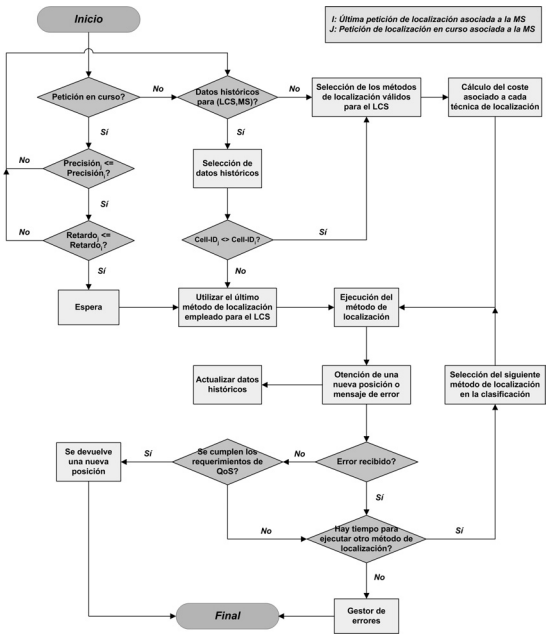


Figura 4: Algoritmo propuesto para la Smart Layer

forma, esta sección se centra en estudiar las mejoras aportadas por la Smart Layer mediante la gestión concurrente de LCS. Este Nivel de Concurrencia (NC) se define como la probabilidad de recibir una petición LCS mientras otra está siendo servida, al tiempo que el QoS requerido para esta nueva petición es menos restrictivo que el de la petición que está en curso.

La Tabla 1 muestra el valor asociado a PS_1 y PS_2, los cuales se corresponden con la función de probabilidad de tener un LCS de tipo *i* en entorno rural y (sub)urbano respectivamente. El número *i* referente al tipo de LCS, coincide con la clasificación proporcionada en dicha tabla. Debe notarse que, cuanto mayor es el valor del tipo de LCS, menor precisión requiere éste. La Tabla 2 muestra la probabilidad de recibir *i* peticiones de localización mientras se está cursando una. $PN_{1,i}$ y $PN_{2,i}$ expresan el valor de dicha probabilidad en entorno rural y (sub)urbano respectivamente.

El *NC* puede ser calculado mediante la Ecuación (9), donde $N_{servicios}$ indica el número de LCS considerados en el sistema de localización, e *i* hace referencia al tipo de escenario (es decir, 1 para escenarios rurales y 2 para escenarios (sub)urbanos).

$$NC_i = \sum_{j=1}^{\infty}\left[PN_{i,j}\left(\sum_{k=1}^{N_{Servicios}} PS_{i,k}\left(1-\sum_{l=1}^{k-1} PS_{i,l}\right)\right)\right]. \quad (9)$$

Teniendo en cuenta la Ecuación (9) y los datos mostrados en las Tablas 1 y 2, las cuales han sido extraídos de [14], el *NC* resultante es del 0.75% y del 12.68% para el caso de escenarios rurales y (sub)urbanos respectivamente.

Pese a que los resultados en entorno rural no plantean una enorme ganancia con respecto a los sistemas actuales, en entornos urbanos la Smart Layer evita que el 12.68% de las peticiones LCS sean llevadas a cabo. Dichas peticiones reaprovechan de esta forma los resultados de otras que sí han sido ejecutadas. De esta forma, la reducción en cuanto a señalización y consumo de potencia en los terminales puede considerarse como más que apreciable. Además de esto, el uso de la función de coste representa también una reducción en el coste asociado a la ejecución del LCS. Por ejemplo, bajo la suposición de que tan sólo los servicios de asistencia y de tipo *pull* requieren de una alta precisión y teniendo en cuenta los datos expuestos en la Tabla 1, en entorno urbano se puede alcanzar una reducción en el coste en hasta el 50% de los casos. Si estas mismas consideraciones son aplicadas en entorno rural, la reducción en coste asciende hasta el 42%.

Tabla 1: Función de probabilidad de los distintos tipos de LCS

Clasificación	*Servicio*	$PS_{1,i}$	$PS_{2,i}$
1	Assistance	0.58	0.35
2	Pull	0.42	0.25
3	Games	0.00	0.25
4	Chat/Community	0.00	0.08
5	Push	0.00	0.07
	Total	1.00	1.00

Tabla 2: Probabilidad de peticiones concurrentes

i	$PN_{1,i}$	$PN_{2,i}$
0	99 %	78.59 %
1	1 %	15.41 %
2	0 %	6 %
>2	0 %	0 %
Total	100 %	100 %

5. Conclusión

Este artículo propone la utilización de una capa de aplicación situada el SMLC, en sistemas tanto de nuevo desarrollo como ya implantados: la Smart Layer. Esta capa opera gestionando los procesos de localización bajo la premisa de maximizar el rendimiento al tiempo que se minimiza el coste asociado a cada uno de los LCS. Para lograr esto, la Smart Layer define una función de coste multi-objetivo, la cual permite a los operadores redefinir el concepto de coste óptimo, teniendo en cuenta únicamente sus intereses.

La Smart Layer implementa nuevas y atractivas características como son la definición sin restricciones de múltiples factores de coste, la identificación de entorno adaptativa y la gestión concurrente de diversos LCS. Esto permite que cualquier sistema de localización vea mejorada sus prestaciones al tiempo que reduce los costes asociados a cada petición de localización. Los resultados numéricos han corroborado esta optimización del sistema, mostrando una reducción en la señalización y en el consumo de potencia (entre otros parámetros) en hasta el 12.68% de los casos. Esta reducción ha sido obtenida únicamente mediante las capacidades de gestión concurrente de LCS de que dispone la Smart Layer. De esta forma, el 12.68% de las peticiones son capaces de reaprovechar los resultados de peticiones de localización que ya se encontraban en curso. Además de esta reducción, en determinadas condiciones, el uso de la función de coste dentro de la Smart Layer puede resultar en una optimización del sistema en hasta el 50% de las ocasiones.

Los resultados derivados del uso de la Smart Layer en escenarios rurales son menos notorios que los obtenidos en el caso (sub)urbano. Pese a ello, la Smart Layer es capaz de obtener una reducción en el coste asociado a cada LCS en hasta un 42% de los casos. Estos ahorros en coste son debidos al uso de la función de coste introducida por la Smart Layer.

Agradecimientos

Los autores quieren agradecer a sus colegas del proyecto EMILY IST-2000-26040 su opinión en diversos aspectos de este artículo.

Referencias

[1] C. Drane, M. Macnaughtan; C. Scott, ”Positioning GSM telephones“, IEEE Communications Magazine, Volume: 36 Issue: 4, April 1998, pp. 46 – 54, 59.

[2] Third Generation Partnership Project, “3G TS 22.071: Location Services (LCS)”.

[3] J. Y. Lee, W. C. Y. Lee, “Optimize CDMA System Capacity with Location”, *Proc. of IEEE PIMRC 2001*, San Diego, USA, October 2001, pp. 17-21.

[4] IST Project, “CELLO: Cellular Network Optimisation based on Mobile Location”, http://www.telecom.ece.ntua.gr/cello.

[5] European Telecommunications Standards Institute, “ETSI 03.71: Location Services (LCS)”.

[6] Third Generation Partnership Project, “3GPP TS 23.271: Functional stage 2 description of LCS”.

[7] D. Kothris, M. Beach, B. Allen, P. Karlsson; “Performance Assessment of Terrestrial and Satellite Based Position Location Systems”, *Proc. of IEE International Conference on 3G Mobile Communications Technology*, March 2001.

[8] D. Porcino; “Performance of a OTDOA-IPDL positioning receiver for 3G-FDD mode”, *Proceedings of IEE Second International Conference on Mobile Communication Technologies*, March 2001.

[9] S. Rooney, P. Chippendale, R. Choony, C. Le Roux, B. Honary, “Accurate Vehicular Positioning using a DAB-GSM Hybrid System”, *Proc. of IEEE VTC-2000*, Volume 1, Spring 2000, pp. 97-101.

[10] M. Spirito, A. Guidarelli; “On the hyperbolic positioning of GSM mobile stations”, *URSI International Symposium on Signals, Systems, and Electronics*, 1998. Issue 98. 1998. pp. 173 – 177.

[11] H. Yin, “Location Based Service”, *T-109.551 Research Seminar on Location Business II*, Helsinki University of Technology, 2002.

[12] Martin-Escalona, F. Barcelo, J. Paradells; “Delivery of Non-Standardized Assistance Data in E-OTD/GNSS Hybrid Location Systems”, *Proceedings of 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, September 2002, pp. 2347 – 2351.

[13] S. Soliman, P. Agashe, I. Fernandez, A. Vayanos, P. Gaal, M. Oljaca; “gpsOne™: A hybrid position location system”, *Proceedings of IEEE 6th International Symposium on Spread Spectrum Techniques and Applications*, volume 1, 2000. pp. 330 – 335.

[14] Emily consortium, *“User and System Requirements Report”*, Deliverable D5 of European Mobile Integrated Location sYstem, http://www.emilypgm.com/publications.asp.

Análisis y Diseño de Políticas de Control de Admisión en Redes Celulares Multiservicio

Vicent Pla Boscà Vicente Casares Giner
Departamento de Communicaciones, E.T.S.I.T.
Universidad Politécnica de Valencia (UPV)
Teléfono: 963879733, Fax: 963877309
(vpla,vcasares)@dcom.upv.es

Abstract *Forthcoming cellular networks will provide multimedia services along with QoS guarantees. In this context admission control is a key aspect in the desing and operation of such systems. In this paper we consider several admission control policies together with appropriate algorithms for their analysis and design. We conclude that the family of randomized stationary policies provide the required versatility to adapt to several goals for both network planning and operation. Appropriate algorithms stem from the fact that randomized stationary policies can be suitably analyzed and adjusted by means of Markov decision processes theory and linear programming techniques.*

1 Introducción

En los últimos años la demanda de servicios de telefonía móvil ha experimentado un enorme crecimiento. Para satisfacer esta demanda creciente, ajustándose a las limitaciones del espectro radioeléctrico, se tiende a una reducción del tamaño de las células, lo que a su vez, implica un mayor impacto de la movilidad de los terminales en las prestaciones de la red, ya que aumenta la frecuencia con la que se producen los traspasos (*handover*). En lo que se refiere a la gestión de recursos, concretamente en la interfaz radio, esto ha llevado a la necesidad de buscar métodos para lograr una *Calidad de Servicio* (QoS) de una forma eficiente, generalmente mediante la priorización adecuada de las peticiones de *handover* frente a las de establecimientos de llamadas nuevas. Sobre esta cuestión puede encontrarse una ingente cantidad de trabajos publicados durante las dos últimas décadas (ver por ejemplo [1, 2, 3, 4]). Por otra parte, el advenimiento de las redes móviles de tercera generación (3G), que se prevé darán servicio a tráfico multimedia, plantea un entorno multiservicio en el que deben coexistir una variedad de servicios con distintos requisitos de QoS y diferentes características de tráfico. En este entorno multiservicio, la complejidad del citado problema de priorización de peticiones o control de admisión aumenta considerablemente.

Como ya se ha comentado, la gestión de recursos radio en redes celulares monoservicio es un tema que ha sido, e incluso continua siendo, profusamente estudiado. Por otra parte, en su momento, la aparición de la RDSI-BA propició el interés por el estudio de las redes fijas multiservicio [5]. Sin embargo, la interacción de estos dos elementos, movilidad y multiservicio, es un materia que no ha recibido la atención de la comunidad investigadora hasta hace relativamente poco. En [6], Li et al. proponen una generalización del conocido mecanismo de prioridad para peticiones de *handover Guard Channel* (GC) [1][1]. En esta propuesta las llamadas de cada tipo dejan de admitirse a partir de un determinado nivel de ocupación del sistema, que puede ser distinto para cada servicio, mientras que la peticiones de *handover* de todos los servicios se admiten siempre que haya suficientes recursos libres. Bartolini y Chlamtac [7] consideran una política de admisión más general que la anterior en la que las peticiones de *handover* (salvo las del servicio más prioritario) también tienen asociado un umbral de ocupación a partir del cual no son admitidas. Recientemente, Heredia et al. [8, 9] plantean una extensión del caso anterior en la que los umbrales pueden ser números no enteros, o lo que es lo mismo, la generalización al caso multiservicio del también conocido *Fractional Guard Channel* (FGC) [10]. Finalmente, en [7] se demuestra que la política de admisión óptima, en cuanto que minimiza una cierta función de coste, no es en general de ninguno de los tipos anteriores, sino que pertenece al grupo más amplio de las políticas *estacionarias* [5].

Naturalmente, cuanto más general sea una política de admisión mayores serán sus posibilidades de cumplir unos determinados objetivos de QoS, pero por otra parte esto implica más grados de libertad, es decir, más parámetros que han de ser ajustados adecuadamente para concretar esa potencialidad mayor. Desde un punto de vista teórico, el análisis de este tipo de sistema y políticas no reviste mayor dificultad, pero desde una vertiente más ingenieril está lejos de ser una cuestión trivial por dos razones: cuando la cantidad de recursos disponibles y/o el número de servicios crecen mínimamente nos encontramos con el problema de la explosión de estados, lo que dificulta su resolución numéri-

[1] Este mismo esquema aparece también bajo una gran variedad de nombres: *Reserve Guard Channel*, *Reserve Margin Algorithm*, *Cutoff Priority*, *Channel Reservation*, ...

ca; estamos ante un problema que no es exactamente de análisis sino de síntesis o diseño, nuestro problema no es evaluar una política concreta con unos parámetros concretos, sino encontrar la política o el valor de los parámetros que satisfagan un cierto QoS, y para esto, es inviable una solución basada en un mero tanteo o búsqueda exhaustiva. Por otra parte, este problema de síntesis ha de resolverse en dos escalas temporales distintas: la de planificación de la red (¿cuál es la cantidad mínima de recursos necesaria para atender el tráfico ofrecido en el caso peor ?) y la de operación (si no estoy en el caso peor, para el tráfico ofrecido en ese momento, ¿cuál es la política de admisión que mejor satisface un determinado objetivo?).

En este artículo se consideran las políticas o familias de políticas siguientes: *acceso total* (o *Complete Sharing*) (CS) [5], *Multiple Guard Channel* (MGC) [7], *Multiple Fractional Guard Channel* (MFGC) [8, 9] y *Radomized Stationary* (RS) [5]. Estas políticas se comparan desde el punto de vista de la capacidad que puede conseguirse con cada una, y se demuestra que la aplicación de la teoría de los *procesos de decisión markovianos* (MDP) [11, 5] junto con las técnicas de la *programación lineal* constituye una herramienta versátil y eficiente para el diseño de las políticas RS, tanto en la fase de planificación como en la de operación.

El resto del artículo está estructurado del siguiente modo: en la sección 2 se describe el modelo del sistema y en la sección 3 las políticas de admisión. En la sección 4 se plantea la problemática que ofrece el diseño de una política de admisión y se analiza el sistema considerando un política del tipo RS. También se proponen métodos operativos de diseño mediante la formulación de distintos problemas de programación lineal. En la sección 5 se proporcionan ejemplos de utilización de estos métodos y se compara numéricamente la capacidad que puede obtenerse utilizando los diferentes grupos de políticas. Finalmente, en la sección 6 se presentan las conclusiones.

2 Descripción del modelo

Nuestro sistema (una célula) dispone de una cantidad total de recursos C y atiende peticiones (llamadas o conexiones) de N tipos distintos de usuarios. Cada uno de estos tipos de usuario tendrá unas características distintas y unos requisitos de calidad de servicio diferentes. Además, para cada tipo de tráfico tenemos que distinguir lo que son las peticiones de establecimiento de conexiones nuevas de las peticiones de establecimiento fruto de un traspaso (*handover*). Por las conocidas razones de tratabilidad matemática del modelo, supondremos que los procesos de llegada son poissonianos y que el tiempo de ocupación de los recursos por una conexión está distribuido exponencialmente. En virtud de estas suposiciones, el sistema tendrá la deseable propiedad de *memoria nula* por lo que su estado estará completamente representado por el vector $\boldsymbol{x} = (x_1, \ldots, x_N)$, donde x_i es el numero de llamadas del tipo i que están siendo cursadas, independientemente de si accedieron al sistema como una llamada nueva o un *handover*. Por su parte, cada tipo de tráfico i ($i = 1, \ldots, N$) estará caracterizado por los siguientes parámetros:

- b_i número de recursos necesarios para cursar la petición.
- $1/\mu_i$ tiempo medio de ocupación de los recursos por la conexión. Nótese que el tiempo de ocupación de los recursos no tiene porqué coincidir con la duración de la conexión ya que esta última comprende la utilización de recursos en una o varias células.
- λ_i^n tasa de llegadas de peticiones correspondientes a llamadas nuevas.
- λ_i^h tasa de llegadas de peticiones originadas por un *handover*.
- p_i^n probabilidad de que una llamada nueva no sea admitida.
- p_i^h probabilidad de que una petición de *handover* no sea admitida.

3 Políticas de Control de Admisión

En está sección se describe el funcionamiento de las políticas de control de admisión consideradas. En general, todas ellas se basan en el estado de ocupación del sistema y el tipo de petición para decidir la aceptación o no de esta última. El tipo de petición está determinado por el tipo de servicio solicitado y la distinción dentro de cada servicio entre llamadas nuevas y llamadas de *handover*. Así, el número total de tipos de petición será $2N$. Obsérvese que no se considera la historia pasada del sistema, cosa que, por otra parte, es lógica si tenemos en cuenta la *memoria nula* de nuestro modelo.

Las políticas o familias de políticas consideradas son, de menos a más general:

CS Complete Sharing
: Se acepta cualquier petición siempre que haya suficientes recursos disponibles.

MGC Multiple Guard Channel
: A cada tipo de petición se le asigna un umbral t, que es un número entero comprendido entre 1 y C. Tendremos por tanto un total de $2N$ umbrales $(t_1^n, \ldots, t_N^n, t_1^h, \ldots, t_N^h)$. Cuando llega una petición se compara el número de recursos utilizados ($b(\boldsymbol{x}) = \sum_{i=1}^{N} x_i b_i$) con el umbral correspondiente y, en caso de que sea estrictamente menor, se acepta la petición; siempre que haya suficientes recursos libres ($b(\boldsymbol{x}) + b_i \leq C$; i es el

servicio al que pertenece la petición). La política anterior (CS) puede verse como un caso particular de esta en el que $t_i^n = t_i^h = C - b_i + 1$ (i=1,...,N) .

MFGC Multiple Fractional Guard Channel
Se trata de una generalización de la política anterior en la que los umbrales se permite que sean números fraccionarios ($0 < t_i^{n,h} \leq C$). Si t es el umbral asociado a una petición cualquiera y $b(\boldsymbol{x})$ el número de recursos utilizados, el criterio seguido para decidir sobre la admisión sería:

- si $b(\boldsymbol{x}) < \lfloor t \rfloor$, aceptar.
- si $b(\boldsymbol{x}) = \lfloor t \rfloor$, aceptar con probabilidad $t - \lfloor t \rfloor$.
- si $b(\boldsymbol{x}) > \lfloor t \rfloor$, rechazar.

Lógicamente, la aceptación está en todos los casos condicionada a que haya suficientes recursos libres.

RS Randomized Stationary
Una política estacionaria aleatorizada es aquella en la que la decisión depende únicamente del estado actual del sistema y de un componente aleatorio [11]. En nuestro caso esto se podría formalizar del siguiente modo: a cada estado del sistema $\boldsymbol{x}$ se le asocian dos N-tuplas $\boldsymbol{\alpha}^n(\boldsymbol{x}), \boldsymbol{\alpha}^h(\boldsymbol{x}) \in [0,1]^N$. Una petición de una llamada nueva (*handover*) que llega al sistema en estado $\boldsymbol{x}$, se aceptará con probabilidad $\alpha_i^n(\boldsymbol{x})$ ($\alpha_i^h(\boldsymbol{x})$). Nótese que las grupos de políticas anteriores están incluidos dentro de éste, pero existen políticas RS que no son de ninguno de los tipos anteriores.

4 Análisis y diseño

Para el análisis de las políticas CS, MGC y MFGC el enfoque aplicado es el tradicional. Fijados los parámetros del sistema y de la política de admisión (valor de los umbrales $t_i^{n,h}$), se plantean las ecuaciones de balance globales del proceso de Markov de donde se obtienen el valor para los parámetros de QoS. En el caso particular de la política CS el proceso de Markov es reversible y, por tanto, las probabilidades de estado tienen forma de producto [5], lo que permite utilizar el *algoritmo de convolución* [12] para calcular éstas de forma más eficiente. En los otros dos casos, en los que no se da esta condición, el sistema de ecuaciones se ha resuelto empleando un método iterativo (*Gauss-Seidel*) que aprovecha el caracter disperso del sistema de ecuaciones.

En cualquier caso, con independencia del método que se utilice para resolver el sistema de ecuaciones, lo que se tiene es una función que devuelve el valor de los parámetros de QoS ($p_i^{n,h}$) en función del valor de los parámetros de la política de admisión ($t_i^{n,h}$), de las características del tráfico ($\lambda_i^{n,h}, \mu_i, b_i$) y de los recursos del sistema (C); ver diagrama de la figura 1. Esto es

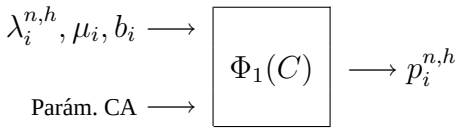


Figura 1: Análisis

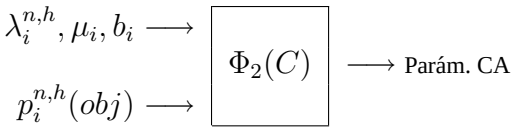


Figura 2: Diseño

justamente lo que se necesita para analizar un sistema con una política de admisión concreta. Sin embargo, en el diseño o ajuste de una política de admisión el problema es el inverso: dados los parámetros del trafico y dados unos valores (o valores límite) para los parámetros de QoS, obtener el valor adecuado para los parámetros de control de admisión; ver diagrama de la figura 2.

El método de análisis anteriormente descrito permite evaluar Φ_1 pero no Φ_2. Por otra parte, se podría obtener numéricamente Φ_2 a partir de Φ_1 mediante una cierta "busqueda inversa". Sin embargo, el carácter multidimensional hace que sea un problema difícil en sí mismo, además de costoso computacionalmente. El coste computacional de evaluar Φ_1 puede llegar a ser muy elevado y el uso de métodos iterativos para realizar esta búsqueda lo aumentaría todavía más. Encontrar algoritmos que realicen la búsqueda es complejo pues si bien p_i^J ($J = n, h$) decrece cuando el umbral correspondiente t_i^J aumenta, el comportamiento con el valor del resto de los umbrales —al contrario de lo que en un principio podría parecer intuitivo— no siempre es monótono. En la figura 3 se representa un ejemplo de esto último.

Debido a la existencia de estos inconvenientes a continuación se considera otra familia de políticas (RS). Al tratarse de una familia de políticas que contiene a las anteriores, teóricamente pueden obtenerse mejores

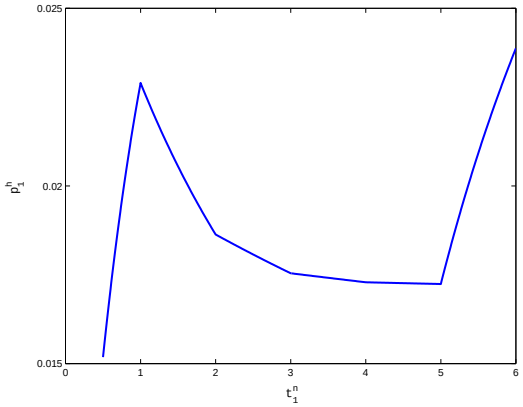


Figura 3: Comportamiento no monótono de p_1^h con t_1^n; $\lambda_1^n = 5,\ \lambda_2^n = 1,\ \lambda_1^h = 0.01,\ \lambda_2^n = 0;\quad \mu_1 = 5,\ \mu_2 = 15;\quad b_1 = 1,\ b_2 = 5;\quad t_1^h = t_2^{n,h} = C = 6.$

prestaciones. Además, y tal vez más importante, está el hecho de que al considerar esta clase más amplia de políticas es posible aplicar unas herramientas que permiten resolver el problema de diseño de una forma más adecuada; se dispone de algoritmos que se ajustan al paradigma de la figura 2.

4.1 Políticas RS

En el análisis de las políticas RS vamos a utilizar el marco teórico de los *procesos de decisión markovianos* (MDP) [11, 5] junto con técnicas de *programación lineal* para la resolución de los problemas de optimización asociados. Con el fin de hacer autocontenido este artículo, a continuación se hace un resumen de la notación y de los conceptos del ámbito de los MDP que van a emplearse.

El espacio de estados es

$$S := \left\{ \boldsymbol{x} : \sum_{i=1}^{N} x_i b_i \leq C;\ x_i \in \mathbb{N} \right\} \qquad (1)$$

Cada estado tiene asociado un conjunto de acciones posibles $A(\boldsymbol{x}) \subseteq A$; donde A es el conjunto de todas la acciones posibles,

$$A := \{\boldsymbol{a} = (a_1, \ldots, a_N) :\ a_i = 0, 1, 2\} \qquad (2)$$

El elemento a_i de una acción $\boldsymbol{a}$ representa el tratamiento que se les da a las peticiones del servicio i; los valores de a_i tienen el significado siguiente: $a_i = 0$, no se admiten llamadas nuevas ni peticiones de *handover*; $a_i = 1$, no se admiten llamadas nuevas pero sí peticiones de *handover*; $a_i = 2$, se admiten llamadas nuevas y peticiones de *handover*. Se supone que para un mismo servicio las peticiones de *handover* siempre tendrán mayor prioridad que las llamadas nuevas por lo que no se ha considerado la opción de admitir llamadas nuevas y no admitir peticiones de *handover*. El número total de acciones en A será 3^N, aunque el numero de acciones posibles en un estado puede ser menor al no haber recursos disponibles; por ejemplo, para un estado en el que todos los recursos están ocupados ($b(\boldsymbol{x}) = C$) el conjunto de acciones posibles en ese estado sólo tiene un elemento, $A(\boldsymbol{x}) = \{(0, \ldots, 0)\}$. En una política RS cada vez que el proceso visita el estado $\boldsymbol{x}$ se elige una de las acciones posibles $A(\boldsymbol{x})$ de manera aleatoria según la distribución de probabilidad $p_x(\boldsymbol{a}),\ \boldsymbol{a} \in A(\boldsymbol{x})$. Las probabilidades asociadas a las acciones y las probabilidades de admisión de cada tipo de petición se relacionan del siguiente modo:

$$\alpha_i^n(\boldsymbol{x}) = \sum_{a_i=2} p_x(\boldsymbol{a}), \quad \alpha_i^h(\boldsymbol{x}) = \sum_{a_i=1,2} p_x(\boldsymbol{a}) \qquad (3)$$

En cada estado ($\boldsymbol{x}$), la elección de una acción ($\boldsymbol{a}$) determina la tasas de transición a los otros estados, $r_{xy}(\boldsymbol{a})$. En nuestro caso tenemos dos tipos de transición, llegadas y salidas del sistema. Las expresiones para sus correspondientes tasas de transición son: para las llegadas ($y = x + \boldsymbol{e}_i$)

$$r_{xy}(\boldsymbol{a}) = \begin{cases} 0 & \text{si } a_i = 0 \\ \lambda_i^h & \text{si } a_i = 1 \\ \lambda_i^n + \lambda_i^h & \text{si } a_i = 2 \end{cases} \qquad (4)$$

y para las salidas ($y = x - \boldsymbol{e}_i,\ x_i > 0$)

$$r_{xy}(\boldsymbol{a}) = x_i \mu_i \qquad (5)$$

Donde i denota el tipo de servicio al que pertenece la llegada o salida y $\boldsymbol{e}_i$ es un vector en el que todas sus entradas son 0 salvo la i-ésima que vale 1.

Para nuestro propósito necesitamos transformar el proceso de Markov en una cadena de Markov equivalente (*uniformizar* el proceso [13]). Esto es posible pues se puede encontrar una cota superior (Γ) para la tasa total de salida de cada uno de los estados,

$$\sum_{y \in S} r_{xy}(\boldsymbol{a}) < \Gamma \quad \forall\, \boldsymbol{x} \in S,\ \boldsymbol{a} \in A(\boldsymbol{x}) \qquad (6)$$

donde

$$\Gamma = \sum_{i=1}^{N} (\lambda_i^n + \lambda_i^h + C\mu_i) \qquad (7)$$

La probabilidades de transición de la cadena de Markov resultante son,

$$p_{xy}(\boldsymbol{a}) = \frac{r_{xy}(\boldsymbol{a})}{\Gamma} \quad \text{si } y \neq x \qquad (8)$$

donde $r_{xy}(\boldsymbol{a})$ esta definido por las expresiones (4) y (5) según se trate de una llegada o una salida, respectivamente. Además, fruto de la transformación de *uniformización* aparecen autolazos (transiciones de un estado a sí mismo), cuya probabilidad es

$$p_{xx}(\boldsymbol{a}) = 1 - \sum_{y \in S} p_{xy}(\boldsymbol{a}) \qquad (9)$$

Se definen además las siguientes funciones de coste[2]

$$c_i^n(\boldsymbol{x}, \boldsymbol{a}) = \begin{cases} 1 & \text{si} \quad a_i = 0, 1 \\ 0 & \text{si} \quad a_i = 2 \end{cases} \qquad (10)$$

$$c_i^h(\boldsymbol{x}, \boldsymbol{a}) = \begin{cases} 1 & \text{si} \quad a_i = 0 \\ 0 & \text{si} \quad a_i = 1, 2 \end{cases} \qquad (11)$$

de modo que el promedio temporal de cada uno de los costes coincide con la probabilidad de bloqueo correspondiente, es decir,

$$p_i^{n,h} = \lim_{k \to \infty} \frac{E\left[\sum_{t=0}^{k} c_i^{n,h}\left(\boldsymbol{x}(t), \boldsymbol{a}(t)\right)\right]}{k+1} \qquad (12)$$

donde $(\boldsymbol{x}(t), \boldsymbol{a}(t))$ representa el estado y la acción en el instante t.

Si $p(\boldsymbol{x})$ representa la probabilidad estacionaria del estado $\boldsymbol{x}$, definimos $p(\boldsymbol{x}, \boldsymbol{a}) = p(\boldsymbol{x}) p_x(\boldsymbol{a})$ por lo que se cumple que

$$p(\boldsymbol{x}) = \sum_{\boldsymbol{a} \in A(\boldsymbol{x})} p(\boldsymbol{x}, \boldsymbol{a}). \qquad (13)$$

[2] Para las funciones de coste se ha respetado la notación que se utiliza para el caso general en el que las funciones de coste dependen del estado y la acción, aunque en nuestro caso únicamente dependen de la acción.

Conjuntos de restricciones A continuación se definen distintos conjuntos de restricciones que posteriormente se utilizan para plantear diferentes criterios de diseño.

R0

$$\sum_{\boldsymbol{a}\in A(\boldsymbol{x})} p(\boldsymbol{x},\boldsymbol{a}) = \sum_{\substack{\boldsymbol{y}\in S\\ \boldsymbol{a}\in A(\boldsymbol{y})}} p_{yx}(\boldsymbol{a})p(\boldsymbol{y},\boldsymbol{a}), \qquad \boldsymbol{x}\in S$$

$$\sum_{\substack{\boldsymbol{x}\in S\\ \boldsymbol{a}\in A(\boldsymbol{x})}} p(\boldsymbol{x},\boldsymbol{a}) = 1$$

$$p(\boldsymbol{x},\boldsymbol{a}) \geq 0, \qquad \boldsymbol{x}\in S,\ \boldsymbol{a}\in A(\boldsymbol{x})$$

Las restricciones de **R0** provienen de las ecuaciones correspondientes a la cadena de Markov asociada al proceso de decisión por lo que estas restricciones serán aplicables en todos los criterios de diseño.

R1 $(J = n, h; \quad i = 1, \ldots, N)$

$$\sum_{\substack{\boldsymbol{x}\in S\\ \boldsymbol{a}\in A(\boldsymbol{x})}} p(\boldsymbol{x},\boldsymbol{a})c_i^n(\boldsymbol{x},\boldsymbol{a}) \leq p_i^J(max)$$

En **R1** se han introducido los parámetros de diseño $p_i^{n,h}(max)$ que corresponden a los valores máximos para las probabilidades de bloqueo (QoS mínima).

R2 $(J = n, h; \quad i = 1, \ldots, N)$

$$\sum_{\substack{\boldsymbol{x}\in S\\ \boldsymbol{a}\in A(\boldsymbol{x})}} p(\boldsymbol{x},\boldsymbol{a})c_i^J(\boldsymbol{x},\boldsymbol{a}) \leq \frac{1}{J_i}z$$

En **R2** se han introducido los pesos n_i, h_i que se utilizarán para ponderar las probabilidades de bloqueo de los distintos tipos de petición. También se ha introducido la variable auxiliar z, que junto con los pesos, se utilizará para aplicar un criterio de equidad del tipo *minimax* ponderado.

$$z = \max\left\{n_i p_i^n,\ h_i p_i^h; \quad i = 1, \ldots, N\right\} \qquad (14)$$

Criterios de Diseño (CD) Los criterios de diseño que se contemplan están formados por una función objetivo a minimizar más uno o varios de los conjuntos de restricciones anteriores. Como tanto la funciones objetivo como las restricciones son lineales, el problema de diseño se convierte en un problema de programación lineal.

CD1

- Minimizar:

$$\sum_{\substack{i=1\\ \boldsymbol{x}\in S\\ \boldsymbol{a}\in A(\boldsymbol{x})}}^{N} p(\boldsymbol{x},\boldsymbol{a})\big(n_i c_i^n(\boldsymbol{x},\boldsymbol{a}) + h_i c_i^h(\boldsymbol{x},\boldsymbol{a})\big)$$

- Sujeto a: **R0**

Este criterio de diseño es el utilizado en [7] y encuentra la política optima en tanto que minimiza el valor de

$$\sum_{i=1}^{N}\left(n_i p_i^n + h_i p_i^h\right)$$

Este criterio tiene la particularidad de que la solución es siempre una política estacionaria pura, es decir no aleatorizada: en cada estado siempre se elige la misma acción ($\forall \boldsymbol{x} \in S,\ \exists!\ \boldsymbol{a} \in A(\boldsymbol{x}) : p(\boldsymbol{x},\boldsymbol{a}) > 0$) Esto podría suponer una ventaja para la implementación. Sin embargo, este criterio tiene el inconveniente de que al hacer optimización global puede ocurrir que sacrifique excesivamente las probabilidades de bloqueo con menor peso dando lugar a problemas de equidad entre distintos servicios. Para solucionar este problema se introducen los criterios siguientes.

CD2

- Minimizar:

$$\sum_{\substack{i=1\\ \boldsymbol{x}\in S\\ \boldsymbol{a}\in A(\boldsymbol{x})}}^{N} p(\boldsymbol{x},\boldsymbol{a})\big(n_i c_i^n(\boldsymbol{x},\boldsymbol{a}) + h_i c_i^h(\boldsymbol{x},\boldsymbol{a})\big)$$

- Sujeto a: **R0, R1**

De este modo se limita el valor máximo que puede alcanzar la probabilidad de bloqueo de cada tipo de petición. En este caso la solución ya no es una política estacionaria pura, aunque puede demostrarse [14] que, si $n_a(\boldsymbol{x})$ es el número de acciones entre las que se elige en el estado $\boldsymbol{x}$, se verifica que

$$\sum_{\boldsymbol{x}\in S}\big(n_a(\boldsymbol{x}) - 1\big) \leq 2N$$

Desde un punto de vista práctico es importante señalar que el problema asociado a **CD2** puede no tener solución si C no es lo suficientemente alto. Encontrar el valor mínimo de C para que el problema tenga solución, o su dual, el valor máximo del trafico ofrecido para que el problema tenga solución con un valor dado de C, son problemas propios de la fase de planificación o dimensionado de la red en los que puede aplicarse el **CD2**.

Durante la operación de la red, en una situación en la que el sistema está en congestión el problema de **CD2** no tiene solución puesto que no es posible cumplir las restricciones **R1**. No obstante, en este caso podría ser importante que el deterioro de QoS se repartiese de forma equitativa entre los distintos servicios. Para este propósito se plantea el siguiente criterio.

CD3

- Minimizar: z

	Configuración				
	A	B	C	D	E
b_1	1	1	1	1	1
b_2	2	4	2	2	2
f_1	0.8	0.8	0.2	0.8	0.8
f_2	0.2	0.2	0.8	0.2	0.2
$p_1^n(max)$ (%)	5	5	5	1	1
$p_2^n(max)$ (%)	1	1	1	2	1
	A,B,C,D,E				
$p_i^h(max)$	$0.1p_i^n(max)$				
λ_i^n	$f_i\lambda$				
λ_i^h	$0.5\lambda_i^n$				
μ_1	1				
μ_2	3				

Tabla 1: Parámetros de las configuraciones

Conf.	C	CS	MGC	MFGC	RS
A	10	1.54	1.88	2.05	2.07
	20	5.61	7.07	7.35	7.38
	40	15.7	19.4	19.7	19.8
B	10	0.36	0.40	0.42	0.44
	20	2.77	3.35	3.46	3.48
	40	10.3	12.5	12.7	12.8
C	10	1.36	1.51	1.65	1.67
	20	5.77	6.91	6.98	7.00
	40	17.6	20.1	20.4	20.5
D	10	1.74	1.97	2.02	2.04
	20	6.04	6.82	6.93	6.94
	40	16.5	18.2	18.4	18.4
E	10	1.5	1.7	1.8	1.8
	20	5.6	6.3	6.4	6.5
	40	15	17	17	17

Tabla 2: Capacidad (λ_{max})

- Sujeto a: **R0, R2**

Este criterio también puede aplicarse en el supuesto contrario al anterior: cuando durante la operación de la red el tráfico ofrecido está por debajo del que el sistema puede cursar manteniendo los requisitos de QoS de **R1**, puede interesar repartir el margen de mejora de QoS de forma equitativa entre los servicios.

Por último, como se verá en la sección siguiente mediante un ejemplo, **CD4** permite definir dos modos de operación distintos dentro de la no congestión (carga normal y carga alta).

CD4

- Minimizar: z
- Sujeto a: **R0, R1, R2**

5 Ejemplos de Aplicación y Resultados Numéricos

En esta sección se consideran varios casos de estudio a los que se les aplica las técnicas y criterios que se han descrito anteriormente.

$\boldsymbol{x}$	$b(\boldsymbol{x})$	$\boldsymbol{\alpha}^n(\boldsymbol{x})$	$\boldsymbol{\alpha}^h(\boldsymbol{x})$	$\boldsymbol{a}$	$p_x(\boldsymbol{a})$
$(*,*)$	$0,\ldots,5$	$(1,1)$	$(1,1)$	$(2,2)$	1
$(6,0)$	6	$(0,1)$	$(1,1)$	$(1,2)$	1
$(4,1)$	6	$(0.47,1)$	$(1,1)$	$(1,2)$	0.25
				$(2,2)$	0.75
$(2,2)$	6	$(1,1)$	$(1,1)$	$(2,2)$	1
$(0,3)$	6	$(1,1)$	$(1,1)$	$(2,2)$	1
$(7,0)$	7	$(0,0)$	$(1,1)$	$(1,1)$	1
$(5,1)$	7	$(0,0.97)$	$(1,1)$	$(1,1)$	0.17
				$(1,2)$	0.83
$(3,2)$	7	$(0,1)$	$(1,1)$	$(1,2)$	1
$(1,3)$	7	$(1,1)$	$(1,1)$	$(2,2)$	1
$(8,0)$	8	$(0,0)$	$(0,1)$	$(0,1)$	1
$(6,1)$	8	$(0,0)$	$(0,1)$	$(0,1)$	1
$(4,2)$	8	$(0,0)$	$(1,1)$	$(1,1)$	1
$(2,3)$	8	$(0,1)$	$(0.86,1)$	$(1,1)$	0.14
				$(1,2)$	0.86
$(0,4)$	8	$(0,1)$	$(1,1)$	$(1,2)$	1
$(7,1)$	9	$(0,0)$	$(1,0)$	$(1,0)$	1
$(5,2)$	9	$(0,0)$	$(1,0)$	$(1,0)$	1
$(3,3)$	9	$(0,0)$	$(1,0)$	$(1,0)$	1
$(1,4)$	9	$(0,0)$	$(1,0)$	$(1,0)$	1
$(8,1)$	10	$(0,0)$	$(0,0)$	$(0,0)$	1
$(6,2)$	10	$(0,0)$	$(0,0)$	$(0,0)$	1
$(4,3)$	10	$(0,0)$	$(0,0)$	$(0,0)$	1
$(2,4)$	10	$(0,0)$	$(0,0)$	$(0,0)$	1
$(0,5)$	10	$(0,0)$	$(0,0)$	$(0,0)$	1

Tabla 3: Ejemplo de política RS

En primer lugar se evalua la capacidad del sistema para las distintas políticas de admisión. Definimos la capacidad del sistema como el valor máximo de la tasa total de llegadas de llamadas nuevas ($\lambda = \sum_{i=1}^{N} \lambda_i^n$) para el que se cumplen los requisitos de QoS (valores máximos para $p_i^{n,h}$). La capacidad se ha evaluado para las distintas configuraciones recogidas en la tabla 1. Los valores de la capacidad se muestran en la tabla 2. Como es lógico, con una disciplina más general se obtiene mayor capacidad. Por otra parte, la ganancia relativa que se obtiene es menor cuando la cantidad de recursos es más alta ($C = 40$). En particular, RS y $MFGC$ tienden a igualarse cuando C aumenta, mientras que para MGC se observa la misma tendencia aunque de forma más lenta. La diferencia entre cualquiera de estas tres políticas y CS también se reduce, aunque aquí el efecto es bastante menor y en todos los casos se mantiene una diferencia superior al 10 %.

A modo de ejemplo, el ajuste de la diferentes políticas para el caso $C = 10$ de la configuración A sería el siguiente: MGC $t_1^n = 7$, $t_2^n = 7$, $t_1^h = 10$, $t_2^h = 10$; MFGC $t_1^n = 6.8963$, $t_2^n = 6.8963$, $t_1^h = 8.3206$, $t_2^h = 10$; RS ver tabla 3. Para ajustar la política RS se ha aplicado el **CD2**. Como puede verse la política resultante no es estacionaria pura aunque sólo está aleatorizada en 3 estados. Como ya se ha comentado, si se quiere obtener una política estacionaria pura se puede aplicar el **CD1**. Si se aplica el **CD1** utilizando como pesos el valor inverso de la máxima probabilidad de bloqueo correspondiente, es decir, $n_i = 1/p_i^n(max)$ y $h_i = 1/p_i^h(max)$, se obtiene una capacidad de 1.81. Esta capacidad es un 4 % menor que con MGC, un

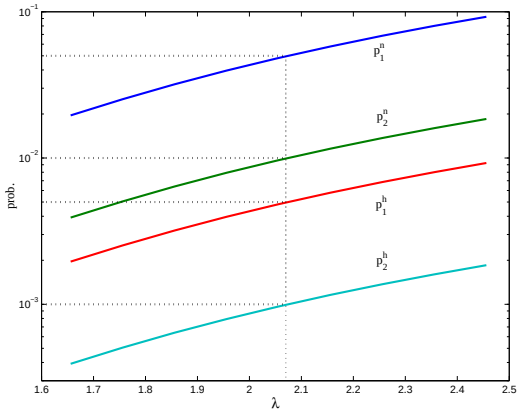


Figura 4: RS

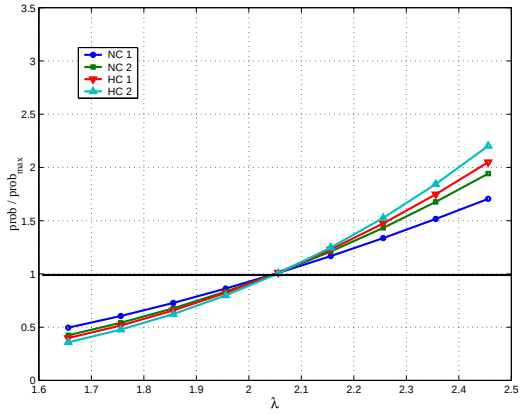


Figura 6: MFGC

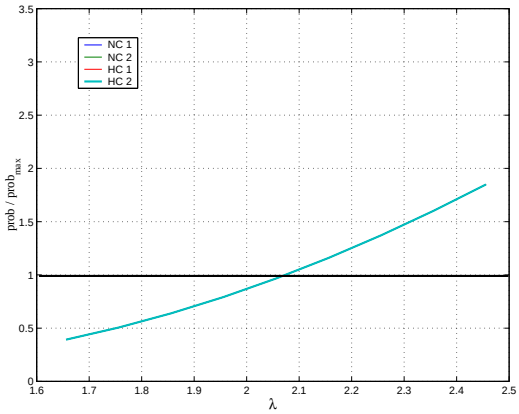


Figura 5: RS

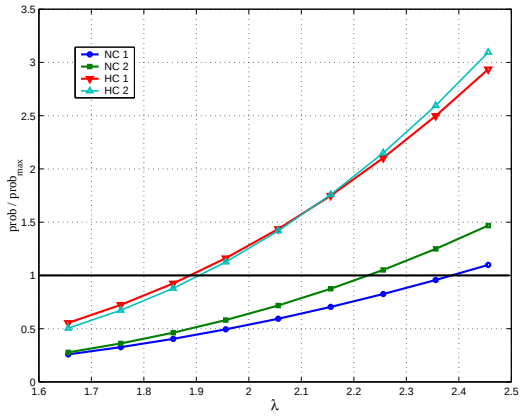


Figura 7: MGC

12 % que con MFGC y un 13 % que con RS. Para mejorar este valor se han ajustado los pesos de la función objetivo empleando un algoritmo heurístico e iterativo. Como resultado se obtiene una capacidad de 2.03, para los siguientes valores de los pesos $n_1 = 0.9697$, $n_2 = 2.265$, $h_1 = 3.991$,$h_2 = 53.355$. Por tanto, manteniéndose dentro de la familia de las políticas estacionarias puras se consigue una capacidad que es un 1 % y un 2 % menor que la que se obtiene con MFGC y RS, respectivamente, y es un 8 % superior a la capacidad que se obtiene con MGC.

En los ejemplos anteriores se han comparado las diferentes familias de políticas desde el punto de vista de la capacidad. Esto es útil en la fase de planificación de la red cuando se necesita conocer el tráfico que puede cursar con una cierta cantidad de recursos cumpliendo unos requisitos de QoS. Para este tipo de cálculos el operador debe considerar el caso peor para el tráfico ofrecido, es decir, una previsión del tráfico durante la hora cargada. Sin embargo, si la previsión es adecuada, la mayor parte del tiempo el tráfico ofrecido estará por debajo del valor previsto para la hora cargada y, aunque con menor frecuencia, podría ocurrir también que se superase este valor. En el primero de los casos habrá una cierta holgura en el cumplimiento de los requisitos de QoS y en el segundo se violará uno o varios de estos requisitos. Cabría plantearse por tanto, si es posible repartir la mejora o el empeoramiento de manera equitativa entre los distintos servicios. La respuesta es afirmativa si se utiliza una política RS. Para lograr esto se necesita reajustar los parámetros de la política en tiempo de operación a partir de los valores del tráfico estimados mediante medida, y aplicar el algoritmo que se deriva del **CD3** utilizando $n_i = 1/p_i^n(max)$ y $h_i = 1/p_i^h(max)$. Para las otras políticas no conocemos la existencia de ningún algoritmo que permita hacer esto. Como ejemplo ilustrativo se ha utilizado el caso $C = 10$ de la configuración A. Se ha tomado el valor máximo de la capacidad (2.07) y se ha variado la tasa total ofrecida (λ) desde un 10 % por debajo de la capacidad a un 10 % por encima. En la figura 4 se representa la evolución de las distintas probabilidades de bloqueo cuando se emplea una política RS y se realiza el reajuste de la misma aplicando el **CD3**. En esta gráfica puede observarse como la distancia entre las diferentes curvas se mantiene constante. Esto mismo puede observarse con mayor exactitud en la figura 5, donde se representa el cociente entre cada probabilidad de bloqueo y su valor máximo; como consecuencia del reparto equitativo las cuatro curvas se solapan en una única. En la figura 6 y la figura 7 se muestra el mismo tipo de representación pero para las políticas MFGC y MGC, en las que no es posible el reajuste. En estos casos se observa que el aumento o el deterioro de la calidad no se distribuye equitativamente.

Por último, se presenta un ejemplo en que se utili-

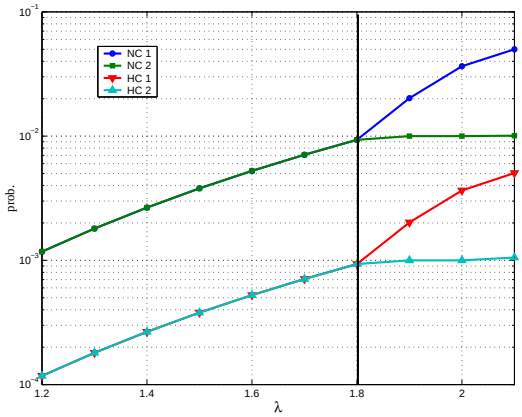


Figura 8: Doble modo de operación: carga normal y carga alta

za el **CD4** para ajustar en tiempo de operación un sistema en el que hay dos modos de funcionamiento dependiendo de la carga, a saber, carga normal y carga alta. En el modo de carga normal el objetivo de QoS es $p^n_{1,2} \leq 0.01$ y $p^h_{1,2} \leq 0.001$, mientras que en el modo de carga alta se permite un deterioro de la QoS del servicio 2 ($p^n_1 \leq 0.05, p^h_1 \leq 0.005$) manteniendo la misma exigencia para el servicio 1 ($p^n_2 \leq 0.01, p^h_2 \leq 0.001$). De nuevo se ha utilizado el caso $C = 10$ de la configuración A. Para ajustar la política se ha aplicado el **CD4** con los parámetros siguientes: $p^n_1(max) = 0.05$, $p^n_2(max) = 0.01$, $p^h_1(max) = 0.005$, $p^h_2(max) = 0.001$, $n_1 = n_2 = 1$, $h_1 = h_2 = 10$. En la figura 8 se representa el resultado.

6 Conclusiones

En este trabajo se han considerado distintas políticas para el control de admisión en redes celulares multiservicio. Entre estas políticas se incluyen la generalización a un entorno multiservicio de las más populares del ámbito monoservicio (GC y FGC) y la familia más general de la políticas estacionarias aleatorizadas (RS). En general, se concluye que las políticas del tipo RS son preferibles al resto, pues, por una parte permiten conseguir una mayor capacidad para una misma cantidad de recursos — especialmente cuando éstos son reducidos— y, por la otra, el ajuste de los parámetros de este tipo de políticas puede formularse de una forma más adecuada para el diseño. En el artículo se demuestra que esto último se consigue mediante la aplicación de la teoría de los procesos de decisión markovianos junto con técnicas de programación lineal. Dentro de este marco proponemos diferentes métodos para el diseño de políticas de admisión del tipo RS que son útiles no sólo para el dimensionado de la red sino también para la fase de operación.

Agradecimientos

El presente trabajo ha sido financiado por el *Ministerio de Ciencia y Tecnología* a través de los proyectos TIC2000-1041-C03-02 y TIC2001-0956-C04-04.

Referencias

[1] D. Hong and S. S. Rappaport, "Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and nonprioritized handoff procedures," *IEEE Transactions on Vehicular Technology*, vol. VT-35, pp. 77–92, Aug. 1986. See also: CEAS Technical Report No. 773, June 1, 1999, College of Engineering and Applied Sciences, State University of New York, Stony Brook, NY 11794, USA.

[2] Y.-B. Lin, S. Mohan, and A. Noerpel, "Queueing priority channel assignment strategies for PCS hand-off and initial access," *IEEE Transactions on Vehicular Technology*, vol. 43, pp. 704–712, Aug. 1994.

[3] V. Pla and V. Casares, "Delay-loss analysis of channel assignment schemes in mobile cellular with handoff priority and hysteresis control," in *Proceedings of 14th ITC Specialist Seminar on Access Networks and Systems*, (Girona, Spain), ITC, 2001.

[4] F. Barceló, "Comparison of handoff resource allocation strategies through the state dependent rejection scheme," in *Proceedings of ITC 17*, 2001.

[5] K. W. Ross, *Multiservice Loss Models for Broadband Telecommunication Networks*. Springer Verlag;, 1995.

[6] B. Li, C. Lin, and S. T. Chanson, "Analysis of a hybrid cutoff priority scheme for multiple classes of traffic in multimedia wireless networks," *ACM/Baltzer Wireless Networks Journal*, vol. 4, no. 4, pp. 279–290, 1998.

[7] N. Bartolini and I. Chlamtac, "Call admission control in wireless multimedia networks," in *Proceedings of IEEE PIMRC*, 2002.

[8] H. Heredia-Ureta, F. A. Cruz-Pérez, and L. Ortigoza-Guerrero, "Multiple fractional channel reservation for optimum system capcity in multi-service cellular networks," *Electronic Letters*, vol. 39, pp. 133–134, Jan. 2003.

[9] H. Heredia-Ureta, F. A. Cruz-Pérez, and L. Ortigoza-Guerrero, "Multiple fractional channel reservation for multiservice cellular networks," in *Proceedings of IEEE ICC*, 2003.

[10] R. Ramjee, R. Nagarajan, and D. Towsley, "On optimal call admission control in cellular networks," *ACM/Baltzer Wireless Networks Journal*, vol. 3, no. 1, pp. 29–41, 1997.

[11] S. M. Ross, *Applied probability models with optimization applications*. Holden-Day, 1970.

[12] V. B. Iversen, *Teletraffic Engineering Handbook*. ITU-D SG 2 and ITC, 2002. URL: http://www.tele.dtu.dk/teletraffic/.

[13] R. W. Wolff, *Stochastic Modeling and the Theory of Queues*. Englewood Cliffs, NJ: Prentice Hall, 1989.

[14] K. W. Ross, "Randomized and past-dependent policies for markov decision processes with multiple constraints," *Operations Research*, vol. 37, no. 3, pp. 474–477, 1989.

Sesión 6B

Aplicaciones cooperativas y plataformas WEB

Combinación de Evidencias para la Recuperación de Información en el Web: Análisis del Ámbito de las Consultas

Fidel Cacheda, Angel Viña
Departamento de Tecnoloxías da Información e as Comunicacións.
Universidad de A Coruña, Campus de Elvira, Facultad de Informática, s/n
15.071 A Coruña
Teléfono: 981 167000 Fax: 981 167160
E-mail: {fidel, avc}@udc.es

Abstract. *Recent research on the hyperlink structure of the Web has improved the overall precision of the top relevant documents for broad and generic queries, although it usually obtains inferior results when the query is precise and specific. Therefore, an optimal combination of content analysis and link analysis (named combination of evidence) based on the query scope is needed to increase precision for both types of queries. This paper analyses several measures of the query scope based on the number of documents in a query and in the collection. Several sets of queries have been defined (with different scopes) and execute over different search engines evaluating the measures defined. The results show that the number of documents retrieved versus the number of documents used in the query provides an appropriate measure of the query scope and independent from the search engine used.*

1 Introducción

Con el inicio del World Wide Web en la década de los noventa, surgieron en paralelo los primeros sistemas de Recuperación de Información (IR) específicamente diseñados para este nuevo entorno. Debido al crecimiento exponencial sufrido por el Web desde su origen estos sistemas de búsqueda pronto se convirtieron en un componente fundamental del WWW.

Estos sistemas de búsqueda (divididos en motores de búsqueda, directorios Web y metabuscadores) surgen para adaptarse a las características especiales del Web (gran volumen de información, dinamismo, heterogeneidad y distribución, entre otros), y con el propósito de gestionar, recuperar y filtrar la información disponible para sus usuarios.

Inicialmente en los sistemas de búsqueda en el Web se realizó una adaptación de las técnicas de recuperación de información tradicionales al entorno del Web. Este modelo tradicional está basado en el análisis del contenido para determinar si un documento es relevante ante una consulta determinada [14].

Más recientemente se han empezado a tener en cuenta las características especiales del Web para la recuperación de información. Especialmente interesantes son las técnicas basadas en el análisis de la estructura de hiperenlaces. Estas técnicas utilizan la información de enlaces entre documentos en combinación con el análisis de contenidos para detectar documentos de alta calidad, denominados documentos autoritarios para una consulta [6].

Sin embargo, diferentes tipos de consultas requieren diferentes combinaciones entre las técnicas de análisis de contenidos y del análisis de enlaces. La experiencia en el Web sugiere que consultas sobre temas muy específicos o sobre temas no muy difundidos en el Web, no se beneficiarán de las técnicas del análisis de enlaces debido a que las páginas relevantes no serán populares (al estar dedicadas a una audiencia muy especializada) y por lo tanto estarán poco conectadas. Por el contrario, ante búsquedas muy genéricas el análisis de enlaces mejora la precisión de los resultados, frente al análisis de contenidos [6].

Por lo tanto se plantea el problema de la combinación óptima de las técnicas de análisis de contenidos y de análisis de enlaces para una consulta determinada (denominado *combinación de evidencias*). Intuitivamente, la contribución del análisis de enlaces deberá ser mayor para aquellas consultas más genéricas y menor para las consultas más específicas. Para poder determinar dinámicamente la combinación óptima ante una consulta se requiere un método para estimar el *ámbito de la consulta*, esto es, una medida de cuán genérica o específica es una búsqueda.

El estudio realizado en este artículo se basa en la teoría de que una consulta genérica es aquella que recupera un gran número de documentos y una consulta específica es aquella que obtiene un número reducido de resultados. Por lo tanto, el problema consiste en determinar cuándo el volumen de documentos recuperados es elevado o reducido, teniendo en cuenta que esto no depende necesariamente del tamaño de la colección sino también del conocimiento sobre la consulta incluido en la colección de documentos.

Para ello se analizan diversas medidas del ámbito de una consulta basadas en el número de documentos recuperados por la consulta, utilizados en la consulta y disponibles en toda la colección. Se utilizan 7 conjuntos de consultas de diferentes niveles de generalidad (evaluados empíricamente) y 3 sistemas de recuperación de información en el Web, midiendo para cada caso los valores de los parámetros definidos. Los resultados obtenidos identifican como medida relevante el número de documentos recuperados frente al número de documentos utilizados en la consulta, al obtenerse valores homogéneos para los sistemas de búsqueda examinados.

Este artículo se estructura de la siguiente manera. La siguiente sección detalla los objetivos y a continuación se hace una breve exposición del estado del arte. En la sección 4 se describe el análisis realizado y a continuación se presentan los resultados obtenidos. Finalmente se presentan las conclusiones y futuros trabajos.

2 Objetivos

El principal objetivo en este trabajo consiste en establecer una medida que permita determinar el nivel de generalidad o especificidad de una determinada consulta.

Se deberá establecer un modelo teórico para el análisis del ámbito de las consultas. En este modelo teórico se deben incluir diferentes tipos de consultas que permitan analizar el comportamiento de diferentes parámetros ante consultas con diferentes niveles de generalidad o especificidad.

Para los posibles parámetros analizados se deberá comprobar su comportamiento sobre diferentes motores de búsqueda (constituyendo idealmente colecciones de diferentes tamaños), examinando los resultados en función de:

- El tipo de búsqueda, sobre la base del modelo teórico y para diferentes niveles de generalidad y especificidad.
- El motor de búsqueda empleado.
- La interacción entre ambos factores.

Los parámetros válidos deberán de proporcionar una medida adecuada del ámbito de la consulta y de manera independiente del tipo de motor de búsqueda o colección utilizada. De esta manera se garantiza la validez de la medida para todo tipo de colecciones, independientemente del número de documentos indexados.

3 Estado del arte

La primera y más importante característica de un buen sistema de recuperación de información es que debe proporcionar al usuario resultados precisos y un número suficiente que coincidan con la consulta realizada.

Las técnicas de recuperación de información tradicionales se aplicaron principalmente sobre colecciones de documentos planos relativamente estáticas (p.e. entornos de gestión bibliotecaria), y por lo tanto están basadas exclusivamente en técnicas de análisis del contenido.

La IR clásica está dividida en tres modelos básicos: booleano, vectorial y probabilístico, sobre los cuales se han definido diferentes variantes [1]. Estos modelos clásicos se basan en un modelo matemático formal para la recuperación, en donde los documentos están formados por conjuntos de términos que pueden ser individualmente ponderados y manipulados. Las consultas son ejecutadas comparando la representación de la consulta frente a la representación del documento en el espacio, pudiendo recuperar documentos que no contengan necesariamente alguno de los términos de búsqueda.

El modelo booleano consiste en la utilización de la teoría de conjuntos y el álgebra Booleana. En este modelo cada término indexado es ponderado como presente o no presente y todas las consultas se basan en expresiones booleanas [12]. Es el más simple de todos los modelos, si bien únicamente clasifica a los documentos como relevantes o no relevantes (sin una ordenación de los resultados).

El modelo vectorial es el más utilizado en los sistemas de IR modernos. En este caso, los documentos, términos y consultas se representan mediante vectores en un espacio multidimensional [11]. Las consultas se resuelven mediante la comparación del vector de consulta frente a los vectores de los documentos, obteniendo una lista ponderada de resultados. La similitud entre un documento y una consulta se calculan en base la función coseno del ángulo entre sus dos vectores. Normalmente, cada término es ponderado según el esquema tf-idf (*term frequency-inverse document frequency*): directamente proporcional a su frecuencia en el documento e inversamente a la frecuencia del término en la colección de documentos.

En el modelo probabilístico en cada consulta se mide para cada documento la probabilidad de que sea relevante para dicha consulta, obteniéndose un primer conjunto de documentos potencialmente relevantes [10]. A continuación el usuario interactuará con el sistema para indicar aquellos documentos que considera relevantes. El sistema usa esta información para refinar los resultados de la búsqueda,

repitiéndose este proceso hasta una adecuada aproximación al conjunto de resultados óptimos.

Inicialmente, las técnicas de recuperación de información en el Web se basaron en los modelos clásicos de análisis de contenidos (especialmente, en el modelo vectorial). Sin embargo, recientemente se han desarrollado nuevas aproximaciones que combinan el análisis de contenidos con el análisis de enlaces entre páginas Web.

Kleinberg en [6] propone un algoritmo, para una consulta concreta, capaz de localizar documentos autoritarios y documentos catálogos en el Web. La entrada del algoritmo es un conjunto de documentos recuperados utilizando técnicas tradicionales de análisis de contenidos. Sobre este algoritmo Bharat et al. en [2] y Chakrabarti et al. en [4], entre otros, han propuesto diversas modificaciones y aplicaciones de esta técnica.

Brin y Page en [3] y [8] describen la arquitectura de Google[1], un motor de búsqueda en donde cada página indexada es ponderada en base al PageRank, midiendo su popularidad en el Web. Para cada consulta se obtienen un conjunto de documentos relevantes que son ordenados en base a su puntuación en el análisis de contenidos y de enlaces.

En ambos casos, el análisis de enlaces complementa al análisis de contenidos mejorando la precisión entre los primeros documentos recuperados en la consulta.

Sin embargo, como se ha comentado previamente, diferentes tipos de consultas requieren diferentes combinaciones entre el análisis de contenidos y el análisis de enlaces. En este sentido, el siguiente paso dentro de la recuperación de información en el Web consiste en la combinación dinámica de ambos factores, en función del ámbito de la consulta (generalidad o especificidad de la consulta).

Existen pocos trabajos a este respecto y uno de los más interesantes es el propuesto por Plachouras y Ounis en [9] donde definen la medida del ámbito de una consulta en base a una estructura jerárquica de conceptos proporcionada por WordNet [7], para combinar el análisis de contenidos y enlaces. Sin embargo, los resultados obtenidos no son totalmente satisfactorios respecto a las mejoras en la precisión de los resultados.

4 Análisis del ámbito de las consultas

El análisis realizado ha sido dividido en dos partes: en primer lugar, se han definido 7 conjuntos de consultas diferentes que abarcan diferentes niveles de generalidad y especificidad. Y en segundo lugar, estas consultas han sido ejecutadas sobre 3 motores de búsqueda (o colecciones) diferentes examinando diferentes parámetros de la búsqueda que permitan medir o estimar el ámbito de la consulta.

La definición de los conjuntos de consultas está basada en diferentes fuentes de información públicas que proporcionan consultas reales o para propósitos de investigación. Los conjuntos de consultas, detallados a continuación, son los siguientes: Silverstein, Lycos 50, Zeitgeist, TREC 8 Títulos, TREC 8 Descripciones, TREC 9 Títulos, TREC 9 Descripciones.

Silverstein et al. en [13] y Kirsch en [5] investigan diferentes aspectos de las consultas realizadas por los usuarios sobre motores de búsqueda en Internet, proporcionando un listado de las búsquedas más frecuentes. El conjunto denominado *Silverstein* ha sido definido en base al top 25 de Silverstein y al top 15 de Kirsch, eliminando elementos duplicados o ininteligibles, obteniéndose un total de 28 consultas.

El servicio The Lycos 50 Daily Report[2] proporciona las 50 búsquedas más frecuentes realizadas en Lycos[3] semanalmente. Tomando los datos correspondientes a la semana del 1 al 8 de Febrero de 2003 se ha constituido el conjunto denominado *Lycos 50*.

El servicio Google Zeitgeist[4] (similar al anterior) proporciona las 10 búsquedas más frecuentes realizadas en Google durante el período de una semana. El conjunto denominado *Zeitgeist* está formado por las 50 búsquedas más frecuentes recopiladas entre los días 18 de Febrero al 13 de Enero de 2003. Es importante destacar que, tanto en este caso como en el anterior, las listas son creadas en base a una selección personal, frente al primer caso que ha sido directamente calculado a partir de las búsquedas archivadas. Esto puede afectar al nivel de generalidad obtenido en cada caso.

Para la evaluación de sistemas de IR la serie de conferencias TREC (Text REtrieval Conference) constituye un punto de referencia a nivel internacional. Por este motivo, los restantes conjuntos de consultas se han obtenido a partir de diferentes tópicos utilizados en las conferencias TREC-8 y TREC-9, en sesiones especialmente dedicadas a la recuperación de información en el Web.

El objetivo de la sesión del Web en TREC-8 fue proporcionar un medio para evaluar nuevas técnicas en el entorno del WWW. Para ello, se dispuso una colección de 2 Gigabytes formada por aproximadamente 250.000 documentos, utilizando las consultas 401-450 [15]. Los títulos de estos tópicos constituyen el conjunto de consultas denominado

[1] http://www.google.com/

[2] http://50.lycos.com/

[3] http://www.lycos.com/

[4] http://www.google.com/press/zeitgeist/archive.html

TREC 8 Títulos, mientras que las descripciones forman el conjunto denominado *TREC 8 Descripciones*.

El propósito de la sesión del Web en TREC-9 consistía en elaborar una mayor colección de documentos para la evaluación [16]. En este caso los tópicos 451-500 de esta sesión, fueron diseñados específicamente para el entorno Web. Al igual que en el caso anterior, los títulos de los tópicos constituyen el conjunto de consultas denominado *TREC 9 Títulos* y las descripciones el conjunto *TREC 9 Descripciones*.

Tanto en las consultas de TREC-8 como en las de TREC-9, el hecho de utilizar las descripciones proporciona a las consultas una mayor especificidad que en el caso de utilizar únicamente los títulos. En el resto de conjuntos, se asume que el modelo Silverstein está constituido por búsquedas más genéricas que los modelos Lycos y Zeitgeist al ser directamente búsquedas reales.

Para analizar el grado de generalidad y especificidad de los conjuntos de consultas creados se ha definido un cuestionario seleccionando 10 consultas al azar de cada uno de los conjuntos. Este cuestionario fue presentado a 11 voluntarios, con experiencia en la navegación por el Web, que puntuaron según el grado de generalidad cada bloque de consultas entre 1 (muy específico) y 5 (muy genérico).

A partir de la *Fig. 1* se observa como los conjuntos *Silverstein* (S), *Lycos 50* (L) y *Zeitgeist* (Z) presentan un mayor nivel de generalidad, destacando el primero como era previsible. A continuación, los conjuntos *TREC-8 Títulos* (T8T) y *TREC-9 Títulos* (T9T), y ya con un nivel de especificidad significativamente elevado los conjuntos *TREC-8 Descripciones* (T8D) y *TREC-9 Descripciones* (T9D).

El objetivo en la segunda fase del análisis es medir una serie de parámetros para cada una de las búsquedas definidas. En concreto, los parámetros a

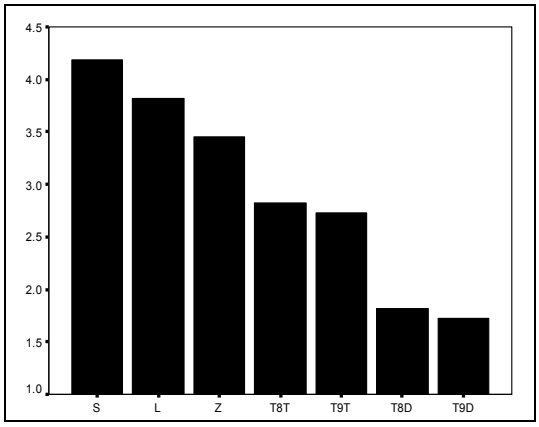


Fig. 1: Nivel de generalidad de los conjuntos de consultas.

determinar son los siguientes:

- *n*, número de resultados obtenidos para la consulta: usando el operador AND en caso de que el buscador no lo emplee por defecto.
- *Q*, número de resultados utilizados en la consulta: obtenido a partir de la consulta original utilizando el operador OR.
- *N*, número de documentos incluidos en toda la colección: este valor es constante para cada colección utilizada.

El análisis del ámbito de las consultas se ha realizado sobre tres motores de búsqueda en Internet que proporcionan información sobre el número de documentos indexados. Los sistemas de búsqueda utilizados han sido:

- AlltheWeb[5]: $N = 2.112.188.990$ documentos.
- Gigablast[6]: $N = 155.069.088$ documentos.
- Google: $N = 3.083.324.652$ documentos.

Idealmente estos tres motores de búsqueda se pueden considerar como tres colecciones de documentos diferentes. Sin embargo, es necesario puntualizar que existen algunas diferencias entre el funcionamiento de sus motores de búsqueda respectivos que pueden afectar a los resultados obtenidos.

En concreto, es relevante determinar la manera en la que los valores de *n* y *Q* son estimados o calculados en cada caso.

En el caso de AlltheWeb por defecto se realiza una búsqueda usando el operador AND calculando exactamente el número de resultados obtenidos. En el caso de las búsquedas disyuntivas también se calcula de manera precisa el número de resultados obtenidos.

Gigablast por el contrario realiza por defecto una búsqueda disyuntiva, mostrando en primer lugar los resultados de una búsqueda basada en el operador AND. Las búsquedas usando el operador AND han sido forzadas, sin embargo se ha observado que en múltiples casos los resultados eran completados con documentos que no contenían todas las palabras buscadas. Esto ha sido detectado y corregido en múltiples búsquedas, sin embargo puede provocar una sobreestimación del valor de *n*. El número de resultados obtenidos se calcula de manera exacta, si bien entre 10 y 500 únicamente se indica que se han obtenido menos de 500 resultados. En este caso se ha tomado el valor 500 como referencia.

[5] http://www.alltheweb.com/
[6] http://www.gigablast.com/

El caso de Google se realiza por defecto una búsqueda utilizando el operador AND. El número de resultados es estimado, con buenas aproximaciones en el caso de búsquedas conjuntivas, mientras que se produce una clara subestimación en el caso de las búsquedas disyuntivas.

Estas diferencias no son especialmente críticas, sin embargo sirven para poner de relieve que en el estudio se pueden plantear diferencias entre los motores de búsqueda, más allá de las derivadas del hecho de trabajar con colecciones de diferentes tamaños.

El estudio realizado se basa en la ejecución de las consultas incluidas en los siete conjuntos definidos (un total de 328 consultas) sobre los tres motores de búsqueda detallados. Cada consulta ha sido ejecutada usando el operador AND y el operador OR, midiendo el número de resultados obtenidos en cada caso, o lo que es lo mismo, midiendo los valores de *n* y *Q*, para cada consulta.

5 Resultados

En esta sección se describen los resultados obtenidos en base al análisis de los parámetros definidos en la sección anterior.

En primer lugar, es importante destacar que no es posible operar con valores absolutos de *n* o *Q* ya que las diferencias entre las colecciones utilizadas afectan directamente a los valores obtenidos en las diferentes consultas.

En la *Fig. 2* se muestran los valores medios de *n* obtenidos para cada motor de búsqueda en cada una de los bloques de consultas definidos. Se utiliza una escala logarítmica para comprimir las grandes diferencias existentes entre los valores de las búsquedas más genéricas y más específicas. Sin embargo, el aspecto más relevante es el relacionado con la gran variación en el número de documentos

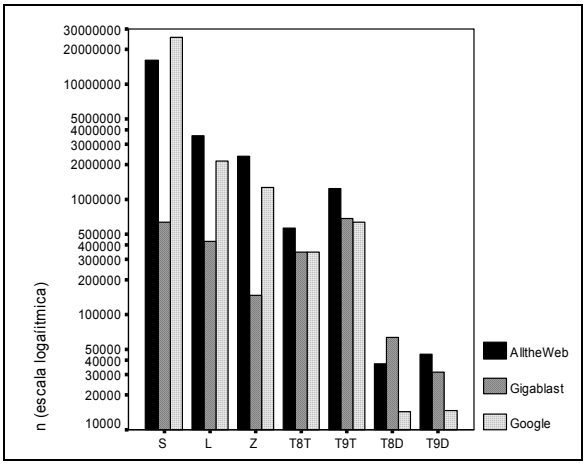


Fig. 2: Valores medios de n para los diferentes motores de búsqueda, en función del tipo de búsqueda.

recuperados (para un mismo conjunto de consultas) entre los diferentes motores de búsqueda. Esto es especialmente patente para las búsquedas que obtienen más resultados (S, L y Z), en donde aquellas colecciones con mayor número de documentos obtienen valores más elevados para *n*. De hecho, para el conjunto *Silverstein*, Google recuperó de media más de 25 millones de documentos, AlltheWeb cerca de 16 millones, mientras que Gigablast únicamente 625 mil documentos.

Por este motivo, para evitar las diferencias existentes entre las diferentes colecciones de documentos utilizadas se han definido las siguientes medidas relativas para estimar el ámbito de las consultas:

- *n/Q*: número de documentos recuperados frente al número de documentos utilizados en la consulta.

- *n/N*: número de documentos recuperados frente al número de documentos en toda la colección.

- *Q/N*: número de documentos utilizados en la consulta frente al número de documentos en toda la colección.

El primero de los parámetros ofrece una medida del porcentaje de documentos recuperados respecto al número de documentos relacionados con esa consulta en la colección (sin importar el volumen total de la colección). Este parámetro asume que la consulta es genérica cuando se están recuperando un gran número de documentos respecto al conocimiento de la consulta incluido en la colección.

El segundo parámetro mide el porcentaje de documentos recuperados respecto al conjunto de documentos en la colección. Este parámetro estima que una consulta es genérica cuando se recuperan un gran número de documentos de la colección.

El tercer parámetro mide el conocimiento incluido en la colección sobre una consulta. A priori, este parámetro no se considera un buen estimador del ámbito de una consulta, aunque podría ser utilizado en combinación con alguno de los anteriores para refinar los resultados obtenidos.

Para cada uno de estos parámetros se debe medir su respuesta en función del tipo de búsqueda realizado y del motor de búsqueda empleado. Obviamente, el valor obtenido deberá variar en función del tipo de búsqueda realizada proporcionando una medida del ámbito de la consulta. E idealmente, el comportamiento del parámetro será independiente del tipo de motor de búsqueda utilizado.

Para cada uno de los parámetros definidos se ha utilizado el test ANOVA para medir la relación de cada parámetro con: el tipo de búsqueda realizada y el motor de búsqueda empleado.

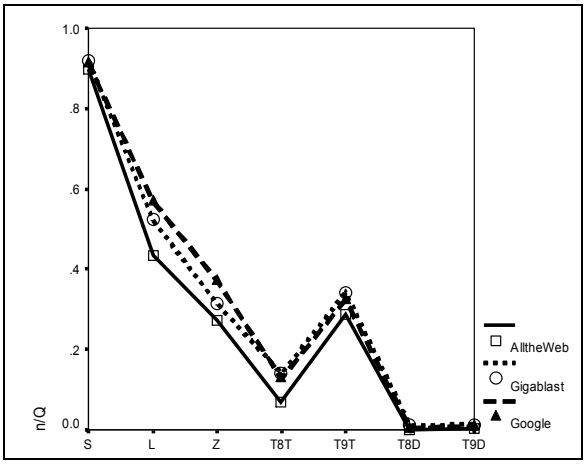


Fig. 3: Comparación de los valores medios de n/Q para los diferentes motores de búsqueda en función del tipo de búsqueda (test ANOVA).

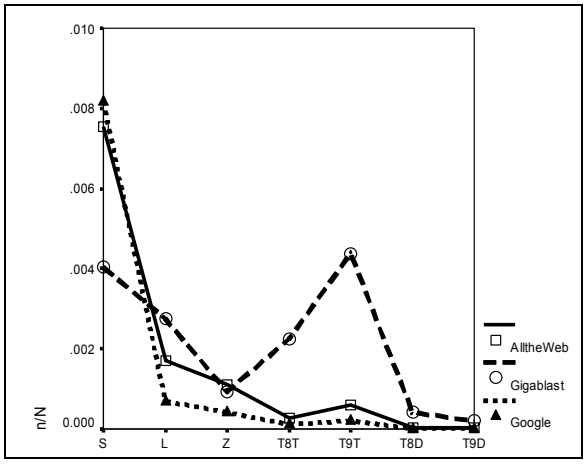


Fig. 4: Comparación de los valores medios de n/N para los diferentes motores de búsqueda en función del tipo de búsqueda (test ANOVA).

El test ANOVA realizado sobre el parámetro *n/Q* refleja un comportamiento bastante homogéneo respecto a los diferentes motores de búsqueda. En la *Fig. 3* se muestran los valores medios en función del tipo de búsqueda para cada motor de búsqueda.

Los resultados del test indican, como era previsible, que el tipo de búsqueda es un parámetro influyente (con un 99.9% de probabilidad). Sin embargo, la repercusión del tipo de motor de búsqueda es diferente. En este caso (como se observa en la *Fig. 3*), los valores que toma el parámetro para cada tipo de búsqueda son muy similares para los tres sistemas de búsqueda analizados, y no existen comportamientos diferentes entre los sistemas para tipos concretos de búsquedas. Esto es patente en los valores obtenidos por el test ANOVA que no permiten aceptar que el tipo de buscador es un parámetro influyente (con un p-valor de 0.098) y rechazando la interacción entre los dos factores estudiados (con una probabilidad del 97.7%).

En el test realizado sobre el parámetro *n/N* se observa como el comportamiento depende en gran medida del tipo de búsquedas realizadas, pero sin embargo, existen grandes diferencias entre los diferentes motores de búsqueda utilizados (ver *Fig. 4*).

Al igual que en el caso anterior, la importancia del tipo de búsqueda es notoria en el test ANOVA al aceptar la influencia de este parámetro en los valores de *n/N* con un una probabilidad del 99.9%. Sin embargo, aunque el tipo de motor de búsqueda no se pueda considerar un factor influyente (con un p-valor de 0.272), sí que existe una clara interacción entre ambos factores (p-valor del 0.025). Esto es, para algún motor de búsqueda ante algún tipo concreto de consulta el comportamiento es diferente del resto de los casos. Este aspecto es fácilmente identificable en la *Fig. 4*, especialmente para las búsquedas S y T9T y el motor de búsqueda Gigablast.

En el test realizado sobre el parámetro *Q/N* respecto a los factores de tipo de búsqueda y motor de búsqueda los resultados obtenidos indican una clara diferencia entre los tres motores de búsqueda (ver *Fig. 5*).

Obviamente, la influencia del tipo de búsqueda sigue estando reflejada en los resultados del test ANOVA (con una probabilidad del 99.9%). Sin embargo, la principal diferencia respecto a los dos parámetros anteriores radica en que en este caso, el motor de búsqueda también es un parámetro influyente y además existe interacción entre los dos factores (en ambos casos, con un 99.9% de probabilidad).

Esto implica que los valores obtenidos en el parámetro *Q/N* para un tipo concreto de búsqueda dependen en gran medida del motor de búsqueda empleado.

Conviene destacar el hecho de que en el análisis de los parámetros *n/Q* y *n/N* se acepta el tipo de motor de búsqueda como un parámetro no influyente, aunque con unas probabilidades no demasiado contundentes. Esto se deriva de los diferentes comportamientos existentes en los tres buscadores a la hora de calcular o estimar los valores obtenidos para *n* y *Q*, tal y como se comentaron en la sección anterior. Sin embargo y a pesar de estas diferencias, el comportamiento de ambos parámetros (especialmente el primero de ellos) es muy similar entre los diferentes sistemas de búsqueda analizados.

A modo de resumen, y en base a los resultados obtenidos a partir de los test ANOVA sobre los parámetros analizados, se observa que el mejor comportamiento se produce en el caso del parámetro *n/Q*. El comportamiento de este parámetro depende únicamente del tipo de búsqueda realizado, obteniéndose resultados similares para los diferentes motores de búsqueda.

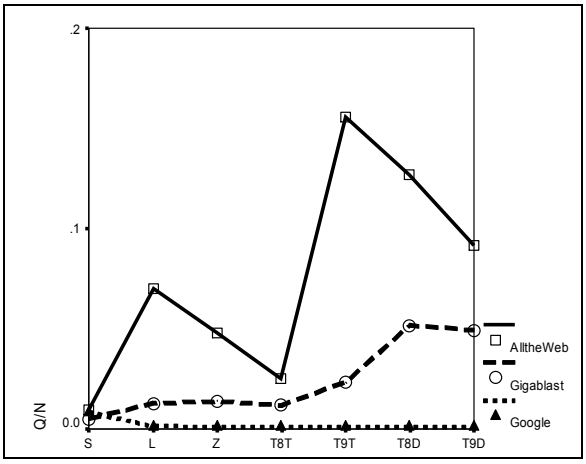


Fig. 5: Comparación de los valores medios de Q/N para los diferentes motores de búsqueda en función del tipo de búsqueda (test ANOVA).

En cambio, en el caso de los otros dos parámetros, los resultados obtenidos no son independientes del motor de búsqueda empleado, especialmente en el caso del parámetro *Q/N*. El caso del parámetro *n/N* no es tan crítico al observarse un modelo de comportamiento similar para los diferentes buscadores, sin embargo la interacción con el tipo de búsqueda en algunos casos reduce la utilidad de este parámetro.

Respecto al ajuste sobre el modelo teórico propuesto en base a los conjuntos de categorías definidos, también el parámetro *n/Q* ofrece un mejor resultado. Si bien no es posible un análisis detallado en base al modelo teórico al estar basado en una clasificación muy básica, si se observa como los valores mayores del parámetro se asocian con aquellas consultas más genéricas, y viceversa. Esto permite un correcto ajuste a los niveles de generalidad y especificidad definidos en el modelo teórico previo.

En el caso del parámetro *n/N* el ajuste es razonablemente adecuado, con valores mayores para las consultas más genéricas (normalmente, inferiores al 1% del número de documentos en la colección). Sin embargo, se observan diferencias significativas entre los diferentes sistemas de búsqueda.

Para el parámetro *Q/N* los resultados analizados son demasiado variables respecto al tipo de motor de búsqueda para ser concluyentes.

6 Conclusiones

En este artículo se centra en el estudio del ámbito de las consultas para una óptima combinación de las técnicas de análisis de contenidos y de análisis de enlaces en la recuperación de información en el Web, en función del tipo de búsqueda realizado.

En concreto, se examinan estos tres parámetros: el porcentaje de documentos recuperados frente a los documentos utilizados en la consulta (*n/Q*), el porcentaje de documentos recuperados frente a los documentos en la colección (*n/N*) y el porcentaje de documentos utilizados en la consulta frente a los documentos en la colección (*Q/N*).

Se han definido siete conjuntos de búsquedas representativos de diferentes niveles de generalidad-especificidad de las consultas, que han sido ejecutados sobre tres sistemas de búsqueda en el Web (constituyendo idealmente tres colecciones de documentos diferentes), midiendo en cada caso los valores obtenidos para los diferentes parámetros a analizar.

Los resultados demuestran que el parámetro *n/Q* representa una medida apropiada del ámbito de una consultas, con un comportamiento independiente del tipo de sistema de búsqueda empleado (a pesar de las diferencias reales existentes entre los sistemas utilizados).

En cambio, los parámetros *n/N* y *Q/N* ofrecen un comportamiento más variable en función del tipo de sistema de búsqueda empleado (especialmente en el caso del segundo), sin ofrecer una medida adecuada respecto al ámbito de las consultas realizadas.

Este trabajo constituye el primer paso en el estudio de la combinación de evidencias en la recuperación de información en el Web. Los trabajos futuros estarán orientados hacia la comparación de la calidad de los resultados obtenidos en un sistema de búsqueda basado en la combinación de evidencias, frente a los modelos tradicionales de análisis de contenidos y de enlaces.

En concreto, mediante los parámetros analizados se podrá estimar el nivel de generalidad-especificidad de cada consulta (especialmente mediante el parámetro *n/Q*, aunque también se consideran fusiones con alguno de los otros parámetros), y combinar de manera dinámica las valoraciones de los documentos obtenidas del análisis de contenidos y del análisis de enlaces. Aquellas consultas más genéricas darán un mayor peso a la componente del análisis de enlaces, mientras que las consultas más específicas harán más hincapié en el análisis de contenidos.

La calidad de los resultados obtenidos (midiendo la calidad en base a la precisión relativa) deberá ser comparada con un sistema de recuperación de información basado exclusivamente en el análisis de contenidos, en el análisis de enlaces y en la combinación estática (independiente del tipo de consulta) de ambos. Este análisis determinará el nivel de mejora obtenido (en caso de obtenerse alguno) por el modelo propuesto para los diferentes tipos de consultas, frente a los sistemas de búsquedas utilizados hasta el momento.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el proyecto CICYT (TIC2001-0547) y en el caso del primer autor también por la Fundación Caixa Galicia (Beca curso 2002/2003 para Estudios de Postgrado en Universidades y en Centros de Investigación de Excelencia Académica)

Los autores de este artículo agradecen al profesor Iadh Ounis del grupo de Information Retrieval de la Universidad de Glasgow la motivación para realizar este trabajo y sus importantes comentarios.

Referencias

[1] R. Baeza-Yates, B. Ribeiro-Neto. "*Modern Information Retrieval*". Addison Wesley, ISBN 0-201-39829-X (1999).

[2] K. Bharat, M. R. Henzinger. "*Improved Algorithms for Topic Distillation in a Hyperlinked Environment*". Research and Development in Information Retrieval, pp: 104-111, 1998.

[3] S. Brin, L. Page. "*The Anatomy of a Large-Scale Hypertextual Web Search Engine*". Computer Networks and ISDN Systems, 30 (1-7): 107-117, 1998.

[4] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, S. Rajagopalan. "*Automatic Resource List Compilation by Analyzing Hyperlink Structure and Associated Text*". 7th International World Wide Web Conference, 1998.

[5] S. Kirsch, "*Infoseek's Experiences Searching the Internet*". ACM SIGIR Forum, vol. 32, no. 2, pp. 3-7, 1998.

[6] J. M. Kleinberg. "*Authoritative Sources in a Hyperlinked Environment*". Journal of the ACM, 46(5): 604-632, 1999.

[7] G. Miller. "*WordNet: A Lexical Database for English*". Communications of the ACM, 38: 39-41, 1995.

[8] L. Page, S. Brin, R. Motwani, T. Winograd. "*The PageRank Citation Ranking: Bringing Order to the Web*". Annual Meeting of the American Society for Information Science, ASIS'98, 1998.

[9] V. Plachouras, I. Ounis. "*Query-Biased Combination of Evidence on the Web*". Mathematical/Formal Methods in Information Retrieval Workshop, ACM SIGIR 2002.

[10] S.E. Robertson, K. Sparck Jones. "*Relevance Weighting of Search Terms*". Journal of the American Society for Information Science, 27:129-146, 1976.

[11] G. Salton, A. Wong, C. S. Yang. "*A Vector Space Model for Automatic Indexing*". Communications of the ACM, 18:613-620, 1975.

[12] G. Salton, M.J. McGill, "*Introduction to Modern Information Retrieval*". McGraw-Hill, New York, ISBN: 0-070-54484-0 (1983).

[13] C. Silverstein, M. Henzinger, H. Marais, M. Moricz. "*Analysis of a Very Large Web Search Engine Query Log*". SIGIR Forum, vol. 33, no. 1, pp. 6-12, 1999.

[14] C. J. Van Rijsbergen. "*Information Retrieval*", 2nd edition. Butterworth-Heinemann. ISBN: 0-408-70929-4 (1979).

[15] E. Voorhees, D. Harman. "*Overview of the Eighth Text Retrieval Conference (TREC-8)*". The Eighth Text REtrieval Conferece (TREC-8), Gaithersburg, Maryland, pp. 1-24, 1999.

[16] E. Voorhees, D. Harman. "*Overview of the Ninth Text Retrieval Conference (TREC-9)*". The Ninth Text REtrieval Conferece (TREC-9), Gaithersburg, Maryland, pp. 1-14, 2000.

Aplicación y Evaluación del Estudio de Casos como Técnica Docente en el Área de Ingeniería Telemática

E. Gómez Sánchez, Y.A. Dimitriadis, J.I. Asensio Pérez
M. Rodríguez Cayetano, M.L. Bote Lorenzo, G. Vega Gorgojo
Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática.
ETSI Telecomunicación, Universidad de Valladolid
Camino del Cementerio, s/n 47011 Valladolid
Teléfono: 983 42 36 66 Fax: 983 42 36 67
E-mail: {edugom,yannis,juaase,manrod,migbot,guiveg}@tel.uva.es

***Abstract**. This paper discusses some experiences on using case studies in courses of Telematics Engineering. In one course a "learn by design" project is carried out, that promotes the development of abilities for searching information, arguing and taking decisions. Moreover, students have stronger motivation and achieve deeper learning of domain concepts, succeeding in exams. After four years applying and evaluating this project, case studies experiences have been proposed for two other courses, with significant differences that required different approaches to the application of case studies. All these experiences suggest a general methodology to define, use and evaluate a case study as an innovative educational technique, which is described in the paper.*

1 Introducción

La docencia en las asignaturas del área de Ingeniería Telemática, como en general en cualquier otra disciplina de ingeniería, está afectada por una aparente contradicción: por una parte tiene un fuerte carácter *aplicado*, es decir, no debe bastar con que el alumno conozca e incluso comprenda perfectamente unos conceptos, sino que se espera que pueda resolver problemas mediante su aplicación; por otra, la *abstracción* es necesaria para no reducir la ingeniería a artesanía [6] y conseguir que el alumno vislumbre soluciones nuevas a problemas que nunca antes se le han planteado.

Con esta visión, la utilización tradicional dada a los créditos prácticos de asignaturas de ingeniería puede presentar serias limitaciones. Las clases de pizarra en las que el profesor presenta soluciones a varios *tipos* de problemas no fomentan la actividad creativa de los alumnos, puesto que se les induce a pensar que el conocimiento de los patrones de solución (las "recetas") es suficiente, y no trabajan relacionando conceptos teóricos abstractos con el problema a resolver. Además, este tipo de problemas suelen estar desconectados del mundo real y restringidos a la disciplina de una asignatura, cuando problemas más realistas ayudarían a conectar en la estructura mental del alumno los conceptos presentados en muchas asignaturas, no sólo tecnológicas. Por otra parte, las prácticas en laboratorio suelen fomentar un mayor grado de creatividad, pero también se restringen habitualmente a escenarios específicos de una asignatura en los que no hay conexión con otros dominios de conocimiento. Por último, existen otras habilidades (manejar documentación dispersa y seleccionar lo importante, criticar soluciones propuestas por terceros, argumentar sobre soluciones a problemas abiertos, hacer y aceptar críticas, llegar a soluciones de compromiso, generar documentos de resumen concisos y correctamente estructurados) que son escasamente fomentadas por la educación tradicional, pese a que su necesidad es reconocida tanto en el mundo académico como profesional.

En este sentido, resulta interesante observar que organismos internacionales como ACM o IEEE publican recomendaciones sobre la organización de programas de ingeniería, en los que esta orientación está presente. Por ejemplo, [10] recomienda que un curso de *Redes y Telecomunicaciones* debe "proporcionar conocimiento de las comunicaciones de datos [...]. El énfasis se dará al análisis y diseño de redes y aplicaciones en red. El análisis coste-beneficio [...] también es evaluado. Los estudiantes deben aprender a evaluar, seleccionar e implementar diferentes opciones de comunicación para una organización". Otra recomendación general, el *Computer Curricula 2001*, plantea que "los alumnos [una vez graduados] deberán presentar capacidades y habilidades que incluyen la capacidad de tener perspectiva de sistema, de abstracción y particularización, adaptabilidad, capacidad de participar en proyectos software no triviales. Para fomentar estas habilidades resulta conveniente plantear a los alumnos casos de estudio que puedan abordar con un cierto grado de libertad [...] para madurar como futuros ingenieros" [1]. Finalmente, también el proceso de convergencia hacia el espacio común europeo promueve una etapa inicial de estudios destinada principalmente a la adquisición de habilidades básicas, de "aprender a aprender", para prepararse para una formación continua a lo largo de la vida.

En realidad, muchas de estas observaciones son válidas en cualquier entorno de aprendizaje, no

estrictamente los de ingeniería. Por ello, se ha sugerido la aplicación de otras técnicas docentes que fomenten estas habilidades, lo que se ha llevado a cabo en distintos niveles educativos y disciplinas [2].

El *estudio de casos* es una de estas técnicas, que enfrenta al alumno con un problema real, que debe comprender, para proponer y discutir una solución. Esta técnica fomenta las capacidades de análisis, aplicación de conocimiento, toma de decisiones, además de incrementar notablemente la motivación del alumno. El estudio de casos suele desarrollarse mediante el trabajo en grupo. El *aprendizaje colaborativo* es un paradigma educativo en el que se pueden situar múltiples técnicas, todas orientadas al aprendizaje del alumno, apoyándose en el hecho de que el trabajo con iguales, dentro del grupo social, permite una mejor selección, comprensión e interiorización de conocimientos [5].

En cualquier caso, la aplicación de estas técnicas tiene notables dificultades: un caso de estudio inadecuado (por ser demasiado complejo o poco versátil) no permitirá el aprendizaje de los alumnos ni fomentará habilidades en ellos, sino que les hará perder tiempo y les causará frustración. Por ello la elección de los casos de estudio es crítica, así como la evaluación del éxito de la experiencia, destinada a corregir aquellos aspectos que impidan o dificulten el aprendizaje, como la dificultad del caso, la falta o exceso de información, el papel del profesor o la carga de trabajo.

En este artículo se discute una metodología para la aplicación del estudio de casos a asignaturas del área de Ingeniería Telemática. En estas asignaturas los alumnos encuentran dificultades para relacionar los conceptos abstractos con la práctica, por lo que un replanteamiento de las actividades prácticas podría ser beneficioso. Además, en la disciplina tienen mucha importancia las habilidades relacionadas con el diseño de soluciones para problemas abiertos, por lo que las distintas variantes del estudio de casos podrían ser adecuadas. Esta metodología se ilustra con tres experiencias en la ETSIT de la Universidad de Valladolid. Uno de los diseños educativos propuestos está relativamente maduro y puede servir como patrón para diseños posteriores pero, como se discutirá con otras experiencias, es necesario contemplar las restricciones específicas de cada asignatura y de sus objetivos de aprendizaje. Se espera que este estudio pueda servir como punto de partida para una discusión más amplia sobre la viabilidad y condiciones de éxito del uso de estas técnicas educativas dentro del área.

Antes de relatar estas experiencias, en la sección 2 se presentan con mayor amplitud las características del estudio de casos, relacionándolo con otras técnicas docentes, y haciendo hincapié en las necesidades docentes en ingeniería. La sección 3 presenta tres experiencias muy distintas. Una de ellas lleva aplicándose y refinándose cuatro años, y ha sido ampliamente evaluada, por lo que puede aportar muchos consejos útiles para el diseño docente. Las otras dos se han aplicado sólo durante un año, pero con diferencias notables entre ellas. Finalmente, la sección 4 recapitula con una visión general y unas reflexiones para el futuro.

2 El estudio de casos como técnica docente

El estudio de casos es una técnica de aprendizaje con la que el alumno se enfrenta a un problema concreto, el *caso*. Para resolver el caso el alumno debe ser capaz de analizar los hechos y los conocimientos referentes a las disciplinas relacionadas, y tomar decisiones razonadas a través de un proceso de discusión, que idealmente debería ser con sus *pares*, es decir, con otros alumnos. Mediante esta técnica el alumno interioriza mejor los conocimientos de la disciplina, al relacionarlos con el mundo real, pese a que por lo general el alumno es expuesto a un menor número de conocimientos que en una clase magistral. Además, el alumno desarrolla o potencia habilidades como la capacidad de organizar la información, de sintetizar, de argumentar o de llegar a consensos. Es además un hecho conocido que la motivación del alumno suele ser mucho mayor [10]. Ahora bien, para que un caso sea útil para el aprendizaje debe seleccionarse con cuidado, intentando que el caso tenga las siguientes características [10]:

- El caso debe plantear una *situación real, pero abordable*. Idealmente, el caso debería ser muy conocido por el profesor, por lo que podría estar tomado de su investigación, siempre que esta tenga relación con la asignatura.

- El caso debe ser *claro y comprensible*, pero no debe sugerir una solución obvia. De hecho, lo ideal es que puedan proponerse varias soluciones válidas, incluso a partir de interpretaciones distintas de los alumnos.

- A pesar de lo anterior, *la información importante y la secundaria deben aparecer mezcladas*, y el alumno debe seleccionar la que considere útil. Esto es una característica de los problemas reales, y sin embargo los enunciados de los problemas académicos frecuentemente listan los datos relevantes, por lo que el alumno no aprende a seleccionar la información útil.

- El caso debe poder *resolverse en un tiempo limitado*, de forma que los alumnos perciban la viabilidad del caso y al mismo tiempo consideren el tiempo como una restricción adicional en la selección de soluciones.

- Por supuesto, el caso debe *poder ser resuelto utilizando conocimientos al alcance del alumno*, típicamente presentados en la asignatura en cuestión, pero también de sus estudios anteriores

en la titulación. De esta forma se garantiza que el alumno aprende los conocimientos y los relaciona entre sí.

Todos estos aspectos permiten intuir que la preparación de un estudio de caso es mucho más compleja para alumnos con conocimientos más reducidos. No obstante, es interesante plantear que el estudio de casos no tiene por qué conllevar necesariamente el diseño de una nueva solución. Las siguientes alternativas son posibles:

- *Comprensión del problema*: Se describe un escenario y se espera que los alumnos sepan identificar los problemas existentes y sus causas. Se busca averiguar "qué va mal y por qué"

- *Diseño de una solución*: Se describe el escenario, indicando el problema y las causas, y se espera que los alumnos sepan proponer soluciones, analizando sus ventajas y desventajas. Se busca encontrar "cuál es la mejor solución y por qué".

- *Análisis de soluciones*: Se describe el escenario, y se muestran varias soluciones predefinidas, y se espera que el alumno encuentre criterios para optar por una de ellas. Se busca encontrar "si las soluciones propuestas resuelven el problema y cómo se ponen en práctica".

Estos tres tipos de estudios de caso requieren distintos conocimientos previos y esfuerzo creativo. En un análisis de soluciones, lo más cercano a un problema académico, el alumno sólo debe elegir entre una de las varias soluciones, pero no proponer creativamente ninguna, ni tampoco averiguar el problema. No obstante, si el caso es real el aprendizaje es mayor que en un ejercicio académico clásico. En el otro extremo, el diseño de soluciones requiere la acción además de la reflexión, por lo que el esfuerzo es mucho mayor, pero también el nivel de interiorización de los conocimientos.

3 Experiencias con estudios de casos

3.1 Metodología propuesta

De lo expuesto hasta ahora parece claro que la selección y desarrollo de un estudio de caso es una labor compleja. Para poder plantear con éxito una actividad de este tipo es necesario llevar una metodología. Tras la aplicación del estudio de casos durante cuatro años en una asignatura, descrita en la siguiente sección, se puede extraer la siguiente metodología, que ha servido para plantear las otras dos experiencias descritas en este artículo:

- *Se comprende el contexto de la asignatura* (madurez de los alumnos, especificidad de los contenidos, carácter práctico, relaciones con otras asignaturas, horas disponibles) para estudiar la viabilidad de la técnica.

- *Se formulan los objetivos de aprendizaje* para el estudio de casos (incluyendo contenidos teóricos así como habilidades de trabajo). Es necesario evaluar si esos objetivos se pueden conseguir mejor con estas técnicas. Por ejemplo, para disciplinas intrínsecamente abstractas no resulta muy adecuado el planteamiento de un estudio de caso.

- *Se elige y elabora el caso.* El caso debe cumplir con los requisitos mencionados anteriormente (real, sin solución obvia, abierto a discusión, con información de diversas fuentes). Después se prepara documentación para los alumnos, que pueden ser textos originales de otras fuentes. Se deben sopesar los problemas que los alumnos puedan encontrar, y pensar soluciones. Además, se debe determinar el tiempo en el que se debe resolver el caso, y la documentación que los alumnos deben generar.

- *Se desarrolla el caso con los alumnos*. Idealmente, los alumnos deben trabajar en grupos pequeños, pero no individualmente. Es muy importante que el profesor de el apoyo suficiente para que los alumnos no se sientan superados por la complejidad del problema, aunque no debería proporcionar soluciones, o expresar preferencias por alguna, porque inhibiría el trabajo creativo de los alumnos y la discusión.

- *Se realiza la evaluación formativa del proceso*, que permite la toma de decisiones de mejora a lo largo del mismo proceso, o para el futuro. La evaluación puede realizarse a partir de muchas fuentes de datos: encuestas individuales o en grupo, entrevistas con voluntarios, observación de la dinámica de la clase por una persona ajena a la asignatura... Por supuesto, tanto recoger como estudiar todos estos datos puede ser muy costoso, y aproximaciones más modestas pueden ser igualmente válidas. En las experiencias de aplicación de estudio de casos de las secciones siguientes se detallan los procedimientos de evaluación utilizados o propuestos.

Esta metodología se ha aplicado a varias asignaturas del área de Ingeniería Telemática, de la que se describen a continuación tres casos significativos. Los motivos para la elección de estas asignaturas son de distinto tipo:

- Por una parte, los profesores que las imparten pueden intercambiar fácilmente sus problemas y experiencias, y relacionarlas con las limitaciones detectadas por una Comisión de Mejora de la Docencia creada en el área (en la ETSIT de Valladolid [11]). Además, muchos de estos profesores pueden aprovechar la experiencia de trabajo en grupos multidisciplinares con profesores de la Facultad de Educación, surgida a

partir de la investigación en Aprendizaje Colaborativo Apoyado por Ordenador (CSCL).

- Por otra parte, las asignaturas tienen características diversas, lo que obligará a adaptar la metodología: hay troncales y optativas, en las que la motivación inicial difiere; están en primer, tercer y cuarto curso, con lo que el nivel de madurez de los alumnos es muy distinto; también varía mucho el número de alumnos (desde 40 hasta 300).

La tabla 1 resume las características más relevantes de las asignaturas, así como las propuestas formuladas para el uso de estudio de casos en ellas, como se detalla en las siguientes secciones.

3.2 Ejemplo 1: Arquitectura de Ordenadores

La asignatura *Arquitectura de Ordenadores* es troncal y en el plan de estudios de Ingeniero de Telecomunicación de la Universidad de Valladolid está en cuarto curso, con 3 créditos teóricos y 6 prácticos. Siguiendo la metodología propuesta en la sección anterior, además de los datos mencionados conviene profundizar en el ***contexto de la asignatura***, para evaluar la viabilidad del estudio de casos para sus créditos prácticos [11]:

- Concluye una rama de asignaturas en las que se han estudiado conceptos básicos de arquitectura de ordenadores, sistemas operativos, y también es posterior a otras en las que se estudian redes de datos, o economía. Así, los alumnos disponen de un abanico de conocimientos que es posible integrar y relacionar con un estudio de casos.

- Los alumnos se acercan al final de la titulación, y esperan actividades más relacionadas con el mundo real.

- La disciplina se presta al diseño y al análisis de alternativas. De hecho, uno de los libros básicos para la asignatura [8] promueve la evaluación cuantitativa y la comparación. La disciplina transmite que no hay solución mejor de forma absoluta, y que hay que buscar compromisos.

- El número de alumnos no es muy elevado, y parcialmente han desarrollado algunas de las habilidades perseguidas.

- El equipo docente lleva impartiendo esta asignatura de manera estable durante varios años.

Con estas condiciones, parece viable plantear un estudio de casos. No obstante, antes hay que formular los ***objetivos de aprendizaje*** para ella:

- En cuanto a los contenidos, los alumnos deben conocer y *comprender* la organización habitual de un ordenador, las relaciones entre subsistemas, así como las técnicas para aumentar el rendimiento de cada uno de ellos. Igualmente, los alumnos deben *comprender* las distintas aproximaciones a la organización de sistemas operativos, y las técnicas básicas para las funciones más importantes de éstos.

- En cuanto a las habilidades más directamente relacionadas con los contenidos, se espera que los alumnos puedan *aplicar* los conocimientos anteriormente mencionados para diseñar y evaluar distintas soluciones informáticas.

- Por último, hay un conjunto de habilidades de ámbito mayor, que los alumnos deben haber desarrollado al terminar la titulación. No son, por lo tanto, objetivos específicos de esta asignatura, pero en ella se incluyen explícitamente. En particular, se presta atención a la selección de información, la colaboración como compartición de información así como debate (crítica y aceptación de críticas) y la elaboración de informes.

Con estos objetivos, se ***eligen los casos y se detalla la técnica docente***. En esta asignatura se desarrolla desde hace cuatro años un *proyecto de diseño y evaluación* a lo largo de todo el cuatrimestre, utilizando la mayor parte de los créditos prácticos. En este proyecto, los alumnos trabajan en parejas asumiendo los roles de consultores y fabricantes que deben asesorar a un cliente sobre la adquisición de un sistema informático para sus necesidades. El profesor adopta por momentos el papel de cliente y el de director del equipo de consultores. En realidad en cada curso existen cinco clientes distintos, también diferentes de los de otros cursos, con lo que hay lugar para el contraste de soluciones. La elección de los casos se hace a partir de hechos de actualidad tecnológica (*Deep Blue Junior* sugiere un caso de juegos), de la literatura técnica (casos de computación ubicua), o de la investigación propia (un sistema distribuido para CSCL). Esta última aproximación a la selección de casos ayuda a realimentar mutuamente docencia e investigación.

El ***desarrollo del proyecto*** es aproximadamente el siguiente:

- El cliente describe su problema y formula unos deseos relativamente vagos (primer enunciado entregado a los alumnos). Los consultores deben averiguar más acerca de las necesidades del cliente. Deben traducir los requisitos del problema en requisitos técnicos o de producción. Durante este periodo los alumnos van estudiando los conceptos teóricos iniciales, pero desarrollan la habilidad de manejar información dispersa e incompleta.

- Los consultores hacen una primera propuesta para la CPU del sistema del cliente. Deben

Tabla 1: Resumen de las características más relevantes de las asignaturas para las que se han propuesto estudios de caso.

Asignatura	**Contexto**	**Objetivos de aprendizaje de la actividad**	**Actividad**	**Observaciones**
Arquitectura de Ordenadores	Troncal, 4º curso, 100 alumnos, última de bloque	Alternativas del diseño de CPU, memoria, E/S, SSOO Selección de información, metodología de evaluación, colaboración, capacidad crítica, elaboración de informes, planificación del trabajo	Proyecto de diseño y evaluación de sistema informático	Solidez en los conocimientos de la disciplina Habilidades potenciadas: redacción de informes, selección de información, argumentación y discusión. Motivación muy alta Mucha carga de trabajo
Arquitectura de Redes, Sistemas y Servicios	Troncal, 1er curso, 300 alumnos, primera de bloque	Alternativas en el diseño de una red de datos Selección de información, capacidad crítica	Estudio y discusión de las soluciones propuestas para dos redes universitarias	Sin evaluar
Ingeniería de Software	Optativa, 3er curso, 40 alumnos, pocas asignaturas relacionadas	Metodología UP y notación UML Selección de información, capacidad de abstracción, planificación del trabajo	Modelado de análisis y diseño dentro de un proyecto software	Sin evaluar

justificarla en función de los requisitos encontrados (no sólo técnicos) y de las tecnologías disponibles. Las propuestas y su evaluación se hacen públicas y los alumnos discuten las diferencias de soluciones para el mismo cliente y para distinto cliente. Durante todo este proceso se están desarrollando buena parte de las habilidades mencionadas en la sección 1 (búsqueda de soluciones de compromiso, defensa argumentada de las soluciones, aceptación de críticas, búsqueda de soluciones de compromiso...). Al terminar esta etapa generan un informe que debe ser formal y estructurado.

- Los consultores revisan su propuesta de CPU y añaden un diseño de jerarquía de memoria. De nuevo las propuestas son evaluadas y los resultados se hacen públicos y se discuten. Finalmente un nuevo informe es generado. Con esta etapa se refuerza el aprendizaje de las habilidades descritas anteriormente.

- Todos los consultores que atienden al mismo cliente se juntan en la última etapa, en la que deben terminar su propuesta de sistema informático, incluyendo la E/S, el sistema operativo y otro software, lo que debe reflejarse en un informe. Al deber llegar a una propuesta única se desarrolla la habilidad de discusión de las ideas para llegar a un consenso.

Este proyecto ha sido refinado con los años para adecuar la carga de trabajo, encauzar las formas de colaboración, y sobre todo introducir elementos que permitan a los profesores evaluar el éxito en la consecución de conocimientos y habilidades, así como proponer mejoras para años posteriores.

Para conseguirlo, se ha llevado a cabo una ***evaluación del proceso*** bastante detallada, que incluye la recopilación de opiniones personales de los alumnos en ocho instantes del cuatrimestre mediante una herramienta de encuestas [4], lo que permite conocer la evolución de sus habilidades, así como la observación de las sesiones de laboratorio por personal externo (investigadores de la Facultad de Educación). Como se motiva a los alumnos para que colaboren a través de herramientas informáticas (como por ejemplo el espacio de trabajo compartido BSCW [7]), también se han estudiado los registros (*logs*) de las interacciones de los alumnos, para evaluar la evolución de la colaboración. Todas estas fuentes de datos permiten un análisis cuantitativo (por ejemplo, cuánto ha aumentado la interacción entre alumnos, cuántas horas dedican a buscar información), mediante estadísticas simples o técnicas más complejas como el análisis de redes sociales [12,15], pero también un análisis cualitativo (por ejemplo, cuál es su percepción de la colaboración, cómo varía su motivación, cómo evoluciona su capacidad para hacer y recibir críticas), con el apoyo de herramientas de análisis cualitativo [13][12]. Entre las conclusiones más interesantes alcanzadas cabe mencionar las siguientes [3,12]:

- Los alumnos opinan que han aprendido más que si se hubiese utilizado una técnica tradicional, y además consideran lo aprendido útil y relacionado con el mundo real. Este hecho se confirma con los exámenes, donde los alumnos demuestran conocimientos sólidos y madurez para resolver los nuevos casos de estudio.

- La motivación es muy alta, y por ello la asistencia a clase y al examen también es cercana al 100% de la matrícula.

- Los alumnos aprecian que han desarrollado habilidad para redactar documentos (confirmada por las revisiones de los profesores). También desarrollan habilidad para seleccionar información (hacia el final de la asignatura manejan con soltura muchas fuentes de

información). Además, reconocen haber mejorado en su técnica para argumentar y discutir y sin embargo, no consideran que haya evolucionado su actitud hacia la colaboración. Esta última opinión es contraria a las observaciones, que indican que el número de interacciones es mucho mayor al final de la asignatura, pero puede explicarse a partir de los prejuicios positivos sobre esta habilidad: se dicen dispuestos a colaborar, son conscientes de que el mundo laboral requiere capacidades de trabajo en grupo, pero su visión de la colaboración es limitada (fundamentalmente, hacer todos juntos una misma cosa al mismo tiempo).

- La carga de trabajo para los alumnos es mucho más elevada. Esto se ha corregido parcialmente en el último curso ayudando a los alumnos en algunas tareas de forma más directiva. Por ejemplo, se les proporcionan plantillas bastante documentadas para los informes. La carga de trabajo para los profesores es también muy elevada, ya que se deben preparar varios casos, supervisar muchos informes, calificar los exámenes tradicionales, y además participar en la evaluación y propuestas de mejora sobre el propio proyecto docente. En este momento dicho proyecto docente tiene ya una cierta estabilidad, por lo que la carga de trabajo es proporcionada, y los profesores consideran rentable, en relación con el aprendizaje, el esfuerzo invertido.

Además de estas conclusiones, se ha investigado en mucho mayor detalle la evolución de la colaboración. Aunque esta investigación está más allá del objetivo de este artículo, el lector interesado puede consultar [12].

3.3 Ejemplo 2: Arquitectura de Redes, Sistemas y Servicios

Como se concluyó en la sección anterior, los resultados de esta técnica docente son satisfactorios pero tendrían un mayor impacto en la formación del alumno si se extendiese la técnica a otras asignaturas. En esta y la siguiente sección se hace esto, pero se observará que las diferencias de contexto y objetivos hacen que el planteamiento general deba rediseñarse con cuidado. En la discusión se seguirá la metodología propuesta en la sección 3.1.

El ***contexto de la asignatura*** *Arquitectura de Redes, Sistemas y Servicios* está determinado por el hecho de que es troncal y en el plan de estudios de Ingeniero de Telecomunicación de la Universidad de Valladolid está en primer curso, con 4'5 créditos teóricos y 4'5 prácticos. La aplicación del estudio de casos como técnica docente sería deseable por los contenidos de la asignatura, pero sus características justifican una mayor precaución al respecto:

- La asignatura comienza una rama dedicada al estudio de las redes de comunicaciones, el núcleo de la docencia del área [11]. Por lo tanto, los alumnos adquieren en esta asignatura los conocimientos iniciales de la disciplina, y además apenas han tomado contacto con disciplinas tecnológicas que puedan conectar en su estructura mental de conocimientos.

- El número de alumnos es muy elevado, y el nivel de conocimientos y capacidades es heterogéneo.

- Pese a lo anterior, la asignatura introduce cuestiones de diseño de soluciones para problemas abiertos (por ejemplo, el cableado de un edificio) que plantean por primera vez la búsqueda de compromisos [11].

La última característica mencionada justificaría el estudio de casos como técnica docente de manera similar a como se hace en *Arquitectura de Ordenadores*, pero en el contexto de esta asignatura dicho planteamiento fracasaría casi con certeza: los alumnos encontrarían grandes dificultades para manejar información muy poco estructurada, a veces contradictoria, y encontrarían desconcertante proponer soluciones para problemas poco determinados, para los que el profesor se niega a proporcionar una solución "correcta" (*la* solución). Por añadidura, todo este esfuerzo difícilmente les haría estructurar personalmente los contenidos aprendidos hasta el momento, que suelen estar débilmente interiorizados.

Cabe plantearse, sin embargo, otra posible variante de la técnica de estudios de caso. En la sección 2 se comentaba que la evaluación y crítica a una solución definida para un determinado problema permite desarrollar muchas de estas habilidades con un esfuerzo distinto, más adecuado quizá para contextos como el de esta asignatura de primer curso. Los ***objetivos de aprendizaje*** deben estar en concordancia con las posibilidades de esta aproximación: aquí se espera potenciar la capacidad de distinguir la información importante de los detalles superficiales, y buscar argumentos de apoyo o crítica para soluciones determinadas a problemas abiertos. Para conseguirlos se han de ***determinar la técnica y los casos adecuados***. Por las dificultades previsibles de llevar a cabo un estudio de casos en el primer curso, se ha seleccionado una actividad simple que utiliza sólo unas pocas horas de la asignatura, y según sus resultados en el futuro se rediseñará para tener un mayor peso en la asignatura. Esta actividad se lleva hacia el final de la asignatura, cuando los conocimientos previos son suficientes, y ***el desarrollo*** consiste en lo siguiente:

- Se propone analizar una solución *ya proporcionada* para la red de datos de una universidad. En particular, se proporcionará a los alumnos información sobre la evolución de la red de datos de la ETSIT de Valladolid (comenzó en un edificio temporal, con diseño inicial marcado por un presupuesto muy reducido, pero luego se

rediseñó para el edificio definitivo, en el que se diseñó el cableado estructurado) [9], así como información sobre la red de la University of New México que está bien documentada en su sitio web [14]. Ambos casos son significativos para el profesor, y parcialmente para los alumnos.

- Los alumnos se organizarán en grupos en los que repartir la información: unos alumnos serán *expertos* en una red y otros en la otra. Después los grupos se reunirán para compartir el conocimiento. Como las soluciones difieren, es de esperar que surja el debate y se originen argumentos de crítica o justificación.

- Los alumnos contestarán un cuestionario sobre el diseño, en el que las preguntas irán destinadas a buscar los argumentos encontrados a favor y en contra de las soluciones proporcionadas.

Como este estudio todavía no se ha llevado a cabo, la ***evaluación del proceso*** aún se está discutiendo. Para una experiencia tan breve, posiblemente sea suficiente con un cuestionario tras la actividad, que indague sobre la consecución de los objetivos (apreciación subjetiva por parte de los alumnos) y sobre las dificultades encontradas en el proceso.

3.4 Ejemplo 3: Ingeniería de Software

La disciplina de ingeniería de software se imparte en la ETSIT de Valladolid en una asignatura optativa denominada *Sistemas Telemáticos I*, con 1'5 créditos teóricos y 4'5 prácticos [11]. El ***contexto de esta asignatura*** es más favorable que el de la anterior para el estudio de casos: está en tercer curso, el número de alumnos es reducido, y muchos de los conocimientos útiles para contextualizar la ingeniería de software ya han sido estudiados formalmente (programación, economía) o informalmente (visión de sistemas y arquitecturas). Además, el peso práctico en la asignatura es significativo, tanto en el número de créditos, como en el carácter de la disciplina tratada.

Por otra parte, los ***objetivos de aprendizaje*** vienen dictados por la propia disciplina: se espera que el alumno se acostumbre a aproximarse a un proyecto software con una metodología de ingeniería, que incluye el recorrido de distintas actividades y la generación de distintos productos. En realidad, además de ayudar a aprender unos conocimientos básicos (por ejemplo, fases y actividades del proceso unificado o notación UML), el énfasis de esta asignatura debería estar fundamentalmente en desarrollar habilidades: de análisis (selección de información), de modelado (abstracción) y de diseño (elección argumentada de soluciones). Adicionalmente, se espera que los alumnos desarrollen cierta capacidad de planificación de su trabajo.

La ***técnica y caso planteados*** para la asignatura son un diseño de proyecto informático para una Unidad de Cuidados Intensivos en un hospital cercano. El ***desarrollo del proyecto*** sigue las pautas del proceso unificado (*Unified Process*, UP), bajo la suposición de que los alumnos se organizan en grupos de tres y toman el papel de ingenieros de software (analistas, arquitectos, diseñadores) y los profesores el papel de clientes. Médicos de este hospital ayudaron a los profesores a preparar el caso, y después han intervenido en clase para añadir realidad al caso. Los profesores intervienen determinando que artefacto UP debe producirse en cada sesión, y explicando la sintaxis UML relacionada, pero los alumnos deciden su propio plan de trabajo dentro de esa sesión.

Al igual que en la asignatura anterior, el ***proceso de evaluación*** está todavía en elaboración. Sin embargo, la experiencia de *Arquitectura de Ordenadores* sugiere no basar la evaluación en datos recogidos *al final* del proceso, y por ello está planteado un primer cuestionario intermedio en el que se explorará la visión subjetiva de los alumnos de su motivación, su disciplina y organización, interiorización de la metodología, formas de colaboración, problemas al recolectar información y nivel de conocimientos adquiridos. En contraste con las apreciaciones de los profesores, estas opiniones permitirán evaluar el éxito y las limitaciones de la primera mitad de la actividad, y corregir estas últimas en la medida de lo posible.

4 Conclusiones y discusión

En este artículo se han descrito tres experiencias de la aplicación del estudio de casos en asignaturas de Ingeniería Telemática. Esta técnica docente permite conseguir una mayor interiorización de los conocimientos de la disciplina, pero además consigue una mayor motivación de los alumnos, que dejan de ser pasivos, por lo que se potencia el desarrollo de sus habilidades, entre las que se han documentado la de seleccionar información, discutir, llegar a consensos o estructurar un informe.

Para superar las dificultades en la aplicación de esta técnica, se ha propuesto una metodología, validada en las distintas experiencias descritas. El estudio del contexto de las distintas asignaturas ha mostrado como los grandes proyectos de diseño y evaluación son viables pero requieren una cierta madurez por parte de los alumnos, mientras que otras alternativas, como el análisis de soluciones pueden ser factibles para cursos más bajos. La elección del caso y la preparación de la documentación deben hacerse con cuidado, intentando que sea real, abierto a distintas soluciones, con restricciones de tiempo, y por supuesto relacionado con los contenidos de la asignatura. Una vez elegido se desarrolla el caso, lo que puede llevar de unas horas a un cuatrimestre entero. Pautas generales que debe tomar el profesor para este desarrollo son fomentar el trabajo en grupo, la discusión, y no tomar posturas fuertemente directivas, tanto en la metodología como en los contenidos, que transmitan la sensación de que hay *una* solución correcta (la del profesor). Finalmente,

es muy importante recordar que el proceso debe ser evaluado. Esta evaluación debe ir orientada a comprobar que los objetivos se han cumplido (los de contenidos y los de habilidades), así como a capturar la visión subjetiva de los alumnos sobre *el proceso*, con especial atención a los problemas encontrados. La experiencia de *Arquitectura de Ordenadores* descrita muestra que la evaluación puede mejorar notablemente el proceso en los cursos siguientes, ayudando al profesor a seleccionar actividades en función de los objetivos y de su viabilidad, mejorando el aprendizaje, lo que es el objetivo último. Por supuesto, el proceso de evaluación no tiene por qué ser siempre tan detallado como en este caso (de hecho no lo es en las otras dos actividades descritas).

En cualquier caso, el esfuerzo de todos los participantes (profesores, alumnos y observadores) es bastante mayor con estas técnicas que con la tradicional clase magistral y sus variantes. Cabe por lo tanto discutir si este esfuerzo genera un beneficio suficiente. En general, la apreciación subjetiva derivada de estas experiencias es que en los primeros cursos de su aplicación el esfuerzo es superior porque requiere mucha preparación y es necesaria una evaluación del proceso para proponer mejoras. Sin embargo, una vez que el proceso alcanza cierta estabilidad, la carga de trabajo sigue siendo mayor que con las clases tradicionales, pero es proporcionada y rentable, teniendo en cuenta que el aprendizaje de contenidos y habilidades es mayor.

El trabajo futuro inmediato consiste en evaluar y mejorar el planteamiento del estudio de casos en estas asignaturas. Además, en línea con la investigación de los autores, se están desarrollando herramientas CSCL para fomentar la colaboración y el trabajo en grupo.

Agradecimientos

El trabajo presentado en este artículo ha sido financiado con fondos del proyecto de la Junta de Castilla y León “UV38/02” y parcialmente de los proyectos del MCyT “TIC2000-1054” y “TIC2002-04258-C03-02”. Los autores agradecen la colaboración de A. Martínez, B. Rubia e I. Jorrín.

Referencias

[1] ACM y IEEE Computer Society. Computing Curricula: http://www.computer.org/education/cc2001, 2001

[2] B.G. Wilson. *Constructivist learning environments: case studies in instructional design*, Englewood Cliffs, NJ, EEUU: Educational Technology Publications (1996).

[3] Y. A. Dimitriadis, A. Martínez, B. Rubia, M. Gallego. "Cooperative learning in Computer Architecture: and educational project and its telematic support". Proc. of the Conference of Frontiers in Education. Reno,NE, EEUU, 2001

[4] E. Gómez, B. Rubia, Y.A. Dimitriadis, A. Martínez. "Quest, a telematic tool for automatic management of student questionnaires in educational research". Proc. of the Second European Conference on Technology, Information, Education and Citizenship. Barcelona, 2002

[5] C.A. Ellis, S.J. Gibbs, G.L. Rein. "Groupware: some issues and experiences". Comm. of the ACM, pp. 9-28, vol. 34 (1) (1991).

[6] G. Fernández. *Conceptos básicos de arquitectura y sistemas operativos*, Madrid: Sistemas y Servicios de Comunicación (1998).

[7] GMD-FIT, Basic Support for Cooperative Work, v. 4.0: http://bscw.gmd.de, 2002.

[8] J. L. Hennessy, D. A. Patterson. *Computer Architecture: a quantitative approach*, San Francisco, CA, EEUU: Morgan-Kaufmann (2002).

[9] A. Izquierdo , Y. A. Dimitriadis. "La red de datos de la ETSIT: evolución histórica". En: *Introducción práctica a la administración de sistemas en Internet*, eds. Y.A. Dimitriadis, F.J. Díaz. Secretariado de publicaciones e intercambio científico, Universidad de Valladolid (1998).

[10] J.A. Sánchez. El estudio de casos: Talleres de técnica docente para el profesorado universitario, Universidad de Valladolid, 2002.

[11] J.I Asensio. Resúmenes de las reuniones de la comisión de docencia del área de Ingeniería Telemática, ETSIT, Universidad de Valladolid: http://www.personal.tel.uva.es/~juaase/docencia, 2002.

[12] A. Martínez, Y.A. Dimitriadis, B. Rubia, E. Gómez, P. Fuente. "Combining qualitative evaluation and social network analysis for the study of classroom social interactions". Computers and Education (por aparecer).

[13] QSR. *Nud*IST, software for qualitative data analysis*, Thousand Oaks, CA, EEUU: Scolari (1997).

[14] UNM. Network presentation & training: http://www.unm.edu/~network/presentations/course/index.html, 1995

[15] S. Wasserman, K. Faust. *Social Network Analysis:methods and applications*, Thousand Oaks,CA, EEUU: Sage Publications (1994).

Aportaciones del proyecto VOTESCRIPT a los esquemas tradicionales de voto electrónico

Ana Gómez Oliva[1], Jesús Moreno Blázquez[1] y Sergio Sánchez García[2]
[1]Departamento de Ingeniería y Arquitecturas Telemáticas. Universidad Politécnica de Madrid
Ctra. Valencia km. 7. 28031 Madrid.
Teléfono: 913 36 78 20. Fax: 913 36 78 17
E-mail[1]: {agomez,jmoreno}@diatel.upm.es
E-mail[2]: ssanche@proyectos.diatel.upm.es

Abstract. *This paper hallmarks the most relevant contributions carried out by the authors in the VOTESCRIPT project (TIC2000-1630-C02). The main goal of this project was the analysis, definition and implementation of a system which copes with every phases and elements existing in a process of electronic voting using computer networks. A summary of the main criticisms of electronic voting is presented to disclose that the most relevant voting schemes only take into account a technological perspective, just trying to imitate the conventional voting schemes. Nevertheless in these proposals important aspects such individual and global verification are not properly undertaken. The paper includes the proposed solutions of the project to solve these mentioned problems.*

1. El voto electrónico ¿una realidad inmediata?

Continuamente los medios de comunicación nos ofrecen noticias sobre el éxito de nuevas experiencias de voto electrónico que se llevan a cabo en todo el mundo. Ante esta profusión de acontecimientos cabe preguntarse si realmente la votación electrónica ha alcanzado ya la madurez y, en breve, podrá sustituir a la votación tradicional.

Sin embargo, un primer análisis de estas experiencias nos lleva a comprobar que el término de *votación electrónica* es un término muy amplio que engloba numerosas actuaciones que pueden ser clasificadas en dos grandes apartados:

- El que sustituye alguno de los elementos físicos del procedimiento de votación clásico por algún tipo de proceso electrónico y

- El que emplea redes telemáticas para comunicar a los votantes con una Mesa Electoral remota.

La casi totalidad de las acciones gubernamentales encaminadas a la automatización de los procesos de votación se encuadran en las actuaciones del primer apartado, siendo la urna electrónica, con o sin papeleta, el dispositivo más comúnmente empleado en todos los casos (la reciente experiencia de Brasil con 135 millones de personas empleando este sistema avalan la validez oficial de este método, a pesar de las nulas garantías de verificación que ofrecía, por tratarse de un sistema de urna sin papeleta).

El segundo apartado, voto a través de Internet, que nosotros hemos dado en llamar **voto telemático**, es el que, a priori, resulta más atractivo desde el punto de vista del ciudadano, ya que le permitiría votar desde casa o desde cualquier punto destinado al efecto, sin necesidad de estar ligado a un determinado Colegio Electoral. Sin embargo, es aquí donde se plantean los mayores retos, no sólo desde el punto de vista técnico, ya que es preciso dotar al sistema de las adecuadas medidas de seguridad, sino también desde el punto de vista social, puesto que el sistema no debe fomentar un desequilibrio en la participación, y por tanto, en la toma de decisiones, en función del nivel de formación informática del ciudadano.

En este apartado, voto telemático, se han realizado escasas experiencias con validez oficial, destacando que en la mayoría de ellas no se reproducen las mismas garantías de seguridad que se proporcionan con el sistema de voto tradicional, como son la posibilidad de que existan interventores para supervisar el proceso o que, en caso de discrepancia, exista la posibilidad de verificar los resultados.

El proyecto VOTESCRIPT[1], finalizado oficialmente en diciembre de 2002, ha abordado la problemática de los sistemas de votación telemática, teniendo como objetivos la modelización y el desarrollo de un prototipo de votación electrónica para realizar votaciones seguras mediante redes de ordenadores públicas y, por tanto, no seguras. Este trabajo ha incluido la realización del análisis, la definición y la implementación de un sistema capaz de soportar los diferentes pasos y elementos existentes en un proceso de votación electrónica, abarcando desde el proceso de emisión del voto hasta el proceso de recuento.

[1] El proyecto VOTESCRIPT (TIC2000-1630-C02) ha sido subvencionado por el Ministerio de Ciencia y Tecnología dentro del Plan Nacional de I+D+I (2000-2003)

Este proyecto ha sido abordado desde una perspectiva integradora que tuviera presente tanto los aspectos técnicos como los sociales, esto es, la solución técnica propuesta debería incluir los mecanismos necesarios para resolver todos los aspectos de seguridad exigibles a un sistema de votación telemática pero, a la vez, esta solución debería ser diseñada de manera que gozara de una amplia aceptación por parte de los ciudadanos.

Este papel destaca las principales aportaciones de este proyecto a la votación electrónica, comparando las soluciones propuestas con las recogidas en los principales esquemas de votación que hoy día sirven de referencia en esta área.

Cabe destacar que durante la realización de este proyecto ha existido una colaboración con la Casa de la Moneda, de manera que parte de las soluciones aquí propuestas han sido trasladas al sistema de votación desarrollado por la Casa de la Moneda, del que se realizó una experiencia piloto en El Hoyo de Pinares (Avila) el pasado mes de marzo.

2. Puntos débiles de los sistemas de votación electrónica

Las principales críticas que se hacen a los sistemas de votación electrónica fueron recogidas por Mercuri[1] en su intervención en la Cámara de Representantes del Comité de Ciencia de EEUU. Pueden resumirse en:

a) Que es imposible superar aspectos tan críticos como son el riesgo de venta de votos, coacción, monitorización clandestina y denegación del derecho a voto.

b) Que no hay forma de ofrecer al votante la seguridad de que el voto se ha registrado tal cual ha sido emitido, o que el recuento es el correcto.

c) Que no ofrece control por parte de los partidos políticos.

d) Que los defectos del sistema pueden ser conocidos años después de la elección y que no hay elementos de auditoría.

e) Que los mecanismos criptográficos se pueden romper tarde o temprano.

f) Y que desde cualquier lugar del mundo se pueden atacar los sistemas telemáticos.

La primera tarea del proyecto fue analizar las soluciones que diversos autores han propuesto para solventar los problemas mencionados. Estas soluciones o *esquemas de votación* definen los agentes, procedimientos y protocolos de seguridad necesarios para llevar a cabo el proceso de votación.

En los esquemas de votación analizados (de los que son una muestra [2] [3] [4] [5] y [6]), la determinación de los requisitos de seguridad que debía reunir el sistema se realizaba reproduciendo las garantías proporcionadas por el voto tradicional, por lo que estos esquemas se centran en garantizar el anonimato del votante.

En este proyecto se ha abordado el desarrollo del sistema desde un punto de vista interdisciplinar, tanto sociológico como telemático, lo que ha propiciado que el sistema de votación se haya diseñado en base a los nuevos requisitos demandados por los ciudadanos y determinados por la investigación sociológica. Entre estos requisitos demandados por los votantes cabe destacar la necesidad de disponer de herramientas para poder verificar el correcto funcionamiento del sistema, no sólo a nivel global, sino también a nivel particular.

En los esquemas citados se observó que, en la mayoría de ellos, no existía la posibilidad de verificar que el sistema operaba correctamente, es decir, que como consecuencia del comportamiento malicioso de algún agente telemático del sistema (Mesa Electoral, Urna o Contador) o la confabulación de varios agentes dentro del sistema, no se estaba produciendo una alteración de los resultados de la votación.

Por tanto, las tareas de este grupo se centraron en el desarrollo de un esquema de votación que ofreciera a los votantes, al menos, las mismas garantías de seguridad que la votación tradicional, poniendo especial énfasis en que además ofreciera pruebas robustas (mediante el empleo de algoritmos criptográficos) del correcto funcionamiento del sistema.

3. Características del sistema definido en el proyecto VOTESCRIPT y resumen de la arquitectura

En el desarrollo del proyecto VOTESCRIPT se decidió emplear tarjetas inteligentes para garantizar la identidad del votante, mediante un certificado personal incluido en la tarjeta. En este punto, se vio la conveniencia de emplear tarjetas criptográficas que permitieran, además, la realización de determinadas funciones de cifrado/descifrado o comprobación de firmas dentro de la propia tarjeta, con objeto de impedir el ataque al sistema. Asimismo, se detectó la necesidad de almacenar cierta información asociada al proceso de emisión del voto, con vistas a una posible verificación posterior.

Respecto a las **críticas a)** y **f)** del apartado anterior se constató que la votación desde casa a través de Internet, aunque puede ofrecerse con las garantías de seguridad adecuadas, hoy día presenta riesgos derivados principalmente de la dificultad de determinar la libertad de acción de la persona que

está utilizando la tarjeta de identificación y por las serias amenazas al sistema que pudieran derivarse de un ataque de denegación de servicio originado por *hackers* que impidieran la celebración de la votación. Por tanto, se consideró que el sistema con más expectativas de éxito a medio plazo era aquel en el que se empleasen puntos de votación que se comunicaran con una urna central mediante una red privada virtual.

En este diseño, se optó por un sistema en el que el votante debe emplear una cabina de votación para emitir su voto, desplazándose a un Colegio Electoral, en lugar de hacerlo desde casa por Internet. De esta forma, se ha pretendido satisfacer de forma más adecuada los requisitos de seguridad necesarios, así como evitar la problemática de la compra de votos, la coacción en el momento de la emisión del voto y la posibilidad de ligar el voto con la ubicación física del votante.

El sistema diseñado se compone de un conjunto de agentes telemáticos (véase Fig. 1), que resumidos brevemente son:

Las Cabinas de Votación, un Sistema Administrador o Autoridad de Identificación encargada de validar al votante como persona autorizada a votar por estar en el censo, un Sistema de Intervención por cada una de las distintas candidaturas que se determine deban participar en la fase de votación, una Urna, un proceso Contador y un conjunto de Puntos de Verificación. El sistema contempla, además, la existencia de una persona jurídica, la Autoridad de Elección, encargada del control general, que se ocupa de atender todas aquellas posibles reclamaciones con respecto al funcionamiento del sistema.

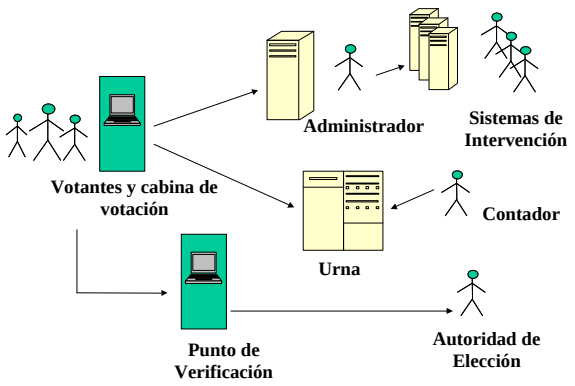


Fig. 1 Agentes telemáticos definidos en el sistema VOTESCRIPT

Como paso previo al comienzo de la votación, se habrá hecho llegar a los votantes una tarjeta inteligente y un identificador de votante que deberá ser conocido por todos los agentes del sistema VOTESCRIPT. La tarjeta inteligente, diseñada especialmente para este proyecto, es capaz tanto de generar claves como de realizar gran parte de los procesos criptográficos necesarios para la seguridad del sistema.

Una vez publicados los resultados de la votación y durante un tiempo limitado, es posible una verificación individual que podrá ser realizada por los votantes a través de los Puntos de Verificación. Asimismo, las distintas candidaturas podrán llevar a cabo una verificación global de los resultados apoyándose en las pruebas criptográficas acumuladas en los Sistemas de Intervención durante el proceso de votación.

4. Aportaciones arquitecturales

En este apartado se comentan las medidas novedosas que han sido incorporadas en la arquitectura del sistema para superar los problemas y críticas mencionados anteriormente.

- Existen unos Sistemas de Intervención para las candidaturas. Cada uno de los Sistemas de Intervención está controlado por un interventor. Estos sistemas, que forman parte del sistema global, serán proporcionados por la Administración Pública y no serán elementos propiedad de las candidaturas. Se prevé que estén ubicados en el mismo entorno que el Administrador y que sus programas estén homologados y sean resistentes ante ataques. Asimismo, se prevé que puedan ser auditados por peritos de confianza de las candidaturas antes del proceso de votación.

  La existencia de los Sistemas de Intervención es una de las principales aportaciones de este sistema, puesto que permite el control, por parte de los partidos políticos, de todo el proceso electoral, a la vez que les dota de la posibilidad de realizar de forma sencilla una auditoría no sólo del resultado final sino de todo el proceso. Esta característica permite contrarrestar la **crítica c)** del apartado 2.

- Existe un conjunto de Puntos de Verificación. Los puntos de verificación son elementos cuya funcionalidad es la de proporcionar a los votantes un lugar en el que llevar a cabo la verificación individual del tratamiento dado a su voto por parte del sistema. Mediante la verificación individual cada votante podrá comprobar, de forma independiente, si su voto se ha tenido en cuenta y ha sido correctamente contabilizado.

  Estos puntos de verificación pueden ser las mismas cabinas de votación u otros sistemas adicionales. El principal requisito de seguridad que se les exige es que no den publicidad a la clave con que se ha emitido el voto para evitar así la compra de votos.

- Existe una Autoridad de Elección encargada del control general del sistema, de velar por su correcto funcionamiento, ocupándose de atender todas las posibles reclamaciones que realicen los votantes. En el caso de que se produzca una reclamación por parte de un votante sobre el tratamiento dado a su voto, ésta descubrirá y comparará todas las pruebas criptográficas presentes en el sistema para comprobar la validez del recuento. Solicitará al votante la tarjeta utilizada para la votación y, a partir de ella, podrá determinar si el votante tiene o no razón, si ha existido o no una falsificación por parte del sistema, y estará en condiciones de llevar a cabo las acciones necesarias en cada caso.

- La Urna, además de recoger y almacenar el voto, genera y envía a la Cabina un comprobante de su entrega, que ésta guarda en la tarjeta inteligente del votante. Mediante este comprobante el votante podrá verificar su voto y reclamar en el caso de que detecte un tratamiento incorrecto de su voto por parte del sistema.

5. Verificación

En todo este proceso es importante la forma en que se realiza la **verificación de los resultados** de la votación. Los esquemas de votación clásicos analizados suponen que todos los agentes telemáticos del sistema operan honestamente, siendo por tanto relativamente sencillo manipular los resultados de la votación, sin que exista ningún método de detección. El sistema diseñado se ha concebido para permitir la verificación tanto a nivel global como a nivel individual.

La verificación global, puesta a disposición de los interventores, proporciona pruebas criptográficas robustas, que permiten demostrar sin ningún tipo de ambigüedad si el sistema ha operado de forma fraudulenta. La verificación individual es otro mecanismo, puesto a disposición de los votantes, que les permite durante un tiempo determinado y en unos lugares concretos, comprobar qué opción de voto se les ha contabilizado. La novedad respecto a otras soluciones radica en que en ningún momento el votante puede demostrar ante terceros no autorizados qué ha votado, lo que impide la compra de votos o la extorsión.

5.1 Verificación individual

La verificación individual podrá ser realizada por los votantes a través de los Puntos de Verificación, una vez finalizado el proceso de votación y durante un tiempo limitado. El votante que desee verificar su voto acudirá a uno de estos Puntos de Verificación y previa identificación se le permitirá acceder, de forma individual, a un sistema en el que introduciendo la tarjeta inteligente, podrá leer en una pantalla el voto que le ha sido contabilizado por el Contador, no entregándosele ninguna prueba de esta comprobación. Caso de que el votante crea que votó por una opción distinta a aquella que le ha sido mostrada podrá iniciar un proceso de reclamación, entregando su tarjeta inteligente a la Autoridad de Elección, la cual descubrirá y comparará todas las pruebas criptográficas presentes en el sistema para comprobar la validez del recuento. De esta manera se proporciona al votante herramientas que contrarresten la **crítica b)** del apartado 2.

5.2 Verificación global

Una vez publicados los resultados de la votación, y con la intención de que las distintas candidaturas obtengan una prueba del correcto funcionamiento del Contador a la hora de abrir y contar votos, se permite que cada una de ellas verifique el procedimiento. Cada candidatura tiene la posibilidad, mediante una serie de procedimientos concretos que se han diseñado, de comparar la información que posee con la que se ha obtenido como resultado final del proceso de recuento. Caso de que ambas informaciones no se correspondieran, podrían proceder a impugnar la votación, presentando para ello pruebas criptográficas robustas. Mediante estas pruebas criptográficas se introducen elementos de auditoría en el sistema que permiten garantizar la validez de todo el proceso, respondiendo así satisfactoriamente a la **crítica d)** del apartado 2.

El sistema diseñado garantiza también que el voto emitido **no podrá ser conocido en el futuro**. Los esquemas de votación clásicos analizados se basan en la presentación del voto debidamente ocultado al Administrador (y Sistemas de Intervención, si los hubiera) del sistema de votación para que verifique que el votante tiene derecho a votar y que no lo ha hecho todavía. El hecho de presentar a los interventores de las candidaturas el voto oculto mediante algoritmos criptográficos garantiza que en la actualidad éstos no puedan conocer su contenido, pero no garantiza que con el avance de las técnicas de criptoanálisis éste no pueda ser conocido en el futuro. El sistema desarrollado en el proyecto VOTESCRIPT aporta como novedad que en la fase de autenticación del votante no se presenta al Administrador o a los Sistemas de Intervención el voto sino la clave que se va a utilizar después para descifrarlo, evitándose que el Administrador lo pueda almacenar para el futuro (el voto únicamente se envía a la urna). Con todo ello se elimina el riesgo mencionado en la **crítica e)** del apartado 2.

6. Conclusiones

Para el diseño del sistema se ha partido de un análisis crítico y exhaustivo de las experiencias y propuestas que habían sido formuladas con anterioridad y se ha optado por una metodología multidisciplinar (tecnológica, sociopolítica y jurídica) tanto para la determinación de los requisitos y condicionantes como para la evaluación del sistema final que se plantea.

Es importante destacar que la fortaleza del sistema se basa en la obtención, por parte de los distintos actores del sistema (entre los que principalmente se encuentran los votantes y los interventores), de piezas de información criptográficamente robustas y seguras que podrán presentar como prueba ante terceros en caso de litigio o disconformidad con los resultados del proceso.

En los apartados precedentes se ha expuesto un resumen de los trabajos desarrollados dentro del proyecto VOTESCRIPT. En la actualidad, y cómo consecuencia de una colaboración con la Fábrica Nacional de Moneda y Timbre – Real Casa de la Moneda, se ha desarrollado un prototipo que se ha utilizado con éxito en una experiencia práctica llevada a cabo el día 16 de marzo pasado en el pueblo abulense de El Hoyo de Pinares y que ha tenido amplia repercusión tanto en la prensa escrita como en televisión y radio por tratarse de la primera experiencia institucional en España de voto electrónico por Internet [7] [8].

Referencias

[1] Mercuri R. *Testimony presented to the U.S. House of Representatives Committee on Science*, Mayo 2001. http://www.house.gov/science/full/may22/mercuri.htm

[2] Fujioka, T. Okamoto, K. Otha. *A Practical Secret Voting Scheme for Large Scale Elections*, Advances in Cryptology, AUSCRYPT´92, Lecture Notes in Computer Science 718. Springer-Verlang, Berlin, pp.244-251 (1993).

[3] Cranor, Lorrie F. y Cytron, Ronald K. *Design and Implementation of a Practical Security-Conscious Electronic Polling System*, WUCS-96-02, Departamento de Informática, Universidad de Washington, St. Louis, Enero 1996.

[4] Herschberg, Mark A. *Secure Electronic Voting Over the World Wide Web*, Tesis doctoral en Ingeniería Eléctrica e Informática, Massachusetts Institute of Technology, 1997.

[5] M.Ohkubo, F. Miura, M. Abe, A. Fujioka, T. Okamoto. *An Improvement on a Practical Secret Voting Scheme*. Lecture Notes in Computer Science 1729, Springer-Verlag, Berlin, pp. 225-234 (1999).

[6] Riera i Jorba, Andreu. *Design of Implementable Solutions for Large Scale Implementable Voting Schemes*, Tesis doctoral Universidad Autónoma de Barcelona, 1999.

[7] http://www.cert.fnmt.es opción Hemeroteca.

[8] Votación en Hoyo de Pinares http://vototelematico.diatel.upm.es

Un Sistema para la Agregación de Contenidos en Internet

Javier López Mato, Ángel Viña
Departamento Tecnologías de la Información y las Comunicaciones
Facultad de Informática. Universidad de A Coruña
15071 A Coruña
Teléfono: 981 167 000 Fax: 981 167 160
E-mail : {jmato,avc}@udc.es

Marcos Casas, Pedro Moreira
Denodo Technologies.
Calle Real 22, 3º
15003 A Coruña
Teléfono: 981 100 200 Fax: 981 100 205
E-mail : {mcasas,pmoreira}@denodo.com

Abstract. *Nowadays, a Internet user can access a great amount of personal information through the Web. Practically all banking organizations allow their clients to obtain information on their accounts, credit cards, etc.;services companies offer invoicing information to their clients; in addition, if we add other services like fidelity cards, webmail, etc., the result is that a user who wants to obtain an updated view of his personal information will have to start a long trip through the involved webs. In addition, usually all of them force the user to follow a certain authentication process. The final scene has little attractive to users that, consequently, will limit the visit to a subgroup of the involved webs. The system proposed automates as much the authentication as the location and recovery of the personal information in each one of the webs, besides generating a unified format that will make easier its presentation and/or processing.*

1 Introducción

"El número de usuarios de agregación financiera está a punto de explotar. Más de 35 millones de consumidores, el 61% de los clientes europeos de banca electrónica, usarán agregación financiera en 2005. La agregación financiera ofrece a los clientes on-line la oportunidad de obtener todas sus cuentas bancarias, además de otros muchos artículos, en un portal personalizado sin importar dónde residen dichas cuentas "

Datamonitor, abril 2002

En este artículo se propone un sistema de agregación genérico, que además de productos financieros es capaz de agregar cualquier información de un usuario que sea accesible a través del Web. Por ejemplo, datos de facturación que ofrecen empresas de servicios (telefonía, electricidad, agua, gas, etc.); programas de fidelización (tarjetas de puntos); correo web, etc. Sobre este sistema se puede implementar cualquier servicio de agregación de contenidos en Internet.

En la implementación del sistema se han seguido los estándares J2EE.

En la arquitectura del sistema (Fig. 1) se diferencian dos áreas funcionales: Una de ellas, constituida por uno o más Módulos Extractores, es la encargada del acceso y recuperación de la información personal en las fuentes Web agregadas; La otra, centrada en el Módulo Agregador, es la responsable de ofrecer el acceso coherente a toda esa información personal agregada.

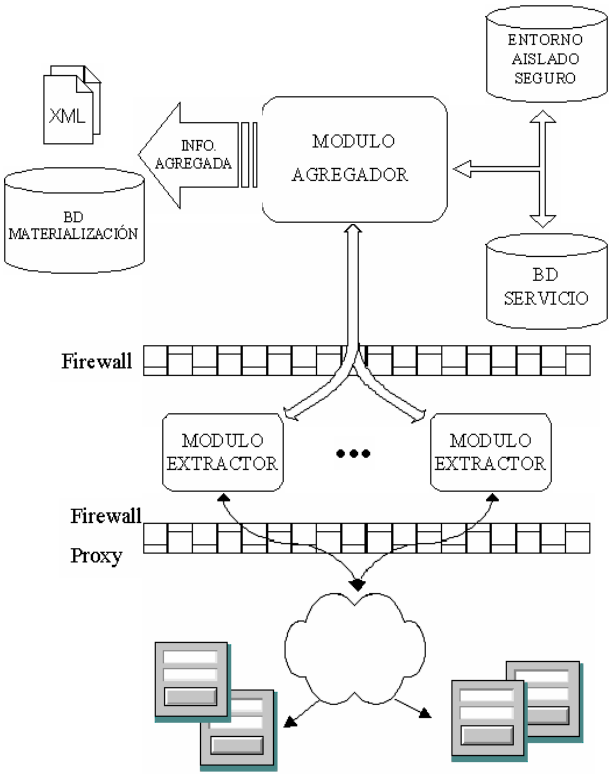


Figura 1: Arquitectura del sistema

En todo el proceso, al manejar información personal, es obvio que se debe tener muy en cuenta la seguridad del sistema. Seguridad que debe abarcar no sólo al almacenamiento físico de la información, sino también a cómo esta información es obtenida y transmitida por la red. El sistema propuesto establece políticas de seguridad en todos estos puntos.

El artículo comienza con la descripción del Módulo Extractor: se introduce una herramienta semi-automática que facilita su operación y mantenimiento, y se detalla la política de seguridad que afecta a su operación.

Continúa después con la descripción del Módulo Agregador: se describe el modelo de datos del sistema, se incide en los distintos enfoques de funcionamiento y finalmente se detalla la política de seguridad que afecta a su operación.

Finaliza con la presentación de la experiencia extraída del desarrollo del sistema, la comparación con otros sistemas similares y las líneas de trabajo futuras.

2 Módulo Extractor

Este módulo se basa en un sistema de mediadores [1] que permite la creación de un esquema global unificado a partir de datos estructurados y semi-estructurados. Más concretamente se basa en la que podemos llamar capa física del sistema de mediadores.

Todas las fuentes WWW que aborda el sistema descrito en este artículo son un ejemplo típico de datos semi-estructurados [2].

En esta sección, presentaremos el sistema de mediadores. Y continuaremos con una discusión de cómo este sistema aborda las fuentes WWW.

2.1 Sistema de mediadores

Los sistemas de mediadores, que fueron propuestos por primera vez en [1], se han empleado con éxito para desarrollar sistemas de integración de datos de forma más rápida, barata y menos intrusiva que otros enfoques tradicionales.

En este sistema, los datos permanecen en sus fuentes y el mediador es el responsable de que los usuarios tengan la ilusión de estar consultando un único y coherente esquema de datos. Por ello este enfoque también recibe el nombre de virtual. Como los datos permanecen en las fuentes, la actualidad está garantizada.

La capa física del sistema de mediadores está constituida por un conjunto de wrappers o envoltorios, cada uno de ellos adaptado a un tipo de fuente concreto (bases de datos relacionales, WWW, ficheros planos, etc.).

Tratándose de datos semi-estructurados, los wrappers envuelven la fuente ofreciendo una visión estructurada de la misma. Cada wrapper debe proporcionar acceso a la relación base de una fuente de tal modo que, a la vista del mediador, sea como una tabla en una base de datos relacional. Por ejemplo, para nuestro mediador, una determinada banca electrónica será vista como un conjunto de tablas con información de los productos que poseen sus clientes. Así podemos tener una tabla cuenta con una serie de atributos (número de cuenta, titular, saldo, etc.), una tabla tarjeta con sus propios atributos, etc.

2.2 Generación de wrappers para fuentes Web

El sistema de mediadores genera de forma semi-automática los wrappers necesarios para sitios Web, JDBC, bases de datos XML y ficheros de texto tanto estructurados como semi-estructurados. Este proceso se realiza a través de una herramienta gráfica que permite que los wrappers sean creados y mantenidos fácilmente.

En el sistema propuesto en este artículo todas las fuentes de información son fuentes Web, así que nos centraremos en este tipo de wrappers.

El acceso y la navegación dentro de fuentes Web se lleva a cabo empleando el navegador Microsoft Internet Explorer (MSIE). Se ha definido el lenguaje NSEQL (Navigate SEQuence especification Language) que permite especificar cualquier secuencia de navegación. Se trabaja por tanto a nivel de navegador en vez de a nivel http (como hacen la mayoría de los sistemas de wrappers para fuentes Web).

NSEQL permite al usuario definir “macros” directamente sobre la interfaz del navegador, así el acceso a una fuente es idéntico al proceso seguido por un usuario que se conectase a la fuente usando MSIE. Esto permite que el wrapper sea totalmente independiente del mecanismo de mantenimiento de sesión empleado por la fuente (que puede ser muy complejo en fuentes Web comerciales, especialmente en bancas electrónicas), código Javascript, HTML dinámico, securización de las comunicaciones con HTTPS, etc. Permite además, generar las secuencias de navegación a partir de ejemplos, es decir, simplemente navegando.

Veamos en un ejemplo, cómo es este lenguaje.

Ejemplo 1: Supongamos que queremos modelar una banca electrónica como una relación CUENTA cuyos atributos sean NUMEROCUENTA, DESCRIPCION, SALDO y DIVISA. Para acceder a esta información normalmente tendremos que entrar en dicha banca electrónica para luego navegar en su interior hasta encontrar el listado de cuentas de un usuario.

```
Navigate (https://www.banco.es/
login, 1);
FindFormByName("ident",0);
SetInputValue("login",0,@LOGIN);
SetInputValue("pwd",0,@PASSWORD);
ClickOnElement("validar","input",1);
FindFrameByName("menu",0,true);
ClickOnAnchorByText("MisCuentas",
0,false,1);
FindFrameByName("menu",0,true);
```

Figura 2: Secuencia NSEQL

En la Fig. 2 se muestra la secuencia de comandos NSEQL capaz de llegar hasta la información requerida.

El comando "Navigate" hace que el navegador navegue a la URL especificada. El efecto es equivalente a un humano escribiendo la URL en la barra de direcciones del navegador. El número expresa el número de páginas que debe esperar a que estén completamente cargadas antes de continuar con la ejecución de la secuencia.

El comando "FindFormByName" busca el primer formulario de la página HTML con el atributo "name" dado. La posición se expresa con el número, siendo el primero el 0.

Entonces, los comandos "SetInputValue (nombreCampo, posicion, valor)" son usados para asignar valores a los campos del formulario. Los valores pueden ser cadenas de texto (encerradas en "") o variables (precedida por '@'). Las variables son sustituidas por su valor en tiempo de ejecución.

Una vez rellenado el formulario, "ClickOnElement" equivale a hacer click en el elemento de tipo "input" y nombre "validar". El número expresa el número de páginas que debe esperar a que estén completamente cargadas antes de continuar con la ejecución de la secuencia. Es importante destacar que no se construye una petición HTTP para enviar el formulario, si no que se delega esa tarea al navegador. De este modo no es necesario tener en cuenta mecanismos habituales como números de sesión ocultos, formularios Javascript, etc. Todo esto es transparente para el sistema, igual que lo es para un humano que use su navegador.

Con esta parte de la secuencia el wrapper ha entrado en la banca electrónica como si fuese un usuario de la misma. El resto de la secuencia sirve para llegar hasta el punto justo donde está la información que precisamos.

El comando "FindFrameByName" busca el primer frame dentro del frame actual de la página HTML con el atributo "name" dado. La posición se expresa con el número, siendo el primero el 0. El boleano indica si exigimos que el nombre dado sea exacto al que figura en el atributo "name" o sirve con que esté contenido.

Dentro del frame, el comando "ClickOnAnchorByText" sigue el primer enlace cuyo texto visible sea el indicado. La posición se expresa con el número, siendo el primero el 0. El boleano indica si exigimos que el texto dado sea exacto al que se muestra en la página o sirve con que esté contenido. El número final expresa el número de páginas que debe esperar a que estén completamente cargadas antes de continuar con la ejecución de la secuencia.

Finalmente el comando "FindFrameByName" se posiciona en el frame que contiene el listado de cuentas del usuario.

En resumen, mediante lenguaje NSQEL se construye una macro que permite al wrapper navegar hasta posicionarse en la página donde se encuentra la información personal del usuario.

Una vez se ha construida la secuencia de navegación del wrapper, el siguiente paso es escribir una especificación que le indique cómo debe extraer tuplas de información de la páginas HTML devueltas por la fuente.

Para extraer la información requerida de las páginas HTML (u otro lenguaje de marcas), la herramienta emplea un lenguaje de especificaciones con el que se generan gramáticas especializadas. Este lenguaje, denominado DEXTL, hace uso de una serie de heurísticas en la presentación de los datos de tal forma que el resultado final es un lenguaje más simple que otros lenguajes similares y sin que por ello se pierda potencia. Además, la herramienta proporciona ayudas gráficas para construir las especificaciones de los wrappers visualmente, a través de un proceso interactivo. Esto permite que los wrappers sean generados por personal sin conocimientos de programación.

Veamos en un ejemplo, cómo es este lenguaje.

Ejemplo 2: Situémonos en el ejemplo anterior. Una vez ha sido generada la secuencia de navegación, el wrapper sabrá cómo llegar a la página donde se encuentra el listado de cuentas del usuario. Ahora debemos generar una especificación que indique al warpper cómo extraer las tuplas de las páginas HTML devueltas por la fuente.

En la Fig. 3 se muestra un fragmento de la página HTML devuelta por una banca electrónica.

Nuestro propósito, recordemos, es modelar la banca electrónica como una relación CUENTA cuyos atributos sean NUMEROCUENTA, DESCRIPCION,

CUENTAS A LA VISTA		
NºCuenta/1er Titular	Modalidad	Saldo
1234 1234 12 1234567890 / Juan Pérez	Cuenta corriente	94.540:Pts (568,20:Eur)
1234 1234 12 1234567890 / Juan Pérez	Cuenta ahorro	3506,00:Eur
1234 1234 12 1234567890 / Juan Pérez	Cuenta vivienda	6900,00:Eur

Figura 3: Fragmento HTML de una banca electrónica

```
PATTERN R
{
FROM
 "N°Cuenta/1er Titular" TABULATOR
 "Modalidad" TABULATOR
 "Saldo" ENDOFLINE
ENDFROM

 ANCHOR :NUMEROCUENTA ENDANCHOR
 IRRELEVANT TABULATOR
 :DESCRIPCION TABULATOR
 :SALDO ":" :DIVISA
 ¿"(" IRRELEVANT ")"? ENDOFLINE
}
```

Figura 4: Especificación del wrapper

SALDO y DIVISA. En la Fig. 4 se muestra una especificación válida.

La idea básica es construir un patrón del que el wrapper buscará coincidencias dentro del documento. El patrón consistirá principalmente en un conjunto de atributos de texto a extraer (en nuestro ejemplo NUMEROCUENTA, DESCRIPCION, SALDO y DIVISA), porciones de texto que no son relevantes (denotados como IRRELEVANT) y separadores.

Los separadores entre atributos pueden ser de tipo texto ("/", ":", etc.) o de tipo "tag". Un separador de tipo tag, ENDOFLINE, representa una expresión regular concerniente a información de formato del documento (usualmente tags HTML).

Los separadores de tipo tag normalmente son definidos para ser reutilizados en múltiples especificaciones. Por ejemplo, ENDOFLINE se define a partir de un expresión regular del tipo ("
" | "</p>" | "</tr>" | "</td></tr>") y puede ser empleada en cualquier especificación que trate documentos HTML. Normalmente este tipo de separadores son definidos por un usuario más experto y agrupados en conjuntos reusables. Posteriormente, estos conjuntos son empleados por usuarios menos cualificados para generar wrappers para muchas fuentes.

Nuestra experiencia nos muestra que la mayoría de las aplicaciones necesitan un número muy reducido de separadores de tipo tag. Por ejemplo, el sistema propuesto emplea un único conjunto de separadores de tipo tag compuesto de los elementos ENDOFLINE, TABULATOR (que se corresponde con el tag HTML "<td>") y ANCHOR (que representa un enlace HTML). De todos modos, la posibilidad de definir nuevos separadores, garantiza que el sistema será capaz de resolver cualquier situación que se pueda producir.

También es posible delimitar partes opcionales de la especificación (encerrándolas entre los símbolos "¿" y "?") y especificar patrones alternativos dentro de la especificación (con el símbolo "|").

Para resolver ambigüedades en el patrón existen una serie de construcciones para limitar la región de búsqueda de un patrón. Las construcciones básicas son FROM y UNTIL, las cuales identifican patrones que delimitan el comienzo y el fin de la región relevante dentro del documento

El sistema también permite la inclusión de acciones (identificándolas con el símbolo "$" seguido del nombre de la acción) que ejecutan cierto código Java cada vez que se produce una coincidencia del patrón en el documento. Estas acciones pueden recuperar el valor de los atributos capturados e incluso modificarlos después de haberlos procesado.

Un ejemplo de acción es aquella que permite la inclusión de secuencias de navegación en los patrones para tratar situaciones en las que los atributos de las tuplas se encuentren distribuidos en diferentes páginas, o aquellos casos en los que se tienen varias páginas de respuesta.

2.3 Seguridad en las comunicaciones

En el apartado anterior se vio que, gracias al uso de MSIE, el Módulo Extractor se conecta a la fuente siguiendo exactamente el mismo proceso que un humano con la ayuda de un navegador. Esto garantiza que el nivel de seguridad con el que el Módulo Extractor accede a la fuente es el mismo que el que la propia fuente considera apropiado para sus clientes, normalmente será HTTPS.

Otro aspecto de seguridad a tener en cuenta es la comunicación entre el Módulo Agregador y el(los) Módulos Extractor(es). Dicha comunicación se lleva a cabo con Java RMI. Entre ambos se transmite la información de autenticación necesaria para iniciar la agregación (viaja del Módulo Agregador al Módulo Extractor), y también la información agregada en dicho proceso (viaja del Módulo Extractor al Módulo Agregador). Esta información se transmite por la red y por tanto puede ser interceptada, así que el sistema de agregación protege dicho intercambio mediante SSL (Secure Sockets Layer).

Además el sistema propuesto es capaz de trabajar en entornos de implantación donde existen firewalls y/o proxies entre los módulos y entre éstos e Internet.

3 Módulo Agregador

En el Módulo Agregador reside toda la lógica del servicio de agregación. Las principales responsabilidades de este módulo son:

- Gestionar los datos del servicio de agregación referentes a los usuarios y a las fuentes agregadas.

- Manejar el formato unificado bajo el cual el usuario del servicio verá toda su información personal. Dos enfoques a este respecto: "materializado" y "virtual".

- Dar soporte a distintos enfoques de agregación: "on-line" y "batch".

- Garantizar la privacidad de los datos del servicio de agregación.

En esta sección, iremos desgranando cada una de la responsabilidades del módulo de agregación.

3.1 Modelo de datos del servicio

El servicio de agregación, para su funcionamiento, precisa de una serie de datos referentes a las fuentes agregadas y a los usuarios del propio servicio.

Estos datos se estructuran como una serie de relaciones de una base de datos relacional. Se han realizado implementaciones del sistema propuesto teniendo como soporte tanto bases de datos Oracle como DB2, aunque serviría cualquier base de datos con soporte JDBC.

La relación USUARIO recoge información relevante respecto a los usuarios del servicio de agregación. Es importante reseñar la situación en la que se encuentra el usuario de cara al servicio. Las tres situaciones manejadas son "activo" (el usuario está en condiciones de utilizar el servicio), "baja" (el usuario se ha dado de baja en el servicio y por tanto no puede hacer uso de él) o "bloqueado" (situación de baja temporal que puede ser debida a múltiples causas). Otro punto importante a reseñar es si el usuario permitirá que el sistema guarde la información necesaria para la autenticación en las distintas fuentes Web, o si por el contrario le será solicitada dicha información cada vez que se inicie el proceso de agregación.

La relación ENTIDAD recoge información relevante respecto a las fuentes agregadas.

La relación USUARIO_ENTIDAD mantiene las relaciones entre los usuarios y las fuentes que éstos agregan. Además, si el usuario lo autoriza, en esta relación se registra la información necesaria para la autenticación del usuario en dichas fuentes.

La relación INFORME_ACTUALIZACION registra, a modo de log, el resultado de todas las agregaciones que tengan lugar en el sistema. Esta información sirve tanto para labores administrativas como para informar al usuario del servicio del resultado de su agregación, previniéndole de los posibles errores producidos.

3.2 Enfoque materializado vs. virtual

Este apartado se centra en el tratamiento de la información agregada. En ella se pueden establecer dos familias: información financiera (cuentas, tarjetas, etc.); y el resto de información que englobaremos como no-financiera (contratos de servicios, facturas, mails, etc.).

Un reto de cualquier sistema de agregación es ofrecer una visión unificada de todo aquello que agrega independientemente del formato en que se encuentre en la fuente original. En el sistema propuesto, los responsables últimos de esta unificación de la información son los propios wrappers que exportarán idénticas relaciones en las distintas fuentes. Una misma fuente podrá exportar al mismo tiempo varias de éstas relaciones. Por ejemplo, una banca electrónica podrá exportar una relación CUENTA (con atributos NUMEROCUENTA, DESCRIPCION, SALDO, DIVISA, etc) y una relación TARJETA (con atributos NUMEROTARJETA, DESCRIPCION, SALDODISPUESTO, etc).

Las relaciones que exportan los wrappers se eligen de modo que cubran las necesidades del servicio de agregación y al mismo tiempo sean comunes a la mayoría de las fuentes agregadas. Por tanto habrá situaciones en las que el formato real en la fuente original no se corresponda directamente con la relación exportada. Para resolver estas situaciones se emplean la "acciones" comentadas en el apartado 2.2. Por ejemplo, el atributo NUMEROCUENTA de la relación CUENTA puede aparecer tal cual en una fuente, y en otra aparecer desglosado (como código de entidad, código de sucursal, dígitos de control y número de cuenta). En el segundo caso, se puede añadir a la especificación del wrapper correspondiente una acción que componga el NUMEROCUENTA a partir de las partes.

En definitiva, el sistema obtiene del Módulo Extractor la información en formato unificado. ¿Cómo se trata dicha información en el Módulo Agregador?. La respuesta admite dos enfoques: "materializado" y "virtual", que se refieren al soporte físico sobre el que se asienta la información agregada.

Con el enfoque materializado, la información personal, una vez extraída de la fuente, se almacena mediante JDBC en una base de datos relacional para

su posterior tratamiento. Por consiguiente, en cada agregación se actualiza la "foto" de la información personal que tenemos en la base de datos. Un servicio de agregación basado en este enfoque recuperará la información agregada directamente de la base de datos. Las relaciones definidas en la base de datos se corresponden con las que exportan los wrappers de las fuentes.

En el momento de escribir este artículo, las relaciones definidas en la base de datos para dar soporte a la agregación de información financiera incluyen cuentas, tarjetas y cuentas de valores, todos ellas con sus respectivos movimientos. Y para la información no-financiera se definen tres relaciones abstractas que dan cabida a todos los productos imaginables de esta categoría: contratos, facturas y sus respectivos movimientos.

La principal ventaja del enfoque materializado está en que, al guardarse copia de la información personal en base de datos, ésta puede presentarse inmediatamente al usuario del servicio aunque no esté totalmente actualizada (advirtiéndole de la circunstancia). La ventaja reside en que el proceso de agregación depende de terceros (las fuentes Web) que no puede controlar, y que en muchos casos son cambiantes e inestables. El enfoque materializado permite al usuario del servicio ver la "foto" de la información personal tal como se encontró la última vez que la fuente estuvo accesible. Además, y tal como veremos en el apartado siguiente, permite dos enfoques de agregación: "on-line" y "batch".

Con el enfoque virtual, la información personal no se almacena después de ser agregada. Esta información tendrá, pues, un tiempo de vida limitado a la duración de la sesión del usuario en el servicio de agregación. Con este enfoque se minimiza la importancia de las consideraciones legales y de seguridad que se derivan del almacenamiento de información personal, y que sin embargo afectan al enfoque materializado.

El formato elegido en el enfoque virtual para representar la información devuelta por los wrappers es XML por su gran aceptación y facilidad de integración con herramientas de terceros. Por cada agregación que se lance sobre una fuente Web se generará un documento XML, es decir, tendremos como máximo un documento XML por cada fuente agregada por el usuario del servicio. Como el enfoque es virtual, dichos documentos no se reflejan en ningún punto del sistema de ficheros, sino que residen en memoria hasta que el usuario abandone el servicio de agregación.

El principal inconveniente del enfoque virtual es que sólo permite seguir el enfoque "on-line" en las agregaciones, tal como veremos en el siguiente apartado.

Por último, destacar que ambos enfoques no son excluyentes. El Módulo Agregador puede emplear ambos al mismo tiempo y, por ejemplo, generar un fichero XML con toda (o una parte) de la información de la fuente y al mismo tiempo almacenar en base de datos toda (o una parte) de la misma.

Se han realizado implementaciónes del sistema propuesto siguiendo cada uno de los enfoques.

3.3 Agregación on-line vs. batch

Este apartado se centra en las dos estrategias de agregación, no excluyentes, que permite el sistema propuesto: "on-line" y "batch".

La estrategia de agregación "on-line" es la más sencilla. La agregación no se inicia hasta que el usuario del servicio la invoca, ya sea explícita o implícitamente. La invocación explícita se produce cuando el usuario, de modo consciente, inicia la agregación de una o varias de sus fuentes agregadas. La invocación implícita se produce cuando, por ejemplo, el propio sistema la inicia automáticamente cada vez que el usuario se conecta al servicio. Ambas formas son posibles en el sistema propuesto.

El proceso de agregación involucra llamadas a los wrappers para que accedan a la fuente remota y recuperen la información personal. Este proceso tiene un coste en tiempo no despreciable y que podemos considerar no-determinista, puesto que depende de la congestión de Internet y del estado de la propia fuente Web. Además está el coste de ancho de banda de la conexión a Internet.

El tipo de invocación (implícita o explícita) de la agregación on-line debe seleccionarse teniendo en cuenta el coste de agregación y el enfoque elegido para el soporte físico de la información agregada (materializado o virtual). Si se ha optado por el enfoque virtual, es lógico pensar que el usuario final, inmediatamente después de acceder al servicio, invocará la agregación de su información personal (puesto que carece de ella). En esta situación puede parecer útil la invocación implícita, y decimos "puede" porque dependiendo del coste de la agregación y del uso que los usuarios hagan del servicio, puede ser más eficiente que el usuario invoque la agregación sólo de aquellas fuentes que le interesen en cada momento.

En la estrategia de agregación "batch" es el sistema el que invoca las agregaciones periódicamente y siguiendo un criterio preestablecido, sin esperar a que el usuario acceda al servicio de agregación. Funcionalmente, esta estrategia está constituida por una serie de tareas que se ejecutan con una periodicidad establecida y que involucran a un conjunto de entradas (definido con una consulta SQL) de la tabla USUARIO_ENTIDAD (ver apartado 3.1, Modelo de datos del servicio).

La complejidad radica en encontrar un criterio eficiente de selección de usuarios a agregar en "batch" . El sistema propuesto utiliza como criterio la periodicidad y grado de utilización del servicio por parte de los usuarios. Así , los usuarios que más utilizan el servicio son candidatos a una agregación "batch" cuya periodicidad se ajustará a la de cada uno de ellos.

La estrategia de agregación "batch" sólo tiene sentido cuando se ha optado por un enfoque materializado o mixto (parte materializada y parte virtual) para la información agregada. Puesto que con ella se persigue ahorrar tiempo al usuario final, de modo que cuando éste acceda al servicio se le pueda mostrar su información personal actualizada convenientemente, sin obligarle a invocar la agregación. Y esto sólo se puede lograr si existe un repositorio físico donde almacenar la información agregada en "batch".

Por último, la estrategia de agregación "batch", al invocar las agregaciones sin intervención alguna por parte del usuario, necesita que éste haya permitido que el sistema mantenga la información necesaria para su autenticación en las fuentes Web (ver apartado 3.1, Modelo de datos del servicio).

Ambas estrategias ("on-line" y "batch") se pueden combinar. Por ejemplo, se pueden agregar en modo batch aquellos usuarios que hagan uso intensivo del servicio (para mejorar las prestaciones que éste les ofrece), y permitir al mismo tiempo que cualquier usuario, en cualquier momento, pueda invocar la agregación "on-line".

3.4 Privacidad

El punto crítico del sistema de agregación en cuanto a la seguridad se encuentra en la información de autenticación del usuario en las distintas fuentes Web.

Como se ha visto, el sistema de agregación precisa esta información si se pretende seguir algún enfoque de tipo "batch" en las agregaciones (es decir, que éstas se realicen sin intervención alguna por parte del usuario) o simplemente por facilitar el uso del mismo a los usuarios de modo que no tengan que aportar en cada agregación la información de autenticación en las fuentes Web.

Garantizar la privacidad de esta información es crucial porque con ella, cualquiera podría suplantar la entidad del usuario en las fuentes Web. Si se tiene en cuenta que entre la fuentes agregadas se encuentran un buen número de entidades bancarias este aspecto se vuelve crítico.

La información de autenticación se almacena, para aquellos usuarios que así lo soliciten, en la relación USUARIO_ENTIDAD (ver apartado 3.1 Modelo de datos del servicio). Sin embargo, no toda la información que registra esta relación es información crítica, y el sistema propuesto hace esa distinción cuando la almacena físicamente. Es decir, la parte no crítica de la información se almacena junto con el resto de datos del servicio de agregación, y la parte crítica se almacena en el denominado "entorno aislado seguro".

El término "aislado" se debe a que el sistema permite aislarlo del resto de información del servicio. Implementando una interfaz Java, se puede proveer al sistema de la clase que servirá de enlace entre el Módulo Agregador y el entorno aislado seguro. El sistema propuesto provee una implementación de referencia que utiliza como soporte una base de datos relacional, pero se podría emplear cualquier otro repositorio: directorio LDAP, ficheros, etc.

El calificativo de "seguro" se debe al hecho de que, independientemente del almacenamiento físico, la información crítica se cifra antes de ser guardada. Es posible utilizar cualquier algoritmo de cifrado implementado por un provider JCE (Java Cryptography Extension), seleccionándolo a través de un fichero de configuración. En las implementaciones actuales del sistema propuesto se ha optado por el algoritmo Triple DES con claves de 128 bits.

3.5 Características adicionales

Algunas características no comentadas anteriormente son:

- Posibilidad de funcionamiento en modo Web Service.

- Soporte transaccional: para transferencias desde las cuentas bancarias agregadas, y en general para cualquier procesamiento automático sobre las fuentes Web.

- Módulo Autologin: permite automatizar la entrada del usuario en la fuente. Por ejemplo, en la agregación de Webmail el sistema se limita a extraer las cabeceras de los correos. Si el usuario desea ver el contenido de un correo en particular el sistema utiliza este mecanismo para automatizar la entrada del usuario en la fuente.

4 Experiencia

La Tabla 1 muestra un resumen de los contenidos y fuentes incluidos actualmente en el sistema propuesto. En la selección, tanto de los dominios como de las fuentes agregadas, ha primado el tratar de cubrir la oferta actual de servicios que se ofrecen en Internet. El sistema, no obstante, admite la inclusión de nuevos dominios o nuevas fuentes.

El sistema propuesto ha servido como base para la implementación de servicios reales de agregación de contenidos que se prestan actualmente desde dos importantes entidades bancarias.

Tabla 1: Resumen de contenidos y fuentes

Dominio		Nº Fuentes Web
información financiera	Cuentas	13
	Tarjetas	11
	Cuentas de Valores	10
información no financiera	Empresas de Servicios	3
	Fidelización	2
	Webmail	12

5 Trabajo Futuro

Las principales líneas de trabajo futuro son:

- Dentro del sistema de generación de wrappers se está trabajando en técnicas de mantenimiento automático que permitan a los wrappers adaptarse a los cambios que se producen en las fuentes Web.

- Completar el servicio con un módulo de alertas al usuario empleando distintas vías (e-mail y SMS principalmente).

- Integración con herramientas de terceros para ofrecer capacidades de análisis financiero a los usuarios del servicio.

- Desarrollo de una versión cliente que funcione desde el ordenador del usuario eliminando la necesidad de un servidor central.

6 Trabajos Relacionados

En los últimos años han surgido diversos sistemas de wrappers basados en el marco de los mediadores ([1], [3] y [4] son referencias básicas para entender el modelo de mediadores). Algunos de estos sistemas de wrappers más relevantes son Ariadne [5], Wargo [6] o Lixto[7].

Estos sistemas han servido para construir gran número de aplicaciones de agregación en Internet en dominios particulares. Especialmente popular ha sido su aplicación a la compra comparativa con sistemas como Mysimon [8], Buscaproductos [9] o el desaparecido Jango (basado en el clásico sistema Shopbot [10]).

Por último, han surgido en la industria sistemas de propósito más general similares al sistema propuesto, y que por tanto, permiten la agregación de un mayor número de contenidos e incluyen soporte para fuentes que requieren autenticación. Uno de estos sistemas, Yodlee [11], tiene el inconveniente de que funciona exclusivamente en modo ASP (Application Service Provider) con una oferta de contenidos determinada. Otros sistemas aparecidos con posterioridad al aquí presentado son Getsee [12] o Tradence [13]. Respecto a éstos, el sistema propuesto tiene una serie de características distintivas como son el uso de herramientas de generación semi-automática de wrappers, flexibilidad en la selección de enfoques de actualización y materialización, y soporte completo de seguridad que abarca datos y comunicaciones.

Referencias

[1] G.Wiederhold. “Mediators in the architecture of future information systems”. Computer, 25(3), March 1992.

[2] Serge Abiteboul. “Querying semi-structured data.” In Proceedings of the International Conference on Database Theory(ICDT), 1997.

[3] Daniela Florescu, Alon Levy, Albert Mendelzon. “Database Techniques for the World-Wide Web: A Survey”. In SIGMOD Record vol.27, nº3. 1998.

[4] H.García Molina, Y.Papakonstantinou, D.Quass, A.Rajaraman, Y.Sagiv, J.Ullman, J.Widow. “The TSIMMIS project: Integration of heterogeneous information sources”. March 1997.

[5] C.A.Knoblock, K.Lerman, S.Minton, I.Muslea. “Accurately and Reliably Extracting Data from the Web: A Machine Learning Approach”. In Bulletin of the IEEE Computer Society Technical Committee on Data Engineering. 1999.

[6] Alberto Pan, Juan Raposo, Manuel Álvarez, Justo Hidalgo, Ángel Viña. “Semi Automatic Wrapper-Generation for Commercial Web Sources”. In Proceedings of IFIP WG8.1 EISIC. 2002.

[7] R.Baumgartner, S.Flesca, G.Gottlob. “Visual Web Information Extraction with Lixto”. In the Proceedings of the 27th International Conference on Very Large Data Bases (VLDB 2001), Rome, Italy, 2001.

[8] Mysimon. http://www.mysimon.com.

[9] Buscaproductos. http://www.buscaproductos.net.

[10] R.B.Doorenbos, O.Etzioni, D.S.Weld. “A Scalable Comparison-Shopping Agent for the World-Wide Web”. In W.L.Johnson and B.Hayes-Roth, (eds.), Proc. Proceedings of the First International Conference on Autonomous Agents (Agents’97), pp. 39-48, Marina del Rey, CA, USA, 1997.

[11] Yodlee. http://www.yodlee.com.

[12] Getsee. http://www.isoco.es.

[12] Tradence. http://www.tradence.com.

Marco de Trabajo de Componentes y Aspectos para el desarrollo de Entornos Virtuales Colaborativos*

M. Amor, L. Fuentes, D. Jiménez, M. Pinto
Departamento de Lenguajes y Ciencias de la Computación. Universidad de Málaga
Campus de Teatinos, s/n. 29071 Málaga (ESPAÑA)
{pinilla,lff,priego,pinto}@lcc.uma.es

Abstract. Current technological advances make possible the use of computers to collaborate and communicate with people who are geographically dispersed. Collaborative Virtual Environments integrate different collaborative applications in a shared space. In this paper we present a framework for deriving virtual environment applications, which has been developed applying advanced software technologies, such as component-based software development and aspect-oriented software development. The framework is implemented in Java and make use of the services provided by CoopTEL, our own component- and aspect-based distributed platform specially suited to develop this kind of applications with strong requirements of configuration, extensibility and adaptability.

1 Introducción

Los avances tecnológicos de los últimos años hacen posible el uso de los ordenadores para colaborar y comunicarse con personas que no se encuentran en la misma localización geográfica. Las empresas pueden beneficiarse de las posibilidades ofrecidas por nuevas herramientas como chats textuales, foros de discusión pizarras electrónicas, editores compartidos y vídeo conferencias, que permiten a un equipo de trabajo distribuido trabajar en grupos, formando equipos virtuales. Para que el trabajo de los miembros de un equipo virtual sea posible sin perder la noción de grupo, el sistema debe ofrecer un *espacio compartido* en el que se puedan ubicar herramientas colaborativas, documentos compartidos, etc. Los *entornos virtuales colaborativos* (EVCs) integran diversas aplicaciones de trabajo colaborativo (CSCW) en un *entorno compartido*. Las necesidades aquí presentadas han motivado un gran interés en el desarrollo de EVCs [1][2] en los últimos años.

Actualmente se está realizando un gran esfuerzo para mejorar el diseño de las aplicaciones colaborativas, pero aún no se dispone de una arquitectura de referencia estándar que ayude en el desarrollo de nuevas aplicaciones sin necesidad de desarrollarlas desde cero. Incluso existiendo algunos MTs colaborativos como es Sametime β], este tipo de MTs no ofrecen mucha más funcionalidad más allá de las herramientas colaborativas típicas, por lo que los programadores siguen necesitando programar la mayor parte de un entorno colaborativo. Esto significa que existe la necesidad de un marco de trabajo (MT) completo para el desarrollo de EVCs. Este MT debería encapsular una arquitectura de referencia para colaborar y navegar a través de espacios compartidos, proporcionando propiedades como presencia, persistencia, seguridad y control de acceso entre otras. Por otro lado, la extensibilidad de los entornos colaborativos actuales es insuficiente, en el sentido de que ofrecen un conjunto limitado de herramientas compartidas y colaborativas, por lo que no es posible añadir nuevas aplicaciones en tiempo de ejecución. Otra limitación de estos entornos es su grado de adaptación, que está normalmente restringido a un conjunto predefinido de preferencias de usuarios. Todas estas limitaciones se deben a que la mayoría de las propuestas existentes no usan tecnologías software avanzadas para conseguir una evolución adecuada de las aplicaciones finales.

Nosotros intentamos solucionar todos estos problemas aplicando de forma conjunta nuevos paradigmas de programación, como son el desarrollo de software basado en componentes (DSBC o en inglés CBSE, Component-Based Software Engineering) y el desarrollo de software orientado a aspectos (DSOA o en inglés AOSD, Aspect-Oriented Software Development). El DSBC [4] impone una nueva forma de desarrollar aplicaciones donde el objetivo principal es la reutilización y la composición de componentes, especialmente los comprados en el mercado de COTS (Commercial Off-The-Shelf). Sin embargo, las tecnologías de DSBC y las plataformas de componentes distribuidos por sí mismos no son suficientes para alcanzar una adecuada modularización del sistema en componentes independientes. Normalmente la misma propiedad ("concern") se encuentra presente en varios componentes creando dependencias no deseadas entre ellos. Estas dependencias dificultan la descomposición funcional de un sistema en componentes autónomos.

En este sentido, las técnicas avanzadas de separación de aspectos son ampliamente aceptadas como las más adecuadas para solucionar el problema de la dependencia entre componentes software, extendiendo el DSBC con nuevas dimensiones al

* Esta investigación ha sido financiada en parte por el proyecto CICYT TIC: 2002-04309-C02-02 y por la organización de telecomunicación "Fundación Auna".

margen de los "objetos" o los "componentes". La programación orientada a aspectos (AOP, Aspect-Oriented Programming) [5], ahora más comúnmente conocida como DSOA [6], es una disciplina muy prometedora, basada en el principio de separación de aspectos. El DSOA introduce una nueva dimensión llamada *aspecto* que modela aquellas características de un sistema que están presentes en múltiples componentes del mismo y pueden cambiar o evolucionar de forma independiente, resultando en sistemas software más desacoplados.

Aunque existen hoy en día propuestas que intentan aplicar el DSBC al trabajo en grupo ("groupware") [7], son pocas las que combinan las tecnologías de DSBC y DSOA para conseguir un MT de EVCs fácilmente configurable, extensible y adaptable. En este artículo, presentamos un MT basado en componentes y aspectos para derivar aplicaciones de EVCs. Este MT se ejecuta sobre CoopTEL, una plataforma distribuida que define la estructura de una aplicación como una colección de componentes y aspectos software, y un conjunto de reglas de composición entre ellos. La característica más relevante de CoopTEL es la independencia entre los componentes y los aspectos, lo que significa que ni los componentes ni los aspectos tienen información sobre cómo son compuestos. Esta información de composición se almacena en una capa middleware que soporta la composición dinámica de componentes y aspectos, especialmente apropiada para aplicaciones con grandes necesidades de adaptación como los EVCs. Nuestro MT ofrece además un conjunto de componentes software que proporcionan la funcionalidad básica de las aplicaciones colaborativas, y un conjunto de aspectos software reutilizables que modelan aquellas propiedades que están presentes en la mayoría de esos componentes.

Usando nuestro MT, un diseñador podrá construir nuevas aplicaciones colaborativas eligiendo los componentes y los aspectos apropiados y definiendo las reglas de composición entre ellos, con poco o ningún esfuerzo de programación. Dado que hemos usado técnicas de DSOA, la aplicación resultante es más modular y extensible que otras. Debido además a la composición dinámica proporcionada por CoopTEL, no proporcionada por otras plataformas de componentes como CORBA y J2EE, las aplicaciones son más adaptables, siendo posible configurar la estructura de una aplicación previamente instanciada, atendiendo a las preferencias o perfil del usuario. Este MT se ha utilizado para desarrollar distintas aplicaciones colaborativas, donde la más representativa es una oficina virtual llamada *Tracom* como parte de un proyecto de investigación. La organización de este artículo es la siguiente. Después de esta introducción, la Sección 2 presenta los componentes y los aspectos que forman parte de nuestro MT y las relaciones entre ellos. En esta sección se detallan una serie de requisitos de implementación que es necesario tener en cuenta para alcanzar un alto grado de reutilización, extensibilidad y adaptabilidad en el desarrollo de los EVCs. En la Sección 3 veremos como todos estos objetivos se obtienen directamente de la utilización de CoopTEL, nuestra plataforma distribuida para el desarrollo de aplicaciones basadas en componentes y aspectos. Veremos las características más importantes de CoopTEL y los servicios que ofrece. En la Sección 4 veremos una aplicación de oficina virtual implementada sobre CoopTEL utilizando el MT descrito anteriormente. Terminaremos con las conclusiones de nuestro trabajo.

2 Marco de Trabajo de EVCs

Nuestro objetivo en esta sección es presentar los componentes y los aspectos que hemos identificado en nuestro MT para derivar EVCs, que proporcionan el comportamiento básico de un EVC, y qué relaciones existen entre ellos. En la Figura 1 los componentes presentan un fondo blanco y los aspectos tienen fondo de color para distinguirlos.

2.1 Componentes básicos en un EVC

En esta sección discutiremos los componentes básicos de nuestro MT. Durante los últimos años se han desarrollado gran cantidad de aplicaciones colaborativas que facilitan la comunicación entre personas que no se encuentran en la misma localización. El problema principal de estas aplicaciones es la falta de *integración* entre ellas. Los EVCs tratan de resolver esta falta de integración proporcionando un *entorno compartido* donde todos los recursos (usuarios, herramientas colaborativas, documentos, etc.) involucrados en la colaboración sean fácilmente accesibles y configurables. El componente Entorno de la Figura 1 representa el "lugar" de entrada a un entorno compartido. Este componente modela la interfaz gráfica, mediante la cual el usuario interacciona con el resto de los recursos del entorno y puede configurar y adaptar su funcionalidad en tiempo de ejecución.

En general, un Entorno estará formado por varios espacios de colaboración, modelados por el componente EspacioColaboracion de la Figura 1. Dependiendo de la aplicación, este componente modelará espacios tan diversos como *habitaciones* en el caso de construir *oficinas virtuales* o *salas de exposición* si se estuviera construyendo un *museo virtual*. Los usuarios podrán navegar de un espacio de colaboración a otro y colaborar con los usuarios localizados en el mismo espacio, usando las herramientas de colaboración disponibles en él. Por tanto, cada componente EspacioColaboración contendrá una serie de recursos como documentos (Documento), herramientas individuales (HIndividual) y colaborativas (HColaborativa). La Figura 1 muestra todos estos componentes y las relaciones entre ellos. Un ejemplo de una herramienta individual es el componente NavegadorDocumentos que permite navegar por los documentos disponibles en un espacio de colaboración. También tendremos componentes que modelan cada una de las aplicaciones colaborativas Chat, Pizarra, Nota, ComparticionAplicaciones, etc.

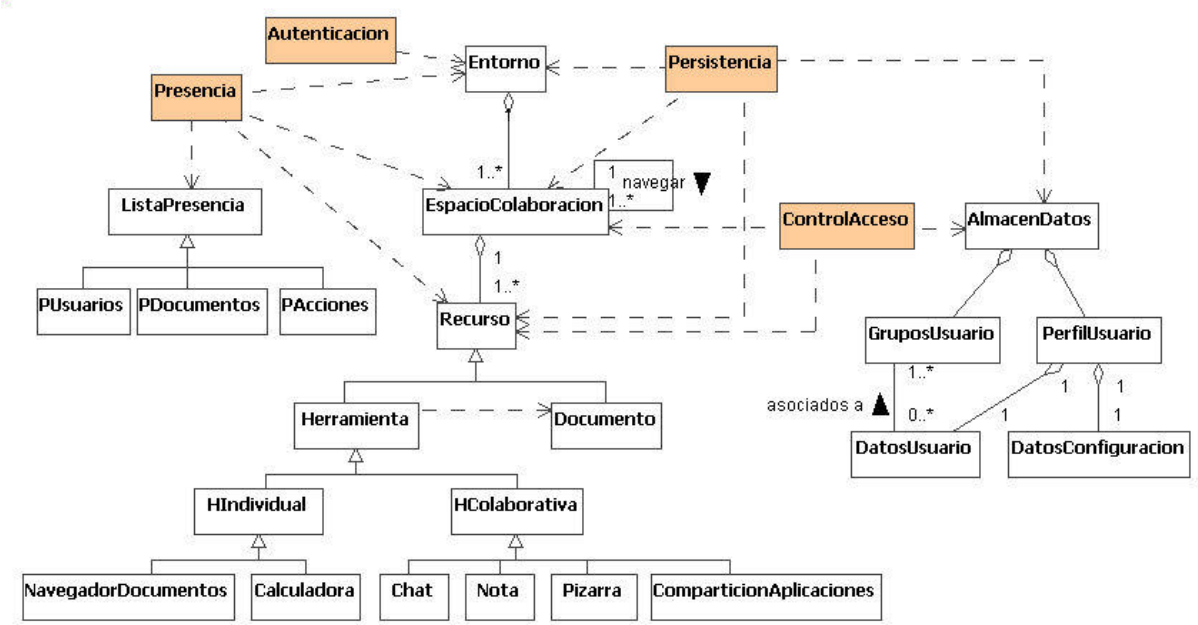


Figura 1. Marco de Trabajo para derivar EVCs

El comportamiento de una aplicación de EVC está claramente dirigido por las acciones y las decisiones que los usuarios toman tras conectarse al entorno. Esto hace patente la importancia de modelar los usuarios de forma adecuada en nuestro MT. Para hacerlo, distinguimos entre el comportamiento de los usuarios en el sistema y la información sobre los usuarios de la que el entorno debe disponer. Básicamente, los usuarios de un EVC interaccionan con el entorno a través de la interfaz gráfica de los diversos componentes proporcionados en el MT (espacio virtual, herramientas colaborativas, editores de documentos, ...), por lo que no es necesario modelar de forma explícita el usuario mediante un componente. Sin embargo, no se puede decir lo mismo sobre la información que el entorno necesita manejar sobre los usuarios. Esta información hace factible que los usuarios se localicen para comunicarse y colaborar y permite la configuración del entorno de acuerdo a las preferencias o el perfil de los usuarios. Esta información de los usuarios, junto a cualquier otra información que el entorno necesite almacenar de forma persistente se guardará en el componente AlmacenDatos de nuestro MT. Los usuarios en un entorno se agruparán en *grupos* de usuarios definiendo cada grupo un tipo de usuario distinto. Además, cada usuario tendrá asociado un *perfil de usuario* con sus datos personales y preferencias de configuración del entorno.

Por último, otros componentes importantes en un EVC son los componentes que proporcionan información de presencia (ListaPresencia) sobre el entorno. Estos componentes proporcionarán información de distinta naturaleza, como la localización y el estado de los usuarios conectados al entorno, información sobre la utilización de forma compartida de los documentos o información sobre las acciones que los distintos usuarios realizan sobre los recursos del entorno.

2.2 Aspectos Básicos en un EVC

Algunos inconvenientes derivados del uso de las aplicaciones colaborativas de forma independiente son: i) los usuarios deben memorizar un número elevado de identificadores de usuario y palabras de paso y deben conectarse de forma separada a cada aplicación; ii) cada aplicación se configura de acuerdo a un conjunto distinto de preferencias de usuario; iii) las listas de presencia, que muestran los usuarios con los que es posible colaborar, son propias de cada aplicación colaborativa.

El origen de estos inconvenientes está en que todas estas aplicaciones colaborativas comparten un conjunto de características como son la *autenticación* de los usuarios antes de que puedan iniciar la aplicación, la *configuración* de las propiedades de la aplicación de acuerdo a un conjunto de *preferencias* de usuario, o la necesidad de mostrar información de *presencia* sobre los usuarios conectados. Aún cuando integremos estas herramientas en un EVC, deben seguir cumpliendo estas características. Por tanto, podemos decir que la *persistencia*, *autenticación*, *control de acceso* y *presencia* son propiedades típicas de un EVC que afectan a la mayoría de los recursos contenidos en él. Siguiendo la filosofía de desarrollo basado en aspectos, estas propiedades serían *aspectos* software. Los aspectos se desarrollan de forma que sean lo más independientes posible de los componentes a los que afectan. Posteriormente son compuestos con los componentes para construir la aplicación final. Esta composición se puede realizar de forma estática durante la compilación, o de forma dinámica en tiempo de ejecución.

Aspectos típicos de cualquier sistema distribuido son *sincronización*, *comunicación*, *coordinación* o *manejo de fallos*. Además, existirá también otro conjunto de aspectos que serán dependientes del dominio de aplicación. En el MT de la Figura 1 se

muestran los aspectos más importantes detectados en el dominio de los EVCs que se corresponden con las propiedades discutidas anteriormente. Se representan mediante relaciones de dependencia indicando los componentes a los que afectan.

Aspecto de Autenticación. La seguridad es un aspecto muy importante en cualquier aplicación distribuida. Dos aspectos relacionados con la seguridad son el de autenticación y el de control de acceso, descrito a continuación. En nuestro MT el aspecto de autenticación se aplica al componente Entorno. Esto significa que el usuario tendrá que "acreditarse" para entrar al entorno virtual, pero no cada vez que acceda a un recurso del entorno. Las ventajas principales de modelar la autenticación como un aspecto son dos. La primera, que el desarrollador de los componentes del entorno no tiene que preocuparse para nada de cuestiones de seguridad, en las que probablemente no sea un experto. Será otro programador experto en cuestiones de seguridad el encargado de implementar esta funcionalidad. La segunda ventaja es que la forma de autenticar a un usuario en el entorno, podrá variar sin afectar para nada al resto de componentes y aspectos que formen parte de una aplicación. Estas variaciones podrán ir desde el medio utilizado para almacenar las credenciales de los usuarios (ej. una base de datos o un servidor de directorios LDAP [8]) hasta el tipo de información utilizada (ej. un par "usuario/palabra de paso" o una firma digital).

Aspecto de Control de Acceso. Otro aspecto relacionado con la seguridad es el de control de acceso. En nuestro MT los componentes podrían ser privados con uno o un conjunto limitado de usuarios. El aspecto de control de acceso se aplica una vez que el usuario está identificado y conectado al entorno para comprobar si tiene los permisos necesarios para interactuar con los recursos contenidos en él.

Aspecto de Persistencia. Otro aspecto importante es el aspecto de persistencia, encargado de almacenar y/o recuperar de un almacén de datos persistente toda la información relacionada con el entorno virtual y todos los recursos disponibles en él. En la Figura 1 podemos ver que el aspecto de persistencia afecta básicamente a todos los componentes del MT. Esto es así porque en una aplicación colaborativa y por tanto también en un entorno que integre un conjunto de herramientas y recursos para colaborar, es necesario almacenar información sobre el estado de los componentes. Esta información será recuperada cuando se conecta un usuario a una aplicación ya iniciada ("latecomers") para disponer del estado actualizado de la misma. Además, un EVC es en sí una aplicación con estado, lo que quiere decir que una vez creado el entorno, éste existe hasta el momento en que es explícitamente destruido, independientemente de que haya o no algún usuario conectado a él en un momento determinado.

Aspecto de Presencia. En una aplicación colaborativa, la colaboración no puede tener lugar si los usuarios conectados a ella no son capaces de localizarse entre sí y conocer si el resto de usuarios están disponibles o no para colaborar. Esta información la proporcionan los componentes de listas de presencia que hemos visto en la sección anterior. Sin embargo, estos componentes son meros contenedores o visualizadores de la información, donde la información de presencia se genera como resultado de la interacción de los usuarios con el resto de componentes del entorno. Ese es el motivo por el que es necesario un aspecto de presencia. Este aspecto se aplica básicamente sobre todos los componentes del MT y se encarga de recoger las modificaciones en el estado de estos componentes y notificarlas a los componentes de presencia de forma completamente transparente a los componentes generadores de la información. Por ejemplo, el aspecto de presencia será el encargado de indicar que un usuario se ha desplazado de un espacio de colaboración a otro sin que el componente EspacioColaboracion tenga que tener nada en su implementación sobre esta notificación.

Aspecto de Colaboración. El aspecto de colaboración o de difusión de mensajes ("broadcast") es un aspecto propio de cualquier aplicación distribuida, que permite la difusión de mensajes a un conjunto de componentes destino. En los entornos colaborativos este aspecto es muy útil para implementar los componentes colaborativos, porque permite desacoplar su funcionalidad de su comportamiento colaborativo, de forma que puedan ser utilizados de forma colaborativa componentes que no fueron implementados con esa finalidad.

Aspecto de Coordinación. Este es uno de los aspectos más importantes de nuestro MT, ya que permite desacoplar la funcionalidad de un componente de las interacciones en las que participa, encapsulando los protocolos de interacción en el aspecto. Las ventajas principales de separar coordinación y computación son: la primera, que el componente es más reutilizable en distintos contextos al no incluir información sobre las interacciones en las que participa; la segunda, que se puede modificar la interacción de un conjunto de componentes modificando en tiempo de ejecución el aspecto de coordinación sin afectar para nada a los componentes.

2.3 Requisitos de Implementación

Aunque la descomposición de la funcionalidad del sistema en componentes y aspectos incremente la reutilización y extensibilidad del sistema, nuestro objetivo es que el MT proporcione aún un mayor grado de adaptabilidad imponiendo una serie de requisitos en la implementación de los componentes y los aspectos del MT.

En primer lugar, los componentes y los aspectos deben ser completamente independientes entre sí. Para conseguir esta independencia, la información de qué aspectos son aplicados y cuando son aplicados no debe codificarse como parte ni de los componentes ni de los aspectos. Esta característica proporciona además la ventaja de que es posible modificar el tipo

y número de aspectos aplicables a un componente sin modificar ningún código. Uno de nuestros objetivos es que los aspectos puedan cambiarse no sólo en tiempo de diseño sino también durante la ejecución.

Otro de nuestros objetivos es poder añadir nuevos componentes y aspectos al MT, y también a una aplicación en ejecución, sin que se vean afectados para nada los componentes y los aspectos que modelan la funcionalidad básica del EVC. Por ejemplo, además de los aspectos mostrados en la Figura 1, otro aspecto propio de un EVC puede ser el de *cifrado*, que añade más seguridad al entorno cifrando la información intercambiada por los usuarios de forma transparente a los componentes que generan o consumen dicha información. Esta característica nos permitiría además añadir en tiempo de ejecución nuevos recursos a los espacios de colaboración, como por ejemplo, nuevas aplicaciones colaborativas no contempladas hasta el momento.

Por otro lado, la información de configuración de los componentes y los aspectos debería ser independiente de una implementación concreta. Para conseguir este objetivo esta información tendrá que estar almacenada en algún almacén de datos externo, de forma que los componentes y los aspectos la recuperen cuando sean instanciados. Una ventaja derivada de esta característica podría ser por ejemplo, disponer de distintas implementaciones del aspecto de control de acceso, donde una implementación simplemente consultará si un usuario tiene o no permisos para realizar determinada acción sobre un recurso sin dar ninguna interfaz de usuario, mientras que otra podría además dar la posibilidad de modificar estos permisos a un usuario autorizado. La información de control de acceso consultada por ambos componentes será la misma.

Para conseguir todos estos objetivos hemos implementado nuestro MT utilizando los servicios ofrecidos por CoopTEL, nuestra propia plataforma distribuida para el desarrollo de aplicaciones basadas en componentes y aspectos. La característica principal de CoopTEL, que lo hace muy adecuado para el desarrollo de nuestro MT, es que proporciona un grado de independencia muy alto entre los componentes y los aspectos en todas las fases del desarrollo de software, incluyendo la ejecución.

3 La Plataforma de Componentes y Aspectos CoopTEL

CoopTEL es una plataforma distribuida que proporciona un mecanismo de composición que permite incorporar la funcionalidad de los aspectos a los componentes de forma dinámica en tiempo de ejecución. CoopTEL está implementado en Java y usa Java/RMI para la comunicación distribuida. Una descripción completa de CoopTEL puede encontrarse en [9]. En esta sección resumimos las características y los servicios que nos ofrece para implementar nuestro MT cumpliendo los objetivos de implementación descritos en la sección anterior.

Los componentes y los aspectos existen en tiempo de ejecución. CoopTEL está basado en un modelo de composición donde los componentes y los aspectos son entidades de primer orden implementadas en el mismo lenguaje de propósito general. La composición de los componentes y los aspectos se establece durante la interacción y está gobernada por un conjunto de restricciones de composición que garantiza su correcta interoperabilidad.

La arquitectura software de una aplicación (AA) se almacena de forma explícita en la plataforma. En CoopTEL la AA se describe en términos de un conjunto de componentes, aspectos y conexiones entre ellos. La AA es definida por el arquitecto del software durante la fase de diseño y se almacena en un almacén de datos que en la implementación actual de CoopTEL es un servidor de directorios LDAP. Cuando un usuario se conecta a una aplicación se descarga esta información (escrita en XML) que define la composición dinámica entre los componentes y los aspectos. Esta es una característica muy relevante ya que la plataforma es capaz de interpretar directamente la AA especificada en UML y pasarla a XML, minimizando el "salto" entre el diseño y la implementación. Ninguna de las plataformas de componentes existentes (CORBA, J2EE, o .NET) ofrece una funcionalidad semejante.

Los componentes y los aspectos están identificados por un nombre de rol único. Además de la composición dinámica, otra característica importante de CoopTEL es que se desacoplan las interfaces de los componentes y los aspectos de sus clases de implementación. Además, no usamos ni las clases que definen la interfaz ni la implementación como identificadores de los componentes y los aspectos. En su lugar, asignamos un nombre de rol único para referenciar a los componentes (ej: "chat") y los aspectos (ej: "persistencia"). Tanto la creación de los componentes y los aspectos como su composición dinámica se realiza en base a estos nombres de rol.

La implementación de los componentes y los aspectos puede modificarse en tiempo de ejecución sin necesidad de recompilar. Esto es posible porque la información necesaria para crear y componer los componentes y los aspectos está descrita en la AA en base a su nombre del rol y no a implementaciones concretas. Esto es muy útil en los EVCs para modificar el comportamiento del sistema de acuerdo a las preferencias de usuario. Por ejemplo, se podría cambiar la forma de autenticar al usuario en el entorno cambiando la implementación del aspecto de autenticación.

Los componentes no tienen referencias directas entre ellos. Otra característica muy importante de CoopTEL es que las referencias entre componentes y aspectos no están codificadas como parte de su implementación. En su lugar, los componentes usan el nombre de rol para especificar el destino de un mensaje. De esta forma los componentes en nuestra plataforma están muy poco acoplados.

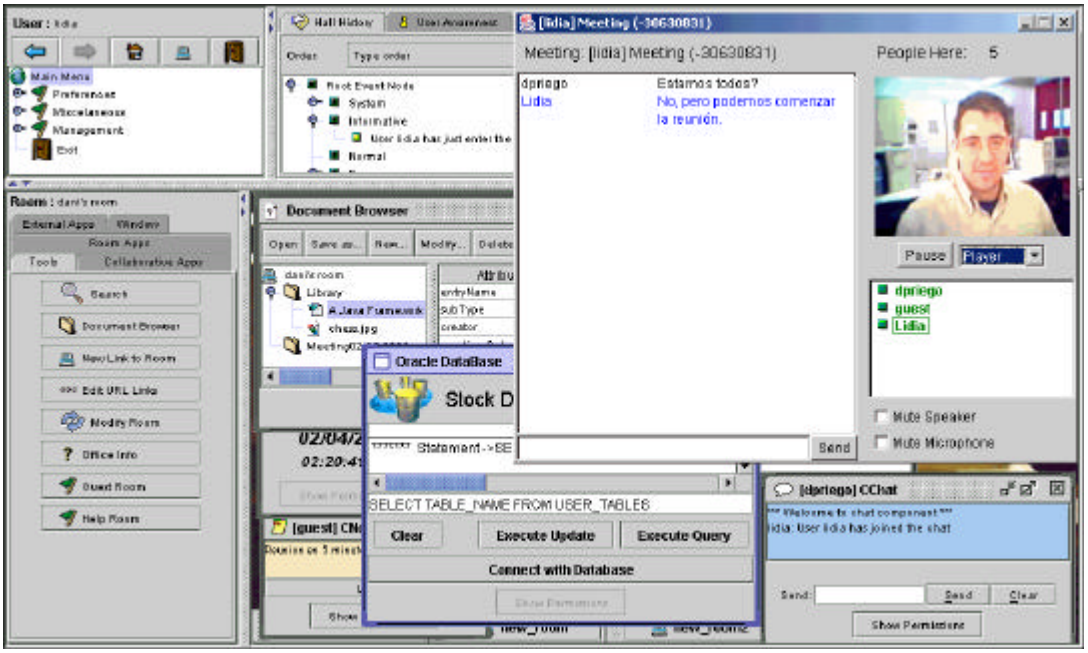


Figura 2. Oficina Virtual en ejecución.

Los componentes y los aspectos son independientes. Para conseguir la independencia entre los componentes y los aspectos, estos no tienen ninguna información sobre como son compuestos, proporcionando un mecanismo muy útil para su enlazado dinámico en tiempo de ejecución ("late binding"). Esta información está almacenada en la AA y se consulta en tiempo de ejecución durante la comunicación de los componentes. Esto significa que los componentes no tienen información sobre los aspectos que se les aplican y que el número y tipo de aspectos asociados a un componente puede variar de forma dinámica. Para cada componente y cada método dentro de un componente, la plataforma almacena la información sobre el tipo de aspectos que se aplican y en que orden.

Servicio de comunicación dinámica de componentes. CoopTEL define un conjunto de primitivas de comunicación para el envío de mensajes síncronos (*execmi()*) y asíncronos (*execute()*), difusión de mensajes (*broadcast()*) y emisión de eventos (*event()*). Estas primitivas tienen como parámetros el nombre de rol del componente destino, el nombre del mensaje enviado y sus argumentos. Utilizando el nombre de rol y consultando la AA la plataforma determina en tiempo de ejecución que instancia de un componente con ese nombre de rol debe recibir el mensaje haciéndoselo llegar. Esto desacopla la información de composición de las clases de implementación haciendo el sistema mucho más extensible y configurable.

Servicio de evaluación dinámica de aspectos. Los aspectos también los evalúa la plataforma de forma dinámica consultando la información de composición definida en la AA. Cuando un componente envía un mensaje se comprueba qué aspectos hay que aplicar a la salida y/o a la entrada de ese mensaje y se invoca el método *eval()* de dichos aspectos. Esto quiere decir que el único requisito que la plataforma impone a los aspectos es que implementen este método que encapsula el comportamiento del aspecto.

Servicio de almacén de datos. En la implementación actual de CoopTEL se utiliza el servidor de directorios LDAP como almacén de datos persistente. En el caso de los EVCs, los componentes y los aspectos pueden usar este servicio para almacenar toda su información de configuración. Por ejemplo, en [8] mostramos como implementar los aspectos de control de acceso y de persistencia haciendo uso de un directorio LDAP.

4 Ejecución de una Oficina Virtual en CoopTEL

Utilizando el MT definido en este artículo, hemos desarrollado una oficina virtual (OV). En la Figura 2 se muestra una imagen de la OV en ejecución donde pueden verse algunas aplicaciones típicas de la oficina, como son: un gestor de documentos, una aplicación de videoconferencia, un componente de presencia de usuarios, una aplicación de consulta a bases de datos, una nota o un chat. El elemento más importante que define la estructura de una OV es la habitación, que sirve para modelar la oficina. El componente habitación extiende la funcionalidad del componente EspacioColaboración de la Figura 1.

Las OVs presentan una estructura jerárquica que debe ser almacenada y recuperada de forma eficiente por el sistema. En la implementación de la OV, hemos optado por almacenar esta información utilizando el servicio de almacén de datos ofrecido por CoopTEL, implementado por un servidor de directorios LDAP. Las ventajas del servidor de directorios LDAP para almacenar la estructura de la oficina son las siguientes: i) la estructura jerárquica del servidor de directorios LDAP nos permite almacenar y recuperar la información sobre los recursos de la oficina de forma muy eficiente.

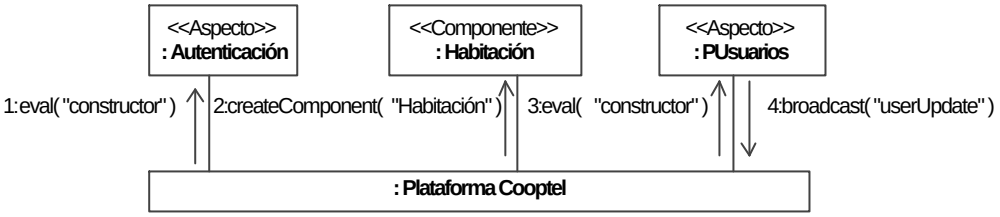


Figura 3. Entrada a la oficina.

Por ejemplo, si una entrada LDAP está asociada a una habitación de la oficina virtual, las entradas hijas guardarán información sobre los recursos contenidos en esa habitación, y ii) el servidor de directorios LDAP permite el almacenamiento de información sobre usuarios. Hemos aprovechado esta característica para almacenar información sobre los usuarios y sus preferencias en la oficina.

Durante el desarrollo de la oficina virtual hemos definido o adaptado una serie de componentes y aspectos del MT con el fin de dotar al sistema con una funcionalidad adecuada a sus fines. En especial hemos logrado modelar e integrar como componentes, aplicaciones desarrolladas por otras compañías como es el caso de las aplicaciones de pizarra compartida o vídeo conferencia de Sametime [3]. También hemos creado o adaptado una serie de aplicaciones generales, tanto individuales (un reloj digital o una herramienta para realizar consultas simples a bases de datos Oracle), como colaborativas (herramientas de votación) para su uso en la oficina. Además, hemos desarrollado un sistema de permisos genérico extendiendo el aspecto de control de acceso ControlAcceso del MT y se ha modelado una herramienta de compartición de documentos. Para finalizar esta sección, mostraremos dos escenarios de la OV donde se utilizan algunos de estos componentes y aspectos, mostrando de esta forma la versatilidad y flexibilidad del MT a la hora de modelar aplicaciones colaborativas complejas como es el caso de una OV.

Entrada a la oficina. La Figura 3 muestra el proceso de entrada de un usuario a la oficina. Cuando el usuario entra, se instancia un componente de tipo Habitación. Antes de la creación del componente (paso 1), la plataforma CoopTEL ejecuta el método *eval()* del aspecto de Autenticación para autenticar al usuario. Al evaluarse, este aspecto solicita al usuario que introduzca su identificador de usuario y contraseña. Este aspecto, se basa en el sistema de autenticación del directorio LDAP para verificar la identidad del usuario. Si los datos de identificación proporcionados son correctos, la ejecución de la oficina continuará y el componente Habitación será creado (paso 2). Finalmente, un aspecto de presencia de usuarios (PUsuarios) será aplicado como aspecto de salida del constructor del componente (paso 3) y distribuirá el mensaje *userUpdate* a través de la plataforma a todos los componentes de presencia de usuario de la oficina (paso 4).

Control de acceso a una aplicación de Chat y aplicación del aspecto de persistencia. En este escenario, mostraremos como se crean las aplicaciones en la oficina virtual y también el uso de los aspectos de control de acceso y de persistencia. Cuando creamos un componente de tipo Chat, como el mostrado por la Figura 4, vemos que antes de proceder a la creación del componente, se evalúa el aspecto de ControlAcceso (paso 1). Este aspecto, verificará si el usuario posee permisos de lectura y/o escritura sobre el recurso accedido. Si el usuario posee los permisos adecuados, la ejecución continuará con normalidad y el siguiente aspecto será evaluado secuencialmente. Si la evaluación falla, se elevará una excepción para notificarlo y se interrumpirá la creación del componente.

El siguiente aspecto a aplicar es el aspecto Persistencia (paso 2), cuya función es la de recuperar el estado del componente Chat desde el servidor de directorios LDAP. Una vez se crea y se inicializa el nuevo componente Chat (paso 3), este lanza el mensaje *userJoin* a todos los otros componentes de tipo Chat de la oficina, para notificar que el nuevo usuario se ha unido a la aplicación. Tras esta acción, un aspecto de presencia (PHistóricoEventos que extiende el aspecto PAcciones), se encargará de notificar a los componentes de presencia de acciones, que el usuario se ha unido al chat (paso 5), distribuyendo el mensaje *userUpdate*, que incluye además información sobre el estado y localización del usuario (paso 6). La aplicación del aspecto PHistóricoEventos, es totalmente opcional, pudiendo ser añadido o eliminado sin afectar la funcionalidad de la oficina virtual ni al componente Chat.

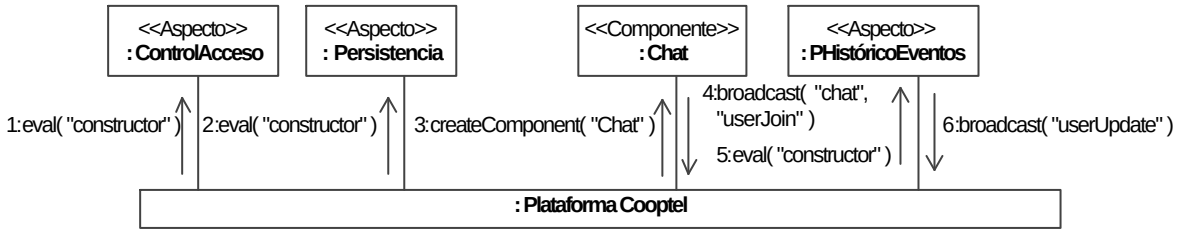


Figura 4. Iniciando un Chat.

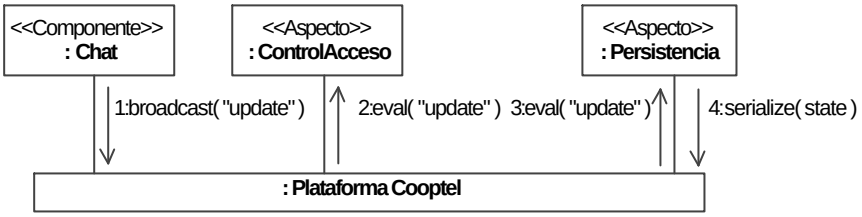


Figura 5. Escribiendo en un Chat.

Supongamos ahora que el usuario escribe un mensaje en el Chat (ver Figura 5). En este caso, el componente Chat enviará un mensaje a los componentes Chat de los otros usuarios para que actualicen su contenido (paso 1). Pero antes de emitir este mensaje, se aplica un aspecto de control de acceso (paso 2) para asegurar que el usuario tiene permiso para realizar esta acción. El aspecto comprobará si el usuario posee permiso de escritura sobre el recurso y en caso afirmativo continuará con el proceso de envío de los mensajes. Aplicamos el control de acceso en este punto ya que haciendo uso del mecanismo de composición dinámica ofrecido por la plataforma CoopTEL, nuestra aplicación permite modificar el control de acceso sobre un recurso en tiempo de ejecución. Si no queremos proporcionar esta funcionalidad, no habría necesidad de incluir este aspecto, ya que el aspecto de control de acceso aplicado a la entrada modifica el comportamiento del componente indicándole cuando un usuario está autorizado a realizar acciones sobre el mismo.

Para finalizar, se aplica un aspecto de persistencia (paso 3), que serializa y almacena en la plataforma el estado del Chat (paso 4) para que cualquier usuario que se incorpore al servicio posteriormente pueda recuperar las últimas líneas escritas.

5 Conclusiones y Trabajo Futuro

En este artículo hemos presentado un MT basado en componentes y aspectos para derivar aplicaciones de EVCs, evitando su implementación desde cero. Aprovechándonos de las ventajas proporcionadas por los aspectos, las aplicaciones resultantes son más modulares, y en consecuencia más reutilizables. La plataforma CoopTEL sobre la que se ejecutan las aplicaciones desarrolladas con nuestro MT proporciona servicios de composición dinámica. De esta forma, tanto los componentes como los aspectos pueden ser considerados componentes COTS, desarrollados de forma independiente. Añadiendo o eliminando componentes o aspectos en tiempo de ejecución, podemos construir aplicaciones altamente adaptables. Actualmente, hemos desarrollado una aplicación de oficina virtual, y después de tres meses de evaluación podemos establecer que su rendimiento es satisfactorio y la sobrecarga introducida por la composición dinámica no es crítica en el desarrollo de este tipo de aplicaciones. Por ejemplo, la plataforma tarda alrededor de 30 ms en la creación de los componentes y los aspectos, y la evaluación de los aspectos depende mucho del tipo de aspecto. Así, por ejemplo, la diferencia de tiempo entre evaluar el aspecto a través de la plataforma y hacerlo directamente es insignificante (unos 20 ms). Esta aplicación está siendo usada en nuestro departamento con gran éxito. Actualmente trabajamos en una segunda versión de CoopTEL donde combinamos composición estática y dinámica, de forma que mejoramos el rendimiento de las aplicaciones finales componiendo de forma dinámica sólo aquellos aspectos que requieren mayor adaptabilidad y modificación en tiempo de ejecución. Además, estamos desarrollando nuevas aplicaciones de entorno virtual usando nuestro MT para EVCs.

Referencias

[1] C. Gutwin, S. Greenberg. “The mechanics of collaboration: developing low cost usability evaluation methods for shared workspaces”, Actas de IEEE 9th International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2000.

[2] S. Sakai, y otros. “An integrated distance learning system capable of supporting interactions for asynchronous distance learning”, Actas del 22 International Conference on Distributed Computing Systems, 2002.

[3] “Real-time collaboration with Sametime”, ftp://ftp.lotus.com/pub/lotusweb/product/sametime/Real-timeCollab.pdf

[4] A.W. Brown, K.C. Wallnau. “The Current State of CBSE”, IEEE Software, Sept/Oct 1998.

[5] G. Kickzales y otros, “Aspect-Oriented Programming”, Actas de ECOOP’97, Jun 1997.

[6] “Aspect-Oriented Software Development Web Site”, http://www.aosd.net.

[7] R. Slagter, H. ter Hofte, O. Stiemerling. “The CSCW2000 workshop on Component-Based Groupware”. Actas de CBG2000 (part of CSCW), Filadelfia, Dic 2000.

[8] M. Amor, M. Pinto, L. Fuentes, J.M. Troya. “El Papel del Servicio de Directorio LDAP en Entornos Virtuales Colaborativos”, Actas de JITEL’01, Barcelona, Sept 2001.

[9] M. Pinto, L. Fuentes, M. Fayad, and J. M. Troya. Separation of Coordination in a Dynamic Aspect-Oriented Framework. Actas de AOSD’02, Holanda, Abr. 2002.

Hacia una Plataforma de Intermediación para Servicios Web en el Ámbito del Aprendizaje Electrónico

Judith S. Rodríguez Estévez, Luis Anido Rifón, Manuel J. Fernández Iglesias
Departamento de Ingeniería Telemática. Universidade de Vigo
E.T.S.I. Telecomunicaciones
Campus Universitario s/n, 36200 – Vigo (Pontevedra)
{jestevez, lanido, manolo}@det.uvigo.es

***Abstract**. The Web services concept has strongly emerged as a new distributed computing paradigm that attempts to make use of existing Internet technology overcoming the drawbacks of other distributed technologies. The E-learning field can take advantage of the new features offered by Web services: integration of heterogeneous applications, publicity of available services, etc. As the adoption of this new technology increases it will become necessary to offer intermediary platforms that make it easier to find and locate services, and compose new services from reuse. Additionally, it will also be necessary to provide added-value services such as auditing, security or management. This paper presents a proposal of a standard-based intermediary framework for Web services.*

1 Introducción

La estandarización de las tecnologías aplicadas al aprendizaje electrónico (*e-learning*) es un proceso que comenzó hace algunos años y que continúa muy activo actualmente. Este proceso está siendo liderado por las principales instituciones y organizaciones involucradas en el desarrollo de software educativo.

Los primeros esfuerzos de este proceso de estandarización estaban centrados en el desarrollo de modelos de información comunes de forma que dieran soporte a la reutilización e interoperabilidad en la capa de datos. El resultado más visible de estos primeros pasos es la definición de un conjunto de modelos de datos muchos de los cuales se han convertido en estándares *de-facto* entre la comunidad de desarrolladores de sistemas de e-learning [1].

Sin embargo, a pesar del incuestionable éxito de estos primeros resultados, el proceso de estandarización debe abordar aún problemas no resueltos, como el de la definición de interfaces comunes que proporcionen interoperabilidad en la capa de servicios. En este sentido, existen actualmente una serie de propuestas relevantes.

Este artículo presenta las propuestas más interesantes en la definición de interfaces y servicios comunes para las plataformas de aprendizaje electrónico (sección 2). La sección 3 analiza algunas de las carencias detectadas tras el análisis de dichas propuestas. La sección 4 está dedicada a la breve descripción del novedoso paradigma de los servicios Web. Asimismo, se analiza en qué modo puede este paradigma ayudar en la resolución de los problemas de interoperabilidad. La sección 5 está dedicada a la presentación de nuestro trabajo, una plataforma de intermediación para servicios Web. El artículo finaliza con la enumeración de una serie de conclusiones y de nuestro trabajo actual.

2 Estado del arte

En esta sección se presentan tres de las propuestas más interesantes en la definición de arquitecturas y servicios estándar para plataformas de aprendizaje electrónico.

2.1 OKI

OKI (*Open Knowledge Initiative*) [2], desarrollado en el Instituto Tecnológico de Massachussets (MIT), define una arquitectura y un conjunto de especificaciones que dan soporte al desarrollo de software educativo. Por un lado, la arquitectura propuesta, modular y por capas, proporciona una plataforma de desarrollo de aplicaciones educativas fácilmente extensible. Por otro lado, el conjunto de especificaciones definidas proporcionan un medio que ayuda a las instituciones a aprovechar la infraestructura existente en el momento de desarrollar las aplicaciones educativas.

Las especificaciones se han desarrollado en forma de Interfaces de Programación de Aplicaciones (API, *Application Program Interfaces*). Esta forma de actuación tiene varias ventajas. En primer lugar, se separa la definición del servicio de su implementación. En segundo lugar, las APIs hacen transparente los detalles de bajo nivel (e.g. protocolo de transporte) a los desarrolladores de las plataformas educativas. En tercer lugar, la funcionalidad de las APIs se puede adaptar fácilmente a la evolución tecnológica sin cambios sustanciales en las aplicaciones.

Dado que la arquitectura está organizada por capas, las APIs OKI también se pueden dividir en dos niveles. La capa inferior engloba una serie de

servicios básicos: autenticación, autorización, gestión de ficheros, conexión de bases de datos, etc. La capa superior define un conjunto de servicios estrictamente ligados a las aplicaciones educativas: gestión de cursos, gestión de contenidos, material de evaluación de conocimientos, etc.

Una primera aplicación de la filosofía propugnada por la iniciativa OKI es la plataforma dotLRN[1] (también conocida como .LRN). Ésta ofrece un marco de desarrollo de software libre y un conjunto de aplicaciones integradas que dan soporte a la gestión de cursos y de comunidades de aprendizaje.

2.2 SIF

La segunda propuesta interesante en la definición de arquitecturas y servicios es SIF (Schools Interoperability Framework) [3]. SIF ha sido concebido con objetivos diferentes a los de OKI. SIF está centrado en el desarrollo de una especificación abierta para lograr interoperabilidad entre las aplicaciones software de gestión y educativas existentes en las instituciones de educación primaria (K-12).

La especificación SIF está basada en XML (*eXtensible Markup Language*) [4] y no está ligada a ningún sistema operativo o plataforma en particular. Además, la definición de un conjunto de reglas para la comunicación y una arquitectura, SIF también define formatos de datos comunes para permitir el intercambio y la compartición de información entre las aplicaciones. De forma más concreta, los objetivos de SIF son los siguientes:

- Definir formatos estándar para los datos compartidos (e.g. información demográfica sobre los alumnos).
- Definir normas estándar para el nombrado de la información compartida.
- Definir las reglas de interacción entre las aplicaciones software.

Las dos propuestas presentadas anteriormente abarcan un amplio espectro de características: desde la definición de modelos de datos hasta la definición de interfaces de programación, aspectos relacionados con la interoperabilidad o definición de arquitecturas. Sin embargo, existen otras propuestas que siguen una metodología diferente a la hora de definir interfaces comunes o arquitecturas. En lugar de realizar la definición de arquitecturas partiendo de cero, toman en consideración los resultados previos tanto del proceso de estandarización de las tecnologías aplicadas al aprendizaje como de otros relacionados con el campo en cuestión. De esta forma realizan la estandarización de las interfaces de los componentes software que deben manejar esos modelos de datos estandarizados.

2.3. IMS

El ejemplo más claro de la segunda forma de actuar presentada anteriormente es la propuesta para la interoperabilidad de almacenes digitales [5] realizada por el consorcio IMS[2]. Esta especificación define las interfaces que deben implementar los componentes software que realizan la función de almacenes digitales permitiendo de esta forma que puedan interoperar. A diferencia de lo que ocurre con SIF no definen nuevos modelos de datos sino que toman en consideración especificaciones existentes sobre metadatos, empaquetado de contenidos, etc. Además, a la hora de su definición, IMS ha intentado dar cabida en su propuesta a los modelos más extendidos de almacenes digitales (e.g. Z39.50).

3 Análisis de las propuestas

Las propuestas presentadas en los apartados anteriores abordan dos problemas existentes en las plataformas de e-learning. Por un lado, la interoperabilidad se logra mediante la definición de modelos de datos e interfaces que las aplicaciones software deben utilizar e implementar, respectivamente. Por otro lado, las propuestas proporcionan herramientas y apoyo para el desarrollo rápido y fácil de plataformas de educación electrónica compatibles con cada una de las especificaciones. Esto es lo que algunos autores califican como los dos niveles de estandarización [6].

Sin embargo, con excepción de la propuesta de almacenes digitales de IMS, las otras dos propuestas asumen el hecho de que las instituciones en las que van a ser implantadas las plataformas de e-learning conformes con las especificaciones ya disponen de la infraestructura necesaria y/o el software propietario necesario para desarrollar completamente las plataformas. Tanto SIF como OKI proporcionan soporte para integrar dichas aplicaciones software.

Además, el hecho de crear entornos de aprendizaje en las que colaboren instituciones con plataformas compatibles con diferentes especificaciones (por ejemplo una con SIF y otra con OKI) parece complicado. De hecho, hasta el momento ninguna de las dos ha publicado definiciones de interfaces externas, es decir, hacia "afuera" de las instituciones. Este problema es doble. En primer lugar, la compartición e intercambio de datos de una plataforma a otra; por ejemplo, los usuarios registrados en una plataforma conforme con la especificación SIF no pueden exportar su perfil personal a una plataforma conforme con OKI debido a que ambas soportan diferentes modelos de datos. El mismo razonamiento se puede aplicar a la

[1] http://dotlrn.org/

[2] http://www.imsglobal.org

reutilización de contenidos educativos. No obstante, esta situación se puede solucionar si se define las correspondencias adecuadas entre diferentes modelos de datos.

En segundo lugar, existe un aspecto básico que no ha sido abordado aún por ninguna de estas propuestas: la reutilización e integración de componentes software de diferentes plataformas. Por ejemplo, el sistema de información de alumnos de una plataforma SIF no se puede comunicar de forma "directa" con el componente equivalente en una plataforma OKI.

En la misma línea, también parece interesante proponer un medio estandarizado de construir almacenes de los servicios educativos disponibles para ayudar a los desarrolladores en la implementación de nuevas plataformas o portales de educación electrónica.

4 Servicios Web

Las tecnologías agrupadas bajo la denominación de Servicios Web proporcionan los medios adecuados para solucionar algunos de los problemas enumerados en el apartado anterior.

Los servicios Web constituyen un conjunto de tecnologías y una arquitectura para computación distribuida compuesta por diferentes ordenadores que se intentan comunicar a través de la red de forma que conjuntamente se comportan como un solo ordenador. El campo de los servicios Web es uno de los campos más prolíficos de investigación y desarrollo en la actualidad, dado que representa la última evolución de la computación modular y distribuida.

El W3C[3] [7] define un servicio Web como: "*[Un servicio Web] es un sistema software identificado mediante un URI cuyas interfaces públicas y asociaciones están definidas utilizando XML. Su definición puede ser descubierta por otros sistemas software. Estos sistemas pueden interactuar con los servicios Web en la forma establecida en su definición, utilizando mensajes basados en XML transmitidos utilizando los protocolos de Internet*"

El trabajo actual en el campo de los servicios Web comprende la identificación y definición de un conjunto de estándares y protocolos para la gestión e implementación de servicios Web. Su propósito se puede resumir en dos grandes líneas. Por un lado, los nuevos estándares permitirán codificar semánticamente las propiedades, funcionalidad, interfaces, y efectos de los servicios Web en un formato inequívoco y "comprensible" por las máquinas, de forma que se automatice el descubrimiento, ejecución y composición de servicios Web. Por otro lado, el acuerdo en los protocolos y modelos de datos impulsará la implementación de servicios interoperables.

4.1 Tecnologías de los Servicios Web

Los estándares y especificaciones de los servicios Web se organizan en una estructura en capas. Las especificaciones básicas que permitirán la integración de aplicaciones software se describen brevemente a continuación:

- XML (*eXtensible Markup Language*) [4] es el lenguaje omnipresente para la creación y definición de diferentes modelos de información y el intercambio de información estructurada entre las aplicaciones software.

- SOAP (*Simple Object Access Protocol*) [8] es un protocolo para la comunicación, basada en mensajes, entre diferentes aplicaciones software. Está basado en XML y utiliza los protocolos de transporte convencionales de Internet (e.g. HTTP) para la transmisión de la información.

- WSDL (*Web Services Description Language*) [9], desarrollado por el W3C, es un formato basado en XML para la descripción abstracta de la funcionalidad (i.e. la interfaz) ofrecida por un servicio Web. WSDL separa la descripción del servicio de los detalles concretos de *cómo* y *dónde* se ofrece su funcionalidad.

- UDDI (*Universal Description Discovery and Integration*) [10] define un protocolo y un modelo de datos para la construcción y consulta de registros de servicios Web de forma que las aplicaciones software pueden descubrir qué servicios Web están disponibles y cómo acceder a su funcionalidad. La información proporcionada por los registros UDDI se puede agrupar conceptualmente en tres grupos: (1) *páginas blancas*, que contienen información de contacto de los proveedores del servicio; (2) *páginas amarillas*, que proporcionan una categorización de los servicios utilizando taxonomías estándar; (3) *páginas verdes*, que proporcionan documentación técnica sobre los servicios Web.

Aunque las especificaciones enumeradas anteriormente constituyen el núcleo tecnológico, existen varias especificaciones adicionales que se están convirtiendo en estándares ampliamente utilizados entre los desarrolladores/investigadores de los servicios Web. Entre ellas, las más importantes son aquellas relacionada con seguridad (WS-Security [11]), gestión de transacciones (WS-Transaction[12]) y la composición y organización de servicios Web compuestos (e.g. BPL4WS [13]).

[3] http://www.w3c.org

4.2 Mejora de los Servicios Web

La tecnología de los servicios Web proporciona el soporte adecuado para tratar de solucionar algunos de los problemas enumerados en la sección 3.

Por un lado, WSDL proporciona un lenguaje basado en XML para proporcionar información, de forma estandarizada, sobre las interfaces de los servicios Web de forma "comprensible" por las máquinas.

Además, UDDI proporcionar no sólo registros de servicios Web, sino también una forma estándar de descubrir servicios, localizarlos y acceder a su información técnica. De esta forma, las plataformas existentes pueden publicar y dar a conocer sus servicios haciéndolos accesibles al resto de la comunidad de desarrolladores. De este modo se soluciona el problema de la localización y registro de servicios.

Dado que los servicios pueden ser localizados fácilmente, el desarrollo de nuevas plataformas o portales de *e-learning* puede ser afrontado desde una perspectiva diferente. En lugar de desarrollar estas plataformas desde cero, los desarrolladores pueden hacer uso de los servicios Web disponibles proporcionados por empresas u organizaciones de confianza.

Este hecho no sólo fomenta la reutilización del software y los servicios existentes, sino que es también posible subsanar posibles carencias de infraestructura y/o tecnología en aquellas instituciones en las que las nuevas plataformas van a ser implantadas (p.e. carencia de software de autenticación, utilización de servicios externos)

Por otro lado, SOAP proporciona un protocolo de mensajería para la transmisión de información serializada descrita en XML sobre protocolos básicos de Internet. De esta forma la comunicación entre los diferentes componentes software pueden ser implementados de forma más sencilla y estandarizada.

5 Plataforma de intermediación

Aunque algunos de los problemas enumerados en la sección 2 podrían resolverse utilizando la tecnología de los servicios Web, es cierto que algunos de ellos aún permanecen y que incluso surgen algunos nuevos. Por ejemplo, la compartición de datos entre plataformas heterogéneas con distintos modelos subyacentes no se puede resolver de forma directa. También aparece la necesidad de herramientas adicionales y apoyo en la utilización de servicios Web: seguridad, transacciones, composición de servicios complejos, etc.

El propósito de nuestro trabajo es la definición de una arquitectura para plataformas de intermediación que ofrezca funcionalidad para que los usuarios o clientes de los servicios Web puedan entrar en contacto con los proveedores, realizando las gestiones de intermediación necesarias. La principal ventaja para los clientes (usuarios humanos o aplicaciones software) es que se les ofrece una plataforma a través de la que pueden encontrar, seleccionar y acceder a los servicios ofrecidos por los proveedores de forma eficiente.

Del mismo modo, los proveedores se pueden beneficiar de servicios como publicidad adaptada y personalizada, contabilidad, etc. Además, la plataforma intermediaria facilita la integración de aplicaciones software individuales proporcionando un *front-end* único tanto para clientes como para proveedores.

Nuestro trabajo tiene dos metas principales que se enumeran a continuación:

- Identificar una arquitectura genérica y guiada por estándares para plataformas de intermediación de servicios Web, teniendo en cuenta las recomendaciones genéricas para arquitecturas de servicios Web [7].

- Analizar, identificar y definir un conjunto de servicios comunes para plataformas de intermediación.

La Figura 1 muestra la primera versión de nuestra propuesta de arquitectura para plataformas de intermediación de servicios Web. Los diferentes componentes y su funcionalidad se enumeran en los siguientes apartados:

- *Broker*. Es el componente núcleo de la plataforma de intermediación. El *broker* implementa de forma global la lógica de negocio de la plataforma. Es el componente responsable de recibir las peticiones de los clientes/proveedores en forma de mensajes SOAP, procesarlos, realizar las peticiones y devolver los resultados correspondientes. El broker actúa como intermediario en todas las transacciones entre clientes y proveedores. De esta forma hace transparente a éstos las posibles transformaciones en los modelos de datos, la localización de los proveedores, o el flujo de mensajes que pueda desencadenarse como respuesta a una petición.

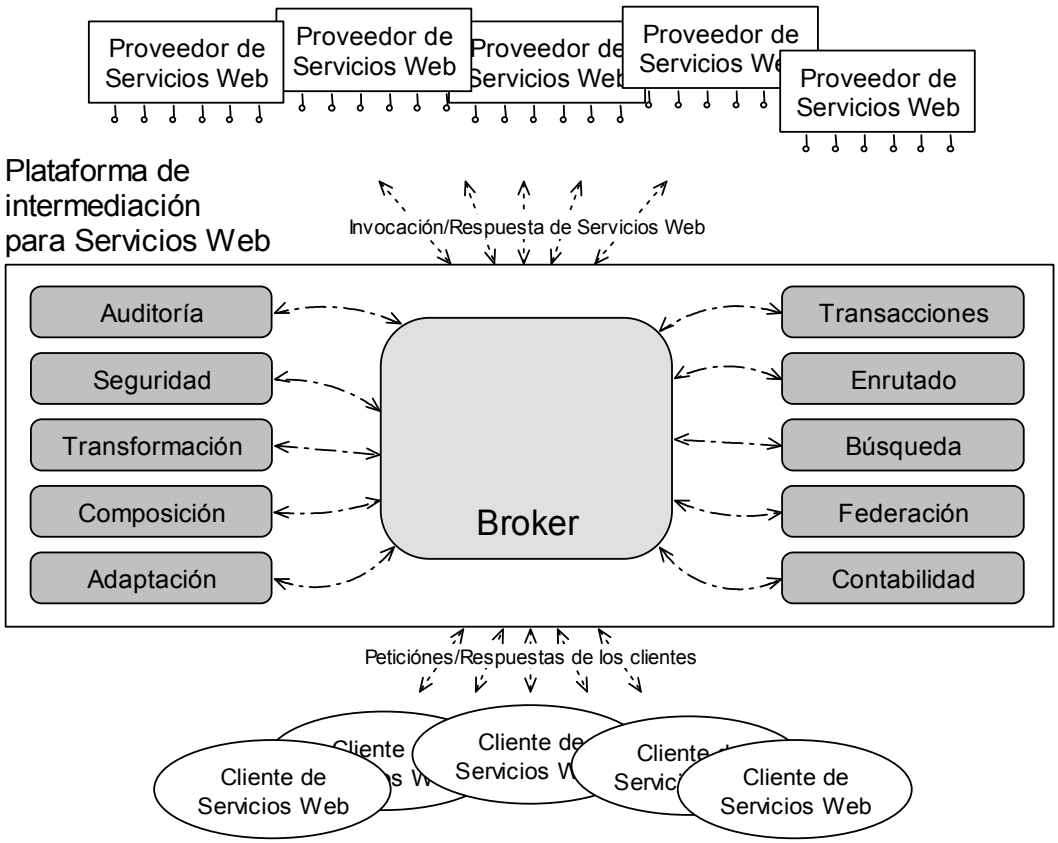


Figura 1: Propuesta de plataforma de intermediación

- *Contabilidad*: El componente de contabilidad gestiona la información necesaria para realizar la contabilidad y el posible cobro de servicios que no son gratuitos. En función de la información existente en el perfil de un proveedor de servicios, este componente también se puede encargar de invocar servicios externos de cobro en representación del proveedor, o bien notificar al proveedor adecuado de que inicie el proceso necesario para que un cliente pueda acceder a un servicio de su propiedad.

- *Auditoría*: El componente de auditoría proporciona servicios de auditoria sobre los servicios Web invocados a través de la plataforma. Este componente gestionará información como calificación de servicios por parte de los cliente, rendimiento, fiabilidad, disponibilidad, etc. Del mismo modo, puede proporcionar análisis estadísticos sobre los servicios Web y sus proveedores. Esta información puede ser de interés para los clientes, por ejemplo a la hora de tener que elegir entre servicios similares proporcionados por diferentes proveedores.

- *Composición de servicios complejos*: El componente de servicios complejos es el encargado de realizar la planificación y el flujo en la invocación de una serie de servicios "sencillos" de forma que conjuntamente den lugar a un solo servicio "complejo".

- *Adaptación*: El componente de adaptación tiene como principal responsabilidad la gestión de los perfiles de los usuarios (clientes y proveedores) y de la adaptación global de la plataforma al perfil correspondiente. Dentro de esta adaptación se incluyen modelos de datos, transformaciones en las peticiones, adaptación de las respuestas, interfaces gráficas adaptadas a preferencias, filtrado de respuestas en función de la capacidad de los usuarios, etc. En resumen, la información almacenada en el perfil guía las acciones que del broker cuando éste recibe las peticiones de los usuarios.

- *Federación*: El componente de federación permite la colaboración entre plataformas de intermediación de servicios Web. De este modo se consigue reenviar peticiones a otros intermediarios en aquellos casos en los que la plataforma receptora de una petición de usuario no es capaz de resolver dicha petición.

- *Enrutado*: El componente de enrutado es el responsable de la gestionar las rutas de los mensajes SOAP y aspectos adicionales relacionados con el enrutado de mensajes de petición y respuesta.

- *Búsqueda*: El componente de descubrimiento y búsqueda se ocupa de la búsqueda de cualquier tipo de información necesaria para cumplimentar las diferentes solicitudes: búsqueda de servicios Web adecuados, búsqueda de proveedores, etc. En esta tarea puede hacer uso de servicios externos como registros UDDI o buscadores. No obstante, también es posible que este componente ofrezca servicios de descubrimientos y búsqueda locales.

- *Seguridad*: Un sistema de seguridad adecuado es una de los aspectos claves para el éxito de la tecnología de los servicios Web. Este componente es el encargado de gestionar todos los aspectos relacionados con la seguridad: autenticación, integridad de los datos, gestión de certificados y claves, implementación de diferentes modelos de seguridad, etc.

- *Transacciones*: El componente de transacciones da soporte a la coordinación de transacciones en aquellos casos en los que la respuesta a una solicitud desencadena varias acciones que deben realizarse de forma atómica. Por ello este componente es responsable del mantenimiento del contexto de las transacciones, completar y descartar transacciones, etc.

- *Transformación*: El componente de transformación es el responsable de realizar las adaptaciones, correspondencias y/o traducciones entre los diferentes modelos de datos gestionados por la plataforma y utilizados por clientes y proveedores.

Además de la identificación de los diferentes componentes que conforman la arquitectura, se ha realizado un análisis detallado para llevar a cabo la definición de los servicios básicos que ofrece la plataforma tanto a clientes como a proveedores. Esta definición se ha realizado utilizando el lenguaje de descripción de servicios WSDL [9].

5.1 Caso de estudio: intermediación en el campo del aprendizaje electrónico

La arquitectura para plataformas de intermediación propuesta anteriormente es una arquitectura genérica que debe ser particularizada (clientes, proveedores y modelos de información subyacentes) para cada ámbito de utilización específico. En nuestro caso este ámbito es el de la educación electrónica.

De este modo, los proveedores serán aquellas aplicaciones software o incluso particulares que ofrezcan servicios de interés para el ámbito del e-learning. Los proveedores pueden ser de dos tipos: (1) proveedores de servicios básicos, como por ejemplo registros UDDI, servicios de seguridad o almacenes de perfiles de usuario. (2) proveedores de servicios educativos específicos: componentes software de entrega de contenidos y almacenes de contenidos digitales, software de evaluación o servicios de certificación de conocimientos.

En cualquier caso, cuando son aplicaciones software las que actúan como proveedores, el único requisito que deben cumplir es ofrecer la descripción de sus interfaces utilizando las tecnologías de los servicios Web y ser capaces de recibir y responder a mensajes SOAP.

Los clientes serán los desarrolladores de plataformas, portales o aplicaciones educativas que utilizarán los servicios de intermediación para la localización de servicios Web adecuados para la funcionalidad que pretenden ofrecer. De esta forma, podrán desarrollar más rápidamente dicho software, dado que podrán partir de servicios ya existentes.

Asimismo, los componentes o aplicaciones software implementados de esta forma, se podrán convertir a su vez en clientes de la plataforma: podrán utilizar la plataforma como intermediaria en las invocaciones a los servicios o en comunicaciones con otras aplicaciones beneficiándose de servicios adicionales como los de transformación y adaptación.

Además, al fijar el ámbito de aplicación también se establecen los modelos de datos de soporte. En nuestro caso, además de las especificaciones estrictamente relacionadas con los servicios Web, se han tenido en cuenta los resultados previos del proceso de estandarización de las tecnologías aplicadas al aprendizaje. Por esta razón, el soporte de información para metadatos, definición de estructuras de contenidos, empaquetado de información, etc. se realiza utilizando modelos de datos estandarizados.

No obstante, ha sido necesaria la identificación y definición de modelos de datos que den soporte a la funcionalidad que implementa la plataforma de intermediación. Básicamente esta tarea ha consistido en la identificación de la información de cabecera para los mensajes SOAP.

A continuación se muestra la descripción de un caso de uso con el que se pretende ejemplificar un escenario de uso de la plataforma de intermediación.

En el primer caso de uso (Figura 2) se representa un escenario en el que un cliente (por ejemplo un desarrollador de un portal) envía al broker una petición SOAP solicitando una búsqueda de proveedores que ofrezcan servicios sobre sistemas de información de alumnos.

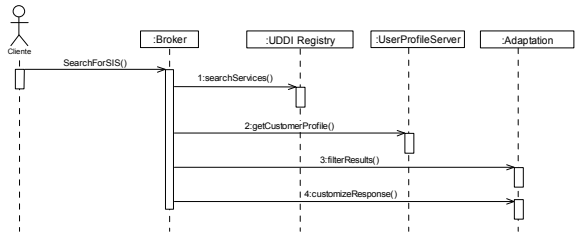


Figura 2: Búsqueda de servicios

Una vez que el *Broker* recibe la petición se desencadenan una serie de mensajes adicionales[4] (que son transparentes para el cliente). El mensaje etiquetado con el número 1 representa una petición de búsqueda sobre un registro UDDI. En el escenario descrito, suponemos que en la petición del cliente también se solicitaba la adaptación de los resultados en función de su perfil. Una vez que ha recibido los resultados, el *Broker* realiza envía un mensaje a un almacén de perfiles con el fin de acceder a las preferencias del cliente (mensaje 2). Posteriormente, el *Broker* realizaría una petición de filtrado (mensaje 3) de resultados al componente de adaptación (por ejemplo para eliminar aquellos servicios cuya utilización suponga un precio fuera del rango seleccionado por el cliente). Finalmente, el *Broker* solicita la adaptación de la respuesta final que se mostrará al cliente (mensaje 4): por ejemplo, el cliente tiene establecido en su perfil que solamente se muestre el nombre del servicio, su proveedor y las condiciones de utilización.

6 Conclusiones y trabajo actual

La interoperabilidad e integración de diferentes aplicaciones software heterogéneas en uno de los campos más activos de investigación y desarrollo actualmente. La identificación y definición de interfaces comunes y modelos de datos y arquitecturas es uno de los aspectos claves para lograr interoperabilidad total.

La tecnología de los servicios Web ofrece un conjunto de especificaciones y protocolos estandarizados que proporcionan soporte para afrontar los aspectos relacionados con la interoperabilidad desde un punto de vista diferente. En esta nueva situación se parte de una serie de especificaciones que tienen en común el hecho de que están siendo desarrolladas de forma conjunta por instituciones y empresas punteras, lo que asegura un amplio consenso desde su concepción. Además, estas nuevas tecnologías no requieren una infraestructura tecnológica subyacente diferente de la ya existente, lo que facilitará su rápida implantación

Sin embargo, hasta que las tecnologías de los servicios Web sean ampliamente adoptadas, parece necesario el desarrollo de una serie de plataformas intermediarias que ayuden a desarrolladores y usuarios finales sacar partido del nuevo "mercado" ofrecido por los servicios Web.

El trabajo actual que estamos desarrollando tiene los siguientes objetivos:

- Análisis adicional y refinamiento de la propuesta

- Enriquecimiento de la plataforma de intermediación con características y funcionalidad adicional de cara a los proveedores de servicios: *marketing*, implementación de otro tipo de políticas para la provisión de información.

- Análisis de la posible introducción de servicios Web "semánticos" siguiendo la filosofía de lo que se ha dado en llamar la Web Inteligente o Web Semántica propuesta por Tim Berners-Lee.

7 Agradecimientos

Queremos agradecer al "Ministerio de Ciencia y Tecnología" su apoyo parcial a este trabajo bajo el proyecto "CORBALearn: Interfaz de Dominio guiada por Estándares para Aprendizaje Electrónico" (TIC2001-3767)

8 Referencias

[1] J. Santos, M. Caeiro, J. Rodríguez, L. Anido. "Standardization in TelE-learning. A Critical Analysis". TelE-Learning. The Challenge for the Third Millenium. 17th IFIP World Computer Congress, Montreal (Canadá). Kluwer Academic Publishers, Agosto 2002, pp. 321-328.

[2] G. Collier and R. Robson. "What is the Open Knowledge InitiativeTM". Eduworks for OKI. Septiembre 2002.

[3] SIIA, "Schools Interoperability Framework Implementation Specification". Version 1.0. Software & Information Industry Association 1090 Vermont Ave., NW 6th Floor Washington, DC 20005. Agosto 2001.

[4] T. Bray, J. Paoli, C. Sperberg-McQueen and E. Maler. "Extensible Markup Language (XML) 1.0 (Second Edition)". W3C, Octubre 2000.

[5] K. Riley, M. McKell. "IMS Digital Repositories Interoperability - Core Functions Information Model". Version 1.0. IMS Global Learning Consortium. Enero 2003

[4] Las comunicaciones entre los diferentes componentes representan en realidad mensajes SOAP, aunque se han etiquetado "significativamente" para que sea más fácil su lectura

[6] Luis E. Anido-Rifón et al. “A Step ahead in E-learning Standardization: Building Learning Systems from Reusable and Interoperable Software Components”, Twelfth World Wide Web Conference, Honolulu, Hawai, EEUU. CDROM, Mayo 2002. Versión electrónica disponible en http://www2002.org/CDROM/alternate/136/

[7] M. Champion, C. Ferris, E. Newcomer and D. Orchard. “Web Services Architecture”. W3C, Noviembre 2002.

[8] M. Gudgin, M. Hadley, N. Mendelsohn, J. Moreau and H. F. Nielsen. “SOAP Version 1.2 Part 1: Messaging Framework”. W3C, Diciembre 2002.

[9] R. Chinnici, M. Gudgin, J. Moreau and S. Weerawarana. “Web Services Description Language (WSDL)”. W3C, Marzo 2003.

[10] T. Bellwood et al. “UDDI (Universal Description, Discovery & Integration). Version 3.0”. UDDI Coallition. July 2002

[11] B. Atkinson, G. Della-Libera, S. Hada, et al. Web Services Security (WS-Security). IBM, Microsoft, VeriSign, Inc., 2002.

[12] F. Cabrera, G. Copeland, B. Cox, et al. Web Services Transaction (WS-Transaction). BEA, IBM, Microsoft, 2002.

[13] Francisco Curbera et al., “Business Process Execution Language for Web Services”. Versión 1.0. BEA Systems, International Business Machines Corporation, Microsoft Corporation, Inc, 2002

Breve 1A: Aplicaciones Telemáticas

Estudio e implementación de una solución de videoconferencia en i2CAT en entornos público y privado con direccionamiento único

Marisol Hurtado, Jesús Alcober
Departamento de Ingeniería Telemática. Universidad Politécnica de Cataluña
Campus del Baix Llobregat. Av. del Canal Olímpic, s/n 08860 Castelldefels
Teléfono: 93 413 70 00 Fax: 93 413 70 07 E-mail: marisol.hurtado@upc.es, alcober@ieee.org

Abstract.. *We focus on one of the main challenges the videoconference has to be deployed in an exploitation IP network, the private and public addressing environment, which has to be transparent to the end user. We provide and compare two generic solutions, which have been deployed in the i2CAT project, since it has a private addressing scheme, and provides videoconferencing services to organizations connected through Internet. Both solutions are valid depending on the scenario they have to be deployed.*

La videoconferencia es una de las aplicaciones multimedia por excelencia. Sin embargo, el uso del estándar H.323, de videoconferencia sobre IP, aún no está extendido y su servicio no se proporciona de forma estable. Ello es debido básicamente a dos aspectos: la provisión de calidad de servicio en la red, por un lado, y el direccionamiento único para cada terminal H.323 existente, por otro. El direccionamiento único podría solventarse con la asignación, por parte de una entidad internacional, por ejemplo Videnet[1]., de una numeración única para cada terminal H.323 que corresponda con su dirección IP El problema se complica cuando las empresas e instituciones optan por utilizar un direccionamiento privado. Es en este momento donde aparece una problemática genérica, que en este artículo propondremos dos soluciones y que se ha implementado en i2CAT[2], proyecto financiado por la Generalitat de Catalunya, como consecuencia del crecimiento natural de la plataforma de videoconferencia. A efectos de direccionamiento, i2CAT es una red con direccionamiento privado cuyos usuarios se quieren comunicar a un entorno con direccionamiento público como Internet.

1 Planteamiento del problema

En líneas generales, el establecimiento de las comunicaciones entre entornos de red IP privado y público requiere de la utilización de traducción de direcciones privadas (NAT, network address translator). En este contexto se presentan dos problemas importantes. En primer lugar, los terminales de videoconferencia determinan dinámicamente los puertos a utilizar a nivel de transporte para mensajes de control de llamada y canales de datos de video y audio. En consecuencia, sería necesario abrir todos los puertos, a nivel de transporte, para dejar pasar el flujo proveniente de Internet hacia la red privada. En segundo lugar, las direcciones privadas no son enrutables en Internet, provocando que los terminales de una red privada no puedan recibir llamadas H.323. Este problema podría resolverse aplicando NAT. Sin embargo, H.323 introduce la dirección IP del terminal destino en los datos de carga de los paquetes H.225 y H.245; así, aunque el problema se resuelva a nivel de red mediante NAT, el problema persiste a nivel aplicación. Ante este problema se estudiaron dos posibles alternativas implementar gateways a nivel de aplicación ALG o proxies [1]. A continuación las comentaremos brevemente. Los gateways a nivel de aplicación (ALG) son dispositivos destinados a interpretar protocolos IP específicos, como es el caso de H.323, realizan un procesamiento minucioso con los datos de carga en los paquetes, de tal forma que son capaces de determinar los puertos que utilizará el terminal para recibir los datos. En consecuencia, el dispositivo abre solo los puertos necesarios por la aplicación y mantiene una lista de las sesiones activas establecidas mediante la técnica conocida como stateful, [2]. Los ALG resuelven el problema del intercambio de tráfico de video y audio entre redes privadas y públicas, pero necesitan además de un NAT para la traducción de direcciones IP privadas. Un proxy H.323 actúa como gateway. Su tarea es establecer conexiones a nivel de aplicación, interceptar el tráfico interno y redirigirlo hacia el exterior, finalmente establecer la llamada como si fueran dos llamadas separadas: una desde el terminal de la red privada al proxy y la segunda desde el proxy al terminal en la red pública. Los proxies requieren que las llamadas se establezcan a través de un gatekeeper interno y uno externo para enrutar adecuadamente los paquetes de voz y datos, ya que no es posible establecer una sesión de videoconferencia directamente entre la red externa e interna en base a un direccionamiento IP. En combinación con los gatekeeper, el esquema de proxy solventa el intercambio de tráfico y resuelve el problema de encaminamiento de direcciones IP privadas. En ambos casos, para conseguir que la infraestructura de comunicación entre la red interna y la externa sea transparente al usuario, se requiere un

[1] Videnet, https://videnet.unc.edu/
[2] I2CAT, http://www.i2cat.net/

direccionamiento único para cada terminal H.323, acorde con el esquema de direccionamiento internacional, como el proporcionado por Videnet, el Global Dialing Scheme (GDS)[3]. El GDS se implementa mediante una estructura de gatekeepers.

2 Soluciones propuestas

Con el fin de evaluar las posibilidades existentes, se implementaron las soluciones de ALG y proxy citadas anteriormente, con la premisa de intentar utilizar software de libre distribución. Para implementar la estructura de proxy se escogió el GNU gatekeeper[4], que es una solución muy completa en funciones de gatekeeper, incorpora funcionalidades de proxy y características avanzadas de H.323 como la redirección de llamada y balanceo de carga, además de ofrecer compatibilidad con la mayoría de productos en el mercado. Para la implementación del ALG, se utilizó el software de Coyote Linux[5], que proporciona funciones de filtrado de paquetes stateful con soporte H.323. El escenario resultante en ambas soluciones consiste en dos gatekeepers, uno en la red privada i2CAT y el otro en la red Internet, conformando dos zonas referenciadas mediante prefijos. Para establecer este escenario se cuenta con una MCU Radvision On Lan con gatekeeper incorporado destinada a la gestión de la zona i2CAT. La zona externa está bajo el control del GNU gatekeeper. La comunicación entre ambas zonas se puede realizar mediante proxy o ALG, dependiendo de uno u otro esquema. Los gatekeepers, en general, gestionan los mensajes de señalización de llamada H.225 de dos formas direct call y gatekeeper routed. La primera (direct call), señalización de llamada directa, consiste en pasar los mensajes de señalización directamente entre los terminales, esta forma no puede utilizarse en ninguno de los esquemas debido a que una dirección privada no es accesible desde Internet, En la segunda forma (gatekeeper routed), la señalización de llamada es direccionada a través de gatekeeper entre los terminales. Esta forma de señalización se utiliza en ambos esquema, en el caso del GNU gatekeeper, habilitando la funcionalidad de proxy, se direcciona además el tráfico adicional correspondiente al flujo de video y audio, no existiendo comunicación directa entre terminales.

En la implementación del ALG, todos los terminales H.323 deben tener configurado el ALG como la puerta de enlace por defecto, a nivel de red. Por otra parte, para que un terminal pueda recibir llamadas desde Internet, las llamadas se deben redireccionar desde el ALG hacia los terminales específicos mediante configuración del ALG. Este hecho, limita el número de llamadas que se pueden recibir del exterior y obliga a crear una estructura rígida, en la que no todos los terminales podrán recibir llamadas externas. Por este motivo esta solución podría ser incompatible con la creación de subredes y, en consecuencia, poco escalable si consideramos. El esquema de ALG sería adecuado para establecer un servicio basado en multiconferencia. Si este es el objetivo, podría conseguirse la escalabilidad al direccionar el tráfico entrante hacia la red privada a un solo terminal, que será una MCU, con la posibilidad de establecer estructuras de MCU en cascada entre las redes interna y externa. Las MCU´s en cascada reducen el consumo de ancho de banda y los niveles de retardo si consideramos que un cascada entre MCU´s se establece mediante una comunicación punto a punto entre ambos terminales. La solución basada en proxy necesita una mayor cantidad de proceso, puesto que se está trabajando a nivel de aplicación, y por tanto, es más lenta que una inspección stateful de un ALG. Otra desventaja es que el hecho de separar la comunicación en dos llamadas (terminal origen-proxy y proxy-terminal destino) induce retardo adicional en la comunicación. En contrapartida, una infraestructura basada en proxy es más escalable y flexible que un ALG, ya que permite establecer llamadas simultáneas entre las redes interna y externa y direccionar las llamadas entrantes hacia cualquier terminal de la red privada.

Una vez implementadas las dos soluciones, se comprobó que ambas resolvían el problema de comunicar i2CAT con Internet a través de un direccionamiento único. En ambos casos se establecieron sesiones punto a punto y punto multipunto a velocidades entre 128 y 768 kbps.

4 Conclusiones

En las redes IP, el servicio de videoconferencia H.323 se encuentra con una problemática a la hora de compaginar el direccionamiento público y privado. En este artículo se han estudiado e implementado dos soluciones, mediante proxy y mediante ALG, al caso concreto de la red del proyecto i2CAT, llegando a la conclusión que el esquema de proxy se ajusta mejor a los requerimientos de i2CAT de cara a los servicios de videoconferencia que ofrece, en los que la premisa es ofrecer conectividad a los usuarios de la manera más escalable posible tanto en sesiones punto a punto como multipunto. Los posibles problemas de retardo inherentes a los proxies no son una limitante en una red con tecnología avanzada como i2CAT. Por último, ambas soluciones resuelven el problema de conectividad y la decisión de implementar alguna de ellas dependerá de las necesidades de la organización.

Referencias

[1] T. Ogletree. *Practical Firewalls*. Que. ISBN: 0789724162 (2000). Ch 7.

[2] B. Chapman, S. Cooper, E. Zwicky. *Building Internet Firewalls*. O'Reilly. ISBN: 1565928717(2000), 2nd Edition, Ch 19.

[3] http://www.unc.edu/video/videnet/help/gds.html
[4] The GNU Gatekeeper, http://www.gnugk.org/
[5] Coyote Linux, http://www.coyotelinux.com/

Herramienta gráfica de configuración y análisis del servicio de tiempo global ofrecido por el protocolo NTP

F. Boronat Seguí, J. Carlos Guerra Cebollada, J. Ramón Mengual Oltra
Area de Ingeniería Telemática, Departamento de Comunicaciones
Universidad Politécnica de Valencia - Escuela Politécnica Superior de Gandía
Ctra. Nazaret-Oliva S/N, 46730 Grao de Gandía (VALENCIA)
Telf: 96 284 93 41, Fax: 96 284 93 13
E-mail: fboronat@dcom.upv.es,

Abstract. *In this article we present a very useful new tool that allows configuring and monitoring the global time service in a Time Server in which the global time is obtained using NTP (Network Time Protocol). This tool is very easy to use and shows graphic and statistic data about the service. We describe the tool and show several windows and graphs obtained with the tool in our laboratory.*

1 Introducción

En general, para un estudio detallado de cualquier servicio distribuido es muy útil el disponer de datos precisos de tiempo de cualquier evento producido en los equipos implicados, bien sea para la detección de problemas de hardware y/o software, así como para el estudio estadístico de los mismos. Para ello se necesitará que los equipos implicados estén sincronizados para disponer de una referencia común de tiempos y, así, poder analizar los datos o históricos de forma adecuada. NTP (*Network Time Protocol*) [1-2] es el protocolo más extendido de sincronización de tiempos en Internet.

En este artículo se presenta una herramienta novedosa, denominada TEMPS_NTP, para configurar y analizar, de forma gráfica e intuitiva, el servicio NTP del servidor de tiempos de nuestra red, basado en sistema operativo Windows NT/2000 o Linux. Todo ello se podrá realizar sin la necesidad de introducir o modificar las líneas de texto en los ficheros de configuración del servicio NTP (Fig. 2), siguiendo las pautas indicadas en las especificaciones de la versión del protocolo instalada.

El protocolo NTP es ampliamente utilizado en Internet y su funcionamiento es sobradamente conocido, por lo que el artículo se limita a presentar la aplicación junto con una serie de gráficas obtenidas con la misma y una serie de conclusiones.

2. Herramienta gráfica de configuración del servicio NTP

La herramienta, cuya ventana principal aparece en la Fig. 1, ha sido programada con Visual C++ 6.0 [3], y permite la configuración del servicio NTP de una forma más cómoda, además del análisis de los resultados obtenidos de la sincronización. Proporciona una plataforma de configuración mucho más fácil de utilizar a través de una interfaz con ventanas y menús. Permite acceder a los archivos de configuración y a las estadísticas a corto y largo plazo obtenidas por el servicio NTP, independientemente del sistema operativo del servidor (Linux o Windows), aunque la herramienta funciona sobre Windows. En la Fig. 1 aparecen, para un servidor NTP sobre Windows 2000, los valores obtenidos del *offset*, *delay* y dispersión en la última sincronización con el servidor NTP elegido como referencia, además de ofrecer sus datos (dirección y *stratum*) [1].

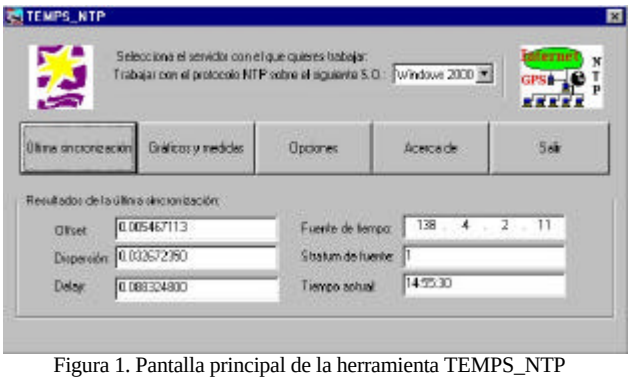


Figura 1. Pantalla principal de la herramienta TEMPS_NTP

```
server      127.127.1.0          # local clock
fudge       127.127.1.0 stratum 10
driftfile c:\winnt\system32\drivers\etc\ntp.drift
authenticate no
server 18.72.0.3
server 138.4.2.11 minpoll 5 maxpoll 5
server 192.93.2.20 minpoll 5 maxpoll 5
peer 138.100.8.3 minpoll 5 maxpoll 5
peer 193.146.145.13 minpoll 4 maxpoll 5
peer  212.95.210.35 minpoll 4 maxpoll 4
enable stats
statistics loopstats
statistics clockstats
statistics peerstats
statsdir C: \NTP\ \Stats\Stats_w2000\
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable
```

Figura 2. Ejemplo de Archivo *ntp.conf para Windows 2000*

Desde la ventana principal, se puede acceder a la hora actual, a la configuración del servicio NTP y al análisis de los resultados de sincronización obtenidos.

2.1 Opciones de configuración

Desde estas opciones (Fig. 3) se permite introducir los servidores de tiempo NTP tomados como referencia de tiempos, si se desea obtener también el tiempo de un receptor GPS local [4], el tipo de estadísticas obtenidas (en períodos de duración diaria, semanal o mensual) y el intervalo de envío de peticiones de sincronización.

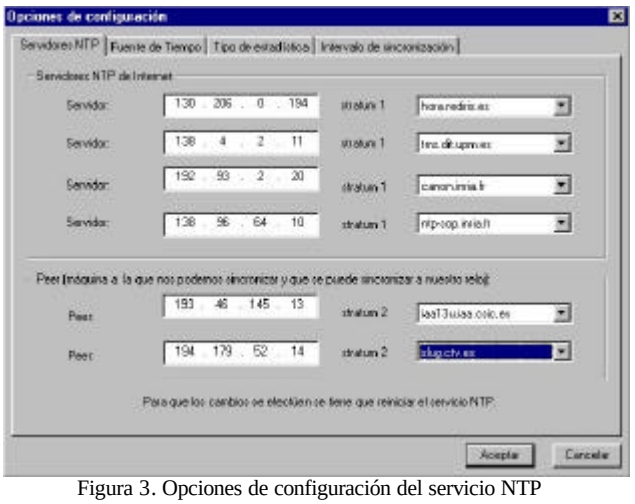


Figura 3. Opciones de configuración del servicio NTP

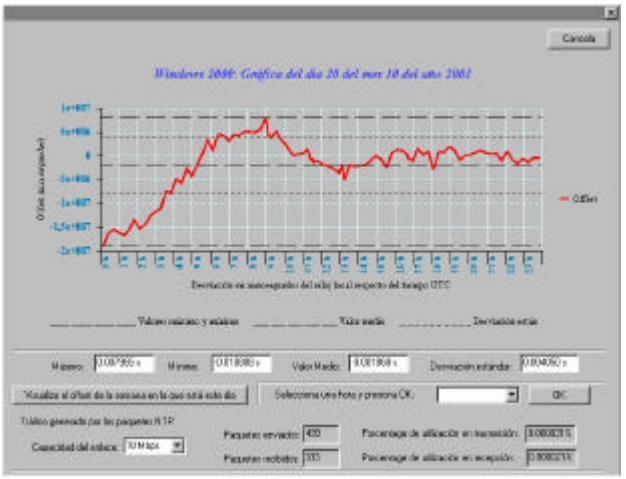


Figura 4. Gráfica de loopstats diaria.

2.2. Gráficos y Medidas

Los datos obtenidos por el servicio se almacenan en ficheros de texto y son recuperados por la aplicación y presentados de forma gráfica al usuario. Existen 4 tipos de ficheros: *clockstats*, *peerstats*, *loopstats* y *serv_emprat* (para obtener éste último tipo se ha tenido que modificar el código de la distribución). En los ficheros *peersats* se guarda el *offset* obtenido con respecto a cada una de las fuentes de tiempo configuradas. En los ficheros *loopstats* se almacena cada una de las actualizaciones que se hacen en el reloj local. En caso de disponer de receptor GPS [4] conectado al equipo, en *clockstats* se guarda las sentencias NMEA [5] recibidas del mismo, de donde se obtiene el tiempo (por ello, estos ficheros sólo se podrán visualizar mediante un editor de texto). Los ficheros *serv_emprats* almacenan el porcentaje de veces que el servicio NTP considera a cada servidor, de los configurados, como mejor fuente de tiempo.

En las Figs. 4, 5 y 6 aparece una gráfica representativa de cada tipo de datos. Aquellas que lo permiten, muestran los valores máximo, mínimo, medio y la desviación estándar de las medidas que representan y, además, se señalan dichos datos mediante diferentes tipos de rectas (Fig. 4). También se muestra el porcentaje de ocupación de la red debida al intercambio de mensajes NTP (Figs. 4 y 5).

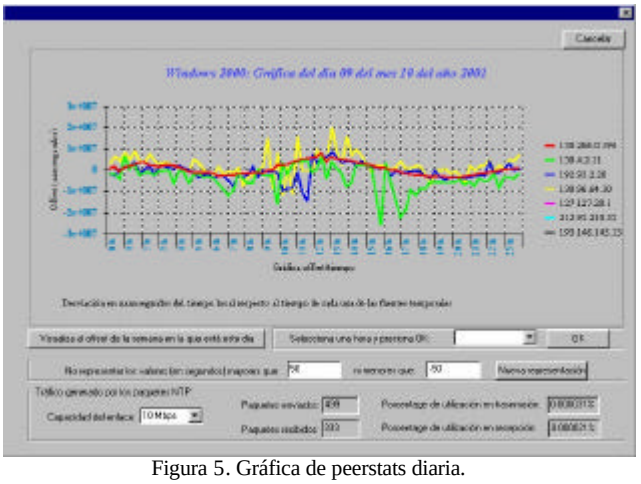


Figura 5. Gráfica de peerstats diaria.

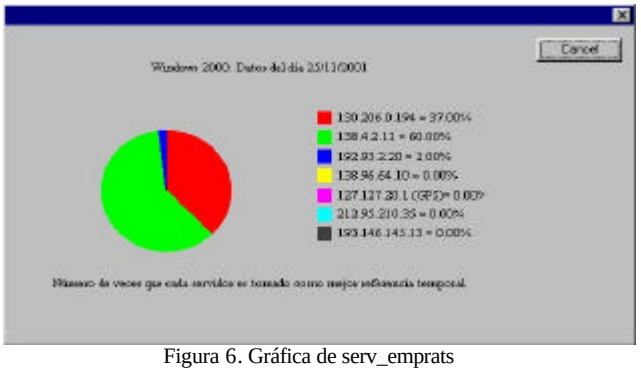


Figura 6. Gráfica de serv_emprats

3 Conclusión

En el presente artículo se ha presentado una herramienta gráfica que facilita la configuración del servicio de tiempo global NTP, independientemente del sistema operativo sobre el que se ejecute el servidor. Como novedad, además del aspecto gráfico, la herramienta también permite un análisis gráfico del comportamiento del servicio, así como la obtención de datos estadísticos, como, por ejemplo, el *offset* medio durante un determinado periodo (una hora, un día, una semana o un mes), así como la carga introducida por el protocolo. Todo ello se realiza de forma gráfica sin tener que acceder a los ficheros de configuración del servicio, facilitando las tareas de administración del mismo.

Referencias

[1] Mills, D., "*Network Time Protocol (Version 3) specification, implementation and analysis*", RFC 1305, University of Delaware, March 1992

[2] FAQ sobre el protocolo NTP en el sitio http://www.eecis.udel.edu/~ntp/ntpfaq/NTP-a-faq.htm

[3] J. Pascual, F. Charte, M. J. Segarra, J.A. Clavijo, "Programación Avanzada en Windows 2000 con Visual C++ y MFC", Ed, McGraw Hill, 1999, ISBN:84-481-1437-0

[4] USA National Park Service, *"Global Positioning Systems (GPS)"*. March 1999. http://www2.cr.nps.gov/gis/gps.htm .

[5] G. Baddeley *"Glenn Baddeley - GPS - NMEA sentence information"*, January 2000.

Soporte a la Evaluación Telemática del Alumno

Juan Carlos Dueñas, Tomás de Miguel, José Luis Ruiz
Departamento de Ingeniería de Sistemas Telemáticos
Escuela Técnica Superior de Ingenieros de Telecomunicación
Universidad Politécnica de Madrid
Teléfono: 91 336 68 31 Fax: 91 336 73 66
E-mail: {jcduenas, tmiguel, jlruiz}@dit.upm.es

***Abstract**. The gradual incorporation of information technologies into the formal education has brought a new world of possibilities for making more effective and ubiquitous the knowledge transference process. The applicability of these new techniques has probably no limits and can be extended to every knowledge domain, but obviously it is in teaching information technologies in which the highest point of usefulness can be achieved, as in this case the teaching means match with the purpose of the subjects. As the department responsible for teaching computer science and computer networks subjects in the Escuela Técnica Superior de Ingenieros de Telecomunicación (ETSIT), DIT (Departamento de Ingeniería de Sistemas Telemáticos) in the Universidad Politécnica de Madrid (UPM) has created SETA (Sistema de soporte a la Evaluación Telemática del Alumno). This paper presents SETA, a system for remotely and automatically tutoring and evaluating students.*

1 Introducción

Durante la última década hemos asistido a un desarrollo vertiginoso en el ámbito de las tecnologías de información y las comunicaciones. Este avance ha derivado en el escenario con el que nos encontramos actualmente: las infraestructuras de comunicaciones han expandido su cobertura, los terminales de los que disponen los usuarios son de altas prestaciones (incluso superiores a estaciones que funcionan como servidoras), son comunes las tecnologías de acceso de banda ancha y todo ello puede conseguirse a costes relativamente razonables, gracias a la libre competencia de operadores y a la aplicación de economías de escala.

Es un momento propicio, por tanto, para la aplicación de estas nuevas tecnologías al dominio de la educación, ya que el encauzamiento de esta nueva ola de progreso debe ser capaz de generar beneficios a los distintos actores involucrados en el mundo de la universidad.

En este documento pretendemos presentar la iniciativa seguida dentro del DIT-UPM para el desarrollo del SETA (Sistema de soporte a la Evaluación Telemática del Alumno). El SETA toma cuerpo en el seno del proyecto RETELEDU (Red Telemática Educativa de la ETSIT), financiado por la Fundación Retevisión, y finalizado recientemente.

La organización del resto del artículo es la siguiente: la sección segunda presenta los motivos para el desarrollo del SETA. En la sección tres se presenta una descripción del sistema. En la última sección se presentan las conclusiones obtenidas al término de este trabajo.

2 Motivación del proyecto

En el ámbito universitario tradicional, los alumnos se veían obligados a desplazarse físicamente hasta los centros educativos para obtener información, realizar gestiones administrativas y, de forma habitual, asistir a las clases.

El proyecto RETELEDU tiene como ámbito de aplicación la introducción de las tecnologías de información a la ETSIT en su conjunto. El objetivo consiste en integrar servicios de tele-educación y de tele-gestión en una plataforma a la cual tanto profesores como alumnos puedan acceder.

En los planes de estudios de carreras técnicas y/o científicas, es cada vez más habitual encontrar asignaturas prácticas de programación de ordenadores o similares. El desarrollo del proyecto SETA se ha dirigido a dar respuesta a las necesidades específicas que aparecen en estas materias:

1. Los contenidos asociados a estas disciplinas son por lo general bastante novedosos para los estudiantes. Los alumnos son neófitos en lo que se refiere a temas como los fundamentos de ordenadores, el uso de herramientas de desarrollo o en lo referente a las metodologías de programación.

2. La realización de un curso de programación exige de la ejecución de una cantidad importante de ejercicios prácticos. Como suele decirse a caminar se aprende andando y esto es algo verdaderamente cierto en el ámbito de la programación.

La evaluación de los ejercicios prácticos queda evidentemente en manos de los profesores, los cuales obtienen de esta forma realimentación sobre el progreso de los estudiantes. Esta información puede ser muy valiosa para los docentes, ya que les permite

tomar a tiempo medidas apropiadas para corregir los errores y mejorar el proceso de aprendizaje.

El enfoque tradicional de corrección manual de prácticas, aparte de convertirse en una tarea tediosa en muchos casos, es difícilmente escalable con respecto al número de alumnos, el número de ejercicios prácticos, el tamaño o la complejidad de las prácticas. En resumen: imposible de aplicar por sus dificultades operativas.

El SETA como sistema tiene como misión ofrecer un soporte automatizado a los procesos de entrega y corrección de prácticas, reduciendo la carga de los profesores en cuanto a estos procesos repetitivos (y permitiendo enfocar el esfuerzo de corrección en los aspectos más creativos de las prácticas).

3 El sistema SETA

El desarrollo del sistema se ha guiado por una arquitectura dividida en tres capas: la lógica de servicio, los datos y la presentación. Tomando como base el patrón de diseño MVC (Model View Controller), se aplicaron en su construcción tecnologías Java 2 Enterprise Edition (Servlet, JSP y JDBC) y servidores de fuente abierta: Apache, Tomcat y MySQL.

La funcionalidad básica del sistema consiste, para los alumnos, en realizar desde Internet la entrega de sus prácticas y acceder de forma inmediata a la evaluación obtenida. Este acceso al sistema se realiza mediante el protocolo http (de forma que puede utilizarse cualquier cliente Web). Existen funcionalidades adicionales para el alumno, entre otras: el acceso al registro completo de sus notas, el envío de reclamaciones/sugerencias o la modificación de su contraseña.

El proceso de corrección de una práctica tiene lugar en tiempo real, por lo que el alumno puede conocer inmediatamente la nota que obtiene en su práctica así como un conjunto de recomendaciones sobre cómo mejorar la práctica entregada. La entrega de cada una de las prácticas se realiza durante un periodo habilitado para ello, durante el cual no se limita inicialmente el número de evaluaciones a las que puede acceder el alumno.

La evaluación de las prácticas de los alumnos se realiza mediante una plantilla de código creada por los profesores y que se aplicará automáticamente sobre el código entregado por cada uno de los alumnos. Los profesores tienen que proporcionar al sistema una batería de pruebas para cada una de las prácticas que componen el curso. En el proceso de evaluación los profesores se encargarán de plantear pruebas al código de los alumnos e ir construyendo la nota y recomendaciones siguiendo un modelo de caja negra.

SETA proporciona una interfaz Web a los profesores para que además de añadir las plantillas de corrección, puedan realizar consultas en cuanto al proceso de entregas de los alumnos, extraer estadísticas de utilización e informes sobre la evolución del aprendizaje.

Adicionalmente los profesores pueden acceder a las reclamaciones y comentarios enviados por los alumnos, lo que permite obtener realimentación a los profesores sobre posibles errores en el proceso de corrección o en el sistema mismo.

4 Conclusiones

SETA es un servicio realmente novedoso en lo que se refiere a la evaluación de los alumnos, ya que permite conocer aspectos prácticos del proceso de aprendizaje del alumno mediante un mecanismo automático, que libera a los profesores del proceso de corrección manual de las prácticas.

Existen muchos otros medios de evaluación del alumno centrados en el aprendizaje de los conceptos teóricos, estando la mayor parte de ellos basados en la realización de tests. Un test permite conocer el grado de conocimiento teórico adquirido por el alumno con bastante fidelidad, pero en los aspectos prácticos únicamente puede comprobar si el alumno ha seleccionado el resultado correcto. Es precisamente en este aspecto en el que el SETA da un paso más allá, ya que para la corrección utiliza la solución completa entregada por el alumno, no únicamente el resultado al que llega, lo que permite llevar a cabo una evaluación más ajustada del trabajo de programación realizado.

La valoración del trabajo del alumno puede extenderse a aspectos de calidad e incluso de estilo, ya que al disponer de la solución completa pueden tomarse medidas de parámetros como: memoria ocupada, tiempo de proceso invertido, número de líneas de código... En la actualidad se está extendiendo para permitir obtener tales medidas de calidad de las soluciones, dotar al sistema de protecciones frente al plagio de prácticas, y proveer de nuevas capacidades de entrega, gestión y corrección de prácticas para otro tipo de laboratorios. Hasta el momento se ha usado en las asignaturas de Fundamentos de Programación de primer curso de la ETSIT, con resultados altamente satisfactorios para los profesores y especialmente para los alumnos.

Referencias

[1] *"The State of Software Engineering Education and Training"* N. Mead, D. Carter, M. Lutz. IEEE Software, 14/6. Noviembre 1997.

[2] *"Preparing Software Engineers for the 'Real World"*. Ed Yourdon. Proceedings of the 15th Conference on Software Engineering 2002. Education and Training (CSEET.02).

[3] *"Tool Support for Teaching the Personal Software Process"*. Margot Postema, Martin Dick, Jan Miller and Simon Cuce. Computer Science Education, 10(2), August, 2000. Swets & Zeitlinger, 179-193

.

Mejora en los protocolos de streaming en redes inalámbricas[1]

Xavier Hesselbach, José Cástor Vallés, Marisa Catalán
Departamento de Ingeniería Telemática. Universidad Politécnica de Cataluña.
Jordi Girona 1 y 3. Campus Nord, Módulo C3, UPC. 08034 Barcelona
E-mail: {xavier.hesselbach,jcvalles,mcatalan}@entel.upc.es

Abstract. *The 2.5G and 3G mobile cellular communication systems improve data rates and are suitable for packet oriented transmission. Within them, both real time and on demand media services can be created and played over the wireless network. In order to play smoothly, multimedia data (in particular, audio and video) need to be available continuously and in the proper sequence without interruption. Results show some undesired behavior that can be interpreted from WLAN and 2.5G towards 3G wireless networks.*

1. Introducción

Los sistemas móviles celulares de generación 2.5G y 3G ofrecen una mayor tasa de transmisión de datos y se ajustan a la transmisión de datos orientada a paquete, a diferencia de los sistemas de segunda generación como GSM que operan en modo circuito. Streaming se define como un modo de transferencia de datos de forma que permite al cliente reproducir los contenidos (audio o video) sin requerir una descarga previa del fichero completo. El equipo fuente o servidor transmite pequeñas unidades del contenido completo en forma de paquetes a través de la red hacia el cliente, el cual accede a ellos tan pronto como son recibidos y como un flujo continuo en tiempo real. Diversos grupos y organizaciones han reconocido la necesidad de estandarización de los servicios de streaming. Los más remarcables son el Internet Streaming Media Alliance (ISMA) [1], el Wireless Multimedia Forum (WMF) [2], el Third-Generation Partnership Project (3GPP) [3] y el 3G Terminals and 3G Services (3GPP2) [4].

2. La negociación del servicio

Windows Media es en la actualidad la aplicación de video más popular debido a la amplia difusión de los sistemas basados en sistema operativo Microsoft Windows. Este motivo ha llevado a estudiar con especial detalle el funcionamiento de los protocolos asociados a la transmisión mediante este software, aunque también se han realizado pruebas con Real Media y Quicktime. Se han llevado a cabo pruebas con la versión 7.1 de Window Media Player para equipo de sobremesa basado en Windows XP y para Pocket PC con WindowsCE versión 3.0 en un equipo Compaq iPAQ 3660. Video streaming es posible mediante el protocolo de transmisión MMS (Microsoft Media Server protocol), en modo UDP o TCP. MMS se encarga del control y la gestión de los diversos flujos basados en ficheros .asf desde un servidor Windows Media.

3. Escenarios de prueba

Se han diseñado y usado tres entornos de prueba para la experimentación con estas aplicaciones: Sobre red Ethernet, sobre una plataforma WLAN con y sin traspasos, y sobre GPRS, también considerando la existencia o no de traspasos.

Fig. 1. Escenario de pruebas simple

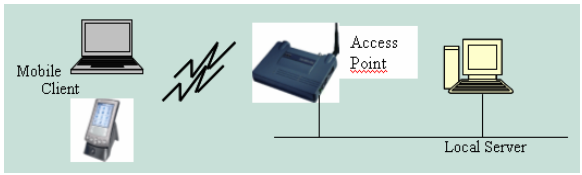


Fig. 2. Escenario basado en WLAN

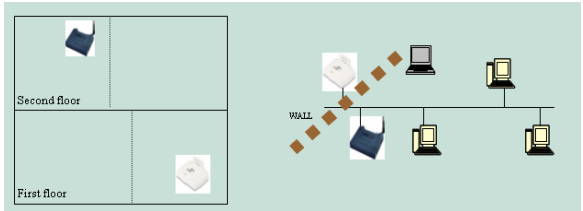


Fig. 3. Escenario complejo WLAN

4. Resultados experimentales

En el empleo de MMS-UDP, las figuras 4 y 5 muestran la tasa de cuadro desde el punto de vista del cliente y el servidor. En los ejes de abscisa, la ventada de tiempo corresponde al orden de los 30 segundos. Mientras el servidor no se ve afectado por el handover, el cliente lo percibe como una interrupción de tráfico. Cada columna representada

[1] Este trabajo ha sido parcialmente financiado por RIU253 - IST-2001-36510.

corresponde a un paquete de datos, y su altura al tamaño de dicho paquete. Por lo tanto, las gráficas permiten apreciar la cadencia de los paquetes y el tamaño de cada uno de ellos. Para estas pruebas, el vídeo ha sido codificado a 15 Kbit/s (no *multirate*) empleando codec Windows Media 7, con buffer de 1 segundo.

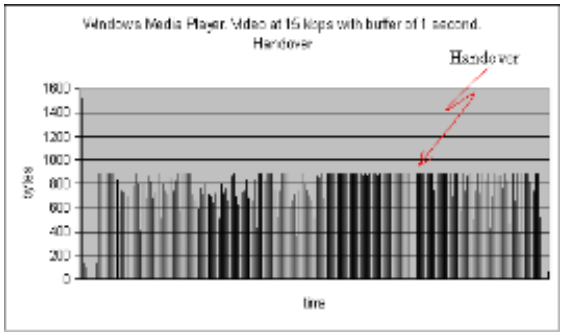


Fig. 4. Video codificado a 15 kbps, buffer = 1 segundo. WLAN con traspaso. MMS-UDP

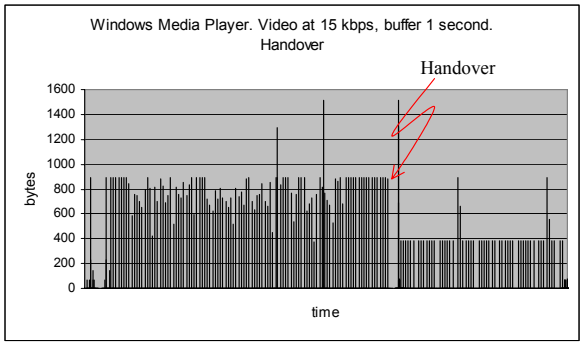


Fig. 5. Video codificado a 15 Kbps, buffer = 1 segundo. WLAN con traspaso. MMS-TCP

4 Recomendaciones para usuarios

Tomando como referencia las pruebas efectuadas, se pueden ofrecer los siguientes resultados, útiles como recomendaciones para los usuarios:

Se identifica un problema existente en las transmisiones empleando Windows Media cuando se produce una interrupción de tráfico, y es que la tasa de transmisión no recupera el valor inicial, aunque realmente el caudal del medio lo permitiría. Debería replantearse el diseño del control de tasa en la aplicación para evitarlo.

Para un tamaño de MTU de 900 bytes, el caudal cursado por una red GPRS es máximo. Este es pues el valor adecuado en la transmisión. Comparado con las plataformas software mencionadas en esta ponencia, Windows Media ofrece un tamaño de paquete muy cercano a este valor.

Con el fin de mantener la calidad tan alta como el canal lo permita, es necesario que a pesar de detectar una interrupción de tráfico, si ésta es debida a un traspaso, el servidor se mantenga a la misma tasa inicial de transmisión del flujo multimedia.

En el caso de que una transmisión supere los 6 segundos, las aplicaciones suelen considerar que existe un fallo de red y en ocasiones se producen desconexiones. Ante esta situación, el tamaño del buffer no presenta ninguna utilidad. En redes GPRS, los traspasos pueden provocar desconexiones debido al valor de los temporizadores. Una recomendación general en este sentido es ajustar los valores de estos parámetros en función de la red subyacente empleada (bajos en Ethernet, valores superiores para WLAN, y altos de hasta unos 10 segundos para GPRS).

5 Conclusiones y trabajos futuros

Esta ponencia muestra algunos resultados experimentales con sistemas comerciales de streaming de amplia difusión, para redes inalámbricas de generación 2.5G y su proyección hacia la 3G. En este sentido, se ha empleado WLAN como plataforma de mayor capacidad, donde los tiempos de traspaso se consideran inferiores a los que UMTS se estima que ofrecerá cuando esté disponible.

De las pruebas se desprenden problemas en algunos protocolos de streaming que afectan al proceso de renegociación ante interrupciones de tráfico. En esta situación, cabe estudiar el funcionamiento de renegociación de tasa a nivel de aplicación. De esta forma, corregir este inconveniente. Otra tarea pendiente es el ajuste de caudal de forma óptima a la capacidad disponible ofrecida por la red. En particular, esto es realmente útil para redes de baja capacidad como GPRS, donde el compromiso entre cadencia de cuadro, tamaño de la imagen, profundidad de bit, tipo de reproductor y formato de codificación se erigen en una decisión subjetiva para conseguir la mejor calidad posible.

Referencias

[1] ISMA, Internet Streaming Media Alliance. http://www.isma.tv

[2] Wireless Multimedia Forum. http://www.wmmforum.com

[3] Third-Generation Partnership Project. http://3gpp.org

[4] 3G Terminals and 3G Services. http://3gpp2.org

[5] D2.1, "Survey on services, terminals and applications available", proyecto RIU253. http://www-riu253.upc.es.

[6] D3.1, "Protocols options and their relation with the wireless packet protocols", proyecto RIU253. http://www-riu253.upc.es.

[7] D5.1, "Information formats for wireless Internet", proyecto RIU253. http://www-riu253.upc.es.

[8] ISO/IEC 14496-2:2001. Information technology - Coding of audio-visual objects - Part 2: Visual.

Simulación de Modulaciones Digitales mediante GUIs de MATLAB

Raúl Llinares, Javier Moya, Andrés Camacho, Jorge Igual
Departamento de Comunicaciones. Universidad de Politécnica de Valencia
Pza Ferrándiz i Carbonell s/n
03801 Alcoy (Alicante)
Teléfono: 96 6528517 Fax: 96 6528461
E-mail: rllinares@dcom.upv.es

***Abstract**. This work shows an application for simulating digital modulations, base-band and pass-band with continuous carrier. The program has been developed with MATLAB in order to get the advantages of its engine relatives to fast simulation and graphical interface. The main objective of the application is provide the user a tool, that shows him the waveform present in any desired point of a simulated system. Also provides the student a practical view of the abstract concepts studied in the theory, like Intersymbol Interference, limitations in band of channel or noise and its undesired effects in the communication process.*

1 Introducción

Se presenta en este artículo un simulador de Modulaciones Digitales que permite reforzar conceptos desarrollados en asignaturas que traten esta materia.

Los objetivos perseguidos fundamentalmente han sido permitir al profesorado mostrar ejemplos reales y representativos a la par que se explican los conceptos teóricos y por otra parte proporcionar a los alumnos una serie de prácticas que se centren en aspectos únicamente de la materia en sí (la sencillez del programa permite que el tiempo en aprender a utilizar el simulador sea nulo).

1.1 Antecedentes y entorno de trabajo

Existen softwares comerciales como CommSim o SystemView que permiten realizar simulaciones de comunicaciones digitales con mucha potencia y versatilidad, pero con el principal incoveniente de los conocimientos requeridos para su manejo.

Otra posibilidad para simular sistemas de comunicaciones la proporciona MATLAB [3] y [4]. MATLAB permite trabajar con funciones llamadas directamente desde el prompt, mediante Simulink, o mediante interfaces gráficos de usuario (GUIs).

Al trabajar desde el prompt con funciones propias, se puede realizar la simulación paso a paso, pero generalmente cuando se quiere realizar cambios en los parámetros, el entorno se vuelve engorroso de manejar.

Por otra parte con Simulink, igual que con los software comerciales, se deben perder sesiones aprendiendo el manejo de la herramienta, con la dificultad añadida de la poca versatilidad de los bloques existentes.

La tercera opción de trabajo con MATLAB es el uso de GUIs, y esa la solución adoptada en este trabajo. La novedad e interés del sistema es que con un único GUI de MATLAB es posible simular todo el sistema de comunicación digital sin necesidad de conocer ninguna herramienta, y sin realizar ninguna llamada a funciones desde el prompt. Tanto los parámetros de simulación como los resultados de interés están detallados en el propio GUI de forma explícita y de forma gráfica, permitiendo una mejor asimilación de las conclusiones obtenidas en cualquier simulación.

2 Simulador

La aplicación permite simular sistemas digitales Banda Base y Paso Banda, tal y como suelen aparecer descritos en los libros de texto [1] y [2].

El simulador consistirá en sí mismo en una ventana en la que fundamentalmente destaca un gráfico con el sistema elegido. Antes de simular, deberán configurarse algunos parámetros. La configuración de los bloques se realiza al hacer clic con el ratón sobre ellos. Se abrirá una subventana en la que se introducirán los valores deseados.

Tras la configuración, el botón de simulación permitirá realizar la simulación en función de los datos de usuario. Es ese momento, el alumno podrá ir pinchando sobre los puntos del sistema indicados y ver la señal que hay en ese punto tanto en el dominio del tiempo como en el de la frecuencia. Para mayor comodidad, se pueden visualizar varias gráficas de forma simultánea.

2.1 Simulación Banda Base

En una simulación en banda base (figura 1), los parámetros que pueden configurarse son la fuente de información, la codificación en línea, los filtros de transmisión y recepción, el canal utilizado y la temporización en recepción.

En cuanto a los resultados mostrados, éstos son la representación de señales en el dominio temporal y frecuencial en todos los puntos del sistema, función de transferencia global del sistema, diagramas de ojos y tasa de errores cometidos.

2.2 Simulación Paso Banda

Para la simulación paso banda, se utilizará el mismo esquema que en el caso banda base, excepto en las modulaciones no lineales (FSK).

En cuanto a los parámetros a elegir, éstos serán el tipo de modulación, el tipo de recepción, los parámetros de los osciladores y el nivel de ruido AWGN.

Los resultados más interesantes serán anchos de banda, constelaciones y probabilidades de error.

3 Validación

La aplicación del simulador se validó en las prácticas de la asignatura Transmisión de Datos durante el curso 2001-2002. Los alumnos se mostraron muy satisfechos por la facilidad de uso y por la claridad en los resultados. Todo ello se vio reflejado en una mejora de un 15% en las encuestas de la asignatura respecto al año anterior.

4 Conclusiones

Se ha presentado una herramienta de simulación de modulaciones digitales ideal para un entorno docente.

Las asignaturas que tratan sobre modulaciones digitales necesitan de un apoyo experimental para su completa comprensión. Conceptos tales como modulación de portadora o Interferencia entre Símbolos, son plenamente asimilados cuando se experimenta visualmente con ellos, por ejemplo con herramientas como la implementada.

Se ha buscado una simplicidad de manejo y visualización de datos que permita al alumno empezar a trabajar sin necesidad de haberse leído previamente ningún tipo de manual. Simplemente se necesitan los conceptos vistos en clase

Por otra parte, se ha hecho uso del lenguaje MATLAB, que permite la ampliación de la herramienta sin ningún tipo de conocimiento de lenguaje de alto nivel. También se aprovechan los interfaces gráficos que proporciona MATLAB (GUIs) para conseguir un interfaz amigable.

Referencias

[1] Sklar, *Digital Communications*, Prentice Hall. ISBN: 0-13-212713-x 025 (1988)

[2] Haykin, *Communications Systems*, John Wiley & Sons. ISBN: 0-471-17-869-1 (2001)

[3] MathWorks, *Communications Toolbox* (2002) .

[4] MathWorks, Creating Graphical User Interfaces (2002).

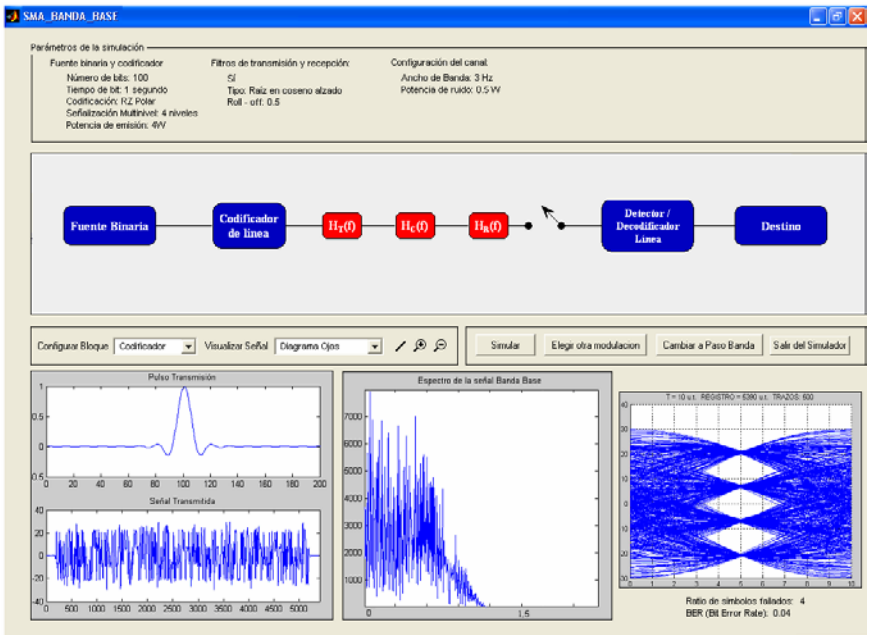


Figura 1: Simulación Banda – Base

Caché Web y Redes de Distribución de Contenidos: Una Visión General

Héctor Ossandón Díaz[1]
Departamento de Computación
Universidad de Tarapacá
18 de Septiembre Nº 2222, Arica-Chile
E-mail: ossandon@dit.upm.es

Encarna Pastor Martín
Dpto. Ingeniería de Sistemas Telemáticos
ETSIT - Universidad Politécnica de Madrid
Ciudad Universitaria, s/n, 28040 Madrid
E-mail: encarna@dit.upm.es

Abstract*. Web caching systems and Content Delivery Networks (CDN) deliver content to end users from network locations which are topologically closer to the user than origin Web servers. The goal of both technologies is to improve performance metrics like, for example, Web access response time, Web server throughput and network bandwidth utilization. This paper presents a study of the most relevant architectural aspects of Web Caching Systems and CDNs.*

1 Introducción y motivación

En la arquitectura clásica Web, todos los clientes que acceden a los mismos contenidos realizan peticiones a un único servidor. La escalabilidad y mejora de las prestaciones se trata de resolver generalmente en el propio sitio Web, creando clusters o granjas de servidores, quedando sin abordar los problemas debidos a la congestión de la red.

Una alternativa a esta arquitectura centralizada consiste en replicar y distribuir la información entre un conjunto de servidores geográficamente dispersos y ubicados en la periferia de la red, topológicamente cercanos a los sistemas de los usuarios finales. El objetivo global de esta red de servidores es evitar que los clientes establezcan conexiones directas con los servidores Web origen y reducir así el tiempo de respuesta de las peticiones. Tanto la tecnología Caché Web como las Redes de Distribución de Contenidos (CDN, *Content Distribution Networks*), buscan precisamente este objetivo, aunque sus formas de operar y sus visiones de negocio son distintas. Por otra parte, una implantación eficiente de estas tecnologías no es trivial. Al ser arquitecturas complejas, existen múltiples problemas que están siendo abordados por la comunidad investigadora.

En este artículo se presenta un resumen de la evolución de los Sistemas Caché Web y CDN y se resaltan los aspectos más relevantes en su diseño. Este estudio constituye el resultado de la primera fase de una investigación más amplia que se está llevando a cabo en el Depto. de Ingeniería de Sistemas Telemáticos, cuya finalidad es aportar soluciones concretas de arquitecturas que permitan organizar redes de distribución de contenidos eficientes, capaces de ofrecer servicios avanzados de acceso a información multimedia.

[1] Actualmente desarrolla su tesis doctoral en el Dpto. Ingeniería de Sistemas Telemáticos, ETSI de Telecomunicación, UPM.

2 Sistemas Caché Web

Un Sistema Caché Web consiste en un conjunto de servidores caché que están diseminados por la red y que colaboran entre sí mediante protocolos, de tal forma que funcionan lógicamente como una gran caché para atender los requerimientos de los usuarios en las respectivas zonas en que sirven. Las **arquitecturas** de los sistemas caché web pueden clasificarse en tres categorías: jerárquica, distribuida e híbrida [1]. Rodríguez et al. [2] proponen un modelo analítico para estudiar los parámetros de prestaciones más significativos de estas arquitecturas, como por ejemplo: la latencia percibida por el usuario, el uso del ancho de banda, la carga de los servidores caché y la utilización del espacio de almacenamiento.

Las **políticas de reemplazo** de la caché determinan los documentos web que se eliminarán de la caché para generar espacio y almacenar nuevos documentos. Una política de reemplazo de la caché busca utilizar de la mejor forma posible los recursos del sistema, de manera que se logre un rendimiento más eficiente del sistema caché en general. Las principales métricas que se utilizan para evaluar el rendimiento de un sistema caché web son: tasa de aciertos de documentos, tasa de aciertos de bytes y tiempo medio de descarga por documento. Existen muchos algoritmos de reemplazo que buscan optimizar las prestaciones de un sistema caché web.

Las **políticas de control de admisión** son algoritmos que deciden si un documento proveniente del servidor web origen se almacena o no en la caché. Su objetivo es que el ingreso de un nuevo documento a la caché no deteriore el rendimiento de ésta. Otra forma de mejorar el desempeño de un sistema caché web es utilizar alguna estrategia de **Prefetching**. El Prefetching consiste en anticiparse a los requerimientos de los usuarios. Se trata de cargar en la caché los documentos que posiblemente solicitará el usuario en un futuro próximo. Una estrategia de Prefetching que haga una buena predicción de los documentos que requerirán los usuarios en un futuro cercano y que se ejecute fuera de las horas de mayor

tráfico de la red, puede mejorar considerablemente las prestaciones de un sistema caché al reducir la tasa de fallos de la caché y disminuir el tiempo promedio de recuperación que perciben los usuarios.

3 CDNs

Al igual que los sistemas caché web, las CDN buscan acercar los contenidos a los usuarios para mejorar los tiempos de acceso y descarga de éstos, sin embargo, su operación y panorámica del negocio es completamente distinta. Desde un punto de vista de operación, una CDN duplica un conjunto de contenidos seleccionados desde uno o varios servidores web origen y los almacena en servidores sustitutos o réplicas (*surrogate servers*) que se encuentran más próximos a los usuarios finales [3]. Así, cuando un usuario solicita alguno de los contenidos duplicados en la CDN, ésta redirige su petición para que sea atendida por el servidor sustituto que le brinde el mejor servicio en ese momento. Para la elección del mejor servidor sustituto, la CDN debe considerar no sólo métricas estáticas como la distancia geográfica o la cantidad de saltos que existen entre el servidor sustituto y el cliente, sino que también debe considerar métricas dinámicas como el estado de congestión de los enlaces o la carga de trabajo que poseen cada uno de los servidores sustitutos [4]. Desde un punto de vista de negocio, una CDN proporciona al proveedor de contenidos un mayor control de las copias de sus contenidos que se encuentran en los servidores sustitutos, dándole así la oportunidad de ofrecer una mejor calidad de servicio en cuanto a contenidos actualizados a sus clientes.

Los elementos arquitecturales de una CDN son: un sistema de distribución, una infraestructura de entrega de contenidos, un sistema de encaminamiento de peticiones y un sistema de tarificación [4].

El **sistema de distribución** es el responsable de trasladar las copias de los contenidos del servidor origen a los servidores sustitutos y mantener la consistencia del sistema.

La **infraestructura de entrega de contenidos** de una CDN es el conjunto de servidores sustitutos que mantienen las copias del contenido del servidor origen y las entregan directamente a los clientes. Decidir sobre la ubicación de los servidores sustitutos en la red es un aspecto crítico. Aunque en la literatura existen varias propuestas de algoritmos para solucionar este problema, no siempre se resuelve de forma satisfactoria.

El **sistema de encaminamiento de peticiones** es uno de los componentes más importantes en la arquitectura. El objetivo final de una CDN es atender las peticiones de los clientes desde los servidores sustitutos que, en ese momento, le brinden un mejor servicio del que se obtendría si se acude directamente al servidor web origen. Los mecanismos más utilizados de encaminamiento de peticiones se pueden clasificar en tres categorías: mecanismos basados en el Servidor de Nombres de Dominio (DNS), mecanismos a nivel de transporte y mecanismos a nivel de aplicación [5]. Un enfoque más novedoso que estamos explorando consiste en el encaminamiento de peticiones a nivel de contenidos. Aquí se ataca el problema de elegir el mejor servidor sustituto como un problema de encaminamiento clásico. La idea central es que los clientes desean conectividad a una pieza de contenido específica y no a una dirección IP o servidor sustituto, de tal forma que los servidores sustitutos que poseen copias del contenido que se solicita, pueden verse como rutas alternativas para acceder al contenido.

4 Conclusiones

El despliegue de los sistemas de caché Web y más recientemente (en 1999) la aparición de las CDNs, sugiere vías de introducir nueva funcionalidad cerca de los extremos pero dentro de la red. Una arquitectura distribuida CDN podría implementarse como una red superpuesta en máquinas situadas en la periferia, entre el núcleo de la red y los usuarios finales (ya se ha hecho con Mbone y 6Bone). Así se podría ofrecer mayor funcionalidad y mejores prestaciones a una amplia comunidad de usuarios.

Como resultado del análisis cuyo resumen presentamos en este artículo, hemos identificado un conjunto de líneas de trabajo en las que se irán abordando los problemas detectados y las lagunas existentes en las propuestas actuales. En concreto, se ha detectado la falta de una base teórica sólida, debido a que estos temas forman parte de una tecnología aún emergente, así que estamos desarrollando modelos teóricos que sirvan de apoyo para analizar y evaluar prestaciones de arquitecturas. En segundo lugar, nos proponemos diseñar estrategias de encaminamiento de peticiones basadas en algoritmos adaptativos que traten de superar las limitaciones detectadas. Finalmente, nos centraremos en el diseño de modelos de distribución en la entrega de contenidos, incluyendo contenidos de medios continuos (flujos de audio y vídeo), ya que es un aspecto sobre el que apenas existen propuestas.

Referencias

[1] G.Lai, M.Liu, F.Wang and D.Zeng, “Web Caching: Architectures and Performance Evaluation Survey”, IEEE International Conference on Systems, Man and Cybernetics, vol.5, pp.3039-3044, 2001.

[2] P.Rodríguez, C.Spanner and W.Biersack, “Analysis of Web Caching Architectures: Hierarchical and Distributed Caching”, IEEE/ACM Transactions on Networking, vol.9, pp.404-418, August 2001.

[3] I.Cooper, I.Melve and G.Tomlinson, “Internet Web Replication and Caching Taxonomy”, RFC 3040, June 2000.

[4] M.Day, B.Cain, G.Tomlinson and P.Rzewski, “A Model for Content Internetworking (CDI)”, RFC 3466, Feb. 2003.

[5] A.Babir, B.Cain, F.Douglis, M.Green, M.Hofmann, R.Nair, D.Potter and O.Spatscheck, “Know CN Request-Routing Mechanisms”, draft-ietf-cdi-known-request-routing-01.txt (work in progress), May 2002.

Nuevas Vías para la Automatización de Procesos de Negocio de Forma Totalmente Descentralizada

Germán Toro del Valle

Telefónica Investigación y Desarrollo
C/ Emilio Vargas, 6.
28043 Madrid (SPAIN)
Teléfono: 91.337.99.59
Fax: 91.337.45.29
e-mail: germantoro@iespana.es

Encarnación Pastor Martín

Departamento de Ingeniería de Sistemas Telemáticos
E.T.S.I. de Telecomunicación
Universidad Politécnica de Madrid
C/ Ciudad Universitaria s/n.
28040 Madrid (SPAIN)
Teléfono: 91.336.73.28
Fax: 91.336.73.33
e-mail: encarna@dit.upm.es

***Abstract**. Workflow Technology includes a set of technics, mechanisms and tools that let any enterprise or organization automate its business processes, obtaining from them a maximum efficiency by means of an optimal use of its (human and not human) resources. In spite of the important benefits that Workflow Technology offers, the number of enterprises and organizations that make use of this technology nowadays is really low. The reasons of this fact are manifold.*
In this paper we present a brief study about the state-of-the-art of Workflow Technology and present an innovative approach to business processes automation based on cutting edge technologies like workflow standards, XML and P2P that will let us eliminate some of the main entrance barriers that Workflow Technology presents today. The main idea behind all this work is our desire of making of Workflow Technology a generally used technology within reach of anyone (enterprise, organization or private individual) interested.

1 Tecnología *Workflow*

La **Tecnología *Workflow*** o **Tecnología de Flujos de Trabajo** es una tecnología de última generación que puede englobarse en el contexto de las Tecnologías de la Información y de las Comunicaciones (TIC). Básicamente, la Tecnología *Workflow* reúne todo un conjunto de técnicas, mecanismos y herramientas que hacen posible la automatización de los procesos de negocio que tienen lugar en las distintas empresas y organizaciones [3].

El elemento central de la Tecnología *Workflow* lo constituyen los denominados **Sistemas Gestores de Flujos de Trabajo** (**SGFT**, del inglés ***Workflow Management System*** (**WfMS**)), gracias a los cuales las empresas y organizaciones pueden alcanzar la denominada **eficiencia operativa**, obteniendo de los diferentes procesos de negocio automatizados un rendimiento máximo mediante la optimización del uso de sus recursos (tanto humanos como no humanos) [1].

1.2 Problema de exclusividad

A pesar de las importantes ventajas que proporciona la utilización de un SGFT, la penetración real de estos sistemas en empresas y organizaciones en la actualidad es muy reducida (o al menos mucho menor de la que cabría esperar), siendo muy pocas las que hoy en día hacen uso de este tipo de sistemas para la automatización de sus procesos de negocio.

La razón fundamental de esta escasa penetración no es otra que el **alto coste de implantación** (tanto de la componente software como de la componente hardware) de las soluciones de flujos de trabajo comerciales existentes actualmente en el mercado. Este alto coste provoca el que, en nuestra opinión, constituye el principal problema o inconveniente de la Tecnología *Workflow* en la actualidad, **problema** al que hemos denominado **de exclusividad de la Tecnología *Workflow***. Y es que la Tecnología *Workflow* es hoy en día una tecnología exclusiva al alcance únicamente de aquellas empresas y organizaciones con un alto potencial económico.

2 Soluciones actuales

El problema de exclusividad de la Tecnología *Workflow* no es un problema nuevo, siendo muy numerosos los grupos de investigación de distintos centros, universidades y empresas que han iniciado trabajos con el objetivo de paliar en la medida de lo posible este problema.

Con todo ello, es posible afirmar que ninguna de las soluciones propuestas hasta la fecha ha terminado finalmente por imponerse, no habiendo obtenido ninguna de ellas una relevancia especialmente importante. Esta escasa repercusión hace que el problema de exclusividad de la Tecnología *Workflow* siga sin resolver, al mismo tiempo que hace plantearnos nuevas vías que hagan posible su solución de forma imaginativa e innovadora.

3 Solución propuesta

3.1 Líneas maestras

La solución que proponemos presenta una característica fundamental como es el hecho de ser una **solución de bajo coste** (tanto en su **componente software** como en su **componente hardware**) al problema de la automatización de procesos de negocio. El objetivo de la misma no es otro que eliminar la principal barrera de entrada que presenta la Tecnología *Workflow* hoy en día, permitiendo de esta forma a cualquier empresa, organización o particular disfrutar de las enormes ventajas que esta tecnología ofrece.

Concretamente, la solución propuesta consiste en la definición de un **protocolo o familia de protocolos** basados en el **intercambio de mensajes XML** que permita resolver de forma **totalmente descentralizada** la problemática asociada a la **automatización de procesos de negocio**, desde su definición en forma de flujo de trabajo, publicación, negociación y configuración, hasta su ejecución propiamente dicha. La idea de hecho no es nueva y consiste básicamente en la definición para el caso de la Tecnología *Workflow* de un protocolo o familia de protocolo similares a los definidos para el caso de la compartición de contenidos en proyectos como, por ejemplo, Gnutella o Freenet [2].

La solución propuesta, aparte de su bajo coste, presenta una serie de características sumamente interesantes que conviene reseñar:

- ✓ **Ruptura con las soluciones de flujos de trabajo comerciales actuales**, eliminando cualquier tipo de dependencia externa con estas soluciones.

- ✓ **Solución abierta basada en estándares establecidos**, especialmente en estándares de *workflow* ofreciendo una solución de flujos de trabajo altamente interoperable no sólo con otras soluciones de flujos de trabajo ya existentes sino también con todo tipo de aplicaciones software.

- ✓ **Solución general**, que permite la automatización de prácticamente cualquier proceso de negocio con independencia de sus características así como del contexto donde dicho proceso tenga lugar.

- ✓ **Solución simple**, en la que prevalecen la simplicidad y la sencillez frente al ofrecimiento de funcionalidades y capacidades avanzadas.

- ✓ **Alto grado de independencia**, prácticamente a todos los niveles (principalmente a nivel de lenguaje de programación, sistema operativo y protocolo de comunicaciones) como consecuencia de la utilización del lenguaje XML que hace posible el desarrollo de implementaciones totalmente interoperables de los protocolos definidos prácticamente para cualquier entorno de ejecución (PC, PDAs, teléfonos móviles, etc.).

- ✓ **Enfoque altamente tecnológico pero también de negocio**, como consecuencia del uso de los últimos avances en cuanto a Tecnologías de la Información y de las Comunicaciones (TIC) se refiere pero también por las medidas adoptadas con objeto de dar publicidad a la solución propuesta. El objetivo final no es otro que involucrar al mayor número de partes interesadas tanto en la evolución y mejora del protocolo o familia de protocolos que se definan como en el desarrollo de implementaciones concretas de los mismos.

3.2 Estado actual y trabajo futuro

La solución propuesta se encuentra actualmente en una fase preliminar de desarrollo como consecuencia de la complejidad del problema que se pretende resolver y de las características propias de la solución propuesta.

Nuestro objetivo más inmediato no es otro que concluir la definición de la que será **primera versión del protocolo o familia de protocolos** definidos, así como proceder al desarrollo de una **primera implementación** de dichos protocolos que permita validar la adecuación de los mismos al problema que se pretende resolver. Concretamente y tras varios estudios realizados, se ha decidido seleccionar la Plataforma JXTA, plataforma *peer-to-peer* (P2P) de código libre desarrollada en el contexto del Proyecto JXTA [4], como plataforma base para el desarrollo de una primera implementación de referencia de los protocolos o familia de protocolos que se definan.

Referencias

[1] Leymann, F., Roller, D. (2000). *Production Workflow. Concepts and Techniques*. Prentice-Hall, Inc. En: Upper Saddle River. New Jersey. Estados Unidos de América. ISBN: 0130217530.

[2] Oram, A. (Editor). (2001). *Peer-to-Peer: Harnessing the Benefits of a Disruptive Technology*. O'Reilly & Associates. En: Estados Unidos de América. ISBN: 059600110X.

[3] Toro, G. (2000). *Análisis y Evaluación de la Arquitectura Estándar para Herramientas de Workflow*. Escuela Técnica Superior de Ingenieros de Telecomunicación. Universidad Politécnica de Madrid. (Proyecto Fin de Carrera)

[4]Toro, G., González, I. (2002). *Plataforma de Última Generación para Trabajo en Grupo Multimedia*. Publicado en las XII Jornadas Telecom I+D 2002 - Innovación en Telecomunicaciones. Madrid, España. 19-21 de Noviembre de 2002.

Sistema de Sindicación de Contenidos en Internet basado en el protocolo ICE (Information and Content Exchange)

Mª Victoria Higuero, Iratxe Etxebarria, Juan José Unzilla, Eduardo Jacob
Departamento de Electrónica y Telecomunicaciones. Área de Ingeniería Telemática.
Escuela Superior de Ingenieros de Bilbao. UPV/EHU. Alda. Urquijo s/n. - 48013 Bilbao (Bizkaia)
Tfno.: 94 601 4206. Fax: 94 601 4259
E-mail: jtphiapm@bi.ehu.es, jtbetsai@bipt106.bi.ehu.es, jtpungaj@bi.ehu.es, jtpjatae@bi.ehu.es

***Abstract**. There are two complementary interests in the content business market of Internet. On one hand appear the content creators or providers, who are interested in finding ways of exploitation of these contents. On the other hand, there are many companies that need to offer quality contents for generating more users traffic in their web pages. In order to satisfy the interests of these two parts, there are intermediaries that carry the contents from the providers of web pages that want to offer them to subscribers. This business model is known as 'content syndication' in Internet, and it has experienced an important increase. This paper presents one syndication model that uses Internet standards based on XML, web application platforms and free-license programs. The result of this design is an open application prepared to be improved and used by no-commercial environments.*

1 Introducción

Internet se ha convertido en una de las mayores fuentes de conocimiento de nuestra sociedad actual. Ante la demanda cada vez mayor de información, aparece la figura del proveedor de contenidos. Las agencias de noticias (EFE, Reuters) que tradicionalmente han suministrado contenidos informativos a medios de comunicación como la radio y la televisión, han ampliado su mercado introduciéndose en el entorno de Internet, generando contenidos digitales para su difusión en la red. Aquí es donde surge la figura del sindicador de contenidos: un intermediario entre las fuentes de contenido (los proveedores) y los portales (los subscriptores) que finalmente quieren ofrecerla a los usuarios de la red.

1.1 Situación actual

En los últimos años se han desarrollado numerosas herramientas de sindicación por parte de importantes compañías como iSyndicate, nFactory, Stars Contents, Interwoven, Vignette Solutions, Adobe o HP, cuyos estudios han contribuido al desarrollo de estándares tales como el protocolo ICE [1] (Information and Content Exchange) utilizado mayoritariamente por los productos de sindicación.

2 Objetivos

El objetivo perseguido es el diseño y desarrollo de un sistema de sindicación de contenidos basado en software libre y apoyado en estándares abiertos, que permita a un sindicador obtener información de varios proveedores de forma que sean clasificados adecuadamente y ofrecidos posteriormente a los subscriptores (clientes del sindicador) según las reglas de envío que se hayan preestablecido, de forma automática o controlada por un administrador, bien a iniciativa del sindicador o del subscriptor.

Se ha tratado de evitar la rigidez de otros sistemas de sindicación y para ello se ha realizado un diseño modular, abierto, ampliable y portable.

3 Descripción de la solución

En la implementación se ha optado por el desarrollo de la aplicación en lenguaje Java, utilizando Servlets y JSPs para atender las peticiones y enviar las respuestas en el lado del servidor, así como el resto de funcionalidades que ofrece la herramienta y para la interfaz a los usuarios. El empleo del motor de aplicaciones gratuito **Tomcat** permite la ejecución de estas aplicaciones.

Se utiliza el **protocolo ICE** para la comunicación entre sindicador y subscriptor. Está basado en XML y define las reglas sintácticas, el formato de los mensajes y las operaciones necesarias para realizar la sindicación. Viaja en el cuerpo de mensajes HTTP POST. El almacenamiento de los datos necesarios para el funcionamiento se realiza con **MySQL**. La implementación TwICE [2] del protocolo ICE, basada en Java y disponible gratuitamente, ha servido de base para la creación del sistema.

4 Arquitectura general

El sistema se compone de tres elementos: el proveedor, el subscriptor y el sindicador. Para la realización de las pruebas de funcionamiento, se ha implementado una aplicación básica de proveedor de contenidos, que sirva como fuente de contenidos digitales para alimentar al sindicador.

4.1 Agente Sindicador

El módulo o agente sindicador es el elemento más importante y centraliza el funcionamiento de toda la aplicación. Es único en el sistema y es el encargado

de recibir los contenidos y documentos de los proveedores, clasificarlos según su tema y ponerlos a disposición de sus clientes, los subscriptores, a modo de ofertas de contenido. Debe mantener una comunicación constante con esos subscriptores, para lo cual se utiliza el protocolo ICE.

El sindicador implementa las operaciones básicas de añadir y eliminar subscriptores, establecer subscripciones, enviar catálogos de ofertas y enviar contenidos de forma automatizada. Así mismo, también es capaz de recibir documentos del proveedor que son almacenados y procesados para determinar a qué subscriptores deben ser enviados. En la Fig. 1 se muestra la arquitectura del módulo.

4.2 Agente Subscriptor

El módulo subscriptor constituye la entidad cliente del sindicador. No es único y pueden existir varios subscriptores con subscripciones establecidas con un mismo sindicador. Cada uno de ellos deberá tener capacidad para comunicarse con el agente sindicador por medio del protocolo ICE y procesar los mensajes que lleguen a iniciativa del sindicador.

Este módulo implementa las operaciones de solicitud de catálogo, establecimiento y cancelación de subscripciones, recepción de contenido en modo push y modo pull, etc. También tiene capacidad de mantener subscripciones con varios sindicadores.

4.3 Agente Proveedor

Para poder validar el sistema se ha desarrollado un agente proveedor con un formato de entrega basado en XML sobre HTTP, que permite una fácil adaptación a proveedores reales. La mayoría de ellos envían los documentos vía email, y algunos permiten personalizar este envío.

5 Funcionamiento

Los tres agentes se comunican entre sí a través de Internet tal y como se muestra en la Fig. 2. La comunicación entre sindicador y subscriptor se realiza mediante el protocolo ICE. El sistema utiliza la versión 1.1 del protocolo, construido en base al lenguaje XML, definido por medio de un documento

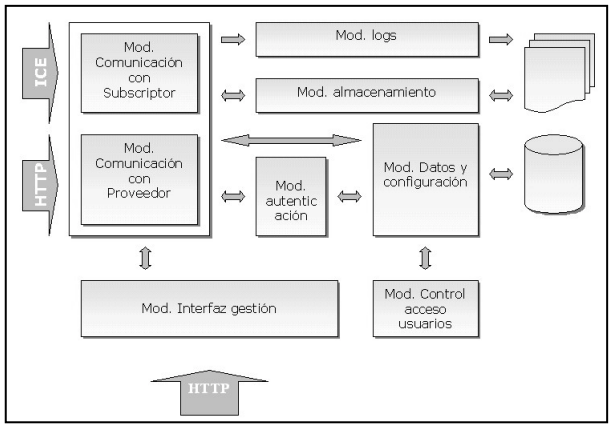


Figura 1: Módulos funcionales del agente sindicador.

DTD denominado ice1_1.dtd [3]. Los mensajes son documentos XML válidos, con un único elemento raíz denominado *ice-payload* y una estructura jerárquica de tags describiendo las operaciones y los datos. Los mensajes ICE viajan dentro del cuerpo de un mensaje HTTP, indicando en el campo content-type de la cabecera HTTP que se trata de un application/x-ice.

Para la comunicación entre proveedor y sindicador, podría utilizarse el mismo protocolo ICE (el sindicador adoptaría el papel de subscriptor), pero esto obliga a suponer que los proveedores implementan el protocolo ICE. Por ello se ha definido un formato de entrega de documentos basando en XML.

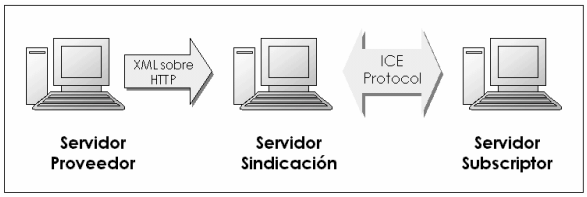


Figura 2: Comunicación entre los tres agentes.

6 Conclusiones y trabajo futuro

El sistema de sindicación presentado permite interaccionar con cualquier sistema de sindicación que haga uso de los estándares más habituales en este tipo de aplicaciones, pudiendo procesar cualquier formato de documento. Su diseño flexible y modular permite incorporar fácilmente cualquier otra forma de captación o envío de documentos, tales como recepción de los documentos por email o FTP.

Como líneas de trabajo futuro cabe citar la conexión con un sistema de pago electrónico para el cobro de los contenidos y la inclusión de un módulo para la gestión de derechos de propiedad intelectual .

Agradecimientos

Este trabajo ha sido financiado en parte por el Gobierno Vasco y por la empresa Sarenet SA en el proyecto OD01UN14, del Plan de Ciencia Tecnología e Innovación (PCTI), 2001.

Referencias

[1] Web oficial del estándar ICE (Information and Content Exchange). http://ww.icestandard.org

[2] Web de entorno de desarrollo de software libre, proyecto Twice. http://twice.sourceforge.net/

[3] N. Webber, C. O'Connell, B. Hunt, R. Levine, L. Popkin, G. Larose. *The Information and Content Exchange (ICE) Protocol version 1.1.* W3C Note 26 October 1998. http://www.icestandard.org/servlet/RetrievePage?site=ice&page=current_specs.

Breve 1B: Miscelánea

Modelado de un sistema de detección de intrusión basado en redes de colas

Armando Ferro, Luis Zabala, Juan José Unzilla, Fidel Liberal
E-mail: {jtpfevaa|jtpzaall|jtpungaj|jtplimaf }@bi.ehu.es
Departamento Electrónica y Telecomunicaciones. Universidad del País Vasco
ETSII e IT de Bilbao. Alameda Urquijo s/n. 48013 Bilbao
Teléfono: 94 6014209 Fax: 94 6014259
Grupo de Ingeniería Telemática

Abstract. *This paper presents an analytical model based in queueing theory which allows to represent the behavior of an intrusion detection system . Besides, an iterative calculation method based in the mean value method has been developed to obtain the most significant parameters of the system, for example throughput. In the future, the main objective of this work is to extend the analytical model of this intrusion detection system to any generic multiprocessor architecture.*

1 Introducción

Las herramientas de seguridad en redes de ordenadores se han convertido en necesarias dentro de los sistemas de empresas u organizaciones. Habitualmente, se utilizan herramientas de prevención como cortafuegos o "firewalls", aunque también existen otros mecanismos como son los sistemas de detección de intrusión. Se necesitan herramientas que monitoricen los sistemas, detecten rupturas, identifiquen malos usos de la red por parte de usuarios legítimos y respondan activamente a los ataques en tiempo real. De todo ello se va a encargar el sistema de detección de intrusos [1].

También son necesarios modelos analíticos que permitan estudiar de forma teórica el comportamiento de un sistema de detección de intrusión. Esto facilitaría la planificación de este tipo de sistemas y el estudio del impacto en la detección de intrusión de las características de las plataformas. En este artículo se presenta un modelo teórico que permite evaluar el rendimiento, desde el punto de vista de capacidad de procesamiento, de un sistema de detección de intrusión en función de las características de la plataforma hardware/software.

2 Modelo analítico para detección de intrusión

Son varias las alternativas para modelar teóricamente un sistema de detección de intrusión [2]. El objetivo final es disponer de un modelo teórico que permita analizar los parámetros fundamentales a la hora de estudiar los rendimientos de un sistema de detección de intrusión. En este trabajo [3] se ha optado por un modelo teórico sencillo basado en redes de colas que permita modelar adecuadamente el comportamiento de un sistema de detección de intrusión y facilite el análisis de los parámetros que se entiende que son los más representativos del sistema.

La red de colas que se propone para modelar un sistema de detección de intrusión es la especificada en la figura 1. Consiste en una red cerrada de colas donde se ha separado en la parte superior una serie de colas multiservidor que representan la capacidad de procesamiento del sistema de detección de intrusión y en la parte inferior se modeliza una cola simple que simula la inyección de tráfico de red con tasa λ. Se han distinguido en la red cerrada cuatro etapas diferentes:

- La etapa de sistema. Consiste en una cola con capacidad m_{KK}. Esta etapa representa el tiempo de tratamiento perdido por el sistema de detección de intrusión a nivel de núcleo de sistema operativo. Incluye el tratamiento de los controladores de dispositivos y la atención del kernel del sistema a la llegada de paquetes.

- La etapa de tratamiento básico. Está formada por dos colas con capacidad m_{Tk} y m_{Tu}. Esta etapa representa el tiempo que dedica el sistema de detección de intrusión al tratamiento básico de los paquetes que captura de la red (estudiar las cabeceras de control y determinar si un paquete es susceptible de análisis o no).

- La etapa de análisis. También está formada por dos colas con capacidad m_{Ak} y m_{Au}. Esta etapa simula el tratamiento de análisis que un sistema de detección de intrusión realiza sobre los paquetes sospechosos. Como no todos los paquetes tienen por qué ser analizados se ha definido una tasa q_a que indica la proporción de paquetes recibidos que están sujetos a análisis.

- La etapa de inserción de tráfico. Formada por una cola simple de capacidad λ. Esta etapa simula la llegada de paquetes al sistema de detección de intrusión con una tasa λ de llegadas.

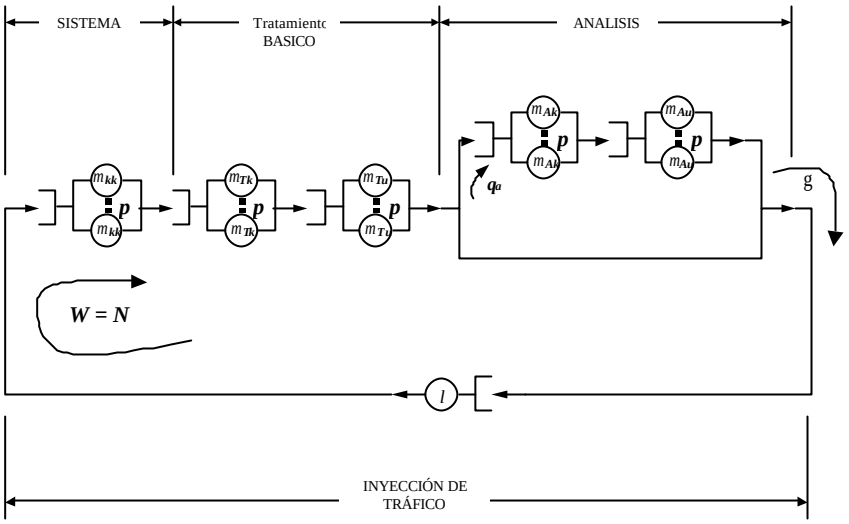


Fig. 1: Modelo general de sistema de detección de intrusión.

Cada cola de servicio se ha representado como una cola con múltiples servidores, tantos como procesadores. Se ha elegido la representación de múltiples servidores para destacar la posibilidad de paralelizar en cada etapa el procesamiento de cada uno de esos procesadores.

Todo el tráfico capturado, pasará por la etapa de sistema y por la de tratamiento básico. Sin embargo, no todo ese tráfico será susceptible de análisis. Esto dependerá de si se detectan evidencias o no de tráfico sospechoso. En este modelo se definirá el coeficiente q_a que indicará la proporción de paquetes capturados que son susceptibles de análisis.

3 Estudio analítico del modelo

La solución analítica del modelo puede plantearse para unas condiciones determinadas considerando eventos de llegada de carácter poissoniano y tasas de servicio exponenciales. También se considera una serie de abstracciones específicas tanto para el caso del cálculo del equivalente de Norton de un sistema de detección de intrusión como para el modelo general con carga de tráfico. El desarrollo de todas estas ecuaciones, incluso teniendo en cuenta estas abstracciones, se hace demasiado complejo. Para simplificar el cálculo se propone desarrollar un método iterativo basado en el análisis del valor medio [4]. La aplicación de este teorema exige tener en cuenta las dependencias entre unos estados y otros dentro de la red cerrada de colas en un diagrama de estados complejo, donde además, las transiciones de estado se pueden realizar con diferentes probabilidades, al tratarse de una red con colas cuya capacidad de servicio es dependiente del estado. Las transiciones de estado no son posibles entre estados cualesquiera, sino que sólo se pueden producir entre estados próximos.

El método de cálculo iterativo de la red cerrada de colas se basa en resolver determinados estadísticos de interés de la red en cada etapa utilizando los datos obtenidos en la etapa anterior. Se pasa de una etapa con N paquetes a la siguiente con N+1 paquetes añadiendo un paquete más a la red cerrada de colas una vez que se esté en situación estable.

Conocidas las relaciones iterativas de las probabilidades entre etapas diferentes y aplicando la fórmula de Little se calcula la distribución de probabilidades de estado para cualquier etapa. Finalmente, partiendo del modelo general sometido a carga de red con una intensidad λ determinada, se puede calcular el volumen de tráfico procesado o throughput por el sistema de detección de intrusión.

Por último, cabe destacar que se han realizado una serie de medidas sobre plataforma Linux, con objeto de comparar el throughput teórico obtenido a partir del método propuesto y el valor obtenido experimentalmente.

Referencias

[1] B. Mukherjee, T. L. Heberlein, K. N. Levitt "Network Intrusion Detection". IEEE Network, 8(3):26-41. Mayo/Junio 1994.

[2] E. D. Lazowska, J. Zahorjan, G. Scott, K. C. Sevcik "Quantitative System Performance. Computer System Analysis Using Queing Network Models". Prentice Hall, 1984.

[3] A. Ferro. "Propuestas de diseño de un sistema de detección de intrusión y definición de un modelo analítico para arquitecturas multiprocesador", Tesis Doctoral. Universidad del País Vasco. Abril, 2002.

[4] M. Reiser. "Mean value analysis and convolution method for queue-dependent servers in closed queing networks", Performance Evaluation, vol. 1, no. 1, pp. 7-18, Enero 1981.

Evaluación del acoplamiento de QoS con HMIP en entornos de micromovilidad integrados con UMTS

Luis A. Galindo[1], Juan M. Vázquez[1], Pedro M. Ruiz[2], Emilio J. García[2]

[1]Telefónica Móviles España S.A.
C/ Cerro de los Gamos, 1, 28224 Madrid
Teléfono: 680 01 93 97 Fax: 680 01 79 57
E-mail: {galindo_la, vazquez_jm1}@tsm.es

[2]Agora Systems, S.A.
C/ Aravaca, 12 3ºB, 28040 Madrid
Teléfono: 915 33 58 57 Fax: 915 34 84 77
E-mail: {pedro.ruiz, emilio.garcia}@agoratechnologies.com

***Abstract**. QoS is one of the big challenges Internet protocol designers have faced during the last years. While overprovisioning of network bandwidth is a usual solution due to the complexity of the problem, this approach is not viable for wireless networks, where spectrum is scarce. Moreover, the adoption of an All-IP paradigm shows the necessity of mechanisms which can deal with the mobility of the nodes. We present an integration of RSVP with HMIP to improve the QoS guarantees given in micro-mobility scenarios. Extensive simulations are used to demonstrating how this approach clearly outperforms both for TCP and UDP the traditional use of RSVP for these scenarios.*

1 Introducción

El mundo de las telecomunicaciones evoluciona hacia redes basadas completamente en IP, en las que terminales móviles equipados con interfaces inalámbricos (GPRS, UMTS o 802.11 en sus diferentes variedades) acceden a aplicaciones antes sólo pensadas para redes fijas.

Los nuevos usuarios esperan obtener un acceso a los nuevos servicios de manera flexible, independiente de la tecnología de acceso y la localización. Sin embargo, las aplicaciones multimedia requieren unas garantías de Calidad de Servicio (QoS) similares a las que proporcionan las redes de conmutación de circuitos, con recursos dedicados. Los mecanismos de QoS a nivel de red como RSVP[1] fueron diseñados para redes fijas, y su aplicación a entornos móviles, donde los terminales varían su punto de acceso a la red con frecuencia, puede dar lugar a violaciones de las reservas (retardos elevados, pérdidas de paquetes o incluso negación del servicio). El reto consiste en mantener el nivel de servicio solicitado inicialmente por la aplicación a medida que el terminal cambia su localización.

Los mecanismos de QoS y movilidad han evolucionado de manera independiente, y por ello se necesitan mejoras. En este artículo se presenta una propuesta de acoplado de protocolos mediante señalización de modo que la interacción hace que el impacto en las reservas sea mínimo.

El resto artículo se organiza como sigue: la **sección 2** describe los mecanismos de micromovilidad y el enfoque propuesto de acoplado. La **sección 3** presenta los resultados de simulaciones. Finalmente, la **sección 4** recoge algunas conclusiones.

2 Micromovilidad y QoS

El uso de protocolos de micro-movilidad mejora la eficiencia de la movilidad cuando se producen traspasos de forma frecuente, pero no se solucionan los problemas a la hora de ofrecer calidad de servicio: es necesario reservar recursos en la nueva ruta, y la característica *Soft State* de RSVP tiene un tiempo de refresco demasiado elevado (30 s), que hace deseable otro mecanismo.

2.1 Acoplado de protocolos

El acoplado de protocolos consiste en implementar un mecanismo de señalización que permita al protocolo de calidad de servicio ser avisado, para que se encargue del restablecimiento de las reservas tan pronto como se detecte un cambio de ruta. En este caso se elige un *acoplamiento débil*, pues la solución de *acoplamiento fuerte*, transportando la información de QoS y micro-movilidad conjuntamente, va en contra del diseño extremo a extremo de Internet.

Para llevar a cabo el estudio, emplearemos los Servicios Integrados (*IntServ* - RSVP), ya que nos centramos en la red de acceso y, al mismo tiempo, IntServ está diseñado para trabajar con los protocolos de encaminamiento actuales y futuros *unicast* y *multicast*. Como protocolo de micro-movilidad emplearemos HMIP[2], desarrollado en el seno de IETF, y con grandes posibilidades de ser el estándar de facto.

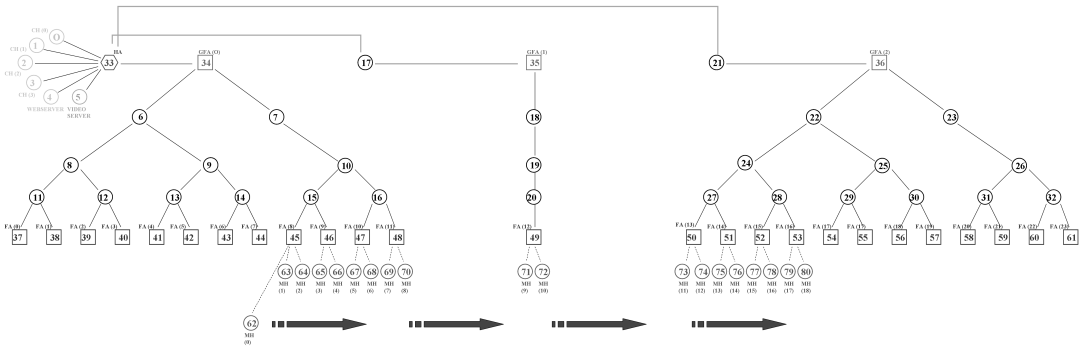


Figura 1: Modelo del parque empresarial en NS

Como técnica complementaria a la que es objeto de estudio, se ha hecho uso de la priorización de la señalización de la calidad de servicio.

3 Resultados

3.1 Escenario de red

Como escenario de red hemos tomado el despliegue en un parque empresarial (Fig. 1). El patrón de tráfico es una videoconferencia (video H.263 sobre UDP), con una tasa media de 275 Kbps. El nodo móvil MH(0) se moverá entre dos redes WLAN/TDD de la empresa, pasando por un nodo intermedio UMTS. Hacia el resto de nodos móviles hay un tráfico interferente de 450 Kbps.

El tráfico del nodo móvil comienza en $t = 5s$. Los nodos adyacentes comienzan a recibir el tráfico interferente. Los traspasos comienzan en $t = 10s$, y se producen cada 40s, usando la micromovilidad, hasta que en $t = 130s$ se pasa a la red UMTS. La simulación acaba en $t = 300s$. La reserva realizada en la red de acceso es de 300 Kbps.

3.2 Análisis de resultados

Los resultados son producto de las simulaciones con NS[3]. Tal como se aprecia en la Fig. 2, donde se muestra el ancho de banda recibido por el nodo móvil en cada situación, el caso ideal se alcanza cuando no hay tráfico interferente. Es interesante notar la comparación entre los casos de reserva sin acoplado y con acoplado en los instantes inmediatamente posteriores a los traspasos: sin acoplado, las reservas se pierden y no se recuperan hasta que se inicia el proceso de reparación de caminos. Al emplear acoplamiento, se aprecia como justo después del traspaso se fuerza la recuperación de la reserva, que a los pocos segundos vuelve a estar operativa. El resultado se acerca más al ideal.

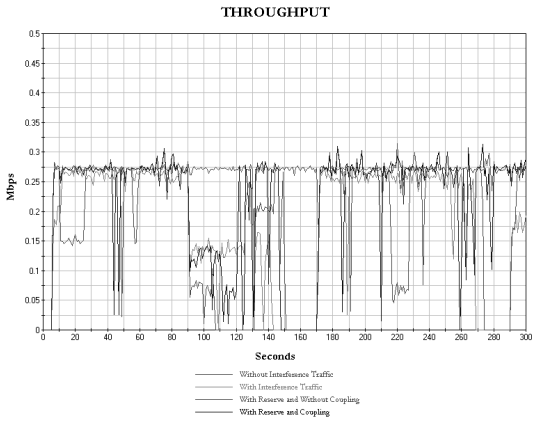


Figura 2: Comparativa del ancho de banda para videoconferencia

4 Conclusiones

Las simulaciones realizadas en nuestro escenario de red empresarial muestran una importante mejora en el rendimiento de las aplicaciones tanto para aplicaciones de tiempo real UDP, como transporte fiable TCP. La principal ventaja del acoplamiento es que, si bien el propio traspaso en sí no puede ser acelerado, sí que se consigue que las reservas se reinstalen tan pronto como el nuevo camino sea estable. Como mejora adicional, se ha empleado también un mecanismo complementario de priorización de paquetes de señalización, que también ha demostrado ser efectivo.

Agradecimientos

Los autores quieren agradecer la colaboración de sus compañeros Alberto López, Juan Antonio Fernández y Almudena Bueno.

Referencias

[1] R. Braden et. Al. “Resource ReSerVation Protocol (RSVP) – Version 1, Functional Specification”. IETF RFC 2205. Sept. 1997.

[2] H. Soliman, C. Castelluccia, K. ElMalki, L. Bellier, “Hierarchical MIPv6 mobility management”, IETF Internet Draft. July 2001.

[3] The NIST Network Simulator version 2, http://www.isi.edu/nsnam/ns, 2002.

Evaluación de Protocolos multicast fiables para distintas topologías Internet

David Salleras Soler y José Luis Melús Moreno

Dpto de Ingeniería Telemática de la Universidad Politécnica de Catalunya

C/Jordi Girona 1-3, Mod. C3, Campus Nord 08034 Barcelona

Tfno 934016021, e-mail: teljmm@mat.upc.es

Abstract. We evaluate two reliable multicast protocols, PGM y SRM, over different network topologies, generated by GT-ITM, when errors are produced during the data transmission. The evaluated metrics are the latency of packet recoveries the average number of repeated retransmissions, analyzing the relationship between them, To do that the simulator-2 is used.

1 Introducción

En esta comunicación se analiza el comportamiento de dos de los protocolos multicast fiable más conocidos, Scalable Reliable Multicast (SRM) [1] y Pretty Good Multicast (PGM) [2] sometidos pérdidas provocadas artificialmente en los enlaces. La comunicación que aquí se presenta estudia la influencia de la topología de la red y el tamaño del grupo analizado. El modelo de pérdidas utilizado aquí admite la perdida de cualquier paquete, ya sea original, de retransmisión o NACK.

2 Protocolos multicast fiables

La elección del protocolo PGM y SRM en este estudio se ha debido a varias razones; su código fuente está en el simulador ns-2. Los dos son del tipo *receiver-initiated error recovery*, sin embargo, la implosión de paquetes hacia la fuente la evitan por mecanismos distintos. En SRM todos los paquetes que intervienen en el procedimiento de recuperación de paquetes perdidos son transmitidos de forma multicast, tanto las *reparaciones* como las *solicitudes correspondientes.* Los temporizadores para las *solicitudes* se escogen a partir de la variable aleatoria uniforme definida entre $2^i[C_1 d_{S,A},(C_1+C_2)d_{S,A}]$. En el caso de *reparaciones* la distancia se elige entre $[D_1 d_{A,B},(D_1+D_2)d_{A,B}]$ siendo $d_{S,A}$ la distancia estimada entre el emisor y el nodo que ha perdido el paquete y $d_{A,B}$ la existente entre el nodo que ha perdido el paquete y el que responde a la pérdida El agente *adaptive,* se adapta para obtener el menor número de retransmisiones, el *deterministic,* actúa sobre la base de la distancia entre nodos y en el *probabilistic,*de una elección aleatoria.. El agente *adaptive,* se adapta para obtener el menor número de retransmisiones, el *deterministic,* actúa sobre la base de la distancia entre nodos y en el *probabilistic,* resulta de una elección aleatoria.

Tabla 1: parámetros de los agentes SRM

	C1	C2	D1	D2
Adaptive	2	2	1	1
Deterministic	2	0	1	0
Probabilistic	0	2	0	1

PGM utiliza los *routers* para bloquear los NACKS duplicados en su camino hacia la fuente y distribuir las *reparaciones* desde la fuente a las ramas donde hay receptores que las hayan solicitado.

3 Escenario de trabajo

Se han utilizado cuatro topologías de red distintas para efectuar la evaluación de estos protocolos: random, hier, transit-stub, representan la estructura *red troncal to-ISP-to-local*. Finalmente, binary-tree, que es regular y fácil de visualizar. La versión del simulador ns-2 utilizada es la ns-2.1b8a. Las redes analizadas están formadas por 100 y 200 nodos y en la generación de pérdidas de paquetes se utiliza un ruido de base del 1% en todos los enlaces de la red. El emisor emite 1 MB con una fuente de bits constante. El tamaño de paquete es de 1024 bytes, la tasa de transmisión de la fuente es de 410kbps y la capacidad de los enlaces es de 1.5Mbps. Se ha variado el tamaño del grupo de receptores para cada topología, incluyendo desde un 5% a un 70% del total considerados. Por último, se varía la ubicación de emisor y receptores. En conclusión, se han generado distintas distribuciones del emisor y los receptores (10) y distintas disposiciones de cada una de ellas, 5 para cada tipo de topología. Los

resultados se han obtenido realizando el promedio de todas ellas.

4 Resultados

La topología binary-tree facilita a PGM obtener el máximo beneficio debido al *router assisted*. Con la estructura jerárquica, solo hay un camino desde los receptores a la fuente, por lo que ésta solo tiene que realizar una retransmisión para un paquete, salvo que se pierda esta retransmisión. Para SRM *adaptive*, los receptores detectan la pérdida a la vez y se sirve la reparación para lo que tienen han perdido el paquete. Él agente utilizado adapta sus temporizadores e intenta aumentarlos para minimizar las retransmisiones, Fig. 1. Esta solución produce un aumento de latencia. La mejor opción es usar el agente *probabilistic,* con temporizadores aleatorios, obteniéndose simultaneamente menor numero de retransmisiones y de latencia. Para las topologías; hier, random y transit-stub el protocolo SRM obtiene resultados muy similares.

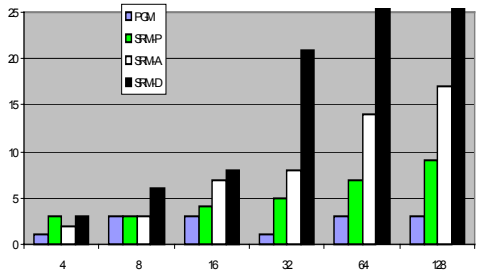


Figura 1: Retransmisiones en una binary-tree

Para PGM, las topologías hier y random presentan menor número de retransmisiones y valor de latencia. La topología transit-stub es más estirada y presenta una mayor distancia entre receptores de distintos clusters y la fuente, por lo que no se produce tanta "ayuda" como en la topología hier. En la topología random, hay muchos caminos con destino a la fuente así el número de retransmisiones se sitúa entre las otras dos.

PGM no obstante, presenta una latencia muy pequeña, en todas las topologías, por debajo incluso de la latencia unicast en hier y random. Este protocolo se muestra siempre independiente del tamaño de grupo. En cambio, para SRM *adaptive* si bien se obtienen menos retransmisiones, la latencia de recuperación es doble e incluso triple de la unicast, ver Fig.2. Los otros agentes SRM (*deterministic* y *probabilistic*), obtienen latencia menor, al precio de aumentar el número de retransmisiones. Dependiendo de cada situación se debería escoger uno u otro agente. Todos los agentes SRM son sensibles al tamaño del grupo no así en el caso del PGM y al aumentar el número de nodos de la red produce un aumento en la latencia, apenas apreciable en el PGM

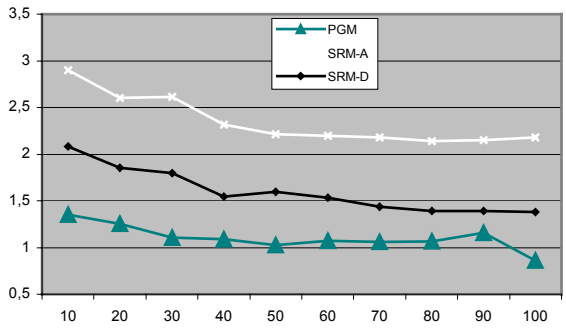


Figura 2: Latencia Adaptive vs Deterministic vs PGM en transit-stub

5 Conclusiones y trabajo futuro

El protocolo PGM presenta menor latencia que SRM, independientemente de la topología y del agente utilizado por este último. Esto le convierte en un protocolo más escalable. No obstante, realiza más retransmisiones que SRM en todas las topologías a excepción de binary-tree. Aquí SRM tiene dificultad para trabajar debido a la regularidad de la topología, ya que todos los receptores tienen la misma distancia a la fuente. En este caso la mejor solución es utilizar un agente *probabilistic*. La latencia y retransmisiones están interrelacionadas. Cuando hay baja latencia hay más retransmisiones y viceversa. Las bajas latencias obtenidas en SRM deben utilizar agentes del tipo *deterministic* y si se quiere bajas retransmisiones, *adaptive*. Existe un equilibrio entre velocidad, menor latencia, que se quiere alcanzar frente al numero de retransmisiones que se puede admitir. Seria interesante estudiar estos protocolos con otro tráfico de background en la red como; FTP, HTTP, video, etc.. Este análisis permitiría conocer el tipo y valor de este tráfico que hace que la latencia aumente o como las retransmisiones repetidas que se produzcan afectan a la adaptación del agente *adaptive* en SRM.

Referencias

[1] Floyd, S., Jacobson, V., Liu, C.-G., McCanne, S., Zhang, L., - *A Reliable Multicast Framework for Light-weight Sessions and Aplication Level Framing.* - IEEE/ACM Transactions on Networking, December 1997.

[2] RFC3208, PGM Reliable Transport Protocol Specification, 2001.

Evaluación de protocolos multicast fiables con tráfico de background FTP

Jorge Lores Ruiz y José Luis Melús Moreno

Dpto de Ingeniería Telemática de la Universidad Politécnica de Catalunya (UPC) C/ Jordi Girona 1-3, Mod C3, Campus Nord, Tfno 934016021, e-mail teljmm@mat.upc.es

Abstract. We use the ns-2 simulator to evaluate the efficiency of SRM (Scalable Reliable Multicast) and PGM (Pragmatic General Multicast) protocols with FTP background traffic in the links. This analysis confirms that MFTP is non real time and PGM presents better performance than SRM for the evaluated latency and the number of duplicated retransmissions.

1 Introducción

Los trabajos analizados por otros autores consideran que solo circula tráfico multicast por la red o que las perdidas se provocan artificialmente mediante funciones de pérdidas, introduciendo determinados porcentajes de paquetes perdidos en enlaces seleccionados. En redes reales uno de los aspectos fundamentales, además de su escalabilidad, es el de la coexistencia con otros tráficos. Aquí el tráfico de background que se considera es FTP. Se realiza una breve descripción de las características básicas de los protocolos, se explica el entorno de trabajo utilizado para desarrollar las simulaciones. Finalmente, se exponen los resultados más relevantes.

2 Protocolos multicast fiables analizados

Se han desarrollado un elevado número de protocolos multicast fiable. En esta comunicación, se comparan entre sí tres de los más destacados, Scalable Reliable Multicast (SRM), [1], y Pretty Good Multicast (PGM), [2] utilizando el simulador ns-2. PGM se diferencia de los otros dos debido al mecanismo *router-assisted* que necesita implementar e cada uno de los *routers*. Todos los protocolos son *receiver-initiated error recovery*, pero se diferencian en la forma de reducir el tráfico de feedback hacia la fuente.

3 Entorno de simulación y escenario de trabajo

Se ha utilizado la versión ns-2.1b8a [3]. SRM esta incluido en el paquete básico del simulador, mientras que PGM fue añadido posteriormente.

La red simulada consiste en 112 nodos, y formada por una parte troncal y una serie de *end-systems* que actúan a modo de redes LAN anexadas a cada uno de los nodostroncales. El emisor envía a los receptores, ubicados de forma aleatoria, un fichero de 1MB de información de forma multicast, con una tasa de 410Kbps. Por la red circula un tráfico de background FTP variable paralelamente al envío de la información multicast. Las sesiones se definen de la siguiente forma:

→ Sesiones Background FTP. Cada sesión FTP está formada por ráfagas. El tamaño medio de la sesión FTP es de 200KB, y el de las ráfagas de 80KB. El número de ráfagas por sesión es una variable aleatoria uniforme entre 1 y 4, con un valor medio de 2. La siguiente tabla resume estas características de las sesiones.

Tabla 1. Características de las sesiones FTP

Parámetros FTP	**Distribución**	**Valor medio**
Session interarrival time	Exponencial $f(\chi) = \lambda' e^{-\lambda'\chi}$	Tiempo_de_simulación/ número_sesiones_ftp
Burst interarrival time	Exponencial $f(\chi) = \lambda e^{-\lambda(\chi-4)}$, $\chi \geq 4$	5 segundos
Burst size	Pareto de primera clase $f(\chi) = \beta a^{\beta} \chi^{-\beta-1}$ $\beta = 1.18$, $a = 12.203$	80.000 bytes

El número de sesiones de tráfico background es de 50 a 2050 y las métricas estudiadas son latencia, número de retransmisiones duplicadas y tiempo total hasta la recepción de la información.

4 Resultados

En la Figura 1 se observan los valores de latencia. Los de PGM son sensiblemente inferiores a los de SRM, para cualquiera de los agentes SRM considerados. La gráfica indica que los receptores SRM han de esperar en media 6 veces más a recibir los datos perdidos que con PGM. En este sentido, la diferencia de eficiencia entre los protocolos SRM y PGM es muy favorable al segundo.

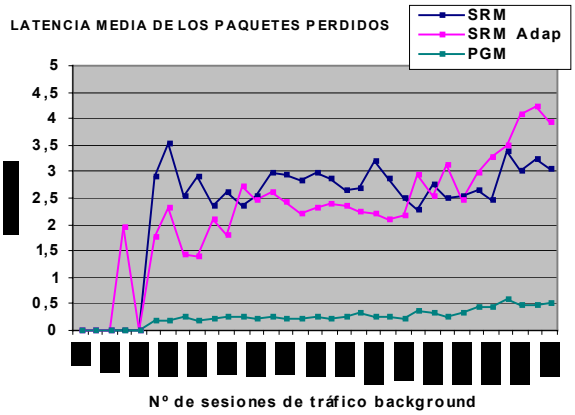


Figura 1. Comparación de la latencia media de los paquetes perdidos

El número de retransmisiones duplicadas para un mismo paquete se presenta en la Figura 2. PGM es mejor A partir de las 1250 o 1350 sesiones de *background,* la introducción de entre 250 – 270 MB de tráfico FTP en la red durante la simulación, los valores de retransmisiones para el PGM empiezan a igualarse a los correspondientes al SRM *adaptive.* El aspecto más importante a destacar es la diferencia de eficiencia entre SRM y PGM. Los valores de SRM *adaptive* son inferiores y es capaz de simultanear una reducción de las retransmisiones con una menor latencia de recuperación. Desde ese momento elige la política de aumentar el valor de los temporizadores para reducir las retransmisiones duplicadas, y aumenta la latencia y el tiempo máximo de recepción de la información por parte del último receptor.

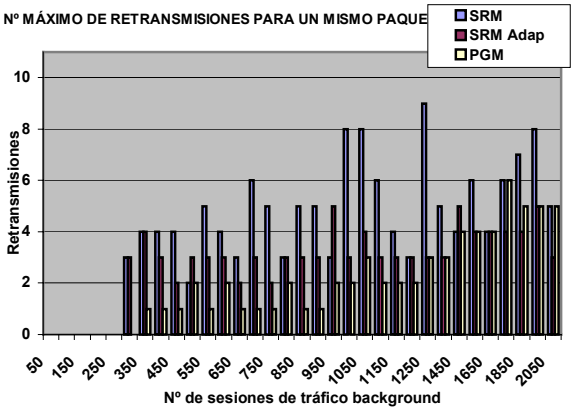


Figura 2. Número máximo de retransmisiones para un mismo paquete

5 Conclusiones y lineas futuras

Se ha realizado un estudio del comportamiento de los protocolos multicast fiable SRM y PGM en un escenario con tráfico background FTP. La latencia en SRM siempre es superior a PGM , cualquiera que sea la implementación adoptada. A medida que se aumenta el tráfico FTP de background se produce un aumento de la métrica correspondiente pero manteniendo PGM los mejores valores. PGM se erige como el protocolo más adecuado para ser implementado en redes reales. Como línea futura se propone analizar el comportamiento de estos protocolos bajo la influencia de otros tipos de tráfico de background, habitual en Internet, como es el tráfico MPEG-4 sobre fuentes UDP o el tráfico WEB.

Referencias

[1] Floyd, S., Jacobson, V., Liu, C.-G., McCanne, S., Zhang, L., *A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing*. IEEE/ACM Transactions on Networking, Diciembre de 1997

[2] RFC 3208, *PGM Reliable Transport Protocol Specification*.2001 http://www.ietf.org/rfc/rfc3208.txt?number=3208

[3] UCB/LBNL/VINT. Network simulator NS-2. http://www.isi.edu/nsnam/ns-2

Red Telemática de Sensorización Medioambiental de Callús

Ricard Moraleda Gareta y Jordi Mataix Oltra
Departament d'Enginyeria Telemàtica. Universitat Politècnica de Catalunya.
Av. Canal Olímpic s/n. Escola Politècnica Superior de Castelldefels
EPSC-UPC. 08860 Castelldefels (Barcelona) Teléfono: 934137000 Fax: 934137007
E-mail: {ricard.moraleda, jordi.mataix}@upc.es

Ramon Fons Vilardell y Carles Cambronero Balaguer
Fundació Privada Aplicació
Ps. Anselm Clavé 19,bajos 08262 Callús (Barcelona) Teléfono: 936930016 Fax: 938361994
E-mail: {ramonfv, ccambronero}@callusdigital.org

Abstract. The aim of this paper is the exposition of a continuous and automated environmental measurement system. This arises from the necessity to measure certain problematic environmental polluting parameters for the health of the citizen of Callus. The propose solution is the study, development and unfold of three independent subnetworks, air, water and noise, forming an only environmental sensorization network. Therefore, a control of the polluting phenomena will be had, for the information to the citizen and the immediate performance, in case the allowed values are exceeded. This document, also makes reference to the unfold of a web cameras monitoring system on the municipality telecommunication network.

1 Introducción

El año 1995 el ayuntamiento de Callús y la escuela del pueblo se sumergieron en unos proyectos educativos que se fundamentaban en la utilización de técnicas telemáticas. Un claro ejemplo de la aplicación de las nuevas tecnologías la encontramos en las interrelaciones existentes entre el medio ambiente y las telecomunicaciones.

En el año 2000 se hizo una Diagnosis Ambiental [1] del municipio de Callús por la Diputación de Barcelona. El análisis global de la información obtenida fue: contaminación mínima de las aguas subterráneas a causa de la salinidad procedente de las minas salinas de Súria, algunas industrias tienen emisiones de humos que superan los valores límite permitidos de opacidad y partículas en suspensión totales (PST), los niveles acústicos generados por la circulación de vehículos a motor, tráfico de camiones, la velocidad, las actividades de ocio del núcleo urbano superan ciertos límites.

Para evitar estos tipos de contaminación mínima se ha estudiado la manera de hacer llegar el conocimiento al ciudadano y a personal interno mediante un proceso de sensorización, con el objetivo de cumplir los niveles impuestos por la normativa. Se basará en tener mecanismos e instrumentos propios para captar datos y ser transmitidos de forma automática, a través de una red telemática a un centro de coordinación municipal. Esta información deberá ser actualizada permanentemente y difundida en el sitio web del municipio.

2 Objetivo

El objetivo del trabajo es proyectar los elementos de medida, la red telemática necesaria para que los datos se reciban en una central y los mecanismos básicos de procesado y representación de dicha información. Es un trabajo fundamentalmente de integración de equipos, redes, protocolos y aplicaciones en las redes ya existentes, FastEthernet cableada y Wireless LAN. El sistema propuesto será escalable para futuras ampliaciones, referentes a equipos y tipos de medida.

3 Tipos de sensorización

Para poder actuar de forma rápida o para evitar desastres mayores en cuanto a la contaminación ambiental (aire, agua y ruido), es muy importante ir midiendo continuamente estos parámetros para saber en cualquier momento cuál es el problema, si lo hay, e identificar la fuente de contaminación.

La sensorización del aire se refiere a la medición de las PST y otros contaminantes atmosféricos.

La sensorización del agua se refiere a la medición de la Salinidad a través de otros parámetros del agua. [2]

La sensorización del ruido se refiere a la medición de los niveles acústicos producidos para que no sobrepasen los márgenes impuestos por las normativas. Se suelen medir los siguientes parámetros: Ls, Lf, Leq, Lcpk, L10, L50 y L90.

3.1 Aire

La sensorización de aire consiste en hacer medidas de éste para poder conocer el grado de cumplimiento de los niveles de los contaminantes atmosféricos, establecidos en la normativa vigente y realizar actuaciones de saneo atmosférico en zonas degradadas. Para ello se utilizará una estación con analizadores de dióxido de azufre, partículas en suspensión, óxidos de nitrógeno, hidrocarburos, monóxido de carbono, sulfhídrico, ozono y benceno para monitorización, a través de una conexión RS-232 en local, o a través de la LAN con una estación central en la Fundació Aplicació o la integración con la Red de Vigilancia y Previsión de la Contaminación Atmosférica de Catalunya – RVPCA [3] de la Generalitat de Catalunya a través de módem.

3.2 Agua

Para la sensorización de los parámetros más relevantes del agua se sigue el siguiente esquema, situando el punto de medición en uno de los depósitos de agua del municipio.

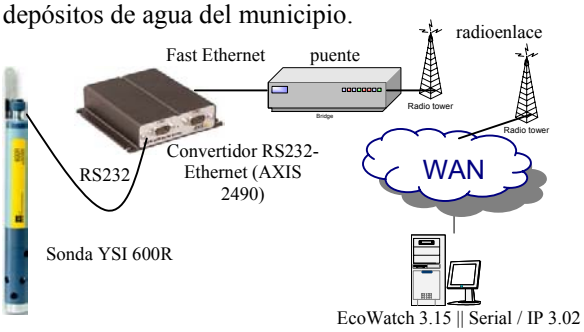


Figura 1: Esquema de la red para medir parámetros del agua

El esquema está basado en una sonda con salida RS-232 conectada a un convertidor Serie-Ethernet y a la red del municipio con la ayuda de un radioenlace punto a punto, ya que actualmente en este punto de medición no existe red cableada ni cobertura inalámbrica. Los datos serie recogidos por el convertidor son encapsulados en paquetes IP, los cuales pueden ser redirigidos y capturados por un PC conectado en cualquier punto de la LAN o WAN con el software de gestión de la sonda y un redirector de puertos Serie/IP, el cual crea un puerto COM virtual en el PC, lo asocia a la dirección IP y puerto TCP del convertidor y encamina los paquetes IP a la aplicación serie y al revés, ya que la aplicación no permite una configuración de red.

3.3 Ruido

Para la sensorización del ruido se utilizará un sonómetro con salida RS-232 y, para dar una mayor movilidad al sistema utilizaremos tecnología Wireless (802.11b). Esto es que para poder enviar estas medidas a través de la interfaz radio, se necesita el convertidor MSS-VIA, de datos RS-232 al estándar 802.11b mediante una tarjeta PCMCIA comercial. Este dispositivo contiene un servidor web en su interior y se le configurará una IP y puerto TCP. Esto hará que desde cualquier PC, conectado a la LAN y con el soporte de un software redirector de puertos Serie a IP (Lantronix Redirector 2.1/1), el cual crea un puerto COM virtual en el PC asociado a la IP y puerto TCP del MSS-VIA, se puedan recoger las medidas y almacenarlas para posteriores estudios con la ayuda del software de gestión del sonómetro y para dar una mayor transparencia en las medidas, referente al ciudadano, se realizará un proceso de puesta de los datos en la página web del pueblo de forma automática siguiendo el esquema siguiente:

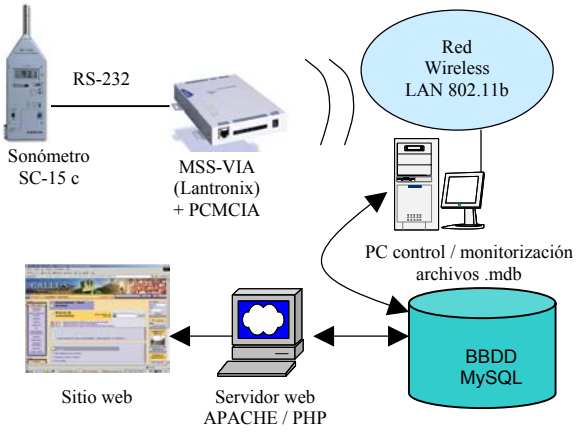


Figura 2: Esquema de la red para medir niveles de ruido

3.4 Televigilancia

Para una mayor seguridad de las instalaciones públicas de Callús se procede a la instalación de videocámaras en edificios públicos del pueblo. Estas videocámaras se conectan directamente a la red FastEthernet o Wireless LAN, y se les configura una dirección IP privada. De esta forma se permite un control remoto mediante un navegador web o el software Milestone XXV, el cual permite grabar secuencias de imágenes por cambios de píxels en la imagen, mandar e-mails de aviso y almacenar las grabaciones en un disco duro local o a través de FTP.

Referencias

[1] Sistema d'indicadors ambientals per a una gestió sostenible del municipi de Callús (febrer 2002). Àrea de Medi Ambient – Servei del Medi Ambient (Diputació Barcelona).

[2] Standard methods for the examination of water and wastewater. American Public Health Association. 18a ed. Washington American Public Health Ass. 1992 (Section 2520 D. Algorithm of Practical Salinity)

[3] Red de Vigilancia y Previsión de la Contaminación Atmosférica de Catalunya (RVPCA): http://www.gencat.es/mediamb/cast/aire/e_cqaire.htm

QoS-Meter: Monitorización de aplicaciones web mediante técnicas de extracción de información en fuentes semi-estructuradas

Vicente Orjales[1], Justo Hidalgo[2], Gustavo López[3], Alberto Pan[3], Victor Carneiro[1]

[1]Departamento de Tecnologías de la Información y las Comunicaciones.
Universidade da Coruña.
Campus de Elviña. 15071. A Coruña
vorjales@denodo.com, viccar@udc.es

[2]Departamento de Ingeniería Informática.
Universidad Antonio de Nebrija.
Campus Dehesa de la Villa. 28040 Madrid
jhidalgo@nebrija.es

[3] Departamento I+D.
Denodo Technologies.
Almirante Fco 5, bajo. 28040 Madrid
glopez@denodo.com, apan@denodo.com

Abstract: *Many current applications and information systems are built by making broad use of Internet technologies, in which the web is used as an access channel. It is defined as an open architecture which facilitates the use of the different applications through a homogeneous interface, from every single geographic place. However, from the point of view of systems management, these applications are much more difficult to be monitored, because of the inherent complexity of a highly-distributed, broadly-accessed architecture. This paper presents a monitoring solution which provides an accurate measurement about the quality of service of any web application from the user's point of view and in a non-intrusive way, by making use of different techniques from heterogeneous and semistructured information retrieval.*

1 Introducción

La mayoría de los sistemas de información y aplicaciones actuales se construyen haciendo amplio uso de tecnologías web; lo que se ha dado en llamar el "Web Oculto" (*Hidden Web* o *Deep Web*) [1][2] y que alberga la mayoría de la información existente en Internet.

Este trabajo describe un sistema que, haciendo uso de técnicas de extracción de información en fuentes semi-estructuradas permite una completa monitorización de cualquier sitio web desde la óptica del usuario final, que las soluciones actuales de gestión son incapaces de ofrecer a pesar de sus evoluciones [3]

2 Agentes de Navegación

La solución desarrollada descansa sobre un agente con el nivel de inteligencia suficiente para navegar a través de la fuente web como si del propio usuario se tratase. El principal problema al que se enfrenta esta técnica es la naturaleza semi-estructurada de las fuentes web a monitorizar: su estructura y la información contenida en ellas no sigue ningún esquema, sino que este está implícito en la propias páginas web estáticas o generadas dinámicamente.

El modelo de información de gestión (MIM) consiste ahora en un repositorio de metainformación que describe a los agentes cómo moverse dentro de la aplicación web. Esta metainformación contiene las acciones que el agente llevará a cabo. Una secuencia podría ser por ejemplo: "*Ir a la página inicial*", "*Autenticarse en la banca electrónica*" y "*Navegar hasta la página de movimientos*", todo ello descrito mediante un lenguaje declarativo construido con XML y basado en NSEQL [4].

```
<sequence id="1" elementName="locax_LOGIN" entityId="3">
   <inputElements>
      <inputElement key="usuario" value="13332912"/>
      <inputElement key="clave" value="0193"/>
   </inputElements>
   <navigation id="0" elementName="URL start">
      <navigate id="1" url="https://lo.cax.es"/>
   </navigation>
   <navigation id="1" elementName="MainPage">
      <hRefClick id="1" text="open line" position="0">
         <findBy what="text" equals="true" pages="2"/>
      </hRefClick>
   </navigation>
   <navigation id="2" elementName="Set Page">
      <frame id="1" name="Inferior" position="0">
         <findBy what="name" equals="true"/>
         <form id="1" name="INPUTS" position="0">
            <findBy what="name"/>
            <setElement>
               <setInput_TextArea name="ID" key="usuario" position="0" type="input"/>
            </setElement>
            <setElement>
               <setInput_TextArea name="B" key="clave" position="0" type="input"/>
            </setElement>
            <hRefClick id="1" text="javascript:submit_form()" position="0">
               <findBy what="href" equals="true" pages="1"/>
            </hRefClick>
         </form>
      </frame>
   </navigation>
</sequence>
```

Figura 1: Implementación de NSEQL basada en XML

El modelado de las fuentes web a monitorizar con este lenguaje oculta aspectos como la gestión de sesión o la interacción con scripts incrustados en las páginas HTML. Esta simplicidad del lenguaje disminuye considerablemente el impacto ante posibles cambios en la fuente, mucho más frecuentes que en sistemas de información tradicionales.

Una vez modelizadas las fuentes a monitorizar, se asigna a los agentes de navegación/monitorización una frecuencia de ejecución. En cada ejecución los agentes realizan las secuencias de navegación descritas en NSEQL midiendo el tiempo invertido en cada uno de los elementos de secuencia. Registran

también: tiempos de consulta DNS, de conexión y de obtención del primer byte entre otros.

Los agentes se denominan "no intrusivos" porque no requieren ninguna instalación sobre la plataforma que alberga la aplicación monitorizada, sino que actúan sobre la misma a través de conexiones HTTP simulando la actividad del usuario, mediante lo que se conoce como "screen-scraping" [5], técnica utilizada originalmente para extraer información.

3 Arquitectura completa del Qos-Meter

A medida que los agentes de navegación se van ejecutando almacenan las medidas en ficheros XML enviados a un repositorio central donde las medidas se consolidad en una base de datos sobre la que se elaboran estadísticas vía web. Al mismo tiempo existe un generador de informes en formato Excel (fig. 2)

El sistema ofrece múltiples posibilidades de despliegue que permiten medir y/o comparar todos los aspectos involucrados en la interacción del usuario. Instalando cada agente sobre una línea de acceso diferente podríamos medir, por ejemplo, como de bueno es un sitio web a través de diferentes proveedores, o, con un conjunto de sitios representativos comparar la calidad de los distintos proveedores a lo largo del tiempo.

4. Conclusiones

La principal aportación de este trabajo es la obtención de medidas que reflejan la calidad del sistema gestionado en cuanto a tiempos de respuesta con un grado de fiabilidad que las herramientas comerciales actuales no proporcionan.

La investigación y desarrollo actual en técnicas de extracción para integración de información es muy notable en contraste con la aplicación de dichas técnicas en el ámbito de gestión y monitorización de sistemas donde en trabajo en esta línea es casi nulo.

5 Trabajo futuro

Este trabajo no sustituye a enfoques tradicionales de gestión, sino que los complementa con información que a través de los mismos no es posible obtener. El siguiente paso obvio es unificar ambas fuentes de información de gestión correlando la información que ambos tipos de soluciones proporcionan logrando un diagnóstico más completo.

Otra mejora importante es dotar a los agentes de la capacidad de regenaración de las órdenes NSEQL ante cambios en las fuentes web, mucho más frecuentes que en aplicaciones no web.

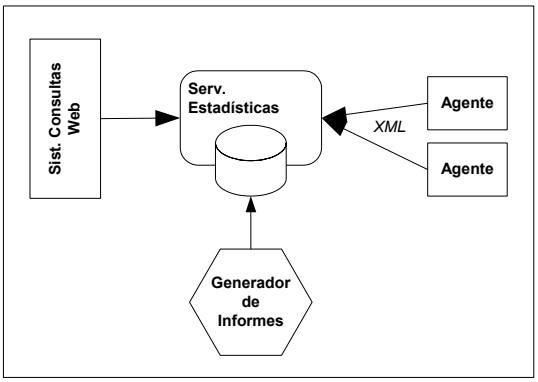


Figura 2: Arquitectura general del sistema

Referencias

[1] "The Deep Web. Surfacing Hidden Value". http://www.completeplanet.com/Tutorials/DeepWeb/

[2] Steve Lawrence, C. Lee Giles. "Accessibility of information on the web". Revista *Nature*, número 400, 1999

[3] V. Orjales, J. Hidalgo, V. Carneiro. "Plataforma Distribuida para la Monitorización y Control de Sistemas B2B con restricciones de Tiempo Real", III Jornadas de Ingeniería Telemática (JITEL 2001), Barcelona, Septiembre de 2001.

[4] Alberto Pan, Juan Raposo, Manuel Álvarez, Justo Hidalgo y Ángel Viña. "Semi-Automatic Wrapper generation for Commercial Web Sources". *Proceedings of the IFIP WG8.1 Conference on Engineering Information Systems in the Internet Context (EISIC)*. 2002.

[5] Thomas P. Vartanian and Robert H. Ledig "Scrape It, Scrub It and Show It: The Battle Over Data Aggregation. ". http://www.ffhsj.com/bancmail/bmarts/aba_art.htm.

Desarrollo y Evaluación de un Servidor Web Multicast

Manuel Cava, Adrián Ridaura, Jesús Sáez, Carlos E. Palau, Juan C. Guerri
Departamento de Comunicaciones. Universidad Politécnica de Valencia
Camino de Vera S/N, Valencia, 46022
Teléfono: 963 87 73 01 Fax: 963 87 73 09
E-mail: {macafer,adrimar}@teleco.upv.es, jesaemo@cfp.upv.es,{cpalau,jcguerri}@dcom.upv.es

***Abstract**. The recent growth in the use of the World Wide Web in the Internet has increased the demand of objects stored in web servers. The load results in worse performance perceived by the users. Several approaches have been developed in order to improve content distribution of web objects. We have developed a web server using cyclic multicast. In this paper we present the implementation of a system with client and server and evaluate the improvement in the performance. The utilisation of multicast provides a significant reduction in bandwidth, resources consumption in the network and in the server and prevents server collapse in extreme scenarios like flash-crowds.*

1. Introducción

El uso del WWW se ha incrementado en los últimos años. Cada vez hay un mayor número de usuarios de este servicio en todo el mundo, el cual se convierte por momentos, y cada vez más, en indispensable a la hora de realizar las tareas más cotidianas. Los administradores se encuentran ante el reto de actualizar y mejorar las prestaciones de la red para evitar que la demanda no la sature: a*ñadiendo hardware, mejorando el protocolo HTTP*, *replicando servidores, utilizando web caching* o m*ulticast*. [1]

La distribución de contenidos multicast de forma fiable y escalable es posible empleando protocolos de transporte adecuados o de forma cíclica. La segunda técnica utiliza un protocolo de transporte no fiable pero la ciclicidad de la transmisión añade una cierta fiabilidad. Esta solución será tanto más efectiva cuanto mayor sea el número de peticiones, y cuando el objeto a distribuir sea más popular. Los estudios teóricos demuestran que la distribución de peticiones sigue un modelo del tipo Zipf.[2]

En este trabajo se ha diseñado y desarrollado un servidor multicast cíclico, incluyendo un módulo para el servidor que automáticamente determina las páginas más populares, y gestiona el envío de multicast cíclico. Un cliente compatible que permite realizar un análisis y evaluación de un sistema completo. El sistema desarrollado utiliza el servidor web abierto Tomcat, siendo compatible el cliente con cualquier navegador al haberse desarrollado en java.

2. Motivación y Trabajo Previo

La distribución de contenidos se ha convertido en uno de los problemas más importantes a resolver en Internet. Se complica sobre todo, cuando el número de clientes que solicita el mismo contenido aumenta, traduciéndose en un mayor tiempo de respuesta percibido por los clientes.

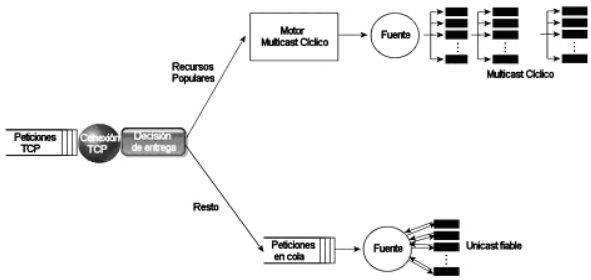


Fig.1 Arquitectura del servidor Web con multicast cíclico

El objetivo ha sido el desarrollo de un servidor web que incluye multicast cíclico, implementando un servicio de distribución de contenidos escalable empleando los conceptos desarrollados para la transmisión multicast cíclica.[1] La utilización de multicast para distribuir contenidos permite descargar al servidor, y reducir el tiempo de respuesta percibido por el cliente. Se ha desarrollado una nueva clase de URL (httpm), utilizando UDP e IP multicast.

El servidor determina los objetos solicitados con mayor frecuencia para distribuirlos mediante multicast cíclico. El sistema se basa en que servidores muy cargados, previsiblemente reciban peticiones similares en un periodo corto de tiempo, de forma que al atenderlas de forma multicast en lugar de unicast permite optimizar las prestaciones percibidas por el usuario (al reducir el ancho de banda consumido y los octetos transmitidos), y en situaciones críticas incluso evitar la caída del servidor por saturación de peticiones (*flash-crowd*).

Trabajos previos relacionados con la utilización de multicast para distribución de contenidos en Internet han sido: BDIS, MUSE, Multimedia Jukebox, o la arquitectura DataCycle. En ninguno de los casos anteriores se ha realizado un desarrollo de un URL multicast, garantizando la escalabilidad y la compatibilidad del servicio, ya que los desarrollos han tenido carácter propietario.

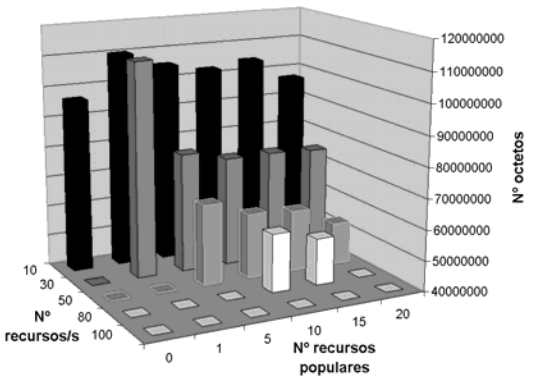


Fig.2 Número de octetos transmitidos

3. Estructura del Sistema

La arquitectura propuesta de servidor web, permite la entrega de objetos mediante multicast cíclico (sobre UDP) y unicast fiable (TCP). La decisión de que protocolo utilizar, se toma en base a la popularidad del objeto solicitado. Las páginas de mayor popularidad se sirven vía multicast cíclico, y las demás utilizando la conexión TCP abierta. El procedimiento empleado para transmitir cada uno de los recursos populares sobre multicast cíclico es:

1. El objeto web a distribuir se divide en bloques.

2. Todos los bloques de un objeto se transmiten secuencialmente desde el servidor al grupo de receptores multicast. Una transmisión de todos los bloques de un objeto constituye un ciclo.

3. Los receptores se unen al grupo multicast, y permanecen unidos hasta haber recibido una copia libre de error de cada uno de los bloques. Si un bloque no se recibe en el primer ciclo en el que se transmite, es recibido en un ciclo posterior.

4. El servidor continúa la transmisión cíclica de un recurso mientras considere que existe al menos un cliente intentando recibir, ya que no recibe realimentación. En caso contrario, el servidor detiene la transmisión cíclica del objeto.

4. Evaluación de Prestaciones

Las pruebas se han realizado en un escenario compuesto por 20 clientes accediendo a un servidor web que implementa el módulo multicast cíclico desarrollado. Se ha empleado un generador de carga que permite configurar los parámetros de petición de recursos al servidor: tasa de peticiones, el grupo de recursos y número de recursos totales.

La figura 2 muestra los octetos trasmitidos por el servidor. Aquellos valores que no se visualizan en la gráfica corresponden a estados de *overflow* del servidor. Se puede observar que la tasa máxima de peticiones aumenta sin desbordamiento del servidor cuando se incorporan los recursos más populares del servidor al motor multicast cíclico. Una reducción del número de octetos transmitidos por el servidor, fruto de la transmisión multicast, proporciona un ahorro del 54% sobre el número de octetos transmitidos, en el mejor de los casos. Existen configuraciones para las que el uso del algoritmo provoca que se envíen un mayor número de octetos debido a que el número total de octetos transmitidos por un flujo multicast es mayor que el de uno unicast al fraccionar los paquetes en bloques y añadirles cabeceras HTTP. Por tanto, si no hay concurrencia de clientes a la escucha, el algoritmo puede incluso empeorar las prestaciones.

El segundo parámetro evaluado es el tiempo de latencia, es decir, el tiempo que los usuarios perciben que tarda en mostrarse el recurso. El tiempo de latencia está íntimamente relacionado con el número de octetos transmitidos. Si se reduce el número de octetos transmitido por la red, implica que el número de flujos de transmisión de un tipo u otro simultáneos del servidor disminuye, por tanto el tiempo de CPU asignado a cada uno de ellos es mayor pudiéndose procesar más rápido, produciéndose una mejora.

En cuanto a la carga de CPU se comprueba que, si bien lo inmediato es pensar que disminuirá al mismo ritmo que lo hacen los flujos, no disminuye considerablemente. Esto es debido a que la carga unitaria de CPU de los flujos multicast es mayor que la de los flujos unicast por la implementación realizada, por tanto la disminución de flujos se compensa por este hecho.

5. Conclusiones

El multicast cíclico es una alternativa viable al web caching en entornos locales y en una futura Internet con soporte multicast, no así en la actual Internet. El ahorro de octetos trasmitidos se relaciona con la concurrencia de un número elevado de clientes solicitando un mismo recurso popular. Los resultados con respecto al tiempo de latencia en recepción se relaciona con la reducción de octetos transmitidos por la red y la menor carga del servidor.

Las mejoras del multicast cíclico desaparecen cuando se pierde la concurrencia de peticiones de un recurso o la tasa de peticiones de un recurso no es suficientemente alta. En situaciones del tipo *flash-crowd* se justifica la utilización de esta técnica.

Referencias

[1] K. C. Almeroth, M. H. Ammar, Z. Fei. “Scalable Delivery of Web Pages Using Cyclic Best-Effort Multicast”, IEEE INFOCOM’98, pp. 1214-1221, San Francisco (CA) USA, Abril 1998.

[2] H. Braun, K. Claffy, “Web traffic characterization: an assessment of the impact of caching documents from NCSA’s web server”, computer Networks and ISDN Systems, 28, pp. 37-51, Diciembre 1995.

Breve 2A: Aplicaciones Cooperativas y plataformas WEB

Xweb: A Framework for Application Network Deployment in a Programmable Internet Service Infrastructure

O. Ardaiz, L. Navarro
Computer Architecture Department
Polytechnic University of Catalunya
Barcelona 08034 Spain
Tel: +34-93-4137102 Fax: +34-93-4017055
{oardaiz, leandro}@ac.upc.es

Abstract: *An application network consists of a number of application servers distributed throughout the Internet, connected and coordinated to provide services with low latency. Adding, removing and migrating servers, application networks adapt to temporal and spatial demand variations. To create new servers anywhere in the Internet a programmable Internet service infrastructure is needed. In addition application network servers must be deployed co-ordinately. We propose a framework for application network deployment that provides basic building blocks for coordinated application network activation at appropriate locations in the programmable infrastructure. We implemented Xweb, a prototype of such framework. We performed experiments to evaluate how fast application networks can be deployed. We conclude discussing current work.*

1. Motivation

Internet services quality can be greatly improved if an application network provides them. Application networks are a set of coordinated application servers distributed throughout the Internet, thereby clients can access a nearby server that provides a low latency service. Application network provide client requests with good quality of service: a request from location X will be provided by nearer server A instead of farther server B; also server A load will not increase beyond a threshold, causing successive requests to be delayed, and some requests are redirected to a distant but less loaded server. Those servers are connected, setting up an application layer topology, and coordinated for request redirection, load balancing and replica consistency. Examples of application networks are content distribution networks, proxy-caching hierarchies, chat server networks or peer-to-peer networks as Gnutella.

Existing application networks are manually created and modified: new servers are manually installed and connections among servers are manually configured. However Internet services have very dynamic demands with temporal and spatial variations [3]; static application networks can not provide these demands with good service quality (unless they are overprovisioned to the worst case: the hot spot service demand f.e. Akamai network[1]). An application network that adapts to those variations will serve such dynamic demands with a good service quality. Adapting to demand variations involves adding and removing servers, migrating servers to locations where new demand arises, and deleting and creating connections among servers. To implement this functionality it is required a programmable Internet services infrastructure. A programmable infrastructure is composed of resources distributed throughout the Internet where third party service providers can remotely activate and stop application servers, and connect and disconnect server instances. Moreover, to facilitate application networks adapt to demand variations maintaining good service quality while consuming few resources, application networks must be deployed, "to spread out or arrange for effective action", instead of being uncoordinatedly activated. As a result of application network deployment, application servers are placed at appropriate locations and coordinated to provide a good service quality.

2. Framework Building Blocks

XWeb building blocks are resource discovery and monitoring, service specification interface, resource mapping and allocation, and code distribution and service composition:

- Resource discovery and monitoring mechanisms provide resource availability information to application provider.
- Service specification interface: so that application providers input their requirements from their expectations of service demand.
- Resource mapping and allocation: service demand and service-wide constraints must be translated or mapped to a resource offer. It involves selection of resources that best match service requirements among those available and putting apart the required resources at each node for sole used by that service.
- Code distribution and service composition: application networks are composed of software entities executed at different nodes. They communicate and coordinate for effective service and efficient resources utilisation. Communication is defined by connectivity among its members and by co-ordination rules each node follows. Application layer network connections are TCP virtual circuits among servers. Rules governing co-ordination among

service instance are very simple rules: "if cannot be processed here, forward to node x", “if coming from region A, process by node y”, etc. Therefore to compose such services, software has to be uploaded, installed, configured and activated at each node where resources have been successfully allocated. Mechanisms required to compose an application network are: 1) Code distribution mechanisms to move code to nodes where it should be activate, 2) Dynamic resource binding mechanisms so that services bind and unbind from resources on the fly, 3) Dynamic remote communication channel set-up and co-ordination rules configuration.

3. Implementation, Experiments and Current Work

We implemented the Xweb programmable infrastructure made up of Linux nodes and Java-Tomcat execution environments (therefore it is possible to deploy Linux service and Servlet based application networks). Each node makes public its resource availability: execution environment, maximum network capacity, storage capacity and adjacent network regions where service can be provided. Xweb is composed of resource agents at resource providers' nodes and deployment managers at service providers' nodes. Resource agent implements resource allocation, code distribution, resource binding and service composition. Deployment managers implement resource discovery, a service specification front-end, resource mapping, and deployment plan creation. Deployment managers command resource agents to allocate resources, obtain service code, bind service programs to allocated resources, and configure inter-server communication channels and application network coordination rules. Resource agents implement an access control mechanism controlled by resource providers which permits only some deployment managers to perform resource allocation and service composition operation. Deployment manager communicate with resource agents by secure SSL connections. Deployment manager and resources are certified by a certification authority.

In the experiments it were deployed web proxy-caching and chat server networks (Web proxy-caching application network provide a caching service for web clients: web pages are cached at intermediate servers and pages that cannot be provided by a server are forward to a parent proxy; chat server application networks are a number of chat servers that are coordinated to share chat channels). Application providers demanded 1 Mbps and 100 Kbps network capacity, and 200 Mbytes and 1 Mbytes storage capacity respectively at each node; clients demand was specified as coming in from all available regions. Therefore at least one server should be activated at each node.

We measured temporal response to a deployment request event because we are interested in finding how quickly application networks can be deployed, and how fast application networks can adapt to demand and resource availability variations. Experiments show that these application networks can be deployed in a period of time on the order of seconds, which is better that what could be obtained previously by manual operations.

There are several techniques to improved Xweb deployment time. We will adapt in the near future some of them, i.e.: reusing installed code (caching), replicating code servers, incrementally loadable application code, and dynamically switching resources binding with virtual containers. Besides we plan to evaluate in future experiments how fast application networks can adapt to demands or resource availability variations. Other future work directions are to study scalability of some mechanisms to the Internet, f.e. resource discovery scalability, and resource mapping algorithm scalability. Further information can be found in our technical report [2].

Acknowledgements

This work was inspired by a research stay in Information Science Institute University of Southern California working in the Xbone [4].

This work has been partially supported by the Spanish MCYT project TIC2002-04258-C03-01 “Global and Peer-to-Peer Computing for Cooperative Learning Environments”.

References

[1] Akamai Inc., http://www.akamai.com, May 2003.

[2] Ardaiz O., Navarro L., “A Framework for Application Network Deployment in a Internet Service Programmable Infrastructure”, UPC-DAC Technical Report UPC-DAC-2003-27.

[3] Santos J.R., Dasgupta K., Janakiraman G. J. and Turner Y., “Understanding service demand for adaptive allocation of distributed resources”, (GLOBECOM '02), Taipei, Taiwan, Nov. 2002.

[4] Touch J., Hotz S., "X-bone: a System for Automatic Network Overlay Deployment", Third Global Internet Mini Conference in conjunction with Globecom´98, Nov. 1998.

Prototipo de un Framework J2EE para la Construcción de Portales Personalizables "Mi" [1]

Fernando Bellas
Departamento de Tecnologías de la Información y las Comunicaciones. Universidad de A Coruña.
Facultad de Informática. Campus de Elviña. CP/15320. A Coruña. España. E-mail: fbellas@udc.es

Pedro Moreira
Denodo Technologies. Departamento de Tecnología. C/ Real, 22 - 3º. CP/15003. A Coruña. España.
E-mail: pmoreira@denodo.com

Abstract. *This paper presents the key ideas of a prototype framework, MyPersonalizer, that facilitates the construction of personalizable "My" portals (e.g my.yahoo.com), that is, portals that allow the user to have one or more pages composed of personalized services. The framework is J2EE-based and is structured according to the Model-View-Controller architectural pattern, providing generic and adaptable model and controller layers. Developers only need to concentrate on framework configuration (e.g. specifying meta-information about the persistent objects in the particular portal being built), implementing service plug-ins for integration of personalized service responses, redefining model and controller policies if necessary, and implementing the portal view as JSP pages.*

1 Introducción

Los portales de Internet (e.g. Yahoo!, Excite, Lycos, etc.) ofrecen un gran número de servicios al usuario, proporcionándole una gran cantidad de información. Para facilitar el acceso a esta información, y atraer más usuarios, muchos de estos portales disponen de versiones personalizadas, los llamados portales "Mi" (e.g. my.yahoo.com, my.excite.com, my.lycos.com, etc.). Este tipo de portales permiten que el usuario disponga de una o varias páginas que agregan la información de un conjunto de servicios, que el usuario elige de una paleta de servicios disponibles y personaliza a su gusto. Dentro de una página, cada servicio se muestra en una venta compuesta de una barra de botones (e.g. editar, maximizar/minimizar, destruir, etc.) y un área de información. El botón de edición permite que el usuario acceda al asistente de personalización del servicio (e.g. seleccionar un conjunto de ciudades y el tipo de unidades en un servicio que ofrezca información meteorológica), de manera que cuando el usuario acceda a la página que lo contiene, el servicio sólo mostrará la información en la que el usuario está explícitamente interesado (e.g. predicciones meteorológicas para las ciudades seleccionadas y en las unidades especificadas). Normalmente el usuario también puede personalizar otros aspectos, como la disposición de los servicios dentro de las páginas y la apariencia de algunos componentes. A pesar de que este tipo de portales nacieron en el contexto de Internet, están cada vez más presentes en las versiones personalizadas de los portales de las intranets, ofreciendo a los usuarios una vista personalizada y restringida de la información de la compañía.

Desde un punto de vista arquitectónico, construir un portal Mi conlleva dos aspectos clave: construir el portal e integrar las respuestas personalizadas de los servicios. Lo primero significa construir una aplicación web que implementa casos de uso tales como, registro de un usuario, autenticación en el portal, creación y destrucción de páginas, selección de servicios y su disposición dentro de la página, etc. Lo segundo implica que por cada servicio es necesario construir un asistente de personalización, implementar soporte para la persistencia de las propiedades personalizables e integrar la respuesta personalizada en el portal. Además, normalmente los servicios ya existen antes de que se decida construir el portal Mi, y seguramente están construidos con diferentes tecnologías (e.g. J2EE, .NET, LAMP, etc.).

Actualmente existe un buen número de productos Java que dan soporte a la construcción de portales Mi (e.g. BEA WebLogic Portal, IBM WebSphere Portal, Oracle 9iAS Portal, Jakarta Jeetspeed, etc.). Normalmente estos productos consisten en un portal ya construido en el que los desarrolladores integran los servicios (portlets). El inconveniente de este enfoque es su falta de generalidad, dado que no es posible hacer grandes variaciones en el portal que "viene de fábrica", pudiendo variar únicamente algunos aspectos básicos de presentación (e.g. estilos). De hecho, cualquier portal construido con uno de estos productos delata claramente con cuál se ha construido. Por otra parte, los productos actuales suelen dar poco soporte para la integración de los servicios en el portal, no incluyendo soporte genérico para la construcción de los asistentes de personalización y la automatización de la persistencia

[1] Este trabajo ha sido financiado parcialmente por CICYT (TIC2001-0547).

de la personalización del servicio, siendo el desarrollador el que se tiene que encargar de estas tareas arduas y propensas a errores. Finalmente, los productos comerciales están totalmente acoplados con los contenedores de aplicaciones J2EE que venden los respectivos fabricantes, por lo que no se pueden usar con otros contenedores J2EE.

2 Arquitectura de MyPersonalizer

MyPersonalizer[2] es un framework J2EE que facilita la construcción de portales Mi y la integración de servicios en el portal, y puede usarse con cualquier contenedor J2EE de aplicaciones web. Arquitectónicamente, el framework está estructurado en capas según el patrón arquitectónico Model-View-Controller (MVC), ofreciendo unas capas modelo y controlador genéricas y adaptables (Fig. 1).

La Fig. 2 muestra un diagrama de clases que ilustra las clases proporcionadas por la capa modelo del framework para modelar los objetos persistentes que se deben almacenar por cada usuario. `ServicePersonalization` representa la personalización de un servicio. Dado que algunos botones pueden tener asociado estado, como por ejemplo, los botones para minimizar/maximizar o mostrar ayuda, también es necesario guardar por cada servicio el estado de sus botones (`ServiceButtonsState`). Un objeto `WorkspaceLayout` representa la disposición de los servicios dentro de una página y contiene las claves de los objetos `ServicePersonalization` y `ServiceButtonsState` correspondientes a los servicios contenidos en dicha página. Similarmente, un objeto `DesktopLayout` contiene una lista ordenada de las claves de los objetos `WorkspaceLayout` poseídos por un usuario. Finalmente, un objeto `UserRegistrationInformation` contiene la información de registro de un usuario y la clave de su `DesktopLayout`.

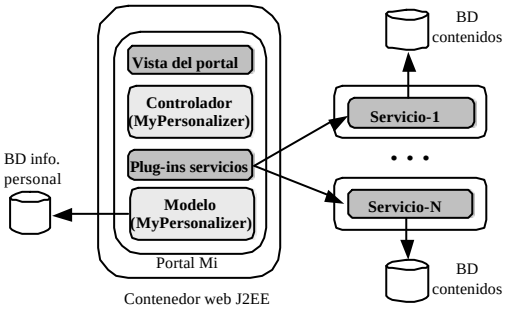


Figura 1: Arquitectura de un portal Mi con MyPersonalizer

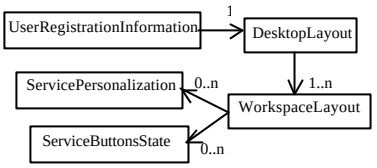


Figura 2: Objetos personales persistentes

Diferentes portales requieren diferentes propiedades para cada uno de estos objetos persistentes. Los portales varían en la información de registro, en los tipos de disposiciones de los servicios dentro de una página (e.g. dos columnas, tres columnas, etc.) y en los tipos de botones que tienen estado persistente. Además, diferentes servicios tienen distintas propiedades de personalización. Por tanto, para que la capa modelo sea genérica, las clases que ilustra la Fig. 2 ofrecen soporte para dotarlas de un conjunto de propiedades, donde cada propiedad tiene un nombre y un valor, univaluado o multivaluado, que puede ser de un tipo simple (`Integer`, `Long`, `String`, etc.) o estructurado (compuesto por otras propiedades).

Para dar soporte a la persistencia de estos objetos en base de datos, la capa modelo proporciona una clase DAO (Data Access Object) para cada objeto persistente, que permite mapearlo a una base de datos relacional. Para que ello sea posible, el desarrollador debe proporcionar meta-información acerca de las propiedades de cada uno de los objetos persistentes del portal concreto que se va a construir. Finalmente, la capa modelo ofrece una fachada que proporciona una operación por cada caso de uso (e.g. registrar un usuario, autenticarse en el portal, especificar los servicios de una página, modificar la personalización de un servicio, etc). La implementación de cada caso de uso hace uso de los correspondientes DAOs.

La capa controlador básicamente proporciona un servlet por cada caso de uso, que recibe los parámetros de la petición HTTP, los valida, y si son correctos, invoca la operación correspondiente sobre la fachada del modelo, pasando finalmente el control a la siguiente página. Por tanto, construir la vista del portal consiste en escribir las páginas JSP correspondientes. Las URLs usadas en los enlaces o en las acciones de los formularios corresponden a los servlets proporcionados por el controlador, que permiten enlazar la vista con el controlador, y éste a su vez con el modelo. El desarrollador proporciona plug-ins como extensiones al controlador para la integración de las respuestas personalizadas de los servicios. Típicamente cada plug-in actúa como un proxy del servicio real. Este proxy recupera la personalización del servicio, invoca una URL del servicio real (pasándole como parámetros la personalización del servicio) que devuelve la respuesta en XML, que finalmente el proxy formatea con una página XSL.

Actualmente estamos trabajando en mejorar la arquitectura del framework para realizar una integración total con Jakarta Struts, el framework MVC “de facto” para aplicaciones web J2EE.

[2] Es posible consultar una versión ampliada de este artículo en http://www.tic.udc.es/~fbellas/mypersonalizer/download/MyPersonalizer-1.0-TechReport.pdf.

Asignador de Recursos Grid para Aplicaciones CSCL basadas en Componentes

Miguel L. Bote Lorenzo, Yannis A. Dimitriadis, Eduardo Gómez Sánchez,
Juan I. Asensio Pérez, Luis M. Vaquero González, Guillermo Vega Gorgojo
Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática
Escuela Técnica Superior de Ingenieros de Telecomunicación, Universidad de Valladolid
Camino del Cementerio s/n, 47011 Valladolid
Teléfono: 983 42 36 66 Fax: 983 42 36 67
E-mail: {migbot@, yannis@, edugom@, juaase@, lvaqgon@ribera, guiveg@}tel.uva.es

Abstract. This paper aims to advance towards enabling grid infrastructures to provide support for component-based Computer Supported Collaborative Learning (CSCL) applications. In order to achieve this goal, a resource assigner able to deploy CSCL applications within the grid must be developed following CSCL domain specific performance criteria. Thus, a generic resource assigner model for CSCL component-based applications is proposed.

1 Introducción

El término *grid* (malla) [1] se emplea habitualmente para referirse a una infraestructura hardware y software que se caracteriza por ser de gran escala, geográficamente distribuida, y compuesta por recursos heterogéneos pertenecientes a múltiples organizaciones administrativas que los comparten con el objetivo de proporcionar soporte computacional a un amplio rango de aplicaciones de forma transparente, de calidad, ubicuo y consistente [2]. El soporte proporcionado por el grid es considerado de gran utilidad para aplicaciones de computación colaborativa dado que el grid permite y fomenta la interacción entre humanos sea de forma síncrona o asíncrona [3] a través de un espacio virtual.

Precisamente es este tipo de interacciones entre humanos las que son empleadas por las aplicaciones de Aprendizaje Colaborativo Apoyado por Ordenador (CSCL – *Computer Supported Collaborative Learning*) para fomentar la colaboración como método de aprendizaje. CSCL es así un campo de estudio dedicado a la investigación de tecnología educativa que se centra en el uso de las tecnologías de información y comunicaciones (TIC) como herramientas de mediación para la aplicación de métodos colaborativos de aprendizaje (ej. aprendizaje entre iguales, enseñanza recíproca, aprendizaje basado en proyectos o problemas, simulaciones, juegos) [4,5].

El grid es una infraestructura que puede dar un soporte adecuado a las aplicaciones CSCL facilitando su despliegue y potenciando su rendimiento. Sin embargo, los autores no tienen conocimiento de que hasta el momento se hayan empleado grids para dar soporte a aplicaciones CSCL a pesar de que las características antes mencionadas de dicha infraestructura pueden proporcionar beneficios importantes para las aplicaciones CSCL. Entre estos beneficios es posible destacar los siguientes. En primer lugar, la gran escala del grid podría permitir la participación en la aplicación CSCL de un elevado número de participantes individuales o en grupo. En segundo lugar, la amplia distribución geográfica de los recursos que forman parte del grid facilitaría la colaboración entre usuarios muy distantes entre sí. También la naturaleza heterogénea de los recursos que son compartidos en el grid podría permitir a los usuarios participar empleando no sólo ordenadores, sino también otros dispositivos como PDAs o pizarras electrónicas además de recursos software como simuladores. Por otra parte, el acceso de calidad proporcionado por la infraestructura grid a través de un servicio garantizado o de mayor esfuerzo puede mejorar el rendimiento de la aplicación cuando ésta es desplegada *ad hoc*.

Este último aspecto es especialmente importante para el dominio CSCL dado que el éxito de la colaboración entre usuarios, y por tanto del aprendizaje como objetivo último, sólo es posible si el rendimiento de la aplicación es bueno. Así, por ejemplo, en el caso de un puzzle que ha de ser resuelto por niños de forma colaborativa, dicha colaboración será imposible si el rendimiento de la aplicación no es lo suficientemente bueno como para distribuir rápidamente a todos los usuarios los cambios que se vayan produciendo en el puzzle.

Para lograr que las aplicaciones puedan aprovechar el potencial de rendimiento, es necesario desarrollar un asignador de recursos, elemento encargado de decidir cómo ha de llevarse a cabo el despliegue de una aplicación de forma que el rendimiento de la misma sea el adecuado de acuerdo con unos criterios predefinidos [6]. Sin embargo, el desarrollo de un asignador de recursos grid es un problema muy dependiente del dominio de aplicación [6].

De este modo, el objetivo es proponer un modelo genérico de asignador de recursos para aplicaciones CSCL basadas en componentes. Dicho modelo genérico permitirá posteriormente desarrollar asignadores de recursos particulares para aplicaciones CSCL concretas.

El resto de este artículo está organizado como se describe a continuación. La sección 2 describe un modelo genérico de asignador de recursos para aplicaciones CSCL basadas en componentes. La sección 3 recoge las principales conclusiones de este documento.

2 Modelo de Asignador

Dada una aplicación CSCL basada en componentes y una lista de recursos grid capaces de albergar dichos componentes, el problema de la asignación de recursos consiste en encontrar una combinación de los siguientes tres aspectos con el objetivo de lograr que la aplicación cumpla con un determinado criterio de rendimiento: una *selección* de los *recursos* del grid en los que se ejecutarán los componentes de la aplicación, una *asignación* de cada uno de los *componentes* que forman la aplicación al recurso seleccionado en el que deba ejecutarse y una *configuración* de las relaciones de comunicación entre *componentes* que establece con qué otros componentes se debe comunicar cada componente de la aplicación. De este modo, se puede considerar que el problema de la asignación de recursos define un espacio de soluciones determinado por todas las posibles combinaciones de selección, asignación y configuración. Para encontrar soluciones válidas en dicho espacio es posible emplear un asignador de recursos de acuerdo al modelo general de la Fig. 1.

En dicho modelo general se identifican cinco elementos necesarios para resolver el problema. Primero, el *modelo de aplicación* describe la descomposición de la aplicación en componentes software. El conjunto de todas las posibles combinaciones de distribuciones de componentes en el grid y configuración de las relaciones de comunicaciones entre componentes que satisfacen el modelo de aplicación conforman el espacio de soluciones. Segundo, el *criterio de selección* define cuáles son las variables empleadas para cuantificar el rendimiento de la aplicación (variables de rendimiento) para cada una de las posibles soluciones del espacio así como las restricciones sobre dichas variables que debe cumplir la solución elegida por el selector. Tercero, las *características de los recursos* definen el estado de cada uno de los recursos que componen el grid. Cuarto, el *modelo de rendimiento* predice el valor de las variables de rendimiento en función de las características de la aplicación así como las de los recursos de la solución evaluada. Finalmente, el *método de exploración* es empleado para recorrer el espacio de soluciones en busca de una solución que cumpla con el criterio de selección definido.

Figura 1: Modelo general de asignador de recursos

Como prueba de concepto, el modelo propuesto ha sido empleado para desarrollar el asignador de recursos de una aplicación CSCL simple, un chat, proporcionando rendimientos mayores que otros modelos de asignación heurísticos. El análisis pormenorizado de estos resultados no se lleva cabo en este artículo por razones de falta de espacio.

3 Conclusiones

El soporte proporcionado por una infraestructura grid puede ser de gran utilidad para las aplicaciones CSCL. Sin embargo, para que dicho soporte sea posible, es necesario desarrollar un asignador de recursos. En este artículo se ha propuesto un modelo genérico de asignador de recursos para aplicaciones CSCL basadas en componentes. Este modelo será empleado en el trabajo futuro para desarrollar asignadores de recursos particulares para aplicaciones CSCL concretas.

Agradecimientos

Los proyectos TIC2002-04258-C03-02, TIC2000-1054 y VA 117/01 han financiado el presente trabajo.

Referencias

[1] I. Foster, C. Kesselman. *The Grid: blueprint for a future computing infrastructure*, San Francisco, CA, USA : Morgan Kaufmann Publishers (1998).

[2] M. L. Bote-Lorenzo, Y. A. Dimitriadis, E. Gómez-Sánchez. "Grid characteristics and uses: a grid definition". Proceedings of the 1st European Across Grids Conference (CD). Santiago de Compostela, Spain, 2003.

[3] I. Foster, C. Kesselman, S. Tuecke. "The anatomy of the Grid: enabling scalable virtual organizations". Int. Journal of Supercomputer Applications, pp. 200-222, vol. 15 (3) (2001).

[4] B. Wasson. "Computer Supported Collaborative Learning: an overview". Lecture notes from IVP 482, University of Bergen, Norway (1998).

[5] T. Koschmann . *CSCL: theory and practice of an emerging paradigm*, Malwah, NJ, USA: Lawrence Erlbaum (1996).

[6] F. Berman. "High-Performance Schedulers". In: [1]

Desarrollo de un Portal Web Integrador de Información Basada en Subastas Online

Mª Eugenia Gonzalo Cabellos, Vicente Luque Centeno
Depto. Ingeniería Telemática, Universidad Carlos III de Madrid
Av. Universidad, 30, E-28911 Leganés (Madrid/Spain)
E-mail: 100012154@alumnos.uc3m.es, vlc@it.uc3m.es

Abstract. *Integrating data from several heterogeneous information sources is a major issue for many commercial and corporative organizations. Less user interaction effort can be achieved if programs automate user's behaviour on the Web. Relevant data has to be properly extracted and managed in order to perform specific tasks for users. This paper proposes an information and services integration system based on information provided by web known, legacy web sites oriented to auctions. Low-cost maintenance techniques have been used to reduce the impact of unexpected changes within retrieved pages.*

1 Introducción

La cantidad de información en la Web está creciendo a una velocidad de vértigo. El increíble éxito de Internet ha propiciado la aparición de nuevos retos tecnológicos que tienen como objetivo principal facilitar y potenciar la forma en que los humanos interactúan con la Web.

Así, resulta de gran interés y utilidad programar aplicaciones que "entiendan inteligentemente" el contenido de las páginas HTML y que dialoguen con un servidor Web siguiendo los enlaces de sus páginas y rellenando sus formularios de una forma "programada" (en lugar de una forma "manual" que es como la hacemos habitualmente en un navegador Web).

En Internet existen muchos portales que ofrecen servicios similares. Englobando varios de estos portales dentro de un mismo portal, se obtiene uno más completo que ofrecerá al usuario la posibilidad de consultar todos simultáneamente, comparar sus informaciones, tener una visión homogénea de sus contenidos,... y todo ello a través de una única interfaz.

En el portal introducido en este artículo, se han integrado tres de los sitios de subastas online más importantes de España, eBay[1], Aucland[2] y MercadoLibre[3].

2 Diseño del sistema

Para poder integrar los contenidos de varias fuentes de información, es necesario hacer un análisis de la estructura de la información proporcionada por cada una de ellas, estudiar cómo extraerla y finalmente buscar la manera de integrar bajo una misma estructura homogénea los datos obtenidos. Cada portal que se pretende integrar es una fuente de información.

Son varios los puntos que se han tenido que estudiar a la hora de diseñar e implementar el sistema.

2.1 Diseño del portal

Para el diseño de este portal se realizó un estudio de los portales que se pretendía integrar, analizando detalladamente los servicios ofrecidos por cada uno de ellos. A partir de este estudio, se decidió qué servicios proporcionaría el portal integrador.

2.2 Integración de datos heterogéneos

Otro punto importante a estudiar es cómo procesar la heterogeneidad de los datos recogidos de los diferentes sitios Web, cómo integrar la información obtenida de los distintos portales dentro de una estructura de datos homogénea para todos ellos.

2.3 Robustez del sistema

El sistema debe ser robusto ante fallos como que algún servidor de los sindicados está caído, o que se produzca un fallo en la red, o los programas que interactúan con la Web no sean capaces de extraer o manejar la información para la que están programados. Es importante que el sistema sea robusto y pueda manejar estas incidencias sin que lleguen estos fallos al usuario.

2.4 Escalabilidad del sistema

Dada la velocidad a la que Internet evoluciona, surgiendo nuevos portales, y muriendo otros, se ha construido el sistema de tal forma que es escalable y fácilmente, sin demasiados cambios, se pueden añadir (o eliminar) portales de subastas.

[1] http://www.ebay.com/es
[2] http://www.aucland.es
[3] http://www.mercadolibre.es

2 Funcionamiento del Sistema

El sistema implementado debe ser capaz de integrar varios servidores Web de tal forma que el usuario tenga la sensación de estar consultando sólo un servidor. A través de una interfaz el usuario estará accediendo a todos los servidores sindicados, y la consulta a éstos le resultará totalmente transparente. La arquitectura más adecuada para implementar este sistema es la de mediador (ver Fig. 1) [1].

Cada wrapper será el encargado de acceder al portal para el que esté programado, interactuar con su servidor, extraer los datos, adaptarlos al formato homogéneo para todas las fuentes (definido en la fase de diseño) y entregarlos al mediador. El mediador integra los datos, ya homogéneos, procedentes de las distintas fuentes (proporcionados por los wrappers), y los combina, filtra y procesa, para posteriormente pasarlos al nivel de usuario o aplicación. En este nivel en este sistema se encuentra un generador de páginas XHTML, que construye la página que se le devolverá al usuario como respuesta a su petición.

3 Generación de Wrappers (Código Envoltorio)

Todo los wrappers del sistema han sido programados manualmente con un lenguaje llamado WebL [3], y son completamente dependientes de las páginas que deben tratar.

Extraer la estructura de los datos de los sitios web no es una tarea trivial. En HTML (al contrario que en un XML concreto) existen muchas formas de representar la misma información. Esto, unido al hecho de que la estructura de las páginas Web puede ser cambiada sin previo aviso, hace que las funciones programadas para extraer información de las páginas obtenidas de la Web, o seguir sus enlaces, o rellenar sus formularios, han de ser lo más robustas posibles.

4 Pruebas y conclusiones

Una vez construido el portal se realizaron muchas pruebas de los distintos servicios que éste ofrecía, con el fin de comprobar el correcto funcionamiento del sistema, sus debilidades y su robustez.

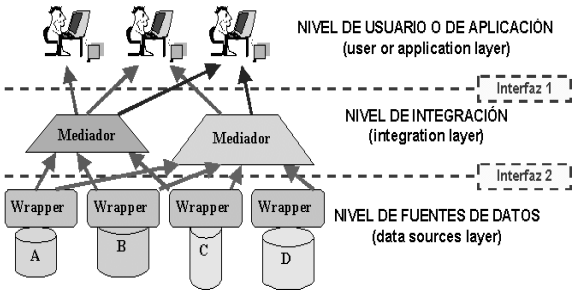


Fig. 1 Arquitectura de mediador

Así, se pudo comprobar que este portal proporciona ***rapidez***, ***comodidad*** y ***sencillez*** al usuario en el acceso a la información de las subastas, ya que es más sencillo, rápido y cómodo acceder desde un portal a varios que acceder por separado a cada uno de ellos. Dada la importancia y la utilidad que gana día a día Internet, todas aquellas aplicaciones que tengan como finalidad facilitar el acceso y utilización de los recursos ofrecidos por la red al usuario, están encaminadas a ser parte del futuro de los servicios ofrecidos a través de la red.

La debilidad más importante, sin duda, es el coste de mantenimiento de los wrappers. Aunque éstos han sido programados de tal forma que resultan lo más robustos posible frente a cambios en los portales sindicados, no se llega a conseguir una robustez del 100%, y hay cambios en las páginas, para las que están programados, que hacen que un wrapper pueda fallar. Así, si un servidor cambia detalles importantes en sus páginas devueltas, la extracción de datos no podrá llevarse a cabo por los wrappers programados para ese servidor, que deberán ser adaptados a estos cambios. Afortunadamente estos cambios no son muy frecuentes, y además se ha proporcionado un servicio de mantenimiento mediante el que el administrador es avisado cada vez un wrapper devuelve un error, de tal forma que este fallo puede ser corregido a la mayor brevedad posible. Normalmente, el wrapper se puede arreglar en unas horas, y en cualquier caso el sistema seguirá funcionando normalmente, excepto ese wrapper. Con el fin de facilitar la creación y mantenimiento de los wrappers hubiera resultado de gran interés y utilidad disponer de una herramienta de generación automática de wrappers en WebL.

Existen varios trabajos relacionados con este, por ejemplo [2], y varias aplicaciones que ya funcionan basadas en [2]. En concreto, hay una aplicación que realiza búsquedas sencillas de subastas[4], pero no implementa un portal de subastas online completo, como el recogido en este artículo.

Referencias

[1] G. Wiederhold. “Mediators in the architecture of future information systems”. Computer, 25(3), March 1992.

[2] Juan Raposo, Manuel Álvarez, Alberto Pan, Ángel Viña. “Un Sistema Mediador en la Práctica: Base de Datos Virtual”. Actas del Segundo Congreso Iberoamericano de Telemática. Merida, Venezuela. 11-13 Septiembre 2002.

[3] Compaq’s Web Language, Compaq Computer http://www.research.digital.com/SRC/WebL/index.html

[4] http://buscaproductos.biwe.es

Una Propuesta Basada en Estándares para la Automatización de Tareas en el Web

Vicente Luque Centeno, Luis Sánchez Fernández, Carlos Delgado Kloos, Peter T. Breuer
Fernando Paniagua Martín, Juan Antonio Herráiz Pérez
Departamento de Ingeniería Telemática. Universidad Carlos III de Madrid
Avda. Universidad, 30, E-28911 Leganés, Madrid, Spain
E-mail: vlc@it.uc3m.es

Abstract *The Web has rapidly become the largest knowledge repository on the world. Manual navigation with browsers results quite expensive when the amount of data to be processed becomes large or the number of links to be followed is high. Being the Web visualization oriented, however, most of its information is heterogeneous and poorly structured, so developing programs that automate the management of data obtained from the Web is quite difficult. Traditional programming techniques result in short-lived programs that can only perform simple tasks and require a great maintenance effort.*

Using well known standards like XPath for information extraction and Message Sequence Charts from ITU, can easily simplify the development and maintenance of these programs. This paper presents a platform based on these standards for developing programs that automate navigation and data extraction from the Web. This approach has been tested over several real well known web sites including auctions, web mail or travel agencies, among others, with successfull results.

1. Planteamiento

Actualmente, millones de usuarios en todo el mundo se ven abocados a realizar cada día tareas en Internet, manejando de forma repetida un conjunto cada vez mayor de aplicaciones y fuentes de información que se encuentran accesibles en el Web. Usar los browsers como herramienta para interactuar con los correspondientes enlaces, formularios y extraer los datos relevantes de las páginas, requiere del usuario una **continua interacción** en el proceso de navegación. Ello conlleva que, cuando el volumen de datos es grande o la complejidad de su manejo requiere muchas interacciones por parte del usuario, la realización de estas tareas resulta prohibitiva por su **elevado coste**.

La utilización de programas que naveguen automatizadamente por el Web, manipulando **inteligentemente** la información disponible en Internet es una necesidad cada vez más demandada en numerosos entornos. Sin embargo, el desarrollo de este tipo de programas no es sencillo por numerosas razones. La automatización de tareas que manipulan información en el Web sería mucho más efectiva si el Web tuviese un nivel elevado de estructuración. Sin embargo, el Web actual es un ente **semiestructurado** donde la búsqueda automática de información tiene importantes limitaciones y las etiquetas HTML no proporcionan adecuadas descripciones acerca de los datos que etiquetan. Los documentos del Web suelen presentar estructuras que no favorecen la automatización. La creación de estos documentos ha tenido unos comienzos donde no se prestaba la atención requerida a su estructura interna, estando eminentemente más orientados a su presentación visual que a la adecuada estructuración de sus contenidos. XML [4] ha sido concebido para permitir esta estructuración en los datos del Web. Sin embargo, el Web está basado en HTML y continuará estándolo durante mucho tiempo.

Tradicionalmente el desarrollo de los programas capaces de automatizar tareas en el Web se ha afrontado con técnicas que implican un alto coste de desarrollo y un elevado coste de mantenimiento. Ello ha provocado que estos programas se caractericen por tener una vida demasiado corta y sean muy sensibles a pequeñas modificaciones en las páginas accedidas, quedando frecuentemente inoperativos por esas modificaciones. En este trabajo se proponen la utilización de dos estándares conocidos para reducir el coste de estos desarrollos y mejorar la estabilidad de estos programas. Estos estándares son:

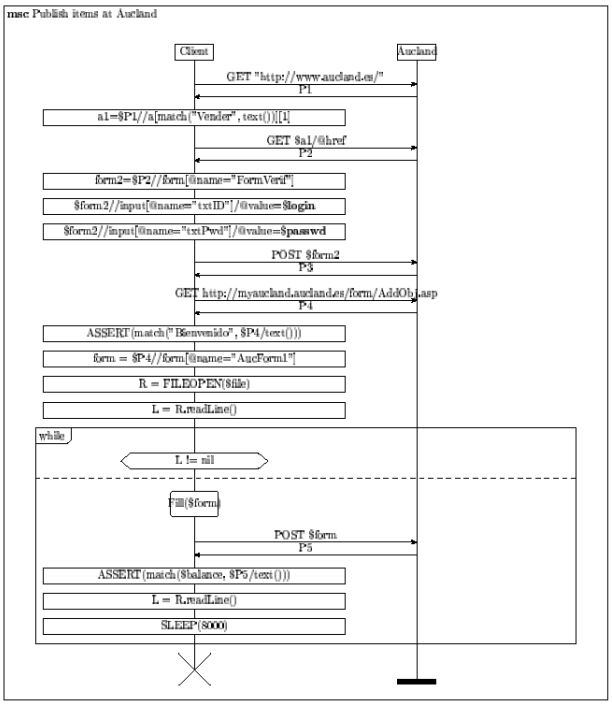
- **XPath** para definir las partes relevantes que deben ser manipuladas en cada página. XPath [3] es un estándar del W3C para el direccionamiento de datos en documentos

XML.

- **MSC** (Message Sequence Charts) para definir las partes relevantes que deben ser manipuladas en cada página. Los MSC [2] son un estándar de la ITU para la representación de escenarios dialogados entre componentes de un sistema distribuido de telecomunicaciones.

2. XPlore: Lenguaje para la navegación y procesamiento de datos en el Web

XPlore es un lenguaje de programación de alto nivel de abstracción para la creación de programas de navegación automática (asistentes de navegación, programas envoltorio, ...), es decir, de programas de navegación, que sirve además como anfitrión a XTendedPath, una extensión del lenguaje XPath 2.0 que aparece detallada en [1]. XPlore es concebido como una adaptación de la representación textual de los MSC enfocada al contexto de la programación de aplicaciones de navegación automatizada, donde, por otro lado, se han añadido extensiones propias. XPlore hereda, por tanto, sus principales características de la representación textual de los MSC, utilizando el subconjunto de elementos más representativos de este formalismo. XPlore es un lenguaje imperativo que permite construcciones típicas de los lenguajes de esa naturaleza, como secuencialización de sentencias, bucles, sentencias condicionales, uso de variables y funciones como aparecen detalladas en las siguientes representaciones gráfica y textual de una sesión en un servidor Web.




```
msc Aucland;

inst aucland;
inst client;

  instance client;

    out GET "http://www.aucland.es/" to aucland;
    in P1 from aucland;

    action a1=$P1//a[match("Vender", text())][1];

    out GET $a1/@href to aucland;
    in P2 from aucland;

    action form2=$P2//form[@name="FormVerif"];
    action $form2//input[@name="txtID"]/@value=$login;
    action $form2//input[@name="txtPwd"]/@value=$passwd;

    out POST $form2 to aucland;
    in P3 from aucland;

    out GET "http://myaucland.aucland.es/form/AddObj.asp" to aucland;
    in P4 from aucland;

    action ASSERT(match("Bienvenido", $P4/text()));
    action form=$P4//form[@name="AucForm1"];
    action $R = FILEOPEN($file);
    action L = $R.readLine();

    while begin;
       condition $L != nil;
    while;
       decomposed Fill($form);
       out POST $form to aucland;
       in P5 from aucland;
       action ASSERT($balance, $P5/text());
       action L = $R.readLine();
       action SLEEP(8000);
    while end;
  endinstance;
endmsc;

msc Fill;
  instance client;
    action $form//input[@name="field"]/@value=value;
    ...
  endinstance;
endmsc;
```

3. Agradecimientos

El trabajo en el que se ha basado este artículo ha recibido el apoyo del proyecto *TEL1999-0207* del Ministerio de Ciencia y Tecnología.

Referencias

[1] V. L. Centeno, L. S. Fernández, C. D. Kloos, P. T. Breuer, and M. E. G. Cabellos. Standards-based languages for programming web navigation assistants. In *5th IEEE International Workshop on Networked Appliances*, pages 70–75, Liverpool, U.K., October 2002.

[2] ITU-T. Recommendation z.120: Message sequence chart (msc). In *Formal description techniques (FDT)*, Geneva, Switzerland, 1997.

[3] W3C. Xml path language (xpath) version 1.0. *W3C Recommendation 16 November 1999*, 1999.

[4] W3C. Extensible markup language (xml) 1.0 (second edition). *W3C Recommendation 6 October 2000*, 2000.

Breve 2B: Modelado y control de tráfico

Encaminamiento con calidad de servicio en redes orientadas a flujo

A. Ariza, E. Casilari y F. Sandoval
Dpto. Tecnología Electrónica. E.T.S.I. de Telecomunicación
Universidad de Málaga. Campus de Teatinos S/N, 29071 Málaga
Teléfono: 952132728 Fax: 9521447
E-mail: alfonso@dte.uma.es

Abstract *A crucial problem in modern telecommunication networks is to define algorithms for QoS (Quality of Service) routers able to optimise the utilisation of network resources. In this work we focus on the analysis of two basic aspects in QoS routing: the election of simple and efficient cost assignment functions to evaluate the cost of link utilisation, and finally the policy for updating the information about the network state in the nodes*

1. Introducción

Uno de los grandes desafíos en las modernas redes de comunicaciones es transportar diferentes tipos de información entre los usuarios, asegurando una cierta calidad de servicio o *QoS* (*Quality of Service*). Para poder satisfacer estos requisitos de *QoS* de forma determinista es preciso usar tráfico orientado a conexión, de forma similar a como han trabajado las redes telefónicas tradicionales. Sin embargo, en contraste con estas últimas, donde sólo se transporta una única clase de tráfico, las nuevas redes deben ser capaces de transportar un tráfico heterogéneo, donde se superponen tráfico con requisitos estrictos de retardo y *Jitter*, con tráficos de datos donde los parámetros de retardo son secundarios y el principal requisito a satisfacer es entregar los datos de forma correcta y sin errores. Para conseguir estos objetivos los algoritmos y protocolos de encaminamiento son una parte básica para el funcionamiento de cualquier red de comunicaciones.

Hay tres aspectos que es preciso estudiar para resolver el problema del encaminamiento orientado a flujo con requisitos de calidad de servicio y que son Algoritmo de búsqueda de caminos, Funciones de asignación de costes y Mecanismos de actualización de estados.

De estos tres aspectos nos centraremos por razones de espacio en los dos últimos.

2. Funciones de asignación de costes

Debido al alto grado de imprecisión de los datos que cabe esperar encontrar en las modernas redes de comunicaciones, creemos que es necesario emplear algoritmos sencillos de asignación de coste en lugar de complejos algoritmos con multitud de parámetros, cada uno de los cuales sujeto a imprecisión, circunstancia que aumentaría el grado de desconocimiento con el que trabajaría el algoritmo de encaminamiento. Entre los algoritmos de asignación de coste que cumple el requisito de simplicidad podemos destacar los siguientes:

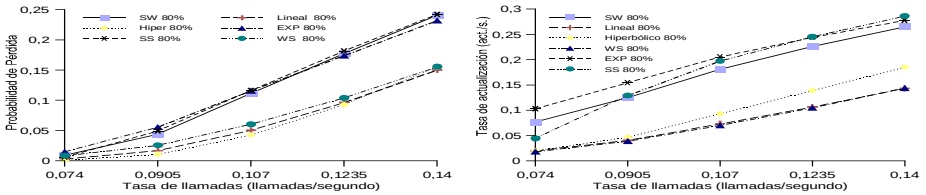
Algoritmo *Shortest-Widest* (SW) [4],
Algoritmo *Widest-shortest* (WS)[4],
Algoritmo *Exp-MC* (Exponencial) [3],
Algoritmo Lineal
Algoritmo *Shortest Safest*(SS)[1].

En la figura 1 se muestra el rendimiento relativo de las distintas funciones de coste. En este experimento el estado de la red se actualiza mediante el método del umbral proporcional, comentado posteriormente en la sección 3, con un umbral (Th) de $0,8$.

De los resultados obtenidos podemos conjeturar que las funciones de asignación que mejor rendimiento presentan, en condiciones de imprecisión en los datos, son la función hiperbólica y la lineal seguida muy de cerca por la WS. En todos los casos el cumplimiento de los parámetros *QoS* de la conexión se ha obtenido mediante el empleo de algoritmos de búsqueda de camino con restricciones. Se observa que es más importante, para el rendimiento de la red, conservar los recursos que el balanceo de la carga entre los distintos enlaces de la red. Los algoritmos Lineal e Hiperbólico, que proporcionan un compromiso entre conservación de recursos y distribución de carga de tráfico son los que mejor rendimiento presentan.

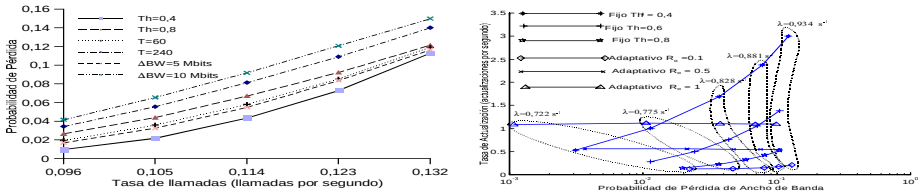
3. Mecanismos de actualización de estados

El encaminamiento con *QoS* exige conocer con la mayor precisión posible el estado de la red. Así pues, un factor a tener en cuenta y que incidirá en el rendimiento final del sistema será el mecanismo empleado para actualizar el estado de la red.



(a) Probabilidad de Pérdida (b) Tasa de actualiaciones generada

Figura 1: Comparación de rendimiento de las funciones de asignación de costes



(a) Probabilidad de Pérdida (b) Actualización adaptativa versus proporcional

Figura 2: Comparación del rendimiento de los algoritmo de actualización de estado más usados

Los mecanismos habitualmente empleados para actualizar el estado de los enlaces han sido, en concreto *actualización por umbral proporcional*, *actualización mediante temporizador*, *actualización mediante umbrales fijos* y *actualización por umbral proporcional* [2].

En la figura 2a se muestra el rendimiento de los mecanismos de actualización de costes descritos. En este experimento comparamos los rendimientos obtenidos al actualizar mediante umbrales de 40 % y 80 % ($Th = 0,4$ y $0,8$), actualización mediante temporizadores de 60 y 240 segundos, y actualización por umbrales fijos de 5 y 10 Mbits/s y en la figura 2b se muestra el rendimiento de la actualización por umbral adaptativo frente a la actualización por umbral proporcional. Se observa claramente que el mecanismo de actualización por umbral adaptativo es el que mejor rendimiento presenta.

4. Conclusiones

La conclusión obvia es la necesidad de introducir mecanismos de encaminamiento orientado a flujos y con encaminamiento en origen si se desea que los algoritmos y protocolos de encaminamiento mejoren el rendimiento de red y soporten de manera activa capacidades de calidad de servicio. En cuanto a los puntos a estudiar para mejorar el comportamiento de los algoritmos y protocolos de encaminamiento hay que señalar tres:

1. Algoritmos de búsqueda de camino. Es necesario encontrar algoritmos más eficientes desde este punto de vista.
2. Funciones de coste. Deben poder trabajar en condiciones de imprecisión en los datos.
3. Mecanismos de actualización. Es necesario usar mecanismos de actualización que verifican dos aspectos parcialmente excluyentes, alto rendimiento en términos de probabilidad de perdida de llamada con una baja tasa de emisión de mensajes de actualización.

Agradecimientos

Este trabajo ha sido parcialmente soportado por el Ministerio de Ciencia y Tecnología través del proyecto TEL99-0755.

Referencias

[1] G. Apostolopoulos, R. Guerin, S. Kamat, and S. Tripathi, *Improving qos routing performance under inaccurate link state information*, Proceedings of the 16th International Teletraffic Congress (ITC'16) (Edimburgo), North Holland Elservier Science Publisher, Junio 1999, pp. 7–11.

[2] A. Ariza, E. Casilari, and F. Sandoval, *Qos routing with adaptive updating of link states*, Electronics Letters **37** (2001), no. 9, 604–606.

[3] R. Gawlick, *Admission control and routing: Theory and practice*, Ph.D. thesis, MIT, 1995.

[4] Q. Ma, *Quality-of-service Routing in integrated Service Networks*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, January 1998.

Estudio de la tasa binaria en comunicaciones Web

E. Casilari, D. F. Castellar y F. Sandoval
Dpto. Tecnología Electrónica, E.T.S.I. Telecomunicación, Universidad de Málaga,
Campus de Teatinos, 29071 Málaga. Tfno.: 34-952132755; FAX 34-952131447
E-mail: ecasilari@uma.es

Abstract. *This work analyses the statistical behaviour of the bandwidth and duration of TCP connections caused by Web transactions. In particular, the paper focuses on the impact of the connection size and Round Trip Time (RTT) on the binary rate. The study is performed basing on synthetic traffic traces which are created by systematic visits to a predefined set of Web pages. The different traces are generated by altering the time interval between page loads and the time of the day in which the experiment is executed. So, this report investigates the possible influence of the user activity pattern on the variability of the connections bandwidth.*

1 Introducción

Las conexiones de datos generadas por las comunicaciones Web constituyen todavía la mayor fuente de tráfico en Internet. De ahí que sean numerosos los estudios realizados en este campo de investigación. Sin embargo, a pesar de la importancia que puede poseer para aspectos como el dimensionamiento del ancho de banda de la red, son relativamente escasos los trabajos, como [1], en los que se estudian con detalle las características de la tasa binaria de las conexiones Web. En este trabajo se analizan las muestras de tráfico obtenidas mediante visitas sistemáticas y programadas a páginas Web. En concreto, se estudia el efecto que poseen el RTT y el tamaño de la conexión sobre la tasa binaria de las conexiones generadas por estas visitas.

2 Descripción de las muestras

Las muestras analizadas se corresponden con el tráfico generado por consultas programadas desde un terminal situado en la red académica de la Universidad de Málaga (UMA) a una lista prefijada de páginas Web. Estas páginas (unas 150 en total, de entre las más populares de la Web) se encontraban situadas en servidores distintos dentro del dominio ".com". El experimento se repitió definiendo tiempos entre visitas de 10, 40 y 120 segundos. Las muestras fueron capturadas mediante Tcpdump y post-procesadas mediante Tcptrace, software que descompone y analiza el tráfico a nivel de conexión, dando estimaciones como el *Round Trip Time* (RTT).

3 Análisis de la tasa binaria

Las muestras arrojaron un valor medio del tamaño de las conexiones en torno a 10 Kbytes. De resultas de este valor medio y el Maximum Segment Size (MSS) típico de 1460 bytes, así como de la naturaleza *heavy tailed* del tamaño de las conexiones, la mayor parte de las conexiones Web poseen un número escaso de paquetes. Esta brevedad de las conexiones da una especial relevancia al mecanismo de arranque lento o *slow-start*, típico de TCP. Así, en las muestras analizadas apenas se detectó la entrada en *congestion avoidance*, ya que la mayoría de las conexiones se encuentra por debajo de los 8 paquetes (más del 75% en todas las muestras). Para estudiar la repercusión de este fenómeno se procedió a analizar la naturaleza estadística de la tasa binaria de las conexiones. De un análisis inicial de la tasa parecería deducirse que la relación entre tamaño de la conexión y tasa es débil y la variabilidad del parámetro es muy elevada. Sin embargo, si se eliminan de la muestra las conexiones que han sido reseteadas (mediante un paquete con el bit de Reset activado) y sólo se consideran aquellas que se cerraron mediante el procedimiento convencional con el bit de FIN (conexiones "completas", véase la Fig. 1), se observan claramente efectos que antes quedaban ocultos. Así, se puede ver cómo influye el mecanismo de *slow-start*, de forma que la tasa binaria crece siguiendo este típico proceso incremental. Aparte, la eliminación de las conexiones reseteadas implica una mayor tasa binaria media y una variabilidad en las mismas mucho menor. Esto supone que una distribución exponencial que caracterizara la tasa de las conexiones completas en función del número de paquetes, no supondría un modelo optimista. Esta distorsión que introduce el mecanismo de Reset en la dinámica de la tasa de las conexiones se puede justificar por el hecho de que este es el procedimiento mayormente empleado para cerrar conexiones reutilizadas para transmitir varios objetos (conexiones persistentes o cerradas con *Keep-Alive*). Por tanto, el cierre mediante Reset se asocia a una mayor variabilidad de la duración de las conexiones, la cual se hace más independiente del propio tamaño de los datos transmitidos y altera la medida de la tasa binaria.

Para corroborar esto, la Tabla 1 compara diversos estadísticos de las conexiones reseteadas con los correspondientes a las acabadas mediante los paquetes con bit de FIN. De la tabla se vuelve a probar la enorme variabilidad en la duración que introduce el procedimiento de las conexiones persistentes (cuya finalización se suele efectuar

mediante un paquete de Reset). Este hecho es especialmente evidente si comparamos la duración media total con la duración de la transmisión de paquetes de datos (eliminando los *handshakes* inicial y final). Por otra parte, de esta comparación deduce que un tiempo entre páginas mayor, aparte de reducir el peso relativo de las conexiones que fueron reseteadas porque no hubo tiempo para cargarlas, permite la finalización de conexiones con servidores de peor acceso, lo que apareja un aumento de la duración y una disminución de la tasa observada.

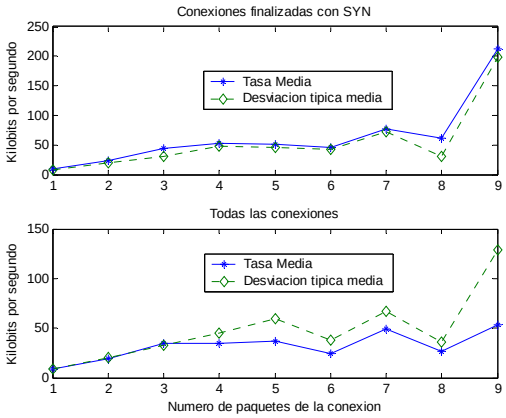


Figura 1. Tasa binaria en función del nº de paquetes de la conexión (T entre páginas de 10 s)

Tabla 1. Comparación entre las propiedades de las conexiones

Tiempo entre páginas	Estadístico (valores medios)	Acabadas con paquetes de FIN	Acabadas con RESET
T=10 s (todas)	Porcentaje sobre el total	59.10%	39.56%
	Ancho de banda (bits/s)	36814	12349
	Duración de la conexión	1.02 s	41.55 s
	Durac. de transm. datos	0.26 s	3.52 s
	Tamaño (bytes)	4590	23840
T=40 s	Porcentaje sobre el total	66.96%	32.37%
	Ancho de banda (bits/s)	32129	9283
	Duración de la conexión	1.24 s	46.36 s
	Durac. de transm. datos	0.39 s	5.52 s
	Tamaño (bytes)	4658	23989
T=120 s	Porcentaje sobre el total	63.25%	33.64%
	Ancho de banda (bits/s)	25561	8156
	Duración de la conexión	2.17 s	47.36 s
	Durac. de transm. datos	0.63 s	4.97 s
	Tamaño (bytes)	4691	23342

Igualmente, se estudió la relación entre tasa y RTT. En [2] Mathis define un límite a la tasa de transferencia (BW_i) que puede llegar a alcanzar una conexión TCP: $BW_i \leq \frac{MSS_i}{RTT_i \cdot \sqrt{p}}$, donde RTT_i es el *Round Trip Time* de la conexión, p la probabilidad de pérdida de paquete y MSS_i el *Maximum Segment Size* negociado por la conexión. En [3] Padhye afina la fórmula anterior teniendo en cuenta aspectos como el tamaño máximo de transmisión o el tiempo inicial para la retransmisión. Ambos modelos, no obstante, asumen conexiones TCP de larga duración más propias de servicios como FTP. La Fig. 2 muestra que para las conexiones de corta duración (la gran mayoría de los flujos *Web*), estos límites aportan valores demasiado optimistas.

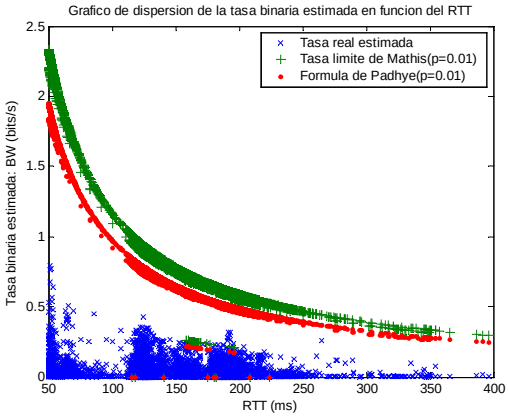


Figura 2. Tasa binaria de las conexiones de todas las muestras en función del RTT y comparación con fórmulas de Mathis y Padhye

En la gráfica anterior, para calcular los límites propuestos por Padhye y Mathis se supuso una probabilidad p=0.01. Este valor fue elegido teniendo en cuenta la probabilidad estimada de retransmisión de paquetes, que oscilaba entre 0.0038 y 0.0084 (0.38% y 0.84%). No obstante, a pesar de que estos límites no llegan a alcanzarse, se puede comprobar la dependencia hiperbólica entre tasa binaria media estimada y el RTT.

3 Conclusiones

Partiendo de muestras sistemáticamente generadas en un navegador, el estudio demuestra la importancia de las conexiones reseteadas sobre la variabilidad de la tasa binaria de las conexiones Web. Esta importancia se justifica por la existencia de conexiones reutilizadas para cuyo cierre normalmente se emplea el mecanismo de Reset. Estas conexiones, que posibilitan la multiplexación de peticiones GET, obligan realmente a replantearse el concepto de tasa en tanto que introducen mecanismos de cierre por *time-out* que alargan la duración de las conexiones.

Agradecimientos

Este trabajo ha sido financiado en parte por el proyecto MCYT Nº TEL99-0755.

Referencias

[1] J. Aracil y D.Morató, “Characterizing Internet Load as a Non-regular Multiplex of TCP Streams”, Proc .of ICCCN 2000, Las Vegas, Nevada (USA), Octubre 2000, pp. 94-99.

[2] M. Mathis et al.,“The macroscopic behavior of the TCP congestion avoidance algorithm”, *Computer Communication Review*, Vol.27, No. 3, Julio 1997, pp. 67-82.

[3] J. Padhye et al. “Modelling TCP throughput: A simple model and its empirical validation” in Proc. SIGCOMM, Agosto 1998, pp. 304-314.

Prestaciones de un Sistema Turbo-CDMA con Códigos Espacio - Tiempo

Alexandre Graell i Amat, Boris Bellalta i Jiménez
Departamento de Tecnologia. Universitat Pompeu Fabra
Edifici Estació de França, Passeig de la Circumval·lació, 8. 08003 Barcelona
Teléfono: 93 542 29 45 Fax: 93 542 24 51
E-mail: {alex.graell,boris.bellalta}@tecn.upf.es

Abstract *In this paper, we analyze the performance of a turbo coded CDMA system with transmit diversity at the mobile station, by employing Space-Time Codes. The effect of code assignment (by user or by antenna) is studied and simulation results of FER performance are presented assuming a fast fading channel. The joint utilization of code division multiple access and Space-Time Codes appears to be a promising approach for multiuser wireless systems.*

1. Introducción

La idea de mejorar la fiabilidad y capacidad de los sistemas de comunicaciones móviles mediante el uso de diversidad ha sido un argumento ampliamente estudiado durante años. Históricamente, se han utilizado distintas técnicas de diversidad (temporal, frecuencial, códigos, espacial) para contrarestar los efectos del fading en los canales inalámbricos, proporcionando significativas mejoras de las prestaciones. En los últimos años, ha habido un interés creciente por combinar diferentes técnicas de diversidad. En este sentido, Tarokh *et al.* han propuesto recientemente los llamados códigos espacio-tiempo [1] (STC), que combinan los beneficios de la codificación a corrección de errores con la diversidad espacial, mediante el diseño conjunto del código de canal, modulación y diversidad espacial. Los códigos espacio-tiempo proporcionan elevadas prestaciones a cambio de un cierto incremento en complejidad.

Motivados por las asombrosas prestaciones de los códigos espacio-tiempo, en este artículo analizamos el impacto de la diversidad espacial en el terminal móvil sobre las prestaciones de un sistema multiusuario. Para ello se evalúa un simple esquema genérico DS-CDMA multiusuario, donde cada usuario utiliza distintas antenas en transmisión.

2. Modelo del sistema

En la Figura 1 se muestra el modelo del sistema considerado en este artículo. Se ha supuesto un sistema con M usuarios, transmitiendo de manera síncrona y simultánea. Para cada usuario, la secuencia de información $\mathbf{u_m}$, $m = 1 \dots M$, se codifica con un código turbo definido en las especificaciones del 3GPP [2]. Concretamente, en este artículo utilizamos el código turbo formado por la concatenación de dos códigos a 8 estados, de tasa global 1/3. La secuencia de código generada, $\mathbf{x}_m$, es permutada por el entrelazador de canal definido en [2]. Actuando sobre tiempo genera a su salida n_T símbolos en cada instante de tiempo, transmitidos simultáneamente por sendas antenas transmisoras, utilizando una modulación QPSK. Finalmente, sobre cada una de las secuencias generadas se aplica una secuencia de ensanchamiento de longitud L, generada aleatoriamente.

Con el objetivo de minimizar la complejidad del sistema, se ha considerado un esquema en el cual cada usuario dispone de 2 antenas en transmisión. El número de antenas receptoras se ha fijado a 2. La transmisión se lleva a cabo sobre un canal con desvanecimientos rápidos. La señal recibida es una superposición filtrada de las señales transmitidas. Para cada antena receptora, j, la señal recibida (L chips) puede expresarse como

$$\mathbf{r}_j(t) = \sum_{m=1}^{M} \sum_{i=1}^{n_T} h_{ij,m}(t) \left[s_{i,m}(t) \mathbf{s}\mathbf{s}_{i,m} \right] + \mathbf{n}_j(t) \qquad (1)$$

donde $h_{ij,m}(t)$ denota los coeficientes de fading de la antena transmisora i a la antena receptora j para el usuario m en el instante de tiempo t. Las antenas se suponen suficientemente espaciadas de modo que los coeficientes de fading $h_{ij,m}$ son incorrelados. $\mathbf{n}_j(t)$ es el vector de longitud L que representa el ruido Gausiano blanco (AWGN), modelado como muestras independientes de un proceso Gausiano complejo de media 0 y varianza $N_0/2$ por dimensión.

3. Simulaciones

En esta sección se presentan algunos resultados del impacto de la diversidad espacial y del criterio de asignación de secuencias de ensanchamiento a cada usuario sobre las prestaciones del sistema DS-CDMA.

En el modelo de canal considerado, los coeficientes

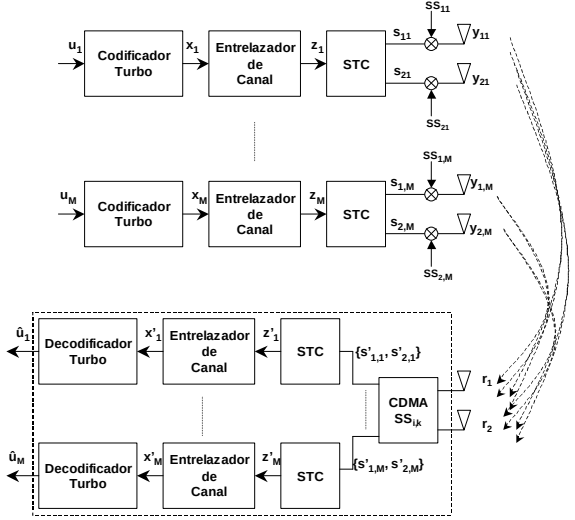


Fig. 1. Modelo del sistema.

chip. Los coeficientes del canal se consideran incorrelados y varían aleatoriamente cada L periodos de chip. Estos, se modelan como un proceso Gaussiano complejo, de media cero y varianza 0.5 por dimensión. Finalmente, la señal recibida se ve afectada por ruido Gausiano blanco.

En las figuras 2 y 3 se muestran las prestaciones del sistema multiusuario con diversidad propuesto en términos de FER para una longitud de bloque de información de $F = 40$ bits y una longitud de secuencia de ensanchamiento $L = 31$ y $L = 127$, respectivamente. Las curvas etiquetadas como NSTC se refieren a un sistema que no utiliza diversidad espacial (una sola antena por usuario). Se han evaluado dos opciones: en la primera, se asignan dos secuencias de ensanchamiento a cada usuario, una para cada una de las antenas transmisoras (1 seq./ant.); en la segunda, una sola secuencia es asignada a cada usuario (1 seq./us.), común a todas las antenas del usuario. Se observa que la utilización de una única secuencia por usuario mejora sensiblemente las prestaciones. Por ejemplo, en el caso de 8 usuarios se observa una ganancia de 1 dB para una probabilidad de error de 10^{-2}. Se aprecia así mismo una importante ganancia de unos 6 dB para 1 seq./ant. en relación al sistema sin diversidad a $FER = 10^{-3}$.

El hecho que la asignación de secuencias de ensanchamiento distintas a cada antena del usuario repercuta negativamente en las prestaciones del sistema, puede explicarse por la naturaleza de los códigos espacio-tiempo. En este sentido, el uso de dos secuencias distintas repercute negativamente reduciendo la ganancia de diversidad inherente del código.

El efecto de aumentar la longitud de la secuencia de ensanchamiento se muestra en la figura 3. Como era de esperar, para el mismo número de usuarios, se aprecia una mejora de las prestaciones. Como en el caso anterior, el uso de diversidad introduce una importante ganancia respecto al sistema sin diversidad, siendo esta más importante cuando se asigna una única secuencia de ensanchamiento a cada usuario. En este caso, se observa una ganancia de unos 6 dB con respecto al sistema sin diversidad a $FER = 10^{-3}$ para un sistema con 8 usuarios. Por otro lado se aprecia que la ganancia respecto al sistema NSTC aumenta con el número de usuarios.

4. Conclusiones

En este trabajo se ha evaluado el impacto de la diversidad espacial en el terminal móvil sobre las prestaciones de un sistema multiusuario. Considerando un código espacio-tiempo sencillo y dos antenas transmisoras por usuario, se han comparado las prestaciones del sistema para dos asignaciones de secuencias de ensanchamiento distintas: una secuencia por usuario común a las dos antenas, o una secuencia independiente por antena. Se ha observado que el uso de diversidad espacial mejora significativamente las prestaciones respecto el mismo sistema carente de diversidad. La mejora es más importante en el caso de asignación de una sola secuencia por usuario. Se observa así mismo que la ganancia en BER y FER introducida por la diversidad aumenta con el número de usuarios.

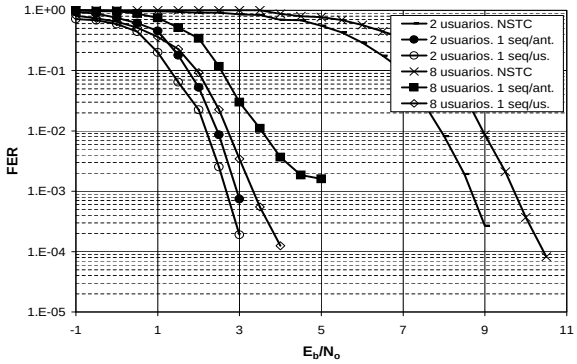


Fig. 2. Prestaciones del sistema en términos de FER. $F = 40$ bit. $L = 31$.

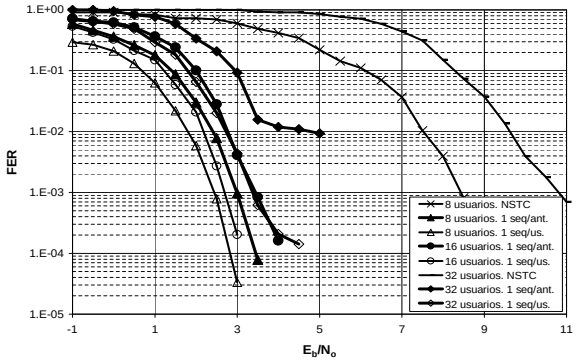


Fig. 3. Prestaciones del sistema en términos de FER. $F = 40$ bit. $L = 127$.

Referencias

[1] V. Tarokh, N. Seshadri, and A. R. Calderbank, *Space-Time Codes for High Data Rate Wireless Communication: Performance Criterion and Code Construction*, IEEE Trans. Inf. Theory, vol. 44, no. 2, March 1998.

[2] 3rd Generation Partnership Project (3GPP), 3G TS 25.212, v3.5.0, *Multiplexing and Channel Coding (FDD)*, Dec 2000.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima, *Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes*, In Proc. IEEE Intl. Conf. Comm., pp. 1064-1079, 1993.

Modelado en SDL de la Capa 2 de UMTS

Beatriz Soret, Victoria Morillo-Velarde, Jesús Colás, Javier Poncela
Departamento de Ingeniería de Comunicaciones. Universidad de Málaga.
ETSI Telecomunicación, s/n. Campus de Teatinos, 29071 Málaga.
Teléfono: 952 13 27 59 Fax: 952 13 20 27
E-mail: javier@ic.uma.es

***Abstract**. Formal visual languages such as SDL help improve the quality of the code, making the maintenance of the code less costly with the help of automatic code generation tools. This work presents the implementation, in SDL, of the layer 2 of the UMTS Radio Access Network front-end (UTRAN). Tests have been carried out to measure the efficiency of the implementation in widely used operating systems.*

1 Introducción

Basado en los protocolos de GSM, UMTS [1] proporcionará una mayor tasa de transferencia de datos, hasta 2 Mbps, y un mejor uso del espectro. La interfaz aire consta de dos elementos: el equipo del usuario (UE) y el terminal de la red de acceso radio (UTRAN).

La arquitectura de protocolos de UMTS se divide en tres niveles: el nivel físico (PHY), el nivel de enlace (MAC, RLC) y el nivel de red (RRC). El modelado del sistema se ha realizado en SDL [2] (*Specification and Description Language*), utilizando la herramienta comercial Tau [3]. Se ha implementado un modelo de las capas MAC y RLC, particularizadas para el caso de una UTRAN. Además, se han realizado emuladores de las capas RRC y PHY para probar el funcionamiento del sistema.

2 MAC

La capa MAC se divide en varias entidades, según se trate de canales compartidos o canales dedicados. El lenguaje SDL ofrece dos alternativas para modelar estas entidades: procesos o servicios. La principal ventaja de utilizar servicios es que estos pueden compartir variables. De esta forma, la información de configuración se puede almacenar en variables compartidas por todos los servicios, evitando el uso de señales con información interna de control. En el caso de la capa MAC la velocidad de procesamiento podía resultar crítica, ya que puede haber canales de transporte de hasta 2 Mbps, por lo que se eligió el modelado mediante servicios, como puede verse en la Fig. 1.

Para realizar el mapeo de canales lógicos y canales de transporte, la capa MAC almacena información de configuración relativa a cómo se mapean canales de transporte, canales lógicos y portadoras.

En UMTS el tiempo se divide en intervalos de transmisión (TTI – *Time Transmission Interval*) de 10ms. El nivel físico es el responsable de la adecuada temporización del sistema. La capa MAC recibe del nivel físico una indicación de inicio de TTI (PHY_STATUS_IND). Como respuesta, se envía una primitiva de datos por cada canal de transporte planificado en ese TTI. En función del volumen de datos que RLC tiene pendientes de transmitir, MAC calcula los formatos de transporte de cada canal, de entre las configuraciones permitidas por RRC, para el próximo TTI y solicita de RLC los datos a transmitir. La selección de formato de transporte también tiene en cuenta la prioridad de los distintos canales lógicos.

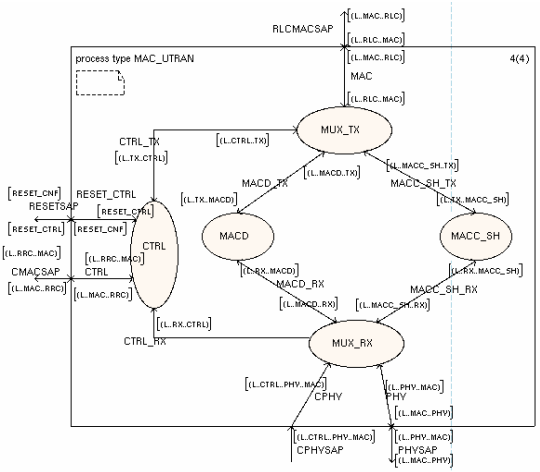


Figura 1. Diseño de la entidad MAC para la UTRAN

3 RLC

Para la construcción de la capa RLC se planteó en primer lugar una implementación basada en servicios similar a la realizada para la capa MAC. La característica de no-concurrencia de los servicios de SDL los hacen adecuados para el tratamiento de las primitivas de datos de los diferentes canales, aunque quizá no para modelar el control de la capa, que podría tardar excesivamente en recibir nuevas configuraciones mientras se procesan los datos que llegan de MAC o de las capas superiores. Una propuesta de estructura de la capa RLC basada en procesos se presenta en la Fig. 2.

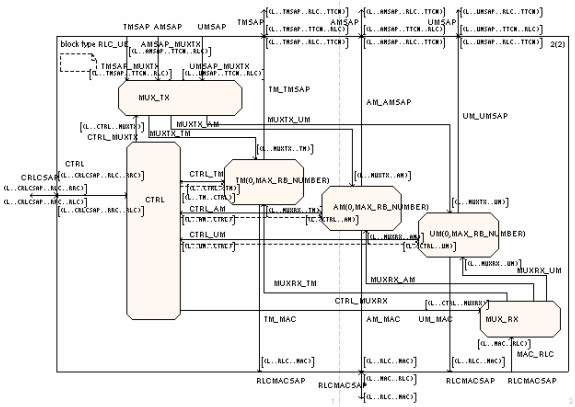


Figura 2. Diseño de la entidad RLC para la UTRAN

El bloque RLC consta de un proceso que maneja las primitivas de control (CTRL), un multiplexor de transmisión, un multiplexor de recepción y un proceso por cada portadora utilizada. La interfaz con las capas adyacentes está compuesta de 5 SAPs: un SAP para cada modo, uno de control y otro de comunicación con la capa MAC.

Los procesos que controlan cada portadora son diferentes según el modo de transferencia (TM, AM, UM). El proceso de control, de forma dinámica, crea los procesos que gestionan, individualmente, cada portadora. Tanto el proceso de control como los multiplexores almacenan, en una copia local, los identificadores de estos procesos. Aunque esto implica una mayor carga de gestión, acelera la multiplexación en recepción. El cifrado se ha implementado tanto para los modos confirmado como no confirmado. En el caso del modo transparente, es la capa MAC la encargada de cifrar la información.

Tabla 1: Eficiencia relativa del uso de servicios o procesos para la comunicación de variables

		Tiempo (s)	miles_sñs/s
Servicios		57.683	18.34
		86.855	18.4
Procesos	Export-import	-	10.86/10.84
	Parámetro	-	10.43
	Módulo C	446.181	23.35

La compartición de información entre procesos se ha implementado mediante la inclusión de Código C externo en el sistema SDL, ya que ofrece mejores prestaciones que el intercambio de señales[1] o el mecanismo export-import (Tabla 1).

4 Pruebas

Se han realizado dos tipos de pruebas: de comportamiento y de rendimiento. Para realizar pruebas de comportamiento se ha empleado el simulador que incorpora la herramienta. En el sistema SDL se ha implementado un emulador de RRC y un emulador de la capa física (Fig. 3), que lleva la sincronización del sistema y recibe y envía primitivas de datos. Se ha realizado también un bloque que emula el comportamiento del UE. Con este entorno de pruebas se han simulado diversas configuraciones, en las que se activan cualquiera de los modos RLC (TM, UM y AM) y los distintos tipos de canales lógicos y de transporte, ascendentes y descendentes.

Para las pruebas de rendimiento se ha generado una aplicación a la que se añadió una interfaz de usuario que controla las medidas. Se quería saber la tasa de bit que alcanzaba el sistema en bps. Para tasas bajas (12.2 kbps) tanto el sistema operativo Windows como Linux se comportan bien. La máxima velocidad de bit probada ha sido de 384 kbps; en este caso, Windows no alcanza la velocidad, produciéndose un deterioro de la tasa de bit conforme se incrementa el intervalo medido. Por su parte, en Linux sí se obtiene la tasa deseada de 384 kbps.

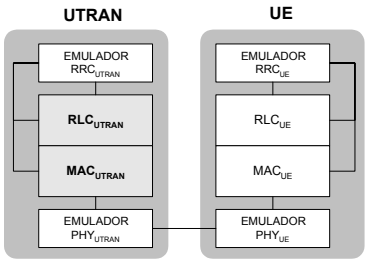


Figura 3. Entorno de pruebas

5 Conclusiones

En este artículo se ha descrito la implementación en SDL de la capa 2 de la entidad UTRAN del sistema de comunicaciones móviles UMTS. Se ha analizado la eficiencia que proporciona el uso de procesos o de servicios, optándose por unos u otros según las necesidades de concurrencia. El sistema diseñado ha sido simulado para verificar su comportamiento, para lo cual se ha construido un emulador de Equipo de Usuario. El código resultante ofrece unas prestaciones adecuadas en sistemas operativos de amplio uso como Linux y Windows en equipos de baja velocidad.

Agradecimientos

Este trabajo se ha realizado en el marco de una colaboración con la empresa CETECOM y ha sido objeto de ayuda con cargo al presupuesto de gastos del Ministerio de Ciencia y Tecnología (proyecto FIT-070000-2002-51)

Referencias

[1] Third Generation Partnership Project. http://www.3gpp.org/.

[2] ITU-T, Recommendation Z.100, *Specification and Description Language (SDL),* (1996).

[3] Telelogic AB. *Tau 4.3 User's Guide, Tau 4.3 Reference Manual*, (2002).

[1] Este efecto es más acusado conforme aumenta el tamaño de los parámetros de la señal.

Detección Activa de Pérdidas de Paquetes en Flujos de Audio en Tiempo Real†

Juan José Ramos Muñoz*, Juan Francisco Núñez Negrillo, Juan Manuel López Soler*
Área de Ingeniería Telemática. Departamento de Electrónica y Tecnología de los Computadores.
Universidad de Granada. ETSI Informática. 18071 Granada
Teléfono(*): +34 958 24 23 03 ; Fax(*): +34 958 24 08 31
E-mail(*): {jjramos,juanma}@ugr.es

Abstract. *Packet loss detection at routers can improve the performance of multicast applications in terms of recovery delay, a critical issue in streams with real-time constraints. In this paper, we describe, analyze and evaluate four techniques by simulation. We demonstrate that mechanisms based on the prediction of packet arrival time outperform classical packet loss detection performance.*

1 Introducción

El impacto de las pérdidas de paquetes en cuanto a la calidad de la reconstrucción de la señal transmitida en flujos continuos con requisitos de tiempo real hace necesario el empleo de técnicas de recuperación que palien dicho efecto. La tecnología de redes activas permite llevar a cabo la detección y la recuperación de pérdidas ([1], [2]) en los *routers* intermedios. La detección temprana de paquetes perdidos en aplicaciones con demandas de tiempo real redunda en la posibilidad de ejecutar mecanismos de recuperación efectivos, dado que la información transmitida es útil durante un tiempo de validez determinado por la restricción de tiempo real.

2 Procedimientos para la detección de pérdidas

2.1 Detección por alteración de secuencia (DAS)

Este método se basa en la numeración secuencial de las tramas de audio generadas en el emisor, tal que una discontinuidad en los números de secuencia se asociará a una ráfaga de pérdidas ([1], [2]).

Este sencillo esquema asume que los paquetes no sufren ningún cambio de orden en la red. Su principal debilidad reside en el tiempo empleado en detectar las ráfagas de pérdidas, ya que se basa en la llegada del primer paquete tras las pérdidas, y hasta que no termine la ráfaga, no se manifestará el problema. Este retardo puede suponer la imposibilidad de recuperar los paquetes perdidos. Ante pérdidas aisladas la detección requiere un intervalo $t_f + t_{jitter}$, medido desde el instante esperado para recibir el paquete perdido, siendo t_f el tiempo que transcurre entre la generación de dos tramas consecutivas en el emisor, y t_{jitter} la fluctuación temporal acumulada para el paquete recibido tras la pérdida

2.2 Detección por expiración de tiempo con umbral fijo (*DET*)

Este mecanismo efectúa una detección de pérdida toda vez que expire un temporizador cuyo valor máximo U será constante y especificado de antemano. Dicho temporizador se reinicia con la llegada del primer paquete y en los posteriores múltiplos de t_f (conocido por el detector).

Para el paquete con número de secuencia i ($i>1$), este procedimiento detectará una pérdida siempre que no se reciba un paquete antes de $(i-1)\cdot t_f + U$ desde la llegada del primer paquete desde el origen.

La principal ventaja de este método es su buen comportamiento frente a ráfagas, ya que detectaría una pérdida con periodicidad t_f, con un retardo para cada detección acotado por U. Las debilidades de este método radican en su dependencia con el *jitter* del primer paquete, y en la elección del valor óptimo de U. Subestimar U provoca FAs (*falsas alarmas*, notificaciones prematuras de pérdidas no producidas). Por contra, un valor para U demasiado elevado puede admitir retrasos inaceptables para la recuperación.

2.3 Detección por expiración de tiempo adaptable (*DET*$^+$)

Para paliar las deficiencias de *DET,* este método adapta dinámicamente un umbral U_a en función de las condiciones de la red estimadas.

† Este trabajo ha sido parcialmente financiado por el proyecto de investigación TIC2002-02798 del Ministerio de Educación y Ciencia, financiado en un 70% por fondos FEDER, y por una beca concedida por la SEEU del PNFPU, referencia AP2002-3895.

La estimación óptima de U_a, implicará que el procedimiento de recuperación de pérdidas se inicie lo más próximo posible al instante en que se produjo la pérdida. Para satisfacer las demandas de retardo extremo a extremo acotado, el umbral adaptable nunca será mayor que U (umbral del método no adaptable).

El procedimiento de adaptación de U_a se basa en la extrapolación lineal de las diferencias entre el instante de llegada y los múltiplos de t_f contados desde la recepción del primer paquete. Se supone así que el tiempo ideal de llegada del paquete i, (con $i>1$), será $(i-1)\cdot t_f$, medido desde la llegada del primer paquete. El retraso considerado para un paquete perdido es U, a efectos de cálculo en la extrapolación.

Este método ofrece tiempos de respuesta mejores que los de *DET* para un mismo valor de U, dado que el umbral de detección U_a será menor o igual que U, pero cabe la posibilidad de que se generen un mayor número de FAs, si los aumentos del retardo crecen más rápidamente que la estimación adoptada.

2.4 Detección por expiración de tiempo adaptable y numeración secuencial (*DET*$^+$+*DAS*)

Este esquema combina los procedimientos *DET*$^+$ y *DAS*, obteniendo simultáneamente las ventajas de ambos, siempre y cuando el umbral del valor máximo U esté bien elegido.

3 Evaluación

Mediante simulación sobre *network simulator* [3] se evalúa el funcionamiento de los detectores en un *router activo,* en términos de tiempo de validez que resta a una trama tras ser detectada como perdida, y del porcentaje de FAs. En la Fig. 1 se representan algunos resultados. Los puntos de las curvas corresponden al porcentaje de detecciones efectuadas según los esquemas, dado el tiempo restante de validez de la trama. En la leyenda se indica para cada método el porcentaje de FAs sobre el número total de detecciones efectuadas.

Los resultados mostrados corresponden a 10 simulaciones de 2000 s, con un total de 640792 paquetes emitidos. Las pérdidas se introducen congestionando un *router* previo al activo, causado por la generación a ráfagas de tráfico UDP y TCP, caracterizadas por la longitud de la ráfaga y el intervalo de envío entre paquetes, que siguen sendas distribuciones normales (en la Fig.1 una normal $N(3,10^{-3})$ y una $N(10^{-4},10^{-4})$, respectivamente). El valor del umbral U considerado para *DET, DET*$^+$ y *DET*$^+$+*DAS* es 80 ms .

Se puede observar que *DET* realiza todas las detecciones en un instante determinado por U. Así pues, todos los paquetes detectados disponen del mismo tiempo de validez para ser recuperados (0,129 (s) en la simulación realizada). La elección conservadora de U evita las FAs, pero las detecciones no siempre se realizan lo antes posible.

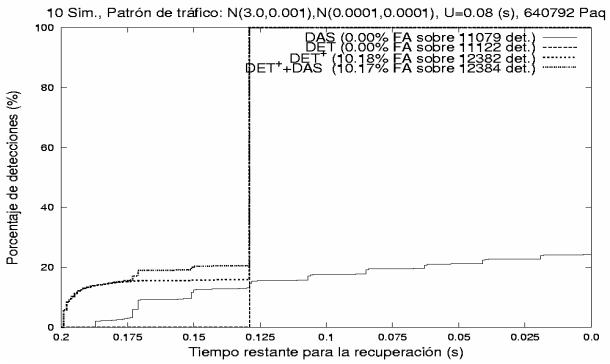


Figura 1: Histograma acumulado del tiempo de validez restante de la trama tras la detección.

DET$^+$ mejora los tiempos de respuesta, como se puede ver en Fig. 1, realizando detecciones antes que *DET*. Sin embargo se producen mayor número de FAs. Y dado que la adaptación en caso de pérdidas es conservadora, converge rápidamente al tiempo de respuesta determinado por U.

Como se puede comprobar en la Fig. 1, *DAS* puede mejorar inicialmente a *DET*, e incluso a *DET*$^+$ en algunos casos (ráfagas que duren menos que U). Sin embargo *DET*$^+$+*DAS* mejora a ambos, obteniendo igual número de FAs que *DET*$^+$.

4 Conclusiones

El empleo de esquemas de detección en nodos intermedios de la red mejora tanto el tiempo de respuesta ante una pérdida, como aumenta el tiempo disponible para realizar labores de recuperación. Los esquemas basados en temporizadores palian el problema de insensibilidad al retardo del que adolece DAS. Sin embargo, para obtener mejoras es necesaria una adaptación menos conservadora, si bien en entornos con fluctuaciones muy cambiantes puede incurrir en un mayor porcentaje de falsas alarmas.

Referencias

[1] C. Pham, M. Maimour. *An analysis of a router-based loss detection service for active reliable multicast protocols*. In IEEE. International Conference on Networks (ICON 2002), pp 140-152. IEEE Computer Society, August 2002.

[2] Calderon, M.; Sedano, M.; Azcorra, A.; Alonso, C.; *Active Network Support for Multicast Applications.* IEEE Network vol. 12, n. 3, pp. 46-52 May/June, 1998.

[3] S. Floyd, S. McCanne. *NS: Network Simulator.* 1997, http://www.isi.edu/nsnam/ns