X Jornadas de Ingeniería Telemática

JITEL 2011

UNIVERSIDAD DE CANTABRIA

28 al 30 de septiembre de 2011

Universidad de Cantabria

ISBN: 978-84-694-5948-5

Editores: Klaus Hackbarth, Ramón Agüero, Roberto Sanz
(Universidad de Cantabria)

Patrocinadores

Colaboradores

Presentación

Las Jornadas de Ingeniería Telemática (JITEL) cumplen su décima edición, desde que se celebraron, por primera vez, en Bilbao en 1997. Periódicamente, las personas e instituciones que trabajan en el área de Ingeniería Telemática se reúnen bajo el patrocinio de la Asociación de Telemática (ATEL), con el principal objetivo de fortalecer el debate y divulgación de las temáticas más vanguardistas en el campo de la Ingeniería Telemática.

La Universidad de Cantabria ha tenido el honor de ser designada responsable de organizar las X Jornadas de Ingeniería Telemática (JITEL 2011); al igual que sucediera el año anterior en Valladolid, la celebración se lleva a cabo compartiendo tiempo y espacio con el mayor evento de investigación, desarrollo e innovación en el ámbito de las tecnologías de la información y las comunicaciones en España, las XXI Jornadas de Telecom I+D, centradas este año en las ciudades inteligentes. Ambos eventos tienen lugar entre los días 28 y 30 de septiembre de 2011, en el Palacio de la Magdalena en Santander, facilitando el intercambio de ideas entre los diferentes participantes y la identificación de sinergias en las distintas temáticas.

Asimismo, teniendo en cuenta el éxito de la primera edición, el programa vuelve a integrar las II Jornadas de Innovación Educativa en Ingeniería Telemática (JIE 2011), un aspecto fundamental, teniendo en cuenta el profundo cambio que está sufriendo la universidad española en la conversión al Espacio Europeo de Educación Superior. Las JIE suponen una oportunidad única de compartir experiencias entre los colegas del área, de manera que todos podamos enriquecer nuestra labor docente.

Este libro recoge las contribuciones que fueron aceptadas para su presentación en las jornadas. Tras un riguroso proceso de revisión, en el que se aseguró, bajo la coordinación del Comité de Programa, que cada artículo recibiera un mínimo de 3 revisiones independientes, se aceptaron 44 artículos para su presentación oral y 14 en formato póster. El programa se ha estructurado en 8 sesiones técnicas, en las que los trabajos se agrupan en función de su contenido, además de la sesión de posters.

Además de la exposición de los trabajos, el programa se completa con dos ponencias invitadas, que se llevan a cabo junto con las Telecom I+D. En la charla inaugural, Paul Moore Olmstead (Atos) analizará las oportunidades que aparecen, principalmente desde la perspectiva de la innovación y la investigación, a partir del paradigma de la *Ciudad Inteligente*. Por su parte, D. José Jiménez (Telefónica) cerrará las jornadas, presentando su visión sobre las nuevas posibilidades que se abren con la *Internet del Futuro*. Además, teniendo en cuenta lo significativo de estas jornadas, se ha organizado una mesa redonda en la que algunos de los principales promotores de la Ingeniería Telemática en España ofrecerán su visión sobre la situación actual del área, así como las perspectivas y retos a afrontar en el futuro.

Es importante destacar el papel de la Asociación de Telemática (ATEL) como entidad organizadora de JITEL 2011, reuniendo personas y organismos, universidades, centros de investigación, empresas o fundaciones del área de Ingeniería Telemática. La asamblea anual y la reunión de socios de ATEL también se llevan a cabo dentro de este espacio de encuentro.

Destacar por otro lado el apoyo de los patrocinadores y colaboradores (el Ministerio de Ciencia e Innovación, el Gobierno de Cantabria, el Ayuntamiento de Santander, la Universidad de Cantabria, así como el Banco Santander y TST); sin su desinteresado apoyo y contribución, las jornadas no se podrían haber llevado a cabo.

Finalmente, hay que destacar el papel jugado por todas las personas que han dedi-

cado su energía, entusiasmo y capacidad profesional para llevar a cabo estos eventos. El Comité de Programa ha realizado una labor fundamental en el proceso de elaboración del programa; los investigadores que han participado en el proceso de revisión han permitido mejorar la calidad científica de las contribuciones; finalmente, todos los miembros del Comité Organizador han hecho de las jornadas una realidad.

Os queremos dar la más cordial bienvenida a las X Jornadas de Ingeniería Telemática (JITEL 2011), esperando que constituyan un verdadero éxito, tanto en sus actividades científicas como sociales. Santander, septiembre de 2011

Santander, septiembre de 2011
Klaus Hackbarth *(Presidente del Comité de Programa)*
Luis Muñoz *(Presidente del Comité Organizador)*

Comité de programa

Amor Pinilla, Mercedes (Universidad de Málaga)
Aracil, Javier (Universidad Autónoma de Madrid)
Bagnulo, Marcelo (Universidad Carlos III de Madrid)
Carneiro, Victor (Universidad de La Coruña)
Corral, Guiomar (Universidad Ramón Llull)
Díaz Verdejo, Jesús (Universidad de Granada)
Dimitriadis, Ioannis (Universidad de Valladolid)
Estepa, Rafael (Universidad de Sevilla)
Felici Castell, Santiago (Universidad de Valencia)
Fernández Navajas, Julián (Universidad de Zaragoza)
García, Roberto (Universidad de Oviedo)
García Galán, Sebastian (Universidad de Jaén)
Giménez Guzmán, José Manuel (Universidad de Alcalá)
Gómez Oliva, Ana (Universidad Politécnica de Madrid)
Gómez Skarmeta, Antonio (Universidad de Murcia)
González Sánchez, José Luis (Universidad de Extremadura)
Gozálvez, Javier (Universidad Miguel Hernández de Elche)
Hackbarth, Klaus (Presidente) (Universidad de Cantabria)
Hesselbach Serra, Xavier (Universidad Politécnica de Cataluña)
Jacob, Eduardo (Universidad del País Vasco)
López Ardao, José Carlos (Universidad de Vigo)
Manzanares López, Pilar (Universidad Politécnica de Cartagena)
Martínez Bauset, Jorge (Universidad Politécnica de Valencia)
Morato, Daniel (Universidad Pública de Navarra)
Oliver, Miquel (Universidad Pompeu Fabra)
Payeras Capellà, María Magdalena (Universidad de las Islas Baleares)
Salvachua, Joaquín (Universidad Politécnica de Madrid)
Sánchez, Luis (Universidad de Cantabria)
Sánchez, Luis (Universidad Carlos III de Madrid)
Suárez Sarmiento, Álvaro (Universidad de Las Palmas de Gran Canaria)
Velasco, Juan (Universidad de Alcalá)
Verdú Pérez, María Jesús (Universidad de Valladolid)

Comité organizador

Agüero, Ramón (Universidad de Cantabria)
García Arranz, Marta (Universidad de Cantabria)
García Gutiérrez, Alberto Eloy (Universidad de Cantabria)
Hackbarth, Klaus Dieter (Universidad de Cantabria)
Irastorza, José Ángel (Universidad de Cantabria)
Lanza Calderón, Jorge (Universidad de Cantabria)
Muñoz, Luis (Presidente) (Universidad de Cantabria)
Sánchez González, Luis (Universidad de Cantabria)
Sanz Gil, Roberto (Universidad de Cantabria)

Revisores

Aguado, Marina (Universidad del País Vasco)
Agüero, Ramón (Universidad de Cantabria)
Aguilar, Mónica (Universidad Politécnica de Cataluña)
Alcober, Jesús (Universidad Politécnica de Cataluña)
Álvarez Campana, Manuel (Universidad Politécnica de Madrid)
Amor Pinilla, Mercedes (Universidad de Málaga)
Arco, José (Universidad de Alcalá)
Asensio Pérez, Juan (Universidad de Valladolid)
Astorga, Jasone (Universidad del País Vasco)
Ayala Viñas, Inmaculada (Universidad de Málaga)
Aznar, José (Universidad de Zaragoza)
Barba, Antonio (Universidad Politécnica de Cataluña)
Barceló, Jaume (Universidad Carlos III de Madrid)
Bauza, Ramón (Universidad Miguel Hernández de Elche)
Bellalta, Boris (Universidad Pompeu Fabra)
Bernal Mor, Elena (Universidad Politécnica de Valencia)
Bote Lorenzo, Miguel (Universidad de Valladolid)
Cabrero, Sergio (Universidad de Oviedo)
Cacheda Seijo, Fidel (Universidad de La Coruña)
Camacho Camacho, José Manuel (Universidad Carlos III de Madrid)
Cano, Cristina (Universidad Pompeu Fabra)
Cano, María Dolores (Universidad Politécnica de Cartagena)
Carmona Murillo, Javier (Universidad de Extremadura)
Carral, Juan (Universidad de Alcalá)
Coll Perales, Baldomero (Universidad Miguel Hernández de Elche)
Corral, Guiomar (Universidad Ramón Llull)
Cortés Polo, David (Universidad de Extremadura)
De la Hoz, Enrique (Universidad de Alcalá)
De la Oliva, Antonio (Universidad Carlos III de Madrid)
Díaz Zayas, Almudena (Universidad de Málaga)
Díaz Verdejo, Jesús (Universidad de Granada)
Díez, David (Universidad Carlos III de Madrid)
Dimitriadis, Ioannis (Universidad de Valladolid)
Draper Gil, Gerard (Universidad de las Islas Baleares)
Egea López, Esteban (Universidad Politécnica de Cartagena)
Estepa, Antonio (Universidad de Sevilla)
Estepa, Rafael (Universidad de Sevilla)
Felici Castell, Santiago (Universidad de Valencia)
Fernández, David (Universidad Politécnica de Madrid)
Fernández Fernández, Gregorio (Universidad Politécnica de Madrid)
Fernández Iglesias, Diego (Universidad de La Coruña)

Fernández Veiga, Manuel (Universidade de Vigo)
Fernández García, Norberto (Universidad Carlos III de Madrid)
Fernández Navajas, Julián (Universidad de Zaragoza)
Formoso López, Vreixo (Universidad de La Coruña)
Freire Veiga, Ana (Universidad de La Coruña)
Galán Jiménez, Jaime (Universidad de Extremadura)
Gállego, José Ramón (Universidad de Zaragoza)
Gálvez, Juan (Universidad de Murcia)
García, Alberto (Universidad de Cantabria)
García, Ana (Universidad Politécnica de Madrid)
García, José (Universidad de Zaragoza)
García, Marta (Universidad de Cantabria)
García, Roberto (Universidad de Oviedo)
García Martín, Ricardo (Universidad de Valladolid)
García Pañeda, Xabiel (Universidad de Oviedo)
García Dorado, José Luis (Universidad Autónoma de Madrid)
García Galán, Sebastián (Universidad de Jaén)
García Manrubia, Belén (Universidad Politécnica de Cartagena)
García Saavedra, Andrés (Universidad Carlos III de Madrid)
García Sánchez, Felipe (Universidad Politécnica de Cartagena)
Gazo Cervero, Alfonso (Universidad de Extremadura)
Gómez Oliva, Ana (Universidad Politécnica de Madrid)
Gómez Sánchez, Eduardo (Universidad de Valladolid)
Gómez Skarmeta, Antonio (Universidad de Murcia)
González, Roberto (Universidad Carlos III de Madrid)
González Barahona, Jesús (Universidad Rey Juan Carlos)
González Sánchez, José Luis (Universidad de Extremadura)
Gorricho, Juan Luis (Universidad Politécnica de Cataluña)
Gozálvez, Javier (Universidad Miguel Hernández de Elche)
Guerri, Juan Carlos (Universidad Politécnica de Valencia)
Hackbarth, Klaus (Universidad de Cantabria)
Hernández, José Alberto (Universidad Carlos III de Madrid)
Hesselbach Serra, Xavier (Universidad Politécnica de Cataluña)
Higuero, María Victoria (Universidad del País Vasco)
Hinarejos, María Francisca (Universidad de las Islas Baleares)
Irastorza, José Angel (Universidad de Cantabria)
Isern Deyà, Andreu Pere (Universidad de las Islas Baleares)
Izal, Mikel (Universidad Pública de Navarra)
Jacob, Eduardo (Universidad del País Vasco)
Liberal, Fidel (Universidad del País Vasco)
Lobo, Ana (Universidad de Oviedo)
López, Víctor (Universidad Autónoma de Madrid)

López Fernández, Luis (Universidad Rey Juan Carlos)
López Mato, Javier (Universidad de La Coruña)
López Millán, Gabriel (Universidad de Murcia)
López Ardao, José Carlos (Universidad de Vigo)
Lucas Estañ, María del Carmen (Universidad Miguel Hernández de Elche)
Maciá Fernández, Gabriel (Universidad de Granada)
Macías López, Elsa (Universidad de Las Palmas de Gran Canaria)
Madinabeitia, Germán (Universidad de Sevilla)
Magaña, Eduardo (Universidad Pública de Navarra)
Manzanares López, Pilar (Universidad Politécnica de Cartagena)
Marrero Marrero, Domingo (Universidad de Las Palmas de Gran Canaria)
Marsá Maestre, Iván (Universidad de Alcalá)
Martínez, Jesús (Universidad de Málaga)
Martínez Bauset, Jorge (Universidad Politécnica de Valencia)
Mata, Felipe (Universidad Autónoma de Madrid)
Mataix, Jorge (Universidad Politécnica de Cataluña)
Melendi, David (Universidad de Oviedo)
Montoto Castelao, Paula (Universidad de La Coruña)
Morato, Daniel (Universidad Pública de Navarra)
Moreno, Ángel (Universidad de Alcalá)
Muñoz, José Luis (Universidad Politécnica de Cataluña)
Muñoz Expósito, José Enrique (Universidad de Jaén)
Navarro Ortiz, Jorge (Universidad de Granada)
Pacheco Paramo, Diego (Universidad Politécnica de Valencia)
Pau, Iván (Universidad Politécnica de Madrid)
Pla, Vicent (Universidad Politécnica de Valencia)
Prado, Rocío (Universidad de Jaén)
Prieto González, Lisardo (Universidad Carlos III de Madrid)
Ramis, Jaume (Universidad de las Islas Baleares)
Ramos, Javier (Universidad Autónoma de Madrid)
Ramos Muñoz, Juan (Universidad de Granada)
Regueras, Luisa (Universidad de Valladolid)
Riera Palou, Felip (Universidad de las Islas Baleares)
Rincón, David (Universidad Politécnica de Cataluña)
Rodríguez Pérez, Francisco Javier (Universidad de Extremadura)
Rodríguez Rubio, Raúl (Universidad de Vigo)
Román, Isabel (Universidad de Sevilla)
Rondinone, Michele (Universidad Miguel Hernández de Elche)
Ros, Francisco (Universidad de Murcia)
Salcedo Campos, Francisco Javier (Universidad de Granada)
Saldaña, José (Universidad de Zaragoza)
Salmerón, Alberto (Universidad de Málaga)

Salvachua, Joaquín (Universidad Politécnica de Madrid)
Sánchez, Luis (Universidad de Cantabria)
Sánchez, Luis (Universidad Carlos III de Madrid)
Sánchez García, Sergio (Universidad Politécnica de Madrid)
Sánchez Aarnoutse, Juan Carlos (Universidad Politécnica de Cartagena)
Santa, José (Universidad de Murcia)
Santiago del Río, Pedro María (Universidad Autónoma de Madrid)
Sanz, Roberto (Universidad de Cantabria)
Sepulcre, Miguel (Universidad Miguel Hernández de Elche)
Sfairopoulou, Anna (Universidad Pompeu Fabra)
Simmross, Federico (Universidad de Valladolid)
Suárez González, Andrés (Universidad de Vigo)
Suárez Sarmiento, Álvaro (Universidad de Las Palmas de Gran Canaria)
Toledo, Nerea (Universidad del País Vasco)
Tomás Gabarrón, Juan Bautista (Universidad Politécnica de Cartagena)
Urbano Fullana, Antonio (Universidad de las Islas Baleares)
Urueña, Manuel (Universidad Carlos III de Madrid)
Vázquez, Enrique (Universidad Politécnica de Madrid)
Vega Gorgojo, Guillermo (Universidad de Valladolid)
Verdú Pérez, María Jesús (Universidad de Valladolid)
Vidal, José (Universidad Politécnica de Valencia)
Villagra, Víctor (Universidad Politécnica de Madrid)
Vozmediano, Juan (Universidad de Sevilla)

Índice

Sesión 1.A: Aplicaciones de redes y servicios telemáticos

Sesión 1.B: Servicios multimedia y web semántica

Sesión 2.A: Criptografía y seguridad en redes

Sesión 2.B: Gestión y optimización en entornos inalámbricos heterogéneos

Sesión 3.A: Modelado y análisis de prestaciones (I)

Sesión 4.A: Protocolos y técnicas para redes no convencionales: MANET, VANET, WSN

Sesión 5.A: Aplicaciones distribuidas y P2P

Sesión 5.B: Modelado y análisis de prestaciones (II)

Sesión de posters

Sesión 1.A
Aplicaciones de redes y servicios telemáticos

Solución estándar y abierta para interoperabilidad de dispositivos médicos personales X73PHD sobre perfil médico Bluetooth HDP

A. Aragüés, J. Escayola, I. Martínez, P. del Valle, P. Muñoz, J.D. Trigo and J. García.
Aragon Institute for Engineering Research (I3A) - Univ. Zaragoza (UZ). c/María de Luna 3, 50018 Zaragoza
{aaragues, javier.escayola, imr, pdelvalle, pmg, jtrigo, jogarmo}@unizar.es

***Resumen*- Este artículo propone una solución estándar y abierta, basada en *software* libre (*open source*) sobre sistema operativo *Linux* para interoperabilidad de dispositivos médicos personales ISO/IEEE11073 *Personal Health Devices* (X73PHD) sobre perfil médico Bluetooth *Health Device Profile* (BT HDP). La solución se ha implementado sobre una arquitectura de capas de abstracción e integrado en un servidor de Historia Clínica Electrónica (HCE) conforme norma internacional UNE-EN/ISO 13606. La arquitectura integra servicios en la nube y, con la incorporación de los paradigmas de e-Accesibilidad y usabilidad, constituye una propuesta completamente estándar para garantizar interoperabilidad de sistemas extremo a extremo.**

***Palabras Clave*- arquitectura de capas de abstracción, Bluetooth *Health Device Profile*, interoperabilidad, ISO/IEEE11073, *Personal Health Devices*, servicios en la nube.**

I. Introducción. Interoperabilidad y estandarización de dispositivos médicos

La evolución más reciente de las nuevas tecnologías y las herramientas de la Sociedad de la Información ha transformado completamente el concepto de e-Salud. La Ingeniería Telemática (IT) juega aquí un papel imprescindible para abordar el reto de la integración de diferentes protocolos de comunicación en soluciones interoperables basadas en estándares. En este contexto, diversas tendencias tecnológicas están irrumpiendo con fuerza en el convulso ecosistema de los nuevos dispositivos (*Tablet* PCs, *Netbooks*, *Smartphones*) donde la aplicación de una norma que defina el intercambio de información entre dispositivos médicos se torna fundamental en un sector tan heterogéneo como la e-Salud. El estándar internacional para este propósito es ISO/IEEE 11073 *Personal Health Devices* (X73PHD) [1] y su implantación comercial viene siendo liderada por diversas iniciativas como *Continua Health Alliance* o *Integrating the Healthcare Enterprise* (IHE). Estas iniciativas buscan la integración de X73PHD en los servicios sanitarios y la certificación de dispositivos X73PHD sobre los perfiles médicos adoptados para las tecnologías de transporte desde el grupo especial de trabajo *Personal Health Devices Working Group* (PHDWG) [2] para X73PHD: USB *Personal Health Device Class* (USB PHDC), Bluetooth *Health Device Profile* (BT HDP) [3], y ZigBee *Health Care* (ZHC).

Este es el primer paso para llegar a una plataforma totalmente interoperable pero, para lograr niveles óptimos en la calidad asistencial y continuidad de cuidado de un paciente, es necesario interactuar con la Historia Clínica Electrónica (HCE) del paciente para el seguimiento y autocontrol de su salud. El estándar internacional para lograr interoperabilidad de HCE entre sistemas sanitarios es UNE-EN/ISO 13606 [4]. Sin embargo, la existencia de normas médicas no garantiza la correcta implementación de una solución homogénea de e-Salud personal dado que la integración de las diferentes normas en soluciones extremo a extremo todavía sigue siendo una tarea compleja e intrincada. Así, y a partir de desarrollos anteriormente publicados [5]-[7], se presenta en este artículo una propuesta de arquitectura abierta y estándar (denominada uz.health) para telemonitorización de pacientes integrando las normas de interoperabilidad ISO/IEEE 11073 y UNE-EN/ISO 13606 sobre sistema operativo Linux (ver Fig. 1). Esta solución se centra en entornos de e-Salud personal y utiliza la más reciente versión X73PHD. Posibilita una comunicación agente-manager entre los dispositivos médicos y un dispositivo *Tablet* PC, *Netbook*, etc. a través del interfaz de red personal (*Personal Area Networks*, PAN) y diferentes tecnologías inalámbricas (particularizando en este trabajo para Bluetooth, que hoy en día es la tecnología inalámbrica más extendida en los dispositivos de salud personal [8]-[9] y por lo tanto la candidata principal para desarrollar soluciones de e-Salud ubicuas). Se homogeneíza la comunicación *Wide Area Network* (WAN) con un servidor de HCE, mediante tecnologías *Web Services* (WS) y formato *eXchange Markup Language* (XML), que garantiza intercambio interoperable de extractos de HCE conforme a UNE-EN/ISO 13606.

En la Sección II se plantean las reglas en las que se sustenta el diseño de la solución, proponiendo una arquitectura de capas de abstracción y detallando el núcleo central conforme al estándar X73PHD. En la Sección III se describe la implementación de la arquitectura propuesta garantizando las especificaciones de X73PHD. En la Sección IV se analiza la integración con el nuevo perfil médico BT HDP como tecnología de transporte recomendada por X73PHD. En la Sección V se discute la integración con el servidor de HCE y los servicios de usuario basados en la nube. Finalmente, en la Sección VI se discuten las conclusiones y líneas futuras de trabajo.

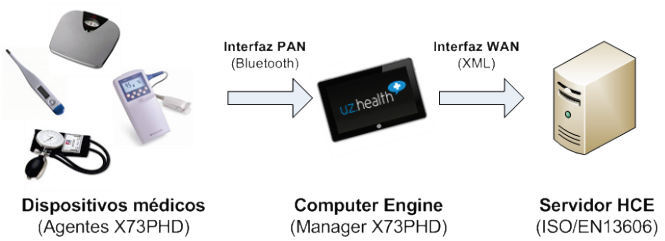


Fig. 1 Esquema de la plataforma de telemonitorización 100% estándar.

ISBN: 978-84-694-5948-5

II. Diseño de la solución basada en ISO/IEEE 11073

El primer paso en el diseño de la arquitectura es el análisis previo [5]-[7] y la detección de sus limitaciones para, como resultado, proponer las siguientes reglas de diseño:

- **Escalabilidad**. La arquitectura es flexible para extender su margen de operaciones sin perder calidad o bien manejar el crecimiento continuo de trabajo de manera fluida. Así, añadir nuevas funcionalidades a la arquitectura propuesta es un proceso simple de modificación del código.
- **Modularidad**. La arquitectura se compone de la unión de varias partes que interactúan entre sí por un objetivo común, realizando las tareas necesarias para su consecución. El código propuesto separa módulos fundamentales como, por ejemplo, las partes que realizan tareas de comunicación del procesado de la información de la parte gráfica.
- **Portabilidad**. El *software* para ejecutarse en diferentes plataformas ya que la solución es compatible con diferentes entornos sin una única pareja arquitectura-sistema operativo.
- **Mantenibilidad**. La arquitectura presenta, a nivel *software*, facilidad para ser modificada, corregir fallos, mejorar su funcionamiento u otros atributos, adaptarse a cambios en el entorno, etc. sin tener que rehacer prácticamente el código.
- **Robustez**. La arquitectura propuesta, además de realizar el trabajo esperado, está preparada para adaptarse a múltiples cambios en el flujo de trabajo y reaccionar apropiadamente ante condiciones excepcionales.
- **e-Accesibilidad y usabilidad**. La arquitectura evoluciona respecto a las propuestas previas ya que se soporta sobre *software* capaz de ser comprendido, usado y atractivo para el usuario. Además, incorpora aplicaciones específicas para que toda persona pueda usar el servicio, independientemente de su capacidad técnica, cognitiva o física.

Con todas las premisas anteriores, se propone un diseño basado en capas de abstracción sobre lenguaje de programación C# y plataforma de desarrollo Mono (la implementación libre de C#) como entorno multiplataforma (Windows/Linux/Mac Os). Esta arquitectura ofrece la posibilidad de crear una o diferentes interfaces de usuario totalmente independientes del núcleo e incluso del lenguaje de programación (con alguna capa de adaptación). Esto permite dotar al núcleo de capacidades de actualización automática (sin necesidad de modificar el resto de capas) o incluso proporcionar conexión y desconexión de diferentes tecnologías de transporte sin tener que modificar el núcleo principal. Este trabajo, como se ha comentado, se soporta sobre un núcleo conforme al estándar internacional 11073-20601 para interoperabilidad de dispositivos médicos, pero su diseño es compatible con otros protocolos de comunicación como HL7. En Fig. 2 se observa el diseño basado en capas de abstracción, que incluye tres capas principales:

- **Capa tecnológica**, donde se integrarán las tecnologías de transporte recomendadas desde el grupo especial de trabajo PHDWG para X73PHD: USB PHDC, BT HDP y ZHC. Esta capa presenta un interfaz homogéneo de comunicación hacia la capa superior a través de canales virtuales.
- **Capa de comunicaciones**, que presenta los diferentes protocolos de comunicaciones; en este caso, X73PHD, pero también, en un futuro, HL7 o protocolos propietarios.
- **Capa de usuario**, que incluye un interfaz gráfico (web o de escritorio), incluso con servicios de valor añadido.

En este tipo de arquitectura es crítico el diseño de la capa central, pues es el bloque que recibe los datos enviados por uno o más sistemas agente y gestiona el proceso de comunicación para asegurar que el intercambio de datos sea de acuerdo al protocolo estándar 11073-20601. Todos los protocolos definidos por X73PHD utilizan sintaxis abstracta (*Abstract Syntax Notation One*, ASN.1) para facilitar la especificación de las estructuras de datos sin hacer referencia a una tecnología de implementación concreta. ASN.1, junto con unas reglas de codificación específicas, facilitan el intercambio de estructuras de datos describiendo esas estructuras de forma independiente de la arquitectura de la máquina y el lenguaje de implementación. 11073-20601 utiliza reglas de codificación optimizadas para ASN.1 denominadas *Medical Devices Encoding Rules* (MDER). MDER fue elaborada en la norma 11073-20101 para minimizar el ancho de banda utilizado en las transferencias de datos médicos personales entre agentes y managers. La ausencia de implementaciones *open source* de estas reglas de codificación motivó la utilización y modificación de *Binary Notes*, un *framework* ASN.1 para C# y Java. Este *framework* presentaba las reglas de codificación más comunes (*Basic/Packet/Data Encoding Rules*, BER, PER o DER). Sin embargo, MDER no era soportado y tuvo que implementarse un nuevo codificador y decodificador para reglas entre dispositivos médicos. MDER es obligatorio tanto para agente como manager aunque los dispositivos pueden opcionalmente establecer otras reglas de codificación como *eXchange Encoding Rules* (XER) o PER. Este bloque central utiliza las librerías de *Binary Notes* para codificar las tramas de datos (*Applications Protocol Data Unit*, APDUs), utilizando las reglas de codificación anteriormente descritas además de las reglas MDER implementadas para esta arquitectura. Fig. 2 ilustra los componentes principales de la implementación:

- ***Domain Information Model* (DIM)**. Constituye el núcleo principal de la arquitectura y define un conjunto de clases para modelar agentes según objetos que pueden contener fuentes de datos (señales biológicas, eventos, informes de alertas, etc.), y los métodos que un manager puede usar para controlar el comportamiento de un sistema agente.
- ***Service Model***. Define el mecanismo de los servicios de intercambio de datos entre manager y agente que están mapeados en mensajes codificados usando ASN.1.
- ***Communication Model***. Se define a través de una máquina de estados finitos (*Finite State Machine*, FSM) que sincroniza los mensajes intercambiados entre manager y agente en las conexiones punto a punto.

Como se ha comentado, para mantener la independencia con la capa de transporte se ha diseñado un modelo abstracto de comunicación basado en canales virtuales. Un canal virtual puede gestionar varios canales simultáneos al mismo tiempo en cada conexión agente-manager. Además, cada uno de esos canales dentro de un canal virtual puede ser de una tecnología o perfil diferente, como Bluetooth *Serial Port*

Profile (BT SPP) o BT HDP. De esta forma, se facilita un interfaz común de comunicación al manager para enviar y recibir APDUs desde o hacia el sistema agente. Para facilitar las tareas de las capas superiores, este núcleo central publica notificaciones sobre eventos internos recibidos desde los agentes, además de facilitar un interfaz para controlar el estado de los agentes conectados al manager. Las aplicaciones que usen este núcleo central pueden suscribirse a eventos de la forma tradicional de C# pudiendo obtener notificaciones sobre eventos genéricos del manager como conexiones o desconexiones de un nuevo agente. Además, este diseño permite ampliar la arquitectura mediante *plug-ins*, que son módulos que se relacionan con una aplicación para aportarle una función nueva y generalmente muy específica. Estos complementos permiten que desarrolladores externos colaboren con la aplicación principal extendiendo sus funciones, reduciendo el tamaño de la aplicación y separando el código fuente de la aplicación por incompatibilidad de las licencias de *software*. Para ello, se ha definido una *Application Programming Interface* (API) que permita a terceros crear complementos que interactúen con el bloque principal propuesto.

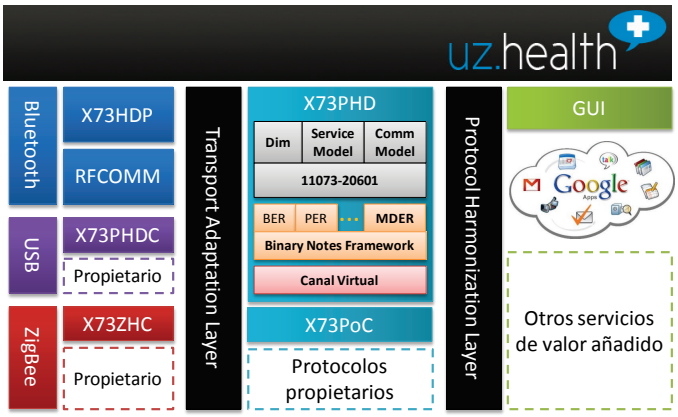


Fig.2. Propuesta de arquitectura de la solución basada en el estándar X73PHD según diseño basado en capas de abstracción.

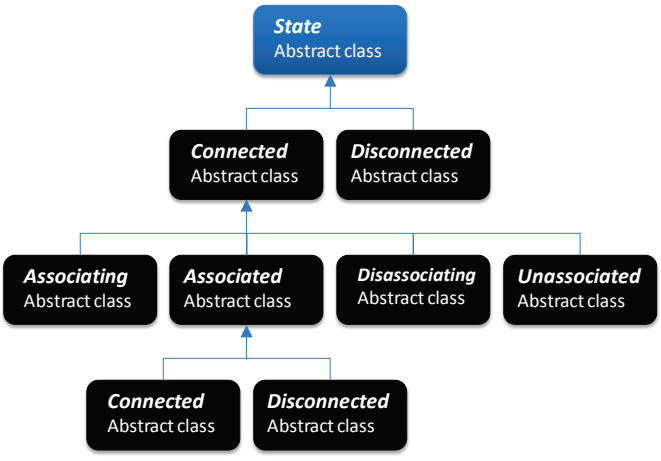


Fig. 3. Propuesta de implementación de la máquina de estados FSM X73PHD.

III. Implementación de la Arquitectura Propuesta

La implementación de la arquitectura está pensada como un cubo de piezas donde cada usuario pueda construir un manager o un agente pieza por pieza de forma modular. Todas estas piezas se pueden obtener de la librería principal (denominada `UZ_ieee_11073`). Esta librería, desarrollada en C#, contiene todos los objetos necesarios para crear un manager X73PHD, depende únicamente de la librería *Binary Notes*, y es compatible con *framework* .NET 2.0 o superiores. Además de lo anteriormente comentado, *Binary Notes* proporciona un compilador ASN.1 (BNCompiler) que construye clases Java o C# a partir de un fichero de especificaciones ASN.1 de entrada. Sin embargo, ya que *Binary Notes* no implementa MDER, algunas clases de la librería se han modificado y creado otras conforme a la norma 11073-20101 (tipos de datos, atributos, objetos, etc.). Con casi 4000 líneas de código, la librería `UZ_ieee_11073.dll` contiene el grueso de implementación de la arquitectura propuesta. El proyecto se ha organizado intentando separar las partes en las que se divide el estándar y todas aquellas clases de apoyo que tienen funcionalidades similares. Esta separación funcional permitirá, en un futuro, modularizar todavía más la solución, moviendo cada una de las partes a librerías independientes y que puedan ser reutilizables por otras aplicaciones o códigos. A continuación se describe brevemente el contenido de cada uno de los espacios de nombres (*namespaces)* que conforman la librería:

- **Config**. Contiene las clases que gestionan la configuración de un dispositivo, como tipo de codificación, versión de protocolo utilizada, etc.
- **Core**. Contiene los objetos a más alto nivel de la librería. Los objetos agente y manager se construyen a partir de canales, objetos *Medical Device System* (MDS), etc.
- **Events**. Contiene las clases que se encargan del sistema de eventos de la implementación ya que el núcleo X73PHD publica eventos para que otras clases u objetos los reciban (nuevas medidas, conexión y desconexión de dispositivos).
- **Exceptions**. Contiene los tipos de excepciones utilizadas explícitamente para esta implementación ya que no sólo pueden generarse eventos sino también excepciones de muchos tipos durante la ejecución del núcleo.
- **Gateway**. Contiene capas de adaptación para comunicar el núcleo X73PHD con el interfaz exterior.
- **Logging**. Contiene la salida de un sistema de *log* de errores robusto, flexible y multi-hilo que genera múltiples tipos de mensajes, ya sea por consola, mensajes remotos, etc.
- **Part_10101**. Contiene, en una única clase, toda la nomenclatura del estándar X73PHD.
- **Part_104xx**. Contiene todas las especializaciones de dispositivos médicos en clases independientes (definidas en la norma 11073-104xx) aprobadas por X73PHD a fecha de redacción de este artículo.
- **Part_20601**. Contiene el grueso de la implementación, la maquina de estados FSM (ver Fig. 3), los canales virtuales, las tramas de datos APDUs, etc.
- **Utils**. Contiene clases de apoyo al resto de clases para realizar operaciones determinadas u objetos que no tienen una clasificación especial.

A continuación se detallan más en profundidad los principales *namespaces* de la librería `UZ_ieee_11073.dll`.

- **Part_10101**: En este *namespace* se definen los códigos de nomenclatura que ofrecen la posibilidad de identificar cláramente objetos y atributos en relación a su código entero *Object Identifier* (OID). Esta nomenclatura está dividida en particiones para organizar los códigos (que se definen mediante constantes estáticas que representan los atributos) dependiendo de su contenido y función.
- **Part_20601**: En este *namespace* se encuentran todos los objetos que definen el comportamiento del núcleo de la implementación según el protocolo 11073-20601. Este *namespace* se compone de otros cuatro *sub-namespaces* que engloban clases homogéneas:
  - **ASN1**. En este *namespace* aparecen los objetos ASN.1 generados mediante el compilador BNCompiler.
  - **FSM**. En este *namespace* se incluyen las clases relacionadas con la máquina de estados FSM que define el modelo de comunicación de ISO/IEEE 11073. En un primer nivel, se definen los interfaces y clases abstractas necesarias para la implementación de la máquina de estados. La implementación de los estados será diferente si el dispositivo es manager o agente. Se muestra en Fig. 3 la implementación de los diferentes estados de FSM, inspirada en el diseño del proyecto Morfeo OpenHealth [10]. Todos los estados derivan de una clase abstracta base denominada *State*. Todo estado tiene un manejador de estados (*IStateHandler*), un nombre identificador del estado y dos métodos (*Process*, para procesar las APDU provenientes de otro dispositivo, y *ProcessEvent*, para procesar diferentes eventos que puedan producirse como desconexión de otro dispositivo, problemas en la red, etc). El interface *IStateHandler* es un elemento clave para manejar los estados y permite enviar la información hacia el interfaz exterior desde cada uno de los estados. Todas las clases de la máquina de estados son clases abstractas por lo que no se pueden instanciar. A modo de ejemplo, se analiza la implementación del estado *Disconnected*. En este estado, el manager X73PHD no procesa ninguna APDU, únicamente los eventos de conexión o desconexión. Así en el método *ProcessEvent*, cuando llega un evento de conexión (*Event.Connection*), se modifica la propiedad *State* del *IStateHandler* indicándole el siguiente estado de FSM: *Unassociated*. En este estado *Unassociated* se procesan tanto eventos como APDUs. Así, si desde *Unassociated* se recibe un evento de desconexión, se mueve *IStateHandler* a estado *Disconnected*. En el caso del procesado de la APDU, si se recibe una confirmación (AARQ), se inicializa el proceso de asociación y, en caso contrario, se envía una APDU de abort por causas no definidas (*AbrtApdeuUndefined*) tal como se define en la norma 11073-20601.
  - **Messages**. En este *namespace* se construyen las APDUs que se utilizan para el intercambio de información en 11073-20601. Todas estas APDUs derivan de una clase base *ApduType* a la que se añade información. Algunos ejemplos son: *AbrtApdu* (APDU para abortar conexión en la que se indica la razón del aborto de conexión y se construye una APDU específica para esta situación), *AbrtApduBufferOverFlow* (APDU para abortar la conexión por desbordamiento de *buffer*, como caso específico heredado del anterior), o *AareRejectApdu* (clase compleja que indica el rechazo a la asociación).
  - **PHD**. En este *namespace* se definen las diferentes clases que conforman el DIM y que caracterizan un dispositivo médico. El DIM caracteriza la información de un agente como un conjunto de objetos. Cada objeto tiene uno o más atributos. Los atributos describen medidas que son comunicadas a un manager, así como elementos que controlan el comportamiento e información del estado del agente. A modo de ejemplo, destaca la clase base *metric* para todos los objetos que representan medida, estado y datos de contexto. Tal como impone X73PHD, esta clase no puede ser instanciada, por ello se trata de una clase abstracta. También destaca la clase MDS, una de las más importantes de toda la librería ya que cada agente es instanciado directamente desde una clase MDS. El objeto MDS representa la identificación y el estado de un agente a través de sus atributos. Y cabe resaltar que este *namespace* contiene la implementación del canal virtual y los diferentes tipos de canales para cada una de las tecnologías de transporte (USB PHDC, BT HDP y ZHC).

Por último, se propone en Fig. 4 un ejemplo de la implementación básica de un manager X73PHD sobre BT HDP utilizando la librería implementada *ad-hoc* `UZ_ieee_11073`. Este ejemplo se ofrece para comprobar la modularidad de la solución propuesta, que permite construir un nuevo manager de forma ágil y rápida, pudiendo añadir nuevos canales a la librería o incluso nuevos protocolos de comunicaciones sin afectar al resto del código.

IV. Integración con Bluetooth HDP

Como se ha comentado, la arquitectura propuesta permite la integración de los nuevos perfiles médicos recomendados por PHDWG para las tecnologías de transporte soportadas sobre X73PHD. A partir de las experiencias anteriores que se soportaban sobre TCP/IP o BT SPP, en esta sección se detalla la integración de BT HDP lo que constituye una solución X73PHD estándar y facilita un *framework* robusto y basado en estándares. Esto permite la interoperabilidad completa entre dispositivos de e-Salud sobre Bluetooth y que ya ha sido implementado en algunos dispositivos médicos del mercado [11]. A fecha de redacción de este artículo, únicamente algunas pilas Bluetooth comerciales presentan compatibilidad con el nuevo perfil médico BT HDP: *Jungo BTware* [12], diseñada para sistemas empotrados de bajo consumo; *Stollmann BlueCode+* [13], diseñada con una arquitectura independiente de la plataforma; *Toshiba* Bluetooth *Stack* [14], diseñada para Windows y certificada por *Continua Health Alliance*; y *Ethermind* Bluetooth *Stack* [15], desarrollada por la empresa *Mindtree* para sistemas empotrados y otras arquitecturas. Todas estas propuestas se apartan del camino de la interoperabilidad por ser soluciones comerciales, cerradas, que no permiten evolucionar con nuevas funcionalidades alejándose de los paradigmas de diseño planteados. Sin embargo, en 2009, desde la Universidad Rey Juan Carlos I, se inició un camino para integrar el perfil médico BT HDP en la pila oficial BlueZ de Linux [16] ofreciendo, así, las nuevas funcionalidades HDP a plataformas como Linux, Android o Meego, todas ellas apoyadas en el *kernel* de Linux.

```
using System;
using System.Collections.Generic;
using UZ_ieee_11073.Core;
using UZ_ieee_11073.Events;
using UZ_ieee_11073.Part_20601.PHD.Channel.HDP;

namespace UZ_ieee_11073_Manager
{
    public class Manager : IEventListener
    {
        private HDPManagerChannel _hdpManagerChannel;
        private List<Agent> _agentsList;

        public Manager() {
           _agentsList = new List<Agent>();
           InitializeChannels();
        }

        private void InitializeChannels() {
           _hdpManagerChannel = new HDPManagerChannel();
           _hdpManagerChannel.AgentAttached +=
AgentAttached;
        }

        void AgentAttached(Agent agent) {
           agent.EventManager.AddEventListener(this);
        }

        public void Start() {
           _hdpManagerChannel.Start();
        }

        public void Stop() {
           Log.Debug("Stopping Manager");
           _hdpManagerChannel.Finish();
           foreach (Agent agent in _agentsList)
           {agent.SendEvent(new
Event(Event.AssociationAbortRequest));
           gent.FreeResources();
           }
           _agentsList.Clear();
        }

        #region IEventListener implementation
        public void StateChanged(Agent agent,
StateChange stateChange) {
           Log.Debug("Agent " + agent.SystemId + "status
has changed. " + stateChange);
        }
        public void NewMeasure(Agent agent,
List<Measure> measure) {
           Debug.Log("The agent " + agent.SystemId +
"has sent new measures:");
           foreach(Measure ms in measure)
                Debug.Log(ms.ToString());
        }
        public void AgentDisconnected(Agent agent) {
           agent.FreeResources();
           _agentsList.Remove(agent);
        }
        public string Name {
           get { return "UZ_ieee11073_Manager"; }
        }
        public void AgentConnected(Agent agent) {
           if (_agentsList.Contains(agent))
           {
               Debug.Log("Agent already connected");
               return;
           }
           _agentsList.Add(agent);
        }
        #endregion
    }
}
```

Fig. 4. Ejemplo de implementación básica de un manager X73PHD.

Todas estas propuestas se apartan del camino de la interoperabilidad por ser soluciones comerciales, cerradas, que no permiten evolucionar el desarrollo con nuevas funcionalidades alejándose de los paradigmas de diseño planteados. Sin embargo, en 2009, desde la Universidad Rey Juan Carlos I, se inició un camino para integrar el perfil médico BT HDP en la pila oficial BlueZ de Linux [16] ofreciendo, así, las nuevas funcionalidades HDP a plataformas como Linux, Android o Meego, todas ellas apoyadas en el *kernel* de Linux. En los primeros pasos de esta iniciativa, la pila BlueZ no contemplaba algunos de los bloques y protocolos necesarios y específicos del perfil médico BT HDP. En Fig. 5 se muestra un diagrama de bloques de la arquitectura BlueZ, mostrando en azul aquellos bloques que no estaban previamente implementados o que necesitaron reescribir parte de su código. Estos bloques específicos del perfil médico se detallan a continuación:

- ***Multichannel Adaptation Protocol* (MCAP)**. Es un protocolo basado en *Logical Link Control and Adaptation Protocol* (L2CAP) que facilita un mecanismo sencillo para manejar múltiples canales de datos. Este protocolo soporta diferentes configuraciones de canal dependiendo de la aplicación o las necesidades de la transmisión como por ejemplo los modos *reliable* o *streaming*. La implementación de MCAP se realizó totalmente compatible con la especificación de Bluetooth *Special Interest Group* (SIG) [17], soportando todas las características obligatorias y algunas de las opcionales, como la re-conexión de canal. Como ya se ha comentado, MCAP requiere los modos de *streaming* y retransmisión mejorada de L2CAP. Estos modos no existían al comienzo del desarrollo del perfil BT HDP y se han ido construyendo paralelamente por los miembros de BlueZ.
- ***Bluetooth Health Device Protocol* (HDP)**. El perfil BT HDP permite a los dispositivos agente (conocidos en Bluetooth como *source* y asociados a los dispositivos médicos) intercambiar datos con los dispositivos manager (conocidos en Bluetooth como *sink* y asociados a móviles, portátiles, *Smartphones*, *Tablet* PCs, etc.). HDP es un perfil que simplifica el uso de MCAP y es el encargado de anunciar a otros dispositivos las características soportadas así como los canales usados por cada perfil. Para estos anuncios hace uso del *Service Discovery Application Profile* (SDP). De esta forma, los dispositivos pueden encontrar al manager y conectar con él o viceversa: el manager puede iniciar una conexión a un dispositivo. La implementación de BT HDP simplifica el uso de MCAP al usuario y registra toda la información necesaria en el registro SDP. Aunque en la primera versión de la implementación se usaban *callbacks* para notificar eventos al usuario, la actual se ha re-escrito para adaptarse a las guías de estilo de BlueZ utilizando el servicio de mensajes de sistema DBus [18], como se detallará más adelante. Esta implementación esconde por completo el mecanismo de control de los canales, facilitando un modelo de abstracción basado en un servicio de notificación de nuevos *MCAP Communication Links* (MCL) tales como nuevos dispositivos remotos. Además, esta implementación gestiona la creación de nuevos canales de datos y controla la correcta configuración de los mismos antes de ser notificados al usuario.

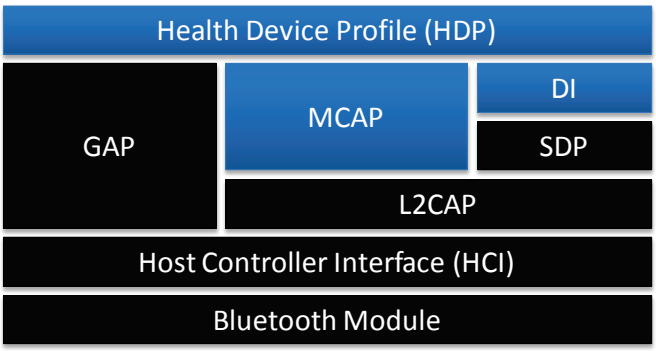


Fig.5. Pila de protocolos BlueZ.

- ***Device ID Profile* (DI)**. Especifica un método por el cual dispositivos Bluetooth comparten información que puede ser usada por otros dispositivos para encontrar imágenes o *software* asociado, como controladores específicos. Toda esta información también se publica en el registro SDP.

Como se ha adelantado, BlueZ utiliza el servicio de mensajes de sistema DBus. DBus es un protocolo de comunicación entre procesos. Ha sido diseñado para que sea ligero y fácilmente utilizado por cualquier programa. La arquitectura de DBus se compone de dos partes básicas: la librería *libdbus*, y un *daemon* que sirve como repetidor de los mensajes. La librería *libdbus* crea conexiones o canales que conectan dos aplicaciones (ver Fig. 6). Lo que hace es usar esa única conexión para conectarse al *daemon* de DBus, que se comporta como un repetidor. De esta forma, todas las aplicaciones que se conecten al *daemon* podrán contactar entre sí. La información se transmite en forma de mensajes que los hay de dos tipos: métodos (que pueden modificar el estado de un objeto o recabar información sobre el mismo) y señales (sirven para notificar un suceso de interés general). DBus es independiente del lenguaje que se use para acceder a él y hace que los objetos sean unas entidades no asociadas a ningún lenguaje concreto. Como un objeto en DBus es una ruta, dichos objetos son direccionados a través de una ruta que equivale a su nombre (un programa publica "objetos-rutas" (*ObjectPath*) a las que se puede acceder) y son equivalentes a las que se emplean en el sistema de ficheros de Linux. Cada aplicación debe tener una ruta única y no es complicado encontrar rutas como "/org/bluez/" ó "/org/freedesktop/DBus". Toda la información que circula a través de DBus puede ser depurada utilizando el comando *dbus-monitor* en una consola de texto y ver cómo se suceden las acciones. Los interfaces son conjuntos de métodos con nombres predefinidos y acciones acordadas que son conceptualmente cercanos y contienen todo lo necesario para reproducir música o buscar texto. La implementación de BT HDP en BlueZ se compone de tres interfaces DBus principales:

- **org.bluez.HealthManager**. Este es el interfaz principal a través del cual se registra una aplicación gestora de BT HDP. El método *CreateApplication* registra la nueva aplicación aceptando un objeto de tipo *Dictionary* como parámetro donde se indica el comportamiento (*sink* o *source*), el *data type*, tipo de canal (fiable o *streaming*), etc.
- **org.bluez.HealthDevice**. Representa al dispositivo médico remoto a través del cual se pueden crear o destruir canales o recibir eventos de conexión o desconexión de canal.

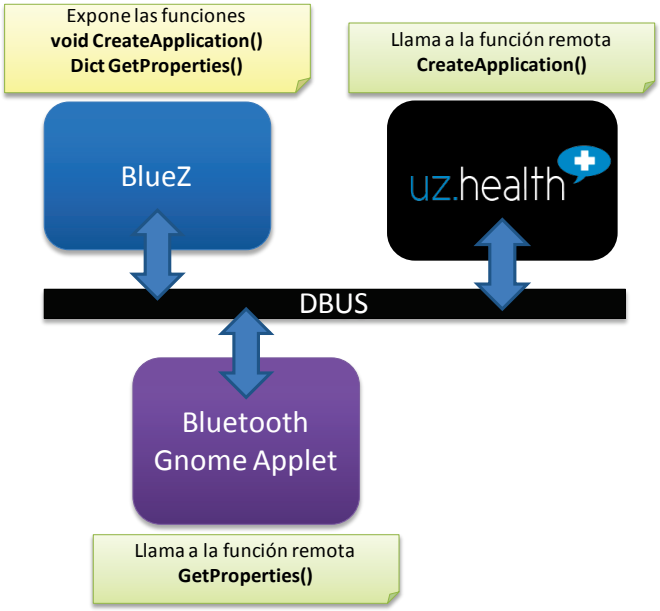


Fig. 6. Ejemplo de conexiones de DBus.

- **org.bluez.HealthChannel**. Representa un canal de datos y a través de este interfaz se obtiene el *stream* sobre el que se intercambia toda la información X73PHD.

La implementación propuesta utiliza un *wrapper* C# sobre DBus para utilizar BT HDP a través de BlueZ. El perfil médico se ha incorporado en la pila BlueZ a partir de la versión 4.71, pero es recomendable utilizar una versión más actual dado el estado semi-experimental de las primeras implementaciones. De la misma forma es necesaria, como mínimo, la versión 1.4.0 de DBus, a partir de la cual se ha incluido la característica "*Unix FD passing*" necesaria para obtener el *stream* de datos desde DBus.

V. Integración con los servicios de usuario

Siguiendo la arquitectura de abstracción de capas, en este apartado se analiza la integración de servicios de usuario. Hasta ahora se ha descrito el núcleo de la plataforma, la capa de comunicaciones y la integración del perfil médico BT HDP. Para esta última capa de la arquitectura, se propone una prueba de concepto incluyendo servicios de valor añadido realizando un acercamiento al diseño en la nube (*cloud computing*). Con este objetivo, se han integrado diversas aplicaciones (que se detallan a continuación), así como la armonización con un servidor de HCE y el desarrollo de un interfaz gráfico básico, que reúne estos servicios y ofrece un acceso intuitivo a la plataforma de telemonitorización.

- **Servicios en la nube**. Entre las nuevas tendencias tecnológicas que están aparecido en el mercado en los últimos años, *cloud computing* ofrece un modelo de pago por uso que permite un acceso cómodo y bajo demanda a un conjunto compartido de recursos informáticos configurables como redes, servidores, almacenamiento, aplicaciones y servicios, que se puede desplegar y utilizar de forma rápida y fácil. El modelo de *cloud computing* ofrece varios tipos generales de servicios: infraestructura (disponibilidad de capacidad de almacenamiento, procesamiento y de red que se factura según el consumo),

plataforma (entorno de desarrollo y herramientas y servicios asociados que se ofrece a los clientes para crear sus propias aplicaciones), y *software* (suministro de aplicaciones que se ofrece en una red y no precisa que los usuarios lo instalen en sus propios ordenadores), entre otros. Este *software como servicio* (*Software as a Service*, SaaS) es, sin duda, el más frecuente en la nube y el que se ha integrado en la arquitectura propuesta. Este modelo de distribución de *software* proporciona a los clientes el acceso a aplicaciones a través de internet, de manera que el usuario no tiene que preocuparse de su mantenimiento. Este modelo permite, para el usuario, optimizar costes y recursos y, para el suministrador (centro de salud, hospital, etc.), implementar economías de escala optimizando costes. La arquitectura propuesta integra *Google Apps* (ver Fig. 7) para ofrecer servicios de valor añadido como alertas, calendario de eventos, recordatorio de tareas (*Calendar*) o mensajería de apoyo mediante aplicaciones de correo electrónico (*Gmail*), mensajería instantánea (*Google Talk*), documentos compartidos (*Google Docs*), gestión de información médica (*Google Health*), entre otros. Esta integración se ha realizado utilizando las librerías de desarrollo para C# facilitadas por Google.

- **Servidor de HCE**. En las secciones anteriores se ha analizado X73PHD como el estándar internacional para interoperabilidad de dispositivos médicos. Este es el primer paso para llegar a una plataforma totalmente interoperable pero, para lograr niveles óptimos en la calidad asistencial y continuidad de cuidado de un paciente, es necesario interactuar con el HCE del paciente para el seguimiento y autocontrol de su salud. Así, como se adelantó en la introducción, X73PHD cubre la comunicación en el interfaz PAN y UNE-EN/ISO 13606 define el intercambio interoperable de HCE. Pero, hasta el momento, no se ha definido un estándar específico para el interfaz WAN entre el manager y el servidor de HCE. Sin embargo, se han propuesto algunas iniciativas, lideradas por *Continua Health Alliance* e IHE, basadas en los perfiles *Device to Enterprise Communication* (DEC, llamado PCD-01) [19], *Subscribe to Patient Data* (llamado PCD-02) [20], IHE *Cross-Enterprise Document Reliable Interchange* (XDR) [21] y, sobre éste último, el documento HL7 *Personal Health Monitoring* (PHM) [22]. Estas propuestas de IHE y *Continua Health Alliance* no implican la definición de nuevos estándares *ad-hoc*. Al contrario, impulsan algunos perfiles y procedimientos de otros estándares, esencialmente HL7 (IHE impulsa mensajes HL7 v2.6 y *Continua Health Alliance* hace uso del perfil HL7-PHM).

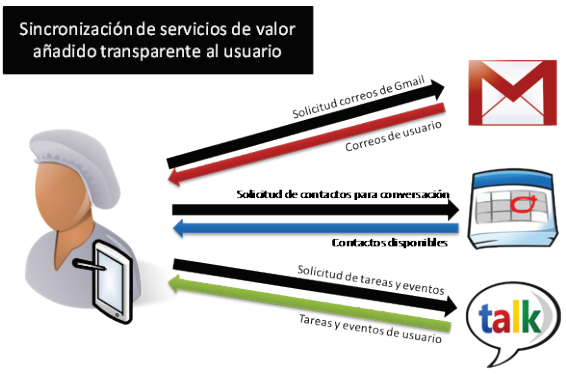


Fig. 7. Servicios en la nube integrados en la arquitectura propuesta.

En la arquitectura propuesta y debido a que la implementación está basada en X73PHD y UNE-EN/ISO 13606, estos enfoques basados en HL7 no son la opción más adecuada por lo que se ha diseñado una arquitectura WS apoyada en documentos XML a diferencia de otras alternativas [23]. Estos documentos XML satisfacen los particulares requerimientos de las implementaciones X73PHD y UNE-EN/ISO13606 para mantener los requisitos de interoperabilidad, seguridad, fiabilidad y privacidad. El contenido y estructura XML depende de un fichero de configuración, personalizado para cada usuario, obtenido del servidor de HCE, y configurado en colaboración con especialistas médicos y a partir de análisis previos de casos de uso [24]. Como muestra Fig. 8, este XML incluye información específica de los pacientes (*idCollecter*), sus dispositivos asociados (*deviceInfo*), el procedimiento de medida (*timeStamp*), y otra información técnica como tipo de dispositivo (*mdc_ attr_id_type*), modelo (*mdc_attr_id_model*), niveles de batería (*mdc_attr_val_batt_charge*), etc.

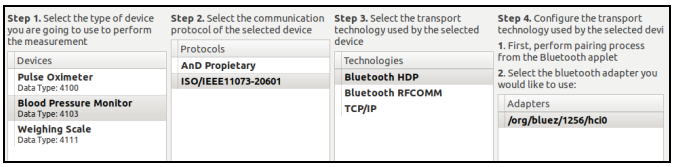
- **Aplicaciones de usuario**. La implementación se completa con el desarrollo de un interfaz como prueba de concepto (ver Fig. 9), pendiente de ser ampliado con funcionalidades de usabilidad, robustez y e-accesibilidad. Se ha construido en GTK# y Mono y su uso está orientado a Linux aunque la portabilidad a entornos Windows es inmediata. En Fig. 9(a) se observa la pantalla principal, donde se presentan las opciones más habituales para el usuario. En la parte superior izquierda se localiza la zona de notificaciones donde se avisará al usuario si tiene algún correo electrónico, alguna tarea pendiente o un mensaje de *chat*.

```
<collecter>
  <idCollecter>1898721281</idCollecter>
  <soc>
    <idPatient>PacTry001</idPatient>
    <deviceReport>
      <deviceInfo>
        <attribute>
          <id>MDC_ATTR_SYS_ID</id>
          <value>00A0960D7B58</value>
        </attribute>
        <attribute>
          <id>MDC_ATTR_SYS_TYPE</id>
          <value>MDC_DEV_SPEC_PROFILE_SCALE</value>
        </attribute>
      </deviceInfo>
      <measurementInfo>
        <timeStamp>28/02/2011 14:05:28</timeStamp>
        <measurement>
          <attribute>
            <id>MDC_ATTR_ID_TYPE</id>
            <value>MDC_MASS_BODY_ACTUAL</value>
          </attribute>
          <attribute>
            <id>MDC_ATTR_NU_VAL_OBS_SIMP</id>
            <value>89</value>
          </attribute>
          <attribute>
            <id>MDC_ATTR_UNIT_CODE</id>
            <value>KILO_G</value>
          </attribute>
        </measurement>
      </measurementInfo>
    </deviceReport>
  </soc>
</collecter>
```

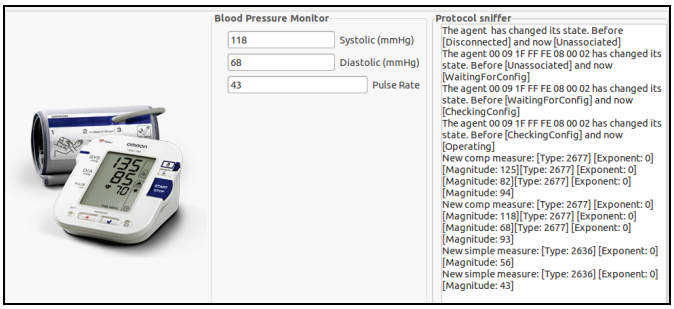
Fig. 8. Esquema XML para envío de datos X73PHD al servidor de HCE.

En la parte inferior izquierda, se localizan los botones para poder contactar con el centro de salud, consultar el HCE o realizar una medida manual que no esté planificada. Además, en la parte superior central, se han incluido una serie de pestañas bajo las que se distribuyen los servicios de usuario: tareas y eventos, correo, mensajes instantáneos y contactos. En Fig. 9(b), se presenta la pantalla de selección de parámetros para el proceso de adquisición de signos vitales desde los dispositivos médicos. Este proceso sigue cuatro pasos: seleccionar el tipo de dispositivo, seleccionar el protocolo de comunicaciones del dispositivo, seleccionar la tecnología de transporte usada por el dispositivo y, en caso de que fuera necesario, configurar los parámetros de la tecnología de transporte. En Fig. 9(b) se muestra un ejemplo de selección de un tensiómetro X73PHD desde un *Tablet* PC mediante tecnología BT HDP. Finalmente, en la última pantalla mostrada en Fig. 9(c), se observa el procedimiento completo de medida conforme al modelo de comunicación X73PHD. En la columna lateral derecha, a modo didáctico, se ha colocado un analizador de protocolo pudiendo observar los diferentes estados por los que pasa el núcleo X73PHD. Una vez completada la medida, ésta se enviará al servidor de HCE en formato el XML siguiendo el esquema detallado en el apartado anterior (ver Fig. 8).

(a) Pantalla principal de la plataforma de telemonitorización.



(b) Pantalla de selección de parámetros para adquisición de medidas.



(c) Ejemplo de medida de presión arterial sobre perfil médico BT HDP.

Fig. 9. Interfaz gráfico implementado como prueba de concepto.

VI. Conclusión

En este artículo se ha propuesto el diseño de una solución estándar y abierta para telemonitorización de pacientes, basada en ISO/IEEE 11073 e UNE-EN/ISO 13606, según una arquitectura de capas de abstracción. La implementación de la arquitectura incluye el nuevo perfil médico Bluetooth HDP y se ha integrado con un servidor de HCE y servicios de usuario basados en la nube. Esta propuesta contribuye, con una alternativa abierta e interoperable, al cerrado mercado de las soluciones comerciales. El diseño propuesto permite a desarrolladores externos trabajar juntos extendiendo las funcionalidades de la solución con nuevos servicios de valor añadido, otras tecnologías de transporte, etc.

Agradecimientos

Los autores agradecen a Santiago Carot-Nemesio del proyecto Morfeo OpenHealth (Universidad Rey Juan Carlos I) su asesoramiento técnico en esta contribución. Este trabajo ha sido parcialmente subvencionado por los proyectos TIN2008-00933/TSI del Ministerio de Ciencia e Innovación (MICINN) y Fondos Europeos para el Desarrollo Regional (FEDER), TSI-020100-2010-277 y TSI-020302-2009-7/Plan Avanza I+D del Ministerio de Industria, Turismo y Comercio, y PI029/09 del Gobierno de Aragón.

Referencias

[1] ISO/IEEE11073 - Personal Health Devices standard (X73PHD). Health Informatics. [P11073-00103. Technical report-Overview] [P11073-104zz. Device specializations] [P11073-20601. Application profile – Optimized exchange protocol]. *http://standards.ieee.org*. [04/11].

[2] Personal Health Devices Working Group (PHDWG). IEEE Standards. *http://standards.ieee.org/PHDworkgroup/*. [04/11].

[3] Bluetooth Health Device Profile (BT HDP) v1.0 rev00 Bluetooth Special interest Group (SIG). *http:// www.bluetooth.com*. [04/11].

[4] ISO/EN13606 - CEN/TC251. EHR Communication Standard. Parts 1-5. *http://www.medicaltech.org*. [04/11].

[5] I. Martínez *et al*. "Implementación integrada de plataforma telemática basada en estándares para monitorización". *JITEL*, pp. 505-512, 2007.

[6] I. Martínez *et al*. "Optimización de una plataforma telemática para monitorización de pacientes para u-Salud". *JITEL*, pp. 374-381, 2008.

[7] I. Martínez *et al*. "Plataforma Telemática de Integración de Estándares End-to-End para Salud Personal", *JITEL* pp. 156-163, 2009.

[8] M. Miyazaki, "Wireless Healthcare – Bluetooth and Beyond". *Business Briefing: Medical Device Manufacturing and Technology*: 65-67, 2005.

[9] X. Zhao et.al, "A Telemedicine System for Wireless Home Healthcare Based on Bluetooth and the Internet" *Telemed J e-Health*:10(s2), 2004.

[10] S. Carot-Nemesio *et al*. "The OpenHealth FLOSS Implementation of the X73-20601 standard", *Open Source Software Workshop* OSEHC, 2010.

[11] Productos BT HDP certificados por Continua: Tensiómetros Omrom/HEM-7081-IT y A&D/UA-767PBT-C, pulsioxímetros Nonin/Onyx-II9560 y Nonin/WristOx2-3150, báscula A&D/UC-321PBT-C, podómetro Omron/HJ-721IT y analizador de masa corporal Omron/HBF-2061T. *http://www.continuaalliance.org*. [04/11].

[12] Jungo BTware. *http://www.jungo.com*. [04/11].

[13] Stollmann BlueCode+. *http://www.stollmann.de*. [04/11].

[14] Toshiba Bluetooth. *http://aps2.toshiba-tro.de/bluetooth*. [04/11].

[15] Ethermind Stack. *http://www.mindtree.com*. [04/11].

[16] BlueZ. *http://www.bluez.org*. [04/11].

[17] Bluetooth Special Interest Group. *http://www.bluetooth.org* [04/11].

[18] DBus. *http://dbus.freedesktop.org/*. [04/11]

[19] Device Enterprise Communication (DEC) Profile PCD-01. IHE-PCD. *http://wiki.ihe.net/index.php?title=pcd_profile_dec_overview*. [04/11].

[20] Subscribe to Patient Data (SPD) Profile PCD-02. IHE-PCD Technical Committee. *http://www.ihe.net/Technical_Framework/upload/IHE_PCD_TF_Supplement_SPD_PC_2007-07-18.pdf* . [04/11].

[21] IHE-XDR. *http://wiki.ihe.net/index.php?title=Cross enterprise_Document_Reliable_Interchange*. [04/11].

[22] HL7 – PHM. *http://www.hl7.org/special/Committees/projman/searchable ProjectIndex.cfm?action=edit&ProjectNumber=209*. [04/11].

[23] A. Mense *et al*.,"Healthy interoperability: A standard based framework for integrating personal monitoring and personal health device data into medical information systems", *J Inf Techn Health*, 7(4):214-221, 2010.

[24] I. Martínez *et al*., "Recent innovative advances in telemedicine: standard-based designs for personal health", *IJBET*, 2010.

Armonización de protocolos de comunicación propietarios y estándares sobre una plataforma integrada de e-Salud para telemonitorización

J. Escayola, I. Martínez, P. del Valle, A. Aragüés, P. Muñoz, J.D. Trigo and J. García.
Aragon Institute for Engineering Research (I3A) - Univ. Zaragoza (UZ). c/María de Luna 3, 50018 Zaragoza
{javier.escayola, imr, pdelvalle, aaragues, pmg, jtrigo, jogarmo}@unizar.es

***Resumen*- Este artículo presenta el diseño e implementación de una plataforma integrada de e-Salud (*Integrated Health Platform*, IHP) para telemonitorización de pacientes que armoniza múltiples estándares de conectividad ofreciendo un sistema de gestión técnico-sanitaria orientada a entornos personales. Entre otros, integra las normas internacionales de interoperabilidad ISO/IEEE 11073 para dispositivos médicos personales e UNE-EN ISO 13606 para intercambio de Historia Clínica Electrónica (HCE). La solución propone diversos módulos para modelado, gestión, operación y actualización de dispositivos médicos. Además, dispone de un protocolo de seguridad para seguimiento remoto por parte de personal autorizado. Esta propuesta de integración de equipos propietarios y estándares garantiza interoperabilidad a todos los niveles y constituye una solución real para la problemática del sistema sanitario.**

***Palabras Clave*- interoperabilidad, modelado y actualización de dispositivos médicos, protocolos de comunicación, seguimiento de pacientes.**

I. INTRODUCCIÓN. INGENIERÍA TELEMÁTICA Y E-SALUD

El estilo de vida del ser humano ha cambiado por completo a lo largo del último siglo. La alimentación, en vez de avanzar hacia una mejor calidad en pro de la salud del organismo acorde con los avances en conocimientos de medicina, parece haberse dirigido progresivamente hacia el exceso y la falta de cuidado. Por otro lado, los horarios de trabajo y el estrés han venido creciendo exponencialmente, llevados por una era en la que únicamente parece importante el tiempo que no se aprovecha. Y, como han demostrado diferentes técnicas, el cuerpo humano no ha sido diseñado para la vida que se intenta vender cada día. Sin embargo, en esta primera década del siglo XXI, los avances en las Tecnologías de la Información y las Comunicaciones (TIC) y especialmente en el ámbito de la Ingeniería Telemática (IT) vuelven a revolucionar la medicina desde que Hipócrates la desvinculó de la mitología en el siglo IV A.C. Las aplicaciones y servicios de soporte orientados al paciente dan el salto definitivo de la salud convencional a la telemedicina [1]. Todo ello no sólo es debido a que la IT se ha optimizado, permitiendo resolver problemas y limitaciones derivadas de algunas tecnologías obsoletas sino que, además, permite plantear nuevos casos de uso para aplicaciones sanitarias no concebibles previamente [2]. En este contexto, el nivel de complejidad que han alcanzado los nuevos dispositivos médicos debido a la escalada tecnológica de los últimos años, ha provocado que incluso los médicos tengan que especializarse además en este conocimiento científico-técnico. Además, han de adquirir las habilidades necesarias para el manejo y operación de los equipos, muchas veces a cargo del personal técnico especializado. Por todo ello, el concepto actualmente de "certificar en Informática Médica" a los profesionales de la salud está teniendo cada vez más repercusión.

Pero no sólo los médicos se ven afectados por esta revolución tecnológica. Con la evolución del mercado de la electrónica y el consumo, los fabricantes han sido capaces de descubrir un nuevo nicho de mercado en la salud personal [3], y los propios pacientes o usuarios adquieren dispositivos médicos asequibles tanto en precio como en configuración y complejidad de uso. Así, surge un nuevo paradigma sanitario que contempla la posibilidad de que el propio paciente pueda tener una actitud, no solo colaborativa en cuanto al seguimiento de las enfermedades, sino *preventiva* llevando a cabo el autocontrol de su propia salud [4]. Desde entonces, la IT ha participado de este nuevo reto proporcionando recursos para el desarrollo de aplicaciones innovadoras donde el paciente es capaz de adquirir la medida de sus señales vitales y transmitirlas con seguridad al centro sanitario para su almacenamiento y posterior procesamiento.

Este nuevo enfoque, basado en la motivación y la búsqueda de la actitud participativa, supone que los ámbitos de aplicación se extiendan a entorno de paciente, desplegando nuevas redes de área personal y corporal (*Personal/Body Area Network*, PAN/BAN) [5], [6]. Las tecnologías sobre las que se sustentan estas PAN/BAN comparten una serie de características: bajo consumo de energía, radios de cobertura del orden de metros y protocolos de comunicación de baja carga sin debilitar la seguridad en la transmisión. Gracias a su implantación en el ámbito cotidiano y su mejora al coexistir varias alternativas en paralelo favoreciendo la competitividad, estas tecnologías han supuesto indiscutiblemente un factor decisivo para la orientación de los dispositivos médicos hacia el usuario y su ecosistema personal [7]. La prueba ha sido la rama de evolución que han tenido los dispositivos médicos hacia los dispositivos de salud personal: portables, adaptables al vestuario y más eficientes (aunque con limitaciones de autonomía, funcionalidad y calidad de los sensores debido a factores computacionales) [8].

En este contexto, los fabricantes de dispositivos médicos han dotado a sus equipos de sistemas y protocolos con el propósito de permitir la comunicación de las medidas obtenidas como valor añadido al producto. Tradicionalmente, las tecnologías de transmisión empleadas han sido: puerto serie (generalmente interfaz RS-232, aunque también eran habituales los interfaces propietarios), inalámbrico (mediante

protocolo IrDA), radiofrecuencia dentro de la banda sanitaria o Ethernet (interfaz RJ-45). Pese a que se trata de tecnologías estandarizadas a nivel internacional, no sucedía lo mismo con la semántica del protocolo, que era propietario. Esto se traducía en un formato desconocido para los datos, las etapas de la comunicación o los mensajes de diálogo y control. Por lo tanto, las posibilidades de desarrollar aplicaciones homogéneas, integrando múltiples dispositivos diferentes, y orientadas a nuevos casos de uso en entornos sanitarios o de paciente, son escasas debido a las dificultades derivadas del proceso de análisis e implementación de estos protocolos propietarios [9]. Hoy en día, las tecnologías de transmisión han recorrido un largo camino, y pueden encontrarse dispositivos médicos con interfaces cableados como *Universal Serial Bus* (USB), e inalámbricos como Bluetooth, ZigBee, WiBree, *Ultra Wide Band* (UWB), etc. Aun así, en un intento por seguir controlando la cuota de mercado y fidelizar a los usuarios, los fabricantes siguen apostando por la utilización de aplicaciones propietarias para exprimir las posibilidades de sus productos de cara al usuario [10].

Pero no ha sido hasta que las tecnologías que componen Internet y el *World Wide Web* (WWW) han alcanzado una extensión y desarrollo suficientes que se ha podido plantear el nuevo paradigma de e-Salud o salud 2.0. La implantación de las tecnologías IT ha revolucionado la metodología de trabajo de los centros sanitarios, especialmente en la automatización y clasificación de los flujos de información, que habían estado hasta entonces siendo gestionadas principalmente por personal administrativo de los hospitales. Además, se establecen nuevos casos de uso centrados en el paciente a partir de las capacidades tecnológicas ofrecidas por los dispositivos médicos y la posibilidad de integración directa de sus datos en los sistemas de información. El historial clínico comienza a implementarse íntegramente en sistemas informáticos basados en servidores, bases de datos y sitios web para el acceso y gestión, dando lugar al Historial Clínico Electrónico (HCE) [11].

Evidentemente, los fabricantes observan un nuevo mercado emergente al cual no tardan en ofrecer una solución rápida: la conectividad en los equipos. Con la conectividad como nuevo ámbito de desarrollo, surgen infinidad de soluciones, sistemas y protocolos basados en diferentes tecnologías que, aunque solventan el problema, lo siguen haciendo de una forma independiente y propietaria. Ahora, el usuario dispone de una gran cantidad de dispositivos sobre los que elegir. Pero, al final, el sistema que emplean éstos para transmitir los datos sigue sin ser innovador, pues todavía depende de su aplicación propietaria para colectar los datos desde los aparatos. Más aún, para su funcionamiento en unos márgenes de garantía proporcionados por el fabricante era, y sigue siendo, habitual la instalación de *software* específico que se encargue de las comunicaciones. Esto introduce un nuevo factor de dificultad a la hora de poder integrar en una misma solución diferentes dispositivos dado que sus aplicaciones son incompatibles y se obliga a adquirir productos de un mismo fabricante sin poder atender a otros criterios más relacionados con el tipo de aplicación sanitaria como calidad, fiabilidad o la estabilidad [12].

Mientras tanto, los usuarios han ido adquiriendo la posibilidad de poder llevar a cabo un seguimiento de sus dolencias, tratamientos o complicaciones de salud desde el propio domicilio, con mayor rango de equipos disponibles. Esto ha impulsado a los fabricantes y proveedores de servicios a ampliar los objetivos y desarrollar gran cantidad de dispositivos médicos para posibilitar nuevos sistemas de telemonitorización orientados a *fitness* (podómetro, medidor de ritmo cardiaco), cuidado de la salud personal (báscula, tensiómetro), nutrición y dieta (analizador de masa corporal), cuidado de ancianos (*hub* domiciliario), seguimiento continuo de enfermedades (analizador de anticoagulante en sangre, glucómetro) así como otros casos futuros de uso (analizador de orina, etilómetro), etc. [13]. Así, el paciente se encuentra ante todo un ecosistema de dispositivos médicos que conforman un amplio abanico de aplicaciones y servicios a su disposición, pero que no siempre son homogéneos.

Es en este punto donde surge la necesidad de garantizar la interoperabilidad entre los dispositivos médicos mediante la armonización de protocolos de comunicación propietarios y estándares. En otros ámbitos tecnológicos, como las comunicaciones ente equipos multimedia o periféricos, la estandarización ha tenido una evolución mucho más rápida que en el campo médico. Si un usuario conecta una memoria USB en un PC, el sistema queda listo para gestionar los contenidos sin tener que realizar ningún paso adicional. Si un usuario compra un equipo "manos libres" Bluetooth para usarlo en sus conferencias vía *skype* en un PC, un *Tablet* PC o un *Smartphone*, el proceso de *pairing* es lo único que lo separa de una conexión como la de USB (y sólo ha de realizarla una sola vez). En e-Salud, como se ha comentado, el modelo de negocio es muy diferente. Actualmente, para recibir una medida en un dispositivo médico, hace falta un *software* específico que se encargue de la comunicación. Pese a que existen propuestas que tratan de abarcar la mayor parte de las aplicaciones de usuario, la ausencia de un protocolo único dificulta su integración en este complejo ecosistema [14] e impide su armonización con otros servicios como gestión de HCE, soporte a la toma de decisiones clínicas o integración de "servicios en la nube".

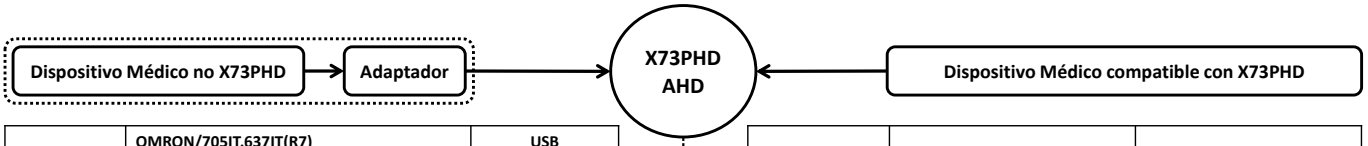
En esta compleja problemática, este artículo propone una plataforma integrada de e-Salud (*Integrated Health Platform*, IHP) para telemonitorización de pacientes que contempla la armonización de protocolos de comunicación propietarios y estándares, con la posibilidad final de poder incorporar la información fisiológica del paciente a su correspondiente HCE alojado en el centro de salud. Para conseguir dicha armonización, se propone un modelo abstracto de representación de dispositivos basado en tecnologías interoperables. En cuanto a protocolos, se establece una capa de abstracción que independice los mecanismos de gestión de las diferentes tecnologías usando, como modelo para la convergencia, la sintaxis y nomenclatura de la familia de estándares ISO/IEEE 11073 (X73) [15]. En la Sección II se analizan los recientes avances en estandarización para e-Salud y las normas internacionales para interoperabilidad de dispositivos médicos. En la Sección III se presenta un detallado estudio de la actualidad de los dispositivos médicos, propietarios y estándares, para soluciones de e-Salud. La Sección IV presenta la plataforma propuesta, detallando cada bloque que la conforma y analizando su integración con tecnologías de comunicación (como Bluetooth) y con un servidor de HCE. Además, se incluyen ejemplos de funcionamiento de cada bloque de la plataforma en fase actual de evaluación piloto para telemonitorización de pacientes. La Sección V analiza las tendencias futuras de la plataforma. Finalmente, en la Sección VI se discuten las conclusiones y líneas futuras de trabajo.

II. Interoperabilidad y estandarización

Los esfuerzos por establecer estandarización a nivel internacional en e-Salud han venido desde instituciones, organizaciones y grupos de desarrollo que han trabajado durante años para ofrecer una solución e introducirla en la sociedad. La mayor parte del desarrollo se ha gestado en el grupo especial para *Personal Health Devices* (PHDs *Working Group,* PHDWG) [16]. Esta organización, creada dentro del *Institute of Electrical and Electronics Engineers* (IEEE), está compuesta por miembros de ámbitos tan diversos como organizaciones orientadas al desarrollo de protocolos (Intel, Texas Instruments, Cisco), universidades (*Waseda University*, *Kyungpook National University*, *University of Applied Sciences Technikum Wien*) o empresas (Phillips, Toshiba), generalmente relacionadas con grupos dedicados al sector e-Salud e IT. PHDWG desarrolla continuamente propuestas y revisiones para ser aprobadas por IEEE, *International Organization for Standardization* (ISO) o el organismo de referencia europeo *Comiteé European Normalisation* (CEN) [17]. En 2006, PHDWG elige la familia de estándares X73 como el protocolo internacional para interoperabilidad de dispositivos médicos. En 2008, PHDWG publica la más reciente versión de la norma X73 para PHDs (X73PHD). Desde entonces, se sigue trabajando activamente en la mejora de su redacción dando como resultado la revisión en 2010 (*Optimized Exchange Protocol Amendment 1*), y la aprobación de nuevas especializaciones de dispositivos médicos. Además, X73PHD ha servido como base de iniciativas para promocionar la estandarización en ese segmento del ecosistema de salud informatizada destacando, entre ellas, *Integrating the Healthcare Enterprise* (IHE) [18] o *Continua Health Alliance* [19]. Esta última impulsa un sistema de certificación de dispositivos que cumplen X73PHD facilitando su implantación en los sistemas existentes y unificando soluciones de e-Salud.

III. Actualidad tecnológica de dispositivos médicos

Como se ha comentado, la apertura de los dispositivos médicos al ámbito personal junto con las nuevas tecnologías IT provoca que muchos proveedores de servicios sanitarios se esfuercen por ser los primeros en dar soporte a personas con un comportamiento pro-activo. En este contexto nace el concepto de *patient-empowerment* [4] lo que dibuja un nuevo escenario de posibilidades para los dispositivos médicos en el que coexisten varias categorías. Por un lado, está el grupo de dispositivos médicos "clásicos" orientados a la medida de parámetros fisiológicos habituales (peso, temperatura, presión arterial, pulso). A este grupo pertenecen los equipos que no ofrecen conectividad (y sólo ofrecen visualización de la medida por pantalla) junto con los dotados de conectividad sobre diversas tecnologías (habitualmente propietarias, de ahí la falta de interoperabilidad). Se muestra, en la tabla izquierda de Fig. 1, un listado de los equipos y modelos de este grupo más frecuentes en servicios sanitarios. Con la iniciativa de *Continua Health Alliance* (entre otras colaboraciones), desde 2008 se dispone de equipos "clásicos" con conectividad pero con formato de comunicaciones estandarizado. Estos nuevos dispositivos certificados pueden integrarse en aplicaciones tanto hospitalarias como personales con intercambio de datos fisiológicos a través de la red. Se muestra, en la tabla derecha de Fig. 1, un listado de los equipos y modelos certificados de este segundo grupo, disponibles en el mercado. Por último, para la armonización tecnológica de ambos grupos, es necesario incluir adaptadores en los dispositivos del primer grupo (ver Fig. 1). Estos adaptadores (en este caso, conforme a X73PHD) permiten la comunicación con su correspondiente dispositivo gestor (*Application Hosting Device*, AHD) de forma análoga al resto de equipos certificados. Así, una propuesta de integración de ambos grupos en una plataforma armonizada garantizaría interoperabilidad a todos los niveles.



Tensiómetro	OMRON/705IT,637IT(R7)	USB
	A&D/UA-767PBT	Bluetooth
	LifeStar/BPPro	Bluetooth
Pulsioxímetro	DatexOhmeda/3900,3775 (3740)	USB
	Diemer-Nellcor/NanoxI-II,Pox10,Capnox	IrDA
	MCC/PO-300	RS-232
Glucómetro	ACCU-CHEK/CompactPlus,Aviva(Sensor)	IrDA
	Telcomed/GL-100	Bluetooth
	MyGlucometer/MyGlucohealth	Bluetooth
Espirómetro	Diemer/D-111,D-120,D-600 (AB-II)	RS-232
	RespironicsNovametrix/610,710,715	USB
	MIR/SpiroBankII,SpiroLabIII	USB, Bluetooth
ECG	Diemer/AR600,AR1200,AR2100	IrDA
	BIOX/CB-2300-A	USB
	CardioLine/D1Plus,D3Plus,D3/6Plus	RS-232
Báscula	A&D/UC-321PBT	Bluetooth
	A&D/UC-321PL	RS-232
	Telcomed/WS-210	Bluetooth

Bluetooth	OMRON/HEM-7081-IT	Tensiómetro
Bluetooth	A&D/UA-767PBT-C	
Bluetooth	Nonin/Onyx® II 9560	Pulsioxímetro
Bluetooth	Nonin/WristOx2™ Model 3150	
USB	Nonin/2500 PalmSAT	
USB	ACCU-CHEK/Smart Pix	Glucómetro
Bluetooth	A&D/UC-321PBT-C	Báscula
Bluetooth	Omron/HBF-206IT	Analizador de masa corporal
Bluetooth	Omron/HJ-721IT	Podómetro

Fig. 1 Dispositivos médicos "clásicos" orientados a la medida de parámetros fisiológicos habituales.

Por otro lado, se encuentra el grupo de "*gadgets* médicos". Son dispositivos que, partiendo del modelo de dispositivo "clásico", comprenden desde prototipos, proyectos o resultados de líneas de investigación que tratan de cubrir (respetando siempre la correcta práctica médica y procedimientos de validación clínica), en algunos casos, aquellos espacios dentro del panorama de e-Salud que todavía no se plantean. Se muestra en Fig. 2 algunos de los ejemplos más recientes y relevantes de este segundo grupo. Estas nuevas tendencias son las que más se acercan a la e-Salud personal y la motivación del paciente. Ofrecen una imagen mucho menos complicada, la posibilidad de extraer información relevante en relación a la salud y estilo de vida del usuario, y se integran en las nuevas tecnologías que ya forman parte, a pasos agigantados, del ecosistema personal. Dentro de los tipos de dispositivos que han surgido en este ámbito cabe destacar, por ejemplo, los analizadores del sueño. Mediante acelerómetros colocados como accesorio sobre el usuario, son capaces de analizar los diferentes estados por los que atraviesa el individuo durante el sueño, su calidad y, además, despertarle dentro de un periodo temporal en el que el estado tras la interrupción sea óptimo. Otras propuestas han aprovechado las posibilidades y penetración de dispositivos *iPhone* o *iPad* para lanzar periféricos como tensiómetros, glucómetros, estetoscopios, monitores de señal ECG o EEG y dermatoscopios (conectables al dispositivo mediante un cable serie propietario). Esto último permite reaprovechar el potencial y conectividad de equipos portables como *iPhone* (o *Smartphones* similares) para el procesado de señales obtenidas a través de sensores intercambiables, dando forma a un dispositivo médico universal.

Litmann 3200	**Estetoscopio** Bluetooth	Bluetooth
Jazz Meter	Glucómetro	Serial + iPhone
ReSound Alera	**Audífono** integrable con dispositivos electrónicos	RFCOMM
Proteus	Monitorización de **ingestión de medicamentos** (pastillas)	RFCOMM
CardioBip	ECG de 12 derivaciones móvil	Serial
PiiX	Frecuencia Cardiaca, respiración, postura, ECG	RFCOMM
ECG Glove	Guante para medición de ECG con 12 derivaciones	Serial
iPhone ECG	Accesorio para iPhone para la obtención de ECG	Serial +iPhone
Withings Body Scale	Báscula	Wi-Fi
CarcioMEMS	Tensiómetro implantable	RFCOMM
Withings Monitor	Tensiómetro	Serial + iPhone
iHealth	Tensiómetro	Serial + iPhone
XWave	EEG monitor	Serial + iPhone
Handyscope	**Dermatoscopio**	Serial + iPhone
IQ	ECG para ensayos clínicos, registro de voz	RFCOMM
i-STAT 1	**Analizador de sangre**	RFCOMM
SickVerify	**Medidor de anticuerpos y cortisol en saliva**	USB
Vscan	**Ecógrafo**	USB
Monica AN24	Monitor **ECG Fetal**	Bluetooth
Symphony tCGM	Glocómetro capilar	RFCOMM
BodyTrace	Báscula	GSM
iRythm	ECG (Holter)	RFCOMM
iBigStar	Glucómetro	Serial + iPhone
MyZeo	**Sleep monitor**	Bluetooth
MDMouse	**Tensiómetro** incorporado en un ratón de PC	USB

Fig. 2 *Gadgets* médicos para nuevos escenarios de e-Salud.

En cuanto a la disponibilidad de repositorios de información, registros e historiales generados a partir de los datos obtenidos mediante todos estos dispositivos, además de los implantados en los mismos hospitales, existen hoy en día propuestas lanzadas por las grandes compañías informáticas como *Microsoft HealthVault* [20] y *Google Health* [21]. *Microsoft HealthVault* propone una herramienta instalable (mediante *drivers*) en el ordenador del usuario con el objetivo de dar una adaptación *software* para una gama de dispositivos de varios fabricantes. Esto sería equivalente a los módulos adaptadores propuestos en este trabajo, pero sin considerar la problemática *hardware* de la conectividad ni garantizar compatibilidad con X73PHD. *Google Health* trata de posibilitar, además del almacenamiento y computación en la nube (*cloud computing*), la integración de los datos médicos del usuario con el resto de sus herramientas como calendario (*Calendar*), correo electrónico (*Gmail*), documentos compartidos (*Google Docs*), entre otros. Sin embargo, no ofrecen *software* de adaptación, ni integración con la HCE en formato interoperable, como se plantea en este trabajo, sino como plataformas independientes.

IV. Diseño e implementación de la plataforma integrada de e-Salud para telemonitorización

A partir de las consideraciones anteriores y siguiendo las premisas establecidas en la introducción, se propone una plataforma integrada de e-Salud (*Integrated Health Platform*, IHP). El diseño de esta plataforma persigue ofrecer los servicios básicos de telemonitorización de pacientes mediante interfaces de gestión para los procesos de obtención de muestras y sus algoritmos inherentes. Los casos de uso que soporta la plataforma están basados en la obtención y envío, desde un emplazamiento fijo, de datos discretos (presión arterial, peso) o continuos (pletismografía, ECG) tanto en modo *store-and-forward* como en tiempo real. En cuanto al acceso remoto, la plataforma dispone de un sistema de subscripción mediante el cual una persona acreditada puede recibir en cualquier momento información relacionada con los parámetros fisiológicos del paciente así como resúmenes estadísticos, si estos están disponibles. En el extremo del sistema se encuentra, ya en el dominio de gestión del centro de salud responsable, el HCE del propio paciente, al cual los datos obtenidos desde la plataforma son incorporados.

Con respecto a la gestión de los dispositivos médicos, se ha propuesto un sistema abstracto de modelado genérico y configuración de los equipos, ateniéndose a sus características técnicas y de representación de datos comunes. El modelo base a partir del cual se ha elaborado la propuesta es el correspondiente al estándar X73PHD, con el objetivo de converger hacia una solución homogénea e interoperable con otros sistemas. Partiendo de las especializaciones definidas en el estándar y de las características de los dispositivos disponibles en el mercado, se ha definido un algoritmo descriptor de modelos basado en la tecnología *eXtensible Markup Language* (XML). El objetivo final es evitar la restricción de uso de la plataforma a un rango predeterminado de dispositivos, aspecto que suele ser característico de los fabricantes. Como resultado, se posibilita la incorporación en una misma aplicación de un número de dispositivos médicos mucho mayor incluyendo, no sólo últimos lanzamientos, sino además aquellos equipos que hayan podido quedar obsoletos (dentro de la categoría de "clásicos" con opciones de conectividad no estándar), pero que siguen presentes en la rutina médica diaria. Esto es fundamental para plantear evaluaciones piloto en entornos sanitarios en los que el

"parque" de dispositivos médicos puede no haber sido renovado en la última década. La idea es que esos equipos puedan seguir usándose, integrados en esta plataforma, agregando los módulos adaptadores necesarios, como se comentó en el apartado anterior.

Con las premisas de diseño y funcionamiento previas, la implementación de la plataforma ha sido planteada como una aplicación *software* modular y escalable donde cada una de las funciones de gestión, modelado, monitorización e interacción del usuario están gobernadas por un núcleo central (IHP). Esta aplicación software, instalada en el equipo del usuario (*Netbook*, *Tablet* PC, *Smartphone*), permite aprovechar los recursos del sistema disponibles para el realizar las funciones de procesado y comunicación con los dispositivos sin tener que optar en una primera aproximación por el desarrollo de una solución dedicada (embebida). Se muestra en Fig. 3 un esquema funcional de la propuesta de plataforma detallando a continuación cada uno de los módulos implementados que la conforman:

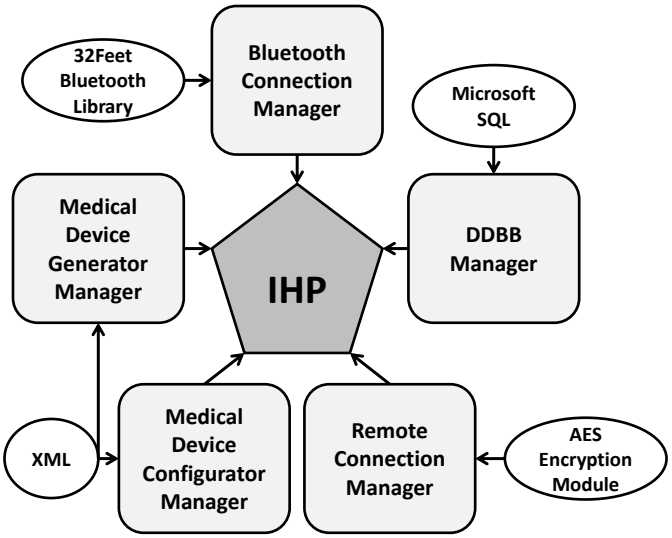


Fig. 3 Esquema funcional de la plataforma integrada de e-Salud (IHP).

- ***Bluetooth Connection Manager*** **(BCM)**. Se encarga de la gestión de las conexiones establecidas con los dispositivos médicos que hagan uso de tecnología inalámbrica Bluetooth para la transmisión de los datos. Dado que se van a gestionar varios dispositivos al mismo tiempo, cada uno de ellos con eventos de conexión lanzados en instantes de tiempo diferentes, en lugar de usar puertos COM virtuales, construidos sobre *Serial Port Profile* (SPP), se consiguen definir, mediante la librería de código libre *In The Hand's 32feet* [22]. Esta librería expone unos *streams* de comunicación que facilitan el manejo de diferentes hilos de gestión, además del procesado de las tramas intercambiadas durante la comunicación. Estos *streams* son vinculados al canal de comunicación lógico de cada dispositivo según se asocian con la plataforma y, una vez verificados, el BCM los asigna al hilo de gestión correspondiente de IHP. Por otro lado, esta librería introduce algunos elementos que en el futuro permitan la integración del nuevo perfil Bluetooth *Health Device Profile* (HDP), el cual usa características como el *Multi-Channel Adaptation Protocol* (MCAP) y algunas de nivel *Logical Link Control and Adaptation Protocol* (L2CAP) como *Enhanced Retransmission Mode* y *Streaming Mode*. El gestor es responsable de relacionar la dirección *Medium Access Control* (MAC) Bluetooth del dispositivo entrante con el objeto correspondiente configurado en la plataforma para iniciar el proceso de comunicación correctamente.
- ***Remote Connection Manager*** **(RCM)**. Su misión es la de proporcionar un servicio de suscripción a la plataforma (actuando como servidor de información), de forma que uno o más usuarios autorizados (clientes) puedan estar recibiendo de forma simultánea datos relativos a la evolución de los valores fisiológicos del paciente obtenidos a través de los dispositivos asociados. Se ha diseñado un protocolo de servicio ligero que soporta las principales funciones básicas del proceso intentando minimizar el *payload*, incorporando una máquina de estados finita minimizada de 5 etapas. Para ello, se aprovecha el factor de que la conexión puede ser interrumpida en cualquier momento sin solicitud previa, al no estar el servicio vinculado a procesos de decisiones clínicas sino a simple monitorización. En cuanto a seguridad, el programa ha de reunir unas características mínimas teniendo en cuenta la naturaleza de la información a transmitir. La estrategia empleada, como prueba de concepto, es el uso de un mecanismo basado en *login* y *password* para identificación del suscriptor y algoritmo *Advanced Encryption Standard*, AES para la codificación. El cliente ha sido diseñado para funcionar ubicuamente en un *Smartphone* (en este caso sobre Windows Mobile aunque exportable a Android o PC), habiéndose realizado pruebas de transmisión a través de redes de área extendida tanto *Wide Area Network* (WAN) 3G y *General Packet Radio System* (GPRS).
- ***Medical Device Configurator Manager*** **(MDCM)**. Este módulo permite gestionar los dispositivos médicos asignados a la plataforma trasladando las características reales de funcionamiento del equipo a un objeto procesable por la plataforma. En este proceso se definen dos conjuntos: los dispositivos que tienen la posibilidad de ser soportados por la plataforma y los que ya están configurados y funcionando. Para la organización se tiene en cuenta, como criterio de diseño, que en los casos de uso planteados el usuario no dispone de más de un dispositivo orientado al mismo propósito (no tendría sentido asignar dos tensiómetros distintos siendo que corresponden a una misma especialización X73PHD). Se muestra en Fig. 4 el interfaz gráfico implementado que incluye la lista de dispositivos soportados (ver lateral izquierdo de Fig. 4), otra lista conteniendo los configurados localmente (ver lateral derecho de Fig. 4) y una serie de botones de control (ver zona inferior de Fig. 4). Al seleccionar un elemento de cualquiera de ambas listas, se proporciona información relacionada con el dispositivo en cuestión, quedando activas las siguientes acciones de control posibles:
  - ADD (agregar dispositivo): si ya existe uno asociado, el módulo no permite añadir otro, solo sustituirlo.
  - REMOVE (eliminar dispositivo): para eliminar un dispositivo asociado.
  - EXCHANGE (intercambiar dispositivos): simplifica la acción de eliminar y asociar un dispositivo en el caso de que los dos dispositivos (soportado y configurado) sean de la misma especialización y no sean complementarios.
  - CONFIG (configurar dispositivo): una vez seleccionado un dispositivo que ya está funcionando en la plataforma, se puede gestionar la configuración de transporte para su funcionamiento (ver Fig. 5).

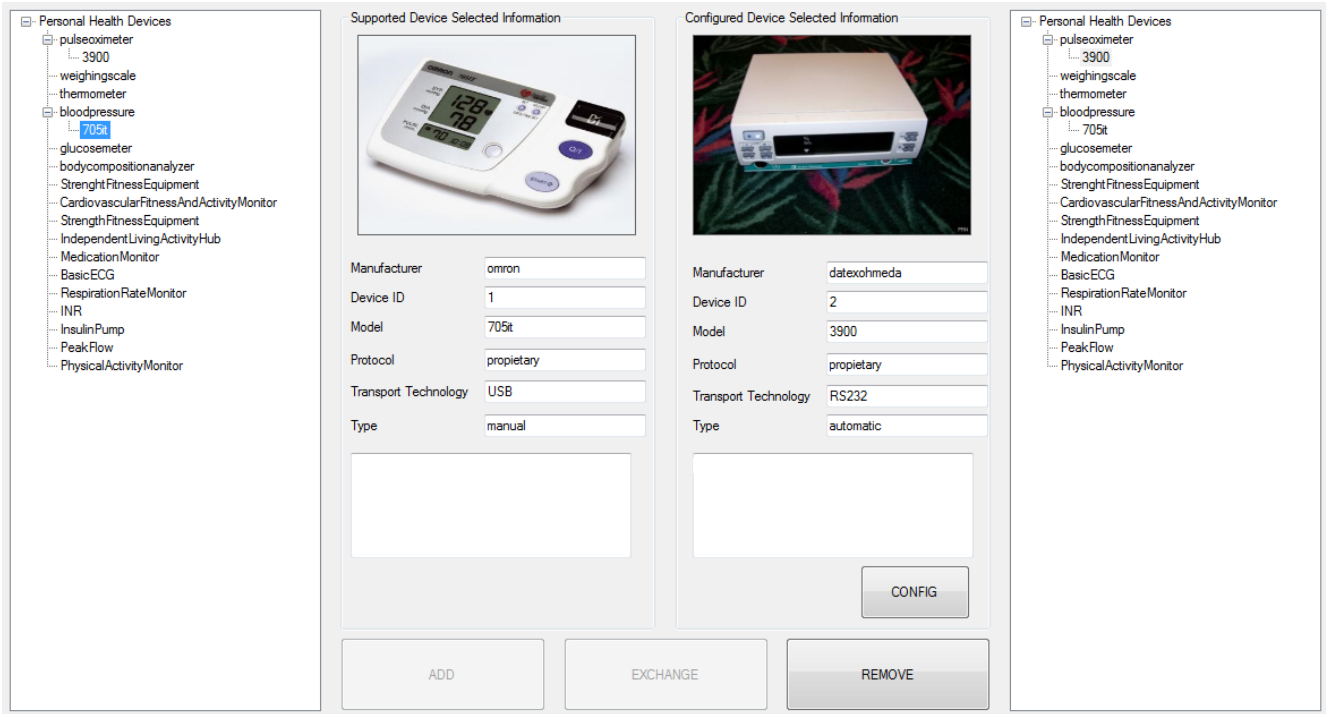


Fig. 4 Interfaz gráfico diseñado para el módulo *Medical Device Configurator Manager* de la plataforma.

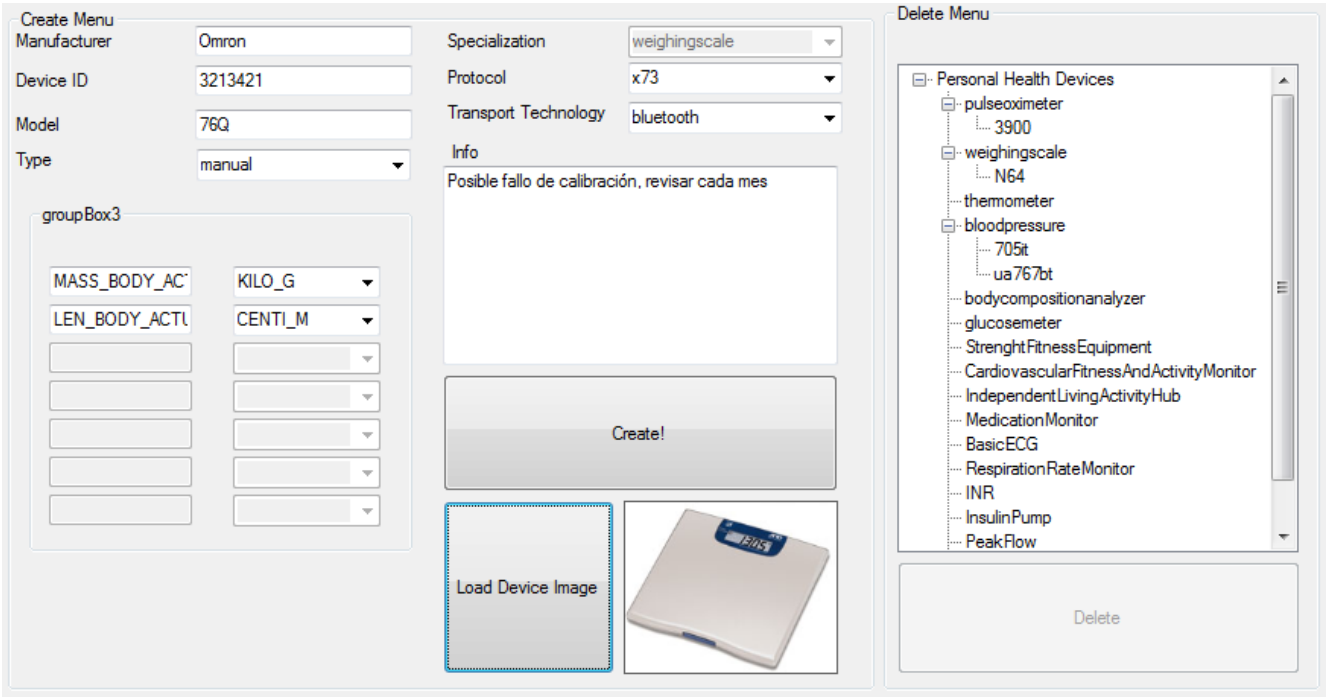


Fig. 5 Interfaz gráfico diseñado para el módulo *Medical Device Generator Manager* de la plataforma integrada.

Todos los cambios realizados mediante este módulo, quedan expuestos a la plataforma y su módulo principal IHP para la gestión de procesos a través de dos ficheros XML: *SupportedDevices* y *ConfiguredDevices*. El modelo de información jerárquica de los dispositivos que es representado de forma abstracta a partir de dichos XML se elabora, como se ha comentado previamente, tomando como base las características básicas definidas por el *Domain Informacion Model* (DIM) de X73PHD [15], de forma que se consiga facilitar su integración en el ecosistema de aplicaciones X73PHD. Además, éste se completa mediante características adicionales para cubrir un abanico más amplio de dispositivos, teniendo en cuenta todo tipo de equipos comerciales aunque no se hayan contemplado todavía en las especializaciones desarrolladas por X73PHD. Así mismo, esta representación permite agregar equipos médicos de tipo "accesorio" o *gadgets* (Fig. 2) que no acaban de ser equipos clásicos, pero sí los complementan ofreciendo funcionalidades importantes. El primero de los ficheros, *SupportedDevices*, contiene aquellos dispositivos conocidos que pueden funcionar en la plataforma, por ejemplo, los equipos médicos cuyos *drivers* están disponibles porque se dispone de su protocolo de comunicaciones. El segundo, *ConfiguredDevices,* contiene aquellos dispositivos que ya han sido configurados para trabajar en la plataforma y actualmente el usuario hace uso normal de ellos para la obtención de medidas. Ambos ficheros tienen una estructura similar, diferenciados únicamente en la configuración específica que se ha introducido sobre un dispositivo una vez que se ha incluido en la plataforma (por ejemplo, configuración del puerto COM). En Fig. 6 se muestra un ejemplo basado en XML de los dispositivos soportados, en cuya descripción se incluyen algunos parámetros como información relativa al dispositivo (especialización, fabricante, identificador y modelo) tecnología e interfaz de transporte, tipo de protocolo de comunicaciones y modo de funcionamiento (manual o automático). Este último es especialmente importante de cara a la usabilidad e interacción con el usuario dado que la aplicación deberá de llevar a cabo un proceso de diálogo para que la realización de las medidas se haga de forma correcta y verificada por el usuario. A nivel de información relativa a parámetros fisiológicos (en Fig. 4 se muestran las medidas de presión arterial y pulsioximetría con sus respectivas unidades) o ambientales, se definen los atributos disponibles haciendo uso de la nomenclatura X73PHD.

- ***Medical Device Generator Manager* (MDGM)**. Complementario a MDCM, facilita la creación de modelos de dispositivos partiendo de un patrón genérico, adaptado a las características particulares del equipo. Si un nuevo dispositivo puede formar parte del ecosistema de la plataforma y no existe un modelo XML disponible, este módulo permite configurar uno (ver Fig. 5). Una vez completado, pasa a formar parte del grupo de dispositivos soportados como entrada al MDCM en *SupportedDevices*. Dentro del conjunto de datos necesarios para la configuración están aquellos relativos al dispositivo a nivel administrativo como, por ejemplo: fabricante, identificador y modelo. Adicionalmente, es necesario introducir el tipo de funcionamiento, ya sea manual o automático y, una vez seleccionada la especialización, los campos relativos a las unidades de medida (ver parte inferior izquierda de Fig. 5). Es importante verificar este parámetro dado que una vez los datos son enviados al HCE remoto, podría producir alertas por falsos positivos (niveles de presión arterial erróneos cuando el paciente se encuentra en perfecto estado). El último parámetro necesario es el tipo de tecnología de transporte, que podrá ser modificada posteriormente a través del módulo MDCM. Otros datos opcionales son la imagen del dispositivo (si se encuentra disponible), información adicional respecto al funcionamiento del equipo u otras peculiaridades.

```
<configuredDevices>
 <device>
  <specialization> MDC_DEV_SPEC_PROFILE_BP </specialization>
  <protocol> propietary </protocol>
  <transport> USB </transport>
  <manufacturer> omron </manufacturer>
  <model> 705it </model>
  <id> 1 </id>
  <type> manual </type>
  <info> Información adicional sobre el dispositivo </info>
  <unitsDefinition>
   <componentName> MDC_PRESS_BLD_NONINV_SYS </componentName>
   <units> MDC_DIM_MMHG </units>
  </unitsDefinition>
 </device>
 <device>
  <specialization> MDC_DEV_SPEC_PROFILE_PULS_OXIM </specialization>
  <protocol> propietary </protocol>
  <transport> RS232 </transport>
  <manufacturer> datexohmeda </manufacturer>
  <model> 3900 </model>
  <id> 000A4F010014 </id>
  <type> automatic </type>
  <info> Información adicional sobre el dispositivo </info>
 </device>
</configuredDevices>
```

Fig. 6 Ejemplo XML de dispositivos soportados.

- ***Database Manager (DDBBM)***. Este módulo tiene la función de gestionar las bases de datos existentes en la plataforma, dando servicio tanto a los módulos de configuración de dispositivos como al gestor de conexiones remotas. Por un lado, la base de datos principal está destinada a almacenar información fisiológica relacionada con el usuario de forma que puedan realizarse búsquedas y elaborarse informes o análisis, siendo este conjunto de datos debidamente protegido mediante cifrado, control y registro de acceso. Dependiendo de la configuración del sistema se ofrece la opción de deshabilitar dicha funcionalidad, eliminando adicionalmente los datos fisiológicos obtenidos tras el envío de informes al HCE. Por otro lado, es posible además disponer de una base de datos que contenga las características más importantes así como los archivos de configuración de los dispositivos médicos que deben integrarse en el sistema X73PHD. Esta base de datos se obtiene como un subconjunto de un repositorio principal alojado en un servidor remoto de telemonitorización, enfocado a proporcionar actualizaciones y copias de seguridad de configuraciones de equipos. A partir de esta copia local se obtienen los ficheros relacionados con los dispositivos en formato XML. Para la implementación de la base de datos se ha utilizado Microsoft MySQL, acorde con todo el entorno de desarrollo de la plataforma, que ha sido Visual Studio 2010.

Por último, IHP implementa la posibilidad de poder realizar un envío manual o programado de la información obtenida del paciente a través de los dispositivos a un servidor remoto de HCE implementado conforme a UNE-EN ISO 13606. A pesar de compartir aspectos de seguridad similares al módulo RCM, éste módulo es en cambio unidireccional y contiene además datos relativos a usuario/paciente, en concreto su identidad, al menos con respecto al centro sanitario correspondiente. El protocolo de envío consiste básicamente en la elaboración de un XML diseñado con el objetico de minimizar el procesado necesario para la verificación y posterior e incorporación en el HCE del paciente. Un esquema funcional de la aplicación completa (IHP, cliente remoto y servidor EHR en el hospital) se muestra en la Fig. 6.

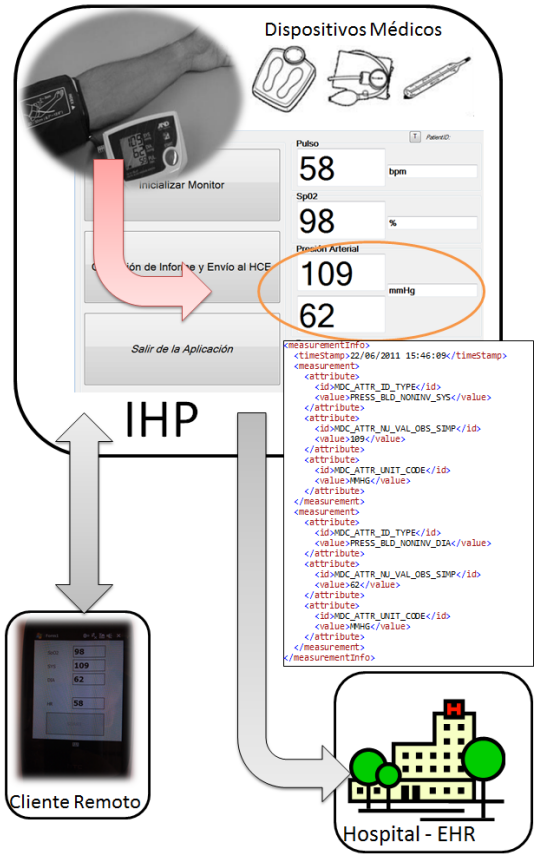


Fig. 6 Esquema funcional completo

V. Tendencias Futuras

Teniendo en cuenta la experiencia obtenida con esta implementación, los hábitos de los usuarios con respecto a las nuevas tecnologías y las posibilidades que éstas ofrecen, se valora la posibilidad de migrar hacia un entorno ubicuo y portátil, en el que las tecnologías de transferencia inalámbricas (especialmente aquellas que disponen de perfiles de interoperabilidad como Bluetooth o ZigBee) juegan un papel importante de cara a la usabilidad y versatilidad del sistema.

Finalmente es necesaria, como evolución natural de la plataforma, la adopción de protocolos de comunicación bidireccionales desde el centro remoto de gestión que, como medio contenedor, ofrezcan no sólo la transferencia de información relacionada con el paciente (monitorización y consulta de datos), sino además un servicio de gestión remota de carácter técnico/sanitario.

VI. Conclusión

La plataforma diseñada propone un nuevo planteamiento de integración que ofrece un ecosistema de dispositivos orientados a diferentes ámbitos de uso y con protocolos de comunicación estándares y propietarios. Tecnológicamente, la propuesta de modelado de dispositivos combina sus posibilidades individuales en torno a una aplicación centrada en el paciente con servicios de gestión homogeneizada. El modelo de dispositivo y la información gestionada están diseñados siguiendo el estándar X73PHD e integrados mediante ficheros XML con un servidor remoto de HCE conforme a UNE-EN ISO 13606. Esta propuesta garantiza interoperabilidad a todos los niveles y constituye una solución real y abierta para la problemática del sistema sanitario.

Agradecimientos

Este trabajo ha sido parcialmente subvencionado por los proyectos TIN2008-00933/TSI del Ministerio de Ciencia e Innovación (MICINN) y Fondos Europeos para el Desarrollo Regional (FEDER), TSI-020100-2010-277 y TSI-020302-2009-7/Plan Avanza I+D del Ministerio de Industria, Turismo y Comercio, y PI029/09 del Gobierno de Aragón.

Referencias

[1] Wooton, R., & Craig, J. (2006). "Introduction to Telemedicine (2nd ed.)". *Oxford: Marston Book Services Limited*.
[2] Korhonen, J., and Parkka, M. (2003). "Health Monitoring in the Home of the Future: Wear it well". *IEEE Eng Med Biol Mag* 22(3), 66-73.
[3] Simons, D. (2008). "Consumer Electronics Opportunities in Remote and Home Healthcare". *IEEE Int Conf on Consumer Electronics*, 1-2.
[4] Anderson, R. M., & Funnell, M. M. (2005). "Patient empowerment: reflections on the challenge of fostering the adoption of a new paradigm". *Patient Education and Counseling* 57(2), 153-157. doi: 10.1016/j.pec.2004.05.008.
[5] Van Langenhove, L. (Ed.). (2007). "Smart Textiles for Medicine and Healthcare: Materials, Systems and Applications". *CRC press, Woodhead Publishing Ltd*.
[6] Yang, G.Z. (2006). "Body Sensor Networks". *Springer-Verlag London Limited*.
[7] F. Axisa *et al*., (2005). "Flexible technologies and smart clothing for citizen medicine, home healthcare, and disease prevention," *IEEE Trans Inf Technol Biomed*, vol. 9, pp. 325-36.
[8] Bonato P. (2010). "Wearable sensors and systems". *IEEE Engineering in Medicine and Biology Magazine* 29(3):25–36.
[9] Reynolds M. *et al*. (2007). "Can telemonitoring systems interoperate? Review of the suitability of existing stantards for adaptable telecare provision". *Healthcare Comp Conf of British Comp Soc*, 104–115.
[10] Withings [On line] *http://www.withings.com/en/index/*. Last visit: 06/2011
[11] Santos MR, Bax MP, Kalra D. (2010). "Building a logical EHR architecture based on ISO 13606 standard and semantic web technologies". *Stud Health Technol Inform*. 160(Pt 1):161-5.
[12] Warren, S. *et al*. (2006). "Lessons learned from applying interoperability and information exchange standards to a wearable point-of-care system". *Transdisciplinary Conf Distr Diagn Home Healthcare*, 101-104.
[13] Jeong I., Jun S., Lee D., Yoon H., (2007). "Development of Bio Signal Measurement System for Vehicles". *Int Conf on Convergence Information Technology,* 1091-1096.
[14] Yao, J., Schmitz, R., & Warren, S. (2005). "A Wearable Point-of-Care System for Home Use That Incorporates Plug-and-Play and Wireless Standards". *IEEE Transactions on Information Technology in Biomedicine*, 9(3): 363-371.
[15] ISO/IEEE11073 - Personal Health Devices standard (X73-PHD). Health informatics. [11073-00103. Technical report - Overview] [11073-104xx. Device specializations] [11073-20601. Application profile - Optimized exchange protocol]. [On line] *http://standards.ieee.org/*. Last visit: 06/2011.
[16] IEEE SA - PHD - Personal Health Device (2010). [On line] *http://standards.ieee.org/develop/wg/PHD.html*. Last visit: 06/2011.
[17] Committee European Normalisation / Tech Committee 251 (CEN/TC251). [On line] *http://www.centc251.org*. Last visit: 06/2011.
[18] *IHE*. [On line] *http://www.ihe.net/*. Last visit: 06/2011.
[19] Continua Health Alliance. [On line] *http://www.continuaalliance.org/*. Last visit: 06/11.
[20] Microsoft HealthVault. [Online] *http://www.healthvault.com/personal/index.* Last visit: 06/2011.
[21] Google Health. [On line] *www.google.com/health*. Last visit: 06/2011.
[22] In The Hand's 32feet. [On line] *http://inthehand.com/content/32feet.aspx*. Last visit: 06/2011.

MOSAIC: Un sistema de intercambio de datos clínicos con soporte para acuerdos multilaterales

Magí Lluch-Ariet, Josep Pegueroles-Vallés
Departament d'Enginyeria Telemàtica (ENTEL)
Universitat Politècnica de Catalunya (UPC)
Edificio C3 - Campus Nord / C. de Jordi Girona, 1-3 / 08034 Barcelona
magi.lluch@entel.upc.edu, josep.pegueroles@upc.edu

***Resumen*—Cuantos más datos clínicos aparecen disponibles en la red, la tarea de acceder y explotar el gran número de repositorios distribuidos deviene cada vez más compleja. Además, acceder a un determinado conjunto de datos en un almacén de datos federado puede tener ciertas restricciones que pueden ser resueltas a través de acuerdos multilaterales. Dichos acuerdos, pueden ser muy complejos de ser resueltos de forma manual.**

Los sistemas actuales para la compartición de datos clínicos no contemplan acuerdos multilaterales. MOSAIC pretende aportar una solución modular y eficiente al problema de intercambio de datos con acuerdos multilaterales. El sistema que se propone aprovecha los sistemas multiagente y los actuales protocolos de interacción entre agentes, además de aquellos propios para la transferencia de datos clínicos.

***Palabras Clave*—Bases de Datos Federadas, Intercambio de datos clínicos y Sistemas Multi-Agente.**

I. Introducción

Cuando un médico busca casos similares para comparar con los datos obtenidos de su paciente, necesita seleccionar los centros clínicos donde buscar, negociar los derechos de acceso a los datos y, ocasionalmente, intercambiar algunos datos locales suyos con este centro. Localizar el centro clínico adecuado y conseguir un buen acuerdo representa un proceso complejo que puede ser parcialmente automatizado.

El principal objetivo del sistema de intercambio de datos propuesto es el de obtener el máximo rendimiento de los casos disponibles en cierto nodo para, mediante acuerdos, conseguir los derechos de acceso a los datos que se necesitan consultar de otros nodos de la red.

Un ejemplo que ilustra como un sistema para el intercambio de datos multilateral puede funcionar es el siguiente. Considere un entorno médico en el que los pacientes están clasificados en diferentes clases (A, B, C, ...) y para cada clase hay un conjunto de posibles diagnósticos (A1, A2, ...).

Nodo-I Aloja un Data Mart (base de datos local del nodo, parte del almacén de datos federado de toda la red) con casos de 'clase A', incluyendo todos los posibles diagnósticos de la clase.
Excepcionalmente, un nuevo paciente 'clase B' necesita ser diagnosticado y el médico quiere compararlo con otros casos (clase B) que ya hayan sido diagnosticados como B1 en nodos externos.

Nodo-II Aloja un Data Mart de pacientes 'clase B' con un excepcional número de casos diagnosticados como B1.
Un nuevo paciente 'clase B' necesita ser diagnosticado y su médico quiere compararlo con algunos otros casos en nodos externos diagnosticados como B2.

Nodo-III Aloja un Data Mart compuesto de datos 'clase B', con un considerable número de casos diagnosticados como B2, pero sin ningún caso diagnosticado como B1. Un nuevo caso 'clase A' ha de ser diagnosticado y el médico quiere compararlo con otros casos ya diagnosticados como A1 en otros centros.

No existe acuerdo bilateral posible para el intercambio de datos en esta red de tres nodos que resuelva las necesidades de acceso a los datos. Sin embargo, como se muestra a continuación, el acuerdo multilateral es posible:

1) El Nodo-III da permisos de acceso al Nodo-II para consultar los casos ya diagnosticados como B2
2) El Nodo-II da permisos de acceso al Nodo-I para consultar los casos ya diagnosticados como B1
3) El Nodo-I da permisos de acceso al Nodo-III para consultar los casos diagnosticados como A1

Cuando los acuerdos bilaterales para el intercambio de datos no son suficientes para conseguir el acceso a los datos deseados, un proceso automático que encuentre posibles acuerdos multilaterales es necesario. Este proceso tiene asociado un protocolo con las siguientes funcionalidades:

- Publicar la referencia de los datos disponibles en un nodo
- Enviar la petición de acceso a los datos deseados
- Enviar la autorización y los derechos de acceso a los datos
- Realizar la transferencia
- Enviar el acuse de recibo de los datos
- Retirar los derechos de acceso si el acuerdo no se satisface plenamente
- Iniciar, Confirmar, o Deshacer la transacción.

El trabajo que se presenta en este artículo cumple con estos requisitos y facilita la realización de acuerdos multilaterales involucrando la compartición de datos entre un cierto número de nodos de una red.

El resto del artículo se estructura como sigue: En la sección 2, se presenta una visión general del estado del arte en almacenes de datos clínicos federados y sistemas para la compartición de estos datos; en la sección 3 se describe la arquitectura de MOSAIC, con sus principales componentes y la arquitectura del protocolo del sistema; en la sección 4 se introduce el diseño del protocolo, explicando cómo los actores del mismo interactúan entre ellos; y en la sección 5, se concluye con los principales retos a los que MOSAIC se enfrenta y se analiza el trabajo futuro en relación al sistema propuesto.

ISBN: 978-84-694-5948-5

II. Estado del Arte en Sistemas Distribuidos y Federados en e-Salud

Desde la disponibilidad de repositorios electrónicos de datos en centros clínicos, varios proyectos e iniciativas desde diferentes disciplinas han contribuido a facilitar el intercambio y compartición de datos entre los centros: i) estándares relacionados con el cómo se tiene que almacenar la información, como DICOM [1], [2] y Historia Clínica Electrónica [3]; ii) estándares relacionados con la interoperabilidad y la transferencia de datos clínicos, como ISO/HL7 21731 [4] e ISO 13606 [5]; iii) protocolos para garantizar los principios básicos de seguridad y privacidad, como SSL; y iv) nuevas áreas de investigación que contribuyen a crear sistemas distribuidos, como la tecnología de Agentes Inteligentes y la Web Semántica, que facilitan la interoperabilidad entre Bases de Datos heterogéneas a través del uso de ontologías. Todas ellas convergen para hacer posibles sistemas y proyectos como los siguientes:

"Cancer Biomedical Informatics Grid" (caBIG) [6], una de las mayores iniciativas para proveer una infraestructura para construir bases de datos distribuidas, compartir datos y conocimiento, incluyendo un conjunto de herramientas de seguridad. "The Cancer Genome Atlas" (TCGA) [7] es uno de los ejemplos donde una base de datos distribuida se construye utilizando caBIG para su desarrollo. A pesar de ello, esta infraestructura no provee capacidades específicas para el intercambio multilateral de datos.

El proyecto Artemis [8] ha estado trabajando - dentro del contexto médico - con la transferencia de datos críticos de pacientes (principalmente datos clínicos). Para ello, ha utilizado una arquitecturas peer-to-peer y servicios web de protocolos de seguridad para la transferencia de registros de pacientes. Como en caBIG, acuerdos multilaterales para el intercambio de datos no están soportados en este proyecto.

HOPE [9] es una plataforma de colaboración en telemedicina para la compartición de datos clínicos. Implementa una infraestructura en malla para bases de datos distribuidas, aportando una interfaz común, independiente de la base de datos de sus nodos, para la gestión de los datos de los pacientes, accediendo a registros clínicos e imágenes (en DICOM) de PACS. En vez de proponer el intercambio de datos, este proyecto facilita la recopilación de información a través de una interfaz integrada.

Un ejemplo de almacén de datos federado y de su asociado sistema de soporte a la decisión lo aporta el proyecto HealthAgents [10], [11], que pretende construir una red mundial de centros clínicos para el diagnóstico de tumores cerebrales. Este proyecto está focalizado en la recolección de datos para la creación de clasificadores que serán usados durante el proceso de diagnóstico, pero no se centra en el intercambio de datos entre los nodos. Entornos de Agentes para propósitos médicos han sido documentados de forma extensiva en la literatura científica [12], [13], [14], pero su uso ha sido tradicionalmente confinado al control e información de los pacientes [15].

Todos estos sistemas implementan y proveen o bien un buen entorno para bases de datos clínicas distribuidas donde almacenar la información en un repositoria virtual, o bien bases de datos federadas donde la recuperación de los datos entre los nodos se realiza después de una consulta al sistema. A pesar de ello, a fecha de hoy, no hay ningún sistema que realice una negociación automática basada en agentes para el intercambio de datos clínicos en un almacén de datos federado.

III. Arquitectura

A. Arquitectura del Sistema

El sistema MOSAIC está compuesto por un conjunto de nodos interconectados, pudiendo tener cada uno de ellos un Data Mart asociado, con datos médicos locales. El propósito de MOSAIC es el de facilitar el intercambio de estos datos entre los nodos. Para soportar esta arquitectura, cada nodo de la red está compuesto por los siguientes componentes: i) Servidor Web (p.ej. Tomcat server); ii) SGBD (p.ej. MySQL); y iii) Plataforma de Agentes compatible con FIPA (p.ej. JADE).

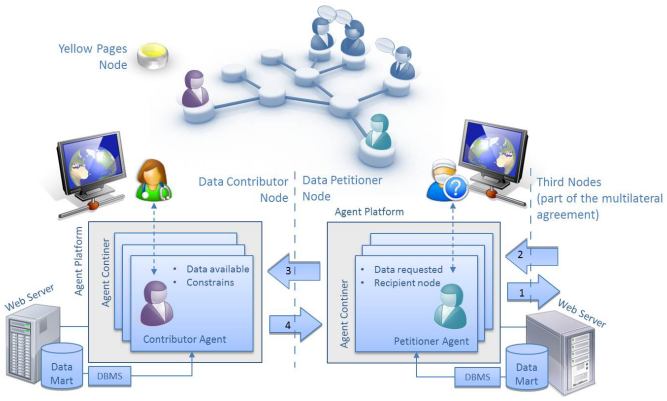


Fig. 1. Arquitectura de MOSAIC, mostrando el flujo de datos entre los Agentes: Entrega de datos a terceros nodos (1); Recolección de datos de terceros nodos, necesaria para satisfacer ciertas condidiones (2); Transferencia de datos a los nodos donde los datos deseados residen (3); y Transferencia de datos a los nodos peticionarios (4).

Los nodos de esta red pueden tener dos roles diferenciados: Uno actuando como "contribuidores de datos" (bautizados como "Contributor Agents") y el otro actuando como "demandantes de datos" (bautizados como "Petitioner Agents"). El protocolo de comunicaciones del sistema MOSAIC permite el intercambio de mensajes e información necesaria durante el proceso de negociación entre los nodos que pretenden lograr acuerdos para el intercambio de sus datos. Finalmente, los agentes "Yellow Pages" son los responsables de mantener la información de la topología de la red, dentro de una Base de Datos que denominaremos las "Páginas Amarillas" del sistema (Fig. 1).

- **El "Petitioner Agent"**. Este Agente es responsable de identificar cuáles son los nodos que pueden contener los datos solicitados, negociar con ellos los derechos de acceso, intentar solventar las condiciones y restricciones de acceso (si existen) y finalmente, recopilar los datos (si ello es posible).
- **El "Contributor Agent"**. Este Agente negociará, si es necesario, las condiciones para proveer el acceso a los datos con el "Petitioner Agent" y los entregará cuando se cumplan las condiciones de acceso a los mismos.
- **El "Yellow Pages Agent"**. Este Agente tiene como objetivo proveer el "servicio de directorio" al resto de

agentes en el sistema. Su principal objetivo es el de informar a los ”Petitioner Agents” de la lista de nodos activos y de sus ”Contributor Agents” a los que pueden intentar lanzar una petición de acceso a sus DataMarts.

B. La Arquitectura de Protocolos

MOSAIC pretende seguir los actuales estándares de Sistemas Multi-Agente, y más concretamente, aquellos definidos por IEEE-FIPA.

Más allá de las características básicas de la simple transferencia de mensajes entre agentes formalizada por el Lenguaje de Comunicación de Agentes (ACL), existe un conjunto de protocolos de los que MOSAIC puede sacar provecho. Entre ellos, los Protocolos de Interacción de IEEE-FIPA permiten implementar parte de los diálogos ente los Agentes ”Petitioner” y ”Contributor”. Adicionalmente, MOSAIC también pretende aplicar y aprovechar aquellos estándares relacionados con la transferencia de datos clínicos, como son los protocolos ISO/HL7 21731, ISO 13606 y DICOM.

En Fig. 2 se muestra el lugar que ocupa MOSAIC dentro de la pila de protocolos de sistemas de Agentes y de transferencia de datos clínicos.

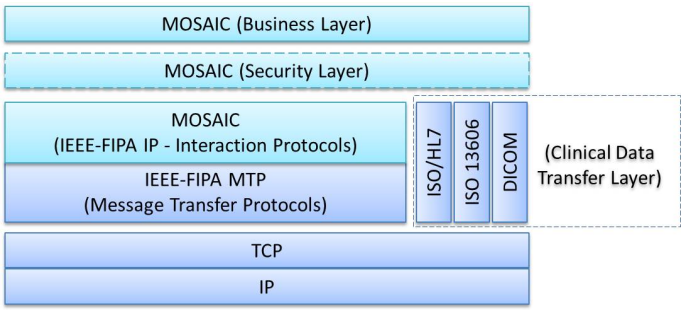


Fig. 2. Pila de los principales protocolos en los que se sustenta MOSAIC

IV. Diseño del Protocolo

Esta sección describe el comportamineto de los Agentes responsables de proveer datos a la red (”Contributor Agents”: CA) y de aquellos que solicitan datos de la red (”Petitioner Agents”: PA) con el objetivo de establecer acuerdos multilaterales entre ellos y lograr con ello el objetivo final de los nodos peticionarios.

A. Petición y recolección de datos: El ”Petitioner Agent”

Después de recibir la petición del usuario, con los detalles del conjunto de datos a recoger de la red, el ”Multicast Petitioner Agent” (PA) consulta las Páginas Amarillas a través del ”Yellow Pages Agent” para saber cuáles son los nodos de la red con ”Contributor Agents” (CA) activos. Una vez identificados, el PA lanza a todos los CA un mensaje con los detalles del conjunto de datos que está buscando, así como la petición de acceso a aquellos nodos que tengan datos que puedan conformar parte de dicho conjunto buscado. El PA se espera a recibir mensajes de los CAs indicando la existencia de los datos solicitados, a los que se podría acceder ya sea con o sin condiciones.

Cuando se hayan identificado un conjunto de opciones para acceder a los datos, un conjunto de agentes (uno por cada opción) será lanzado y, en paralelo y de forma autónoma, explorarán la posibilidad de recolección de estos datos. Cada uno de estos ”Unicast Petitioner Agents” (UPA) preguntará cuáles son las condiciones para acceder a los datos. Para los nodos que no tengan ninguna restricción de acceso al conjunto de datos solicitado y lo ofrezcan de forma abierta, el proceso para la recolección de los datos disponibles será lanzada de forma directa. En caso que se precise cumplir alguna condición previa a la entrega de los datos, el PA preguntará al usuario si accepta o rechaza dichas condiciones y enviará la respuesta al CA del nodo que contiene los datos. Si las condiciones son aceptadas y se cumplen, el PA procederá con la recolección de los datos. Finalmente, es razonable esperar que una posible condición para permitir el acceso a los datos de un nodo sea la entrega de otro conjunto de datos que el nodo esté interesado en recopilar, caso en el que se pueden dar dos situaciones:

- **Intercambio de datos bilateral**. La condición del CA para permitir el acceso a sus datos es recibir un conjunto de datos disponibles en el Data Mart del nodo del PA. Este acuerdo depende exclusivamente del intercambio bilateral entre los nodos de los PA y CA.
- **Intercambio de datos multilateral**. La condición del CA para permitir el acceso a sus datos es recibir un conjunto de datos no disponibles en el Data Mart del nodo del PA. Esta situación fuerza al PA a buscar los datos necesarios para cumplir las condiciones de acceso en otros nodos de la red y la posibilidad de acceder o no a los datos del nodo del CA dependerá de posibles acuerdos multilaterales a realizar entre un conjunto de nodos.
  Se propone crear un parámetro ”tiempo de vida” para permitir definir al usuario del protocolo un límite en el número de nodos que pueden participar en un acuerdo multilateral, evitando así exploraciones de la red inmanejables.

Un PA activado por otro Agente para resolver una condición de acceso a datos tiene que tener en cuenta que el destinatario final de los datos recogidos no es el nodo desde el que el Agente es lanzado y esto debe ser notificado al nodo candidato a ofrecer los datos. Adicionalmente, la petición de acceso a datos tampoco debe ser dirigida a todos los CA activos y tiene que excluir el nodo receptor final de los datos solicitados.

Cuando el acceso a los datos no pueda ser resuelto, el PA esperará un cierto tiempo para permitir una reconfiguración de la red y de su contenido. Pasado este tiempo, el PA iniciará una nueva petición. Este proceso se sucederá un número determinado de veces hasta que un contador que le ponga un límite a esta iteración llegue a cero.

Este comportamiento y el ciclo de vida del Agente se muestra en los Diagramas de Estados representados en Fig. 3 y 4.

- **P1: Iniciar Petición (Start Request)**. Este es el primer estado del Petitioner Agent y donde recibe la petición del usuario con los detalles del conjunto de datos a ser recopilados de la red. El PA consulta en las Páginas Amarillas cuáles son los nodos activos de la red con ”Contributor Agents” ejecutándose y envía a todos ellos un mensaje con los detalles del conjunto de datos que se pretende recopilar y la petición de autorización de acceso

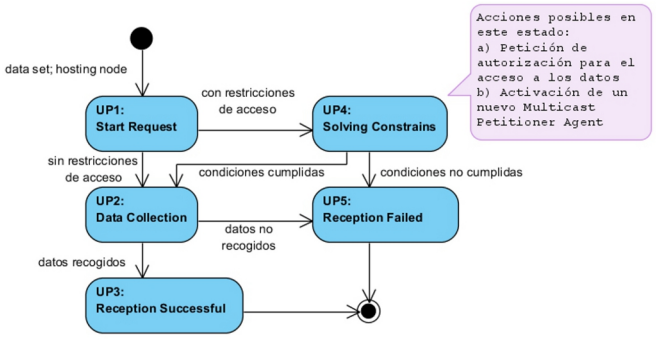


Fig. 3. Diagrama de Estados del Unicast Petitioner Agent

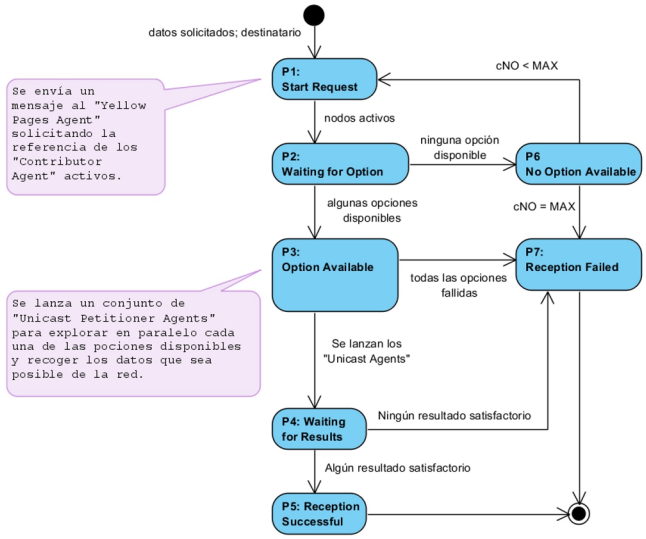


Fig. 4. Diagrama de Estados del Multicast Petitioner Agent

a aquellos que pueden tener parte de ese conjunto de datos. Después de ejecutar estas acciones el Agente pasa al estado P2.

- **P2: Esperando Opciones (Waiting for Option)**. En este estado, el PA espera la recepción de mensajes de los CAs indicándole la existencia de los datos solicitados, a los cuales se podría acceder con o sin condiciones previas, dependiendo de la configuración de cada nodo. Un temporizador (tWO) indicará el tiempo de espera en el que el Agente se mantendrá en este estado. Una vez expirado el tiempo de este temporizador, en caso que exista al menos una respuesta de algún CA, el Agente saltará al estado P3 (Opción Disponible). Si no hay respuesta por parte de ningún CA, pasará al estado P6 (Sin Opcion Disponible).
- **P3: Opción Disponible (Option Available)**. Cuando un conjunto de opciones es localizado, un conjunto de Agentes (uno por opción) será lanzado y en paralelo y de forma autónoma explorará cada una de las posibles opciones de recolección de los datos. Estos nuevos Agentes toman el nombre de ”Unicast Petitioner Agents” (UPA) y serán lanzados con peticiones expresas de explorar el posible acceso a los datos de un nodo específico. Después de lanzar el conjunto de UPAs el Multiast PA salta al estado P4 (Esperando Respuestas).
- **P4: Esperando Respuestas (Waiting for results)**. Este es el estado donde el PA espera por los resultados de los UPAs lanzados en el estado anterior. Un nuevo temporizador (tWR) determinará el tiempo máximo de espera en el que el Agente estará esperando por los resultados de los UPAs. Cuando expire tWR o bien todos los resultados de los UPAs hayan sido recibidos, el Agente examinará dichos resultados para decidir su acción final. En caso que no haya ningún resultado positivo de ningún UPA, el Agente saltará al estado P7 (Recepción Fallida). De otro modo, el Agente saltará al estado P5 (Recepción Exitosa).
- **P5: Recepción Exitosa (Reception Successful)**. Este es el estado final del Agente para peticiones de acceso a datos que resulten exitosas. El PA notifica al usuario el resultado y finaliza.
- **P6: Sin Opción Disponible (No Option Available)**. En este estado el Agente espera un cierto tiempo para así permitir una reconfiguración de la red y su contenido. Este tiempo es definido por el temporizador tNOA. Después de este tiempo, el Agente vuelve al estado P1 (Iniciar Petición). Esto puede suceder un número determinado de veces, hasta que un contador (cNOA) que pone límite a esta iteración llegue a zero. El contador cNOA se inicializa con un valor que decrece en una unidad cada vez que el Agente pasa por este estado.
- **P7: Recepción Fallida (Reception Failed)**. Éste es el estado final cuando no se ha localizado ningún nodo con los datos solicitados o bien aquellos que los tenían han impuesto condiciones para su acceso que no han podido satisfacerse. El PA falla en su objetivo de acceder a los datos, notifica su fracaso al usuario, y finaliza.

El Unicast PA (UPA) es activado de forma automática por un PA para gestionar una opción activa para acceder a un determinado conjunto de datos en un nodo. Recibe la petición de acceso al conjunto de datos, con la referencia del nodo en el que este conjunto de datos se encuentra, y la referencia del nodo destinatario al que se pretenden transferir estos datos. El UPA negociará con el CA del nodo que tiene los datos los derechos de acceso y eventualmente los recogerá si las condiciones son satisfechas.

- **UP1: Iniciar Petición (Start Request)**. Este es el primer estado del UPA y en el que éste recibe la referencia del conjunto de datos a recoger, la referencia del nodo donde estos residen, la referencia del CA con el que negociar los derechos de acceso, y la referencia del destinatario final de los datos (que puede ser tanto el nodo peticionario como un tercero en caso que estos datos sean necesarios para cumplir una condición parte de un acuerdo multilateral. Si no hay restricciones de acceso, el Agente vuelve al estado UP2 (Recolección de Datos). En caso contrario, el Agente salta al estado UP4 (Resolviendo Restricciones).
- **UP2: Recolección de Datos (Data Collection)**. En este estado todas las condiciones, en caso que existan, se han satisfecho y el Unicast PA es autorizado a recoger el conjunto de datos. La trasferencia de los datos se realiza aquí y si se completa con éxito, el Agente salta al estado

UP3 (Recepción Exitosa). En el caso de que por algún motivo la transferencia de datos no pueda concluir con éxito, el Agente salta al estado UP5 (Recepción Fallida).

- **UP3: Recepción Exitosa (Reception successful)**. Éste es el estado final del UPA cuando la recolección de datos finaliza con éxito. Se envía un mensaje al PA notificando la ejecución satisfactoria de la petición de recolección de datos y el Agente finaliza.
- **UP4: Resolviendo Restricciones (Solving Constrains)**. Éste es el estado donde el Agente llega cuando hay restricciones para el acceso a los datos y condiciones a cumplir. En caso que una condición sea la de proveer otro conjunto de datos, el Agente consultará si estos están disponibles en su propio DataMart, en cuyo caso la autorización para ser entregados al CA será solicitada al usuario. Si los datos no pueden ser obtenidos localmente, un nuevo Multicast PA será lanzado. El Multicast PA debe saber si su activación se debe a una necesidad de resolver una restricción de acceso, o no. Si es así, se debe evitar enviar una petición de acceso al CA que podría estar activo en el nodo destinatario final de los datos.
  Cuando el conjunto de datos está disponible, su entrega se realizará lanzando un Unicast Contributor Agent (UCA). Las respuestas de los usuarios y si son necesarias, también de los nuevos PA o de los UAC se gestionan en este estado. En caso que todas las restricciones sean satisfechas, el Agente salta al estado UP2 (Recolección de Datos). Si alguna condición necesaria para obtener el permiso de acceso a los datos no puede ser resuelta, el Agente salta al estado UP5 (Recepción Fallida).
- **UP5: Recepción Fallida (Reception failed)**. Este es el estado final del Unicast PA cuando por alguna razón la recolección de datos no se ha podido resolver de forma satisfactoria. Se envía un mensaje al PA notificándole el fracaso en la ejecución de su petición y el Agente finaliza.

B. Contribución y entrega de datos: El "Contributor Agent"

El usuario responsable del DataMart en un determinado nodo lanza el "Multicast Contributor Agent" (CA) cuando quiere poner un conjunto de datos a disposición de otros usuarios de la red. El usuario indica también al CA cuáles son las condiciones de acceso que deben ser cumplidas antes de autorizar el acceso a los datos. Después de su activación, el CA publica en las Páginas Amarillas su existencia y espera a recibir peticiones de acceso de los PAs activos. El diagrama de estados del CA se muestra en Fig. 5 y se detalla a continuación.

- **C1: Iniciar Contribución (Start Contribution)**. Este es el primer estado del Contributor Agent (CA). Recibe del usuario la referencia del conjunto de datos disponible y de las restricciones de acceso a estos datos, que tendrán que ser cumplidas para poder acceder a ellos. En este estado el Agente publica en las Páginas Amarillas su existencia y pasa al siguiente estado, C2 (Esperando Petición).
- **C2: Esperando Petición (Waiting for Request)**. Como su nombre indica, el CA se mantiene en este estado a la espera de posibles peticiones de acceso que le lleguen de

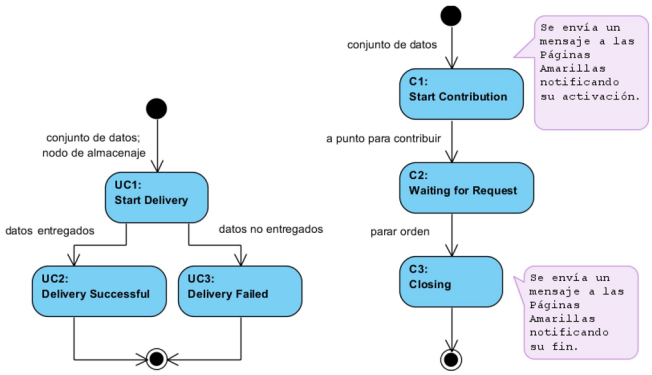


Fig. 5. Diagrama de Estados de los Multicast (izquierda) y Unicast (derecha) Contributor Agents

la red. Cuando llega una nueva petición, el CA lanza un nuevo Agente para procesar la solicitud con el objetivo de validar el cumplimiento de las posibles condiciones de acceso previas a la entrega de los datos y de ejecutar la entrega de los datos en caso que las condiciones sean satisfechas. Este nuevo Agente se denomina "Unicast Contributor Agent" (UCA).
El CA recibe el resultado de la ejecución de cada uno de los UCA y notifica de forma periódica al usuario sobre estos resultados.
Una petición de parada del usuario implica saltar al estado final C3 (Finalizar Contributor Agent). De todas formas, antes de parar el Agente, éste esperará un tiempo para permitir la finalización de los UCA activos. Una vez este tiempo de espera finalice, y en caso que algún UCA permanezca aún activo, será forzado a finalizar antes que el Agente salga de este estado.

- **C3: Finalizar Contributor Agent (Closing CA)**. Éste es el estado final del CA en el que el Agente notifica a las Páginas Amarillas su finalización.
- **UC1: Iniciar Entrega (Start Delivery)**. Éste es el primer estado del Unicast Contributor Agent (UCA), en el que recibe la referencia del conjunto de datos a ser entregados y la referencia del destinatario final donde estos datos serán transferidos, incluyendo también la referencia del CA con el que negociar las condiciones de acceso.
  Si la entrega de los datos concluye de forma satisfactoria, el Agente salta al estado final UC2 (Entrega Satisfactoria). En caso contrario, si la entrega de datos no se ha resuelto por algún motivo, el Agente salta al estado final UC3 (Entrega Fallida).
- **UC2: Entrega Satisfactoria (Delivery successful)**. Éste es el estado final del UCA cuando la entrega de los datos concluye de forma satisfactoria. Un mensaje notificando el éxito de la transferencia de datos se envía al UPA y el Agente finaliza.
- **UC3: Entrega Fallida (Delivery failed)**. Éste es el estado final del UCA cuando por algún motivo la entrega de los datos no se ha resuelto de forma satiisfactoria. Un mensaje notificando el fracaso de la entrega se envía al UPA y el Agente finaliza.

C. Riesgo y prevención de bucles

Un bucle puede aparecer cuando durante una búsqueda multilateral de un cierto conjunto de datos, un PA realiza una petición a un CA en la que i) El mismo conjunto de datos al mismo CA fué solicitado anteriormente y ii) No haya habido ningún cambio en las condiciones para acceder a los datos.

A pesar que la participación de un CA en un acuerdo multilateral más de una vez, en diferentes partes de la cadena, es posible y permitido, los bucles pueden ser detectados y prevenidos. Para ellos, tanto el CA como el PA tiene que actuar como sigue:

- Cada CA creará un recipiente con las referencias de las peticiones de datos activas, incluyendo el identificador de cada petición, el identificador del PA que realiza la petición, y la referencia del conjunto de datos solicitados.
- Cuando un CA recibe una petición recurrente (mismo PA y mismo conjunto de datos), un posible bucle se ha detectado y enviará un mensaje de alerta ("Loop Alert") al PA.
- Cuando un PA reciba un mensaje "Loop Alert" no enviará ningún nuevo PA para resolver las posibles condiciones de acceso y si los datos solicitados al CA no son accesibles, enviará de forma inmediata una respuesta negativa a la petición que se le ha realizado.

D. Seguridad

Diferentes aspectos relacionados con la seguridad de las transacciones y de protección de los datos clínicos en el protocolo MOSAIC han de ser considerados.

- **Características básicas de seguridad**
  Garantizar la identidad de los nodos (autenticidad), y la integridad y confidencialidad de los datos enviados, son las características básicas de seguridad que provee el protocolo SSL a través de las técnicas de clave asimétrica, que ha de formar parte intrínseca de MOSAIC.
- **Derechos de acceso y trazabilidad de los datos**
  Los datos transferidos a un nodo han de marcarse para así poder identificar el destinatario al que se han entregado y para el que se ha autorizado el acceso a ellos. Esto permite identificar el origen de los datos en caso que sean encontrados en cualquier otro lugar, diferente de aquel para el que se ha permitido el acceso. La incorporación de algoritmos de "fingerprinting" en MOSAIC garantizará esta funcionalidad.
  Una característica importante a incorporar es aquella a través de la cual la eliminación de las marcas de "fingerprinting" corrompan la calidad de los datos de tal forma que resulten inservibles.
- **Intercambio justo o "Fair exchange"**
  Cuando el proceso de negociación implica el intercambio de datos, MOSAIC ha de garantizar que los datos enviados pueden ser tan sólo accedidos y utilizados si los recibidos pueden ser a la vez accedidos y utilizados. Técnicas de "Fair Exchange" serán integradas al protocolo de MOSAIC para cubrir esta funcionalidad.

V. Los protocolos de Interacción de IEEE-FIPA

Los Protocolos de Interacción de IEEE-FIPA encajan en el diseño de parte de los diálogos ente los Agentes "Petitioner" y "Contributor". Las especificaciones de estos protocolos son las siguientes:

- **SC00026H** Request Interaction Protocol. [16]
- **SC00027H** Query Interaction Protocol. [17]
- **SC00028H** Request When Interaction Protocol. [18]
- **SC00029H** Contract Net Interaction Protocol. [19]
- **SC00030H** Iterated Contract Net Interaction Protocol. [20]
- **SC00033H** Brokering Interaction Protocol. [21]
- **SC00034H** Recruiting Interaction Protocol. [22]
- **SC00035H** Subscribe Interaction Protocol. [23]
- **SC00036H** Propose Interaction Protocol. [24]

Estos protocolos nos permiten construir el diálogo entre los agentes de MOSAIC de forma estandarizada. Estas especificaciones definen la comunicación entre un tipo de agentes llamados "Initiator" (equivalentes a los agentes Petitioner en MOSAIC) y otro tipo de agentes llamados "Participant" (equivalentes a los agentes Contributor en MOSAIC). En un diálogo basado en cualquiera de los protocolos de interacción, el agente Initiator asignará un parámetro a la comunicación que servirá de identificador único en todos los mensajes ACL, permitiendo identificar aquellos que forman parte de la misma conversación.

De entre estos protocolos, para su uso en MOSAIC, destaca el "Iterated Contract Net" (ver Fig. 6), pero haremos un breve repaso a las funciones de aquellos que mejor encajan en alguna parte del protocolo de MOSAIC.

A. Request Interaction Protocol

Este protocolo permite a un agente Initiator pedirle a un agente Participant que ejecute alguna acción. El agente Participant procesará la petición y tomará la decisión de ejecutar o no la acción asociada, y en caso que la decisión sea de aceptar la petición, ejecutará la acción e informará al Initiator de si la ejecución se ha realizado con éxito y de su resultado.

B. Query Interaction Protocol

El agente Initiator envía una petición al agente Participant. Esta petición puede ser del tipo "query-if" o del tipo "query-ref". La primera a ser utilizada si se quiere saber si cierta afirmación es cierta o falsa, y la segunda para preguntar sobre algún objeto. El agente Participant analiza la petición y decide si aceptarla o no. En caso que la acepte responde con un cierto o falso si la pregunta correspondia a la veracidad o falsedad de una sentencia o bien cualquier otro resultado y la referencia al objeto motivo de la pregunta.

C. Request When Interaction Protocol

En este protocolo, el agente Initiator envía la petición de ejecutar cierta acción al agente Participant cuando cierta precondición se cumpla. Si el Participant entiende la petición y no la rechaza de entrada, esperará a que se cumpla la condición y cuando esto suceda, intentará ejecutar la acción y notificar al agente Initiator de su resultado.

Si a pesar de la aceptación inicial el Participant no puede ejecutar la acción una vez se cumplan los prerequisitos, enviará un mensaje de fallida al Initiator. En caso que pueda ejecutar la acción, notificara de ello al agente Initiator, enviándole acto seguido el resultado de la ejecución de la acción.

D. Contract Net Interaction Protocol

El Agente Initiator solicita 'm' propuestas ("call for proposlas", cpf) a un conjunto de agentes Participant, que son vistos como posibles entidades "a contratar". Los agentes Partcipants receptores de estas propuestas generaran 'n' respuestas, de las que 'j' corresponen a propuestas para ejecutar la tarea. Estas respuestas positivas incluyen las condiciones bajo las que el Participan confirmaria la decisión. Los Participant pueden también rechazar el envio de ninguna propuesta al Initiator. Una vez finaliza el deadline, el Initiator evalua las j propuestas recibidas y selecciona los agentes Participant para realizar la tarea (uno, varios o ninguno). El Initiator enviará un mensaje de aceptación a los Participants elegidos y otro de rechazo a los que haya descartdo. Cuando el Initiator acepte la propuesta, el Participant se compromete a cumplir el pacto. Finalmente, una vez el Participant haya resuelto la tarea, envía un mensaje al Initiator informándole de su final.

E. Iterated Contract Net Interaction Protocol

Como se muestra en Fig. 6, el agente Initiator (equivalente al agente Petitioner en MOSAIC) envía un conjunto de "call for proposals" (cpf) a un conjunto 'm' de agentes tipo Participant (equivalentes a los agentes Contributor en MOSAIC). De los 'n' agentes Participant que responden, algunos (j) rechazan la petición y otros (k) la aceptan (siempre que se cumplan ciertas condiciones). Los 'k' agentes que aceptan explorar la petición, envían las propuestas (o condiciones) al Initiator para su aceptación definitiva. El Initiator puede i) acceptar la propuesta 'p' recibida de algún Participant y rechazar el resto, o bien ii) repetir el proceso enviando una nueva "cpf" revisada al Participant, procurando mejorar el acuerdo y esperando nuevas propuestas (o condiciones) que mejoren la anterior. El proceso termina cuando el Initiator rechaza todas las propuestas y no lanza más cpf, o cuando acepta una o más propuestas de los Participants, o cuando todos los Participants rechazan todas las propuestas.

F. Los protocolos de Interacción en MOSAIC

Los protocolos de Interacción más sencillos pueden ser integrados dentro del protocolo propuesto para MOSAIC pero ninguno de ellos cubre al completo las necesidades de MOSAIC. Existe la posibilidad de implementar el protocolo MOSAIC integrando aquellos protocolos de interacción que resuelven partes del diálogo y complementar el resto con mensajes ACL independientes, o bien se puede diseñar una extensión del protocolo de interacción que más se asemeje al diseño de MOSAIC para que cubra todas las funcionalidades del sistema. Estas opciones no son excluyentes pero para la implementación del sistema se toma la decisión de diseñar una versión adaptada del protocolo "Iterated Contract Net Interaction Protocol" que resuelva al completo los diálogos y comunicaciones entres agentes propuestas para MOSAIC.

VI. Trabajo presente y futuro

El desarrollo del sistema presentado en este artículo es una tarea en curso. Un primer diseño del protocolo MOSAIC y de la arquitectura del sistema está ya disponible y se ha identificado el entorno para su implementación. La evaluación del protocolo necesita de una red con una cantidad de nodos y conexiones significativa, que resulte suficiente para sacar

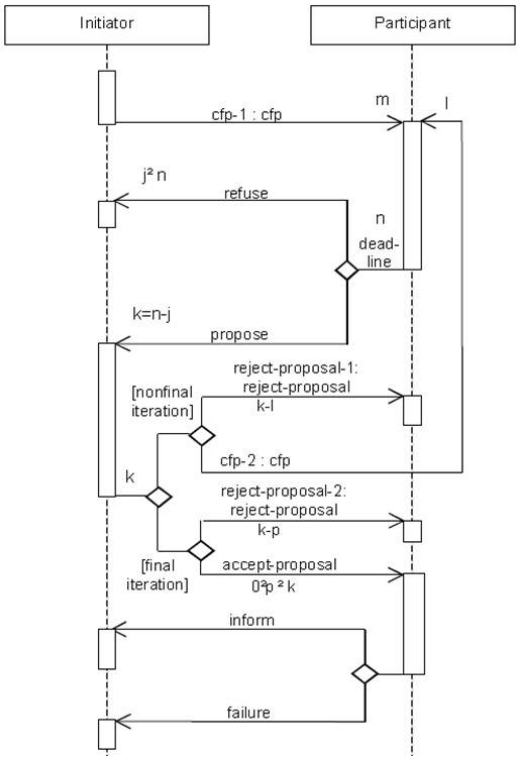


Fig. 6. Diagrama de secuencia del protocolo "Iterated Contract Net"

conclusiones válidas, con lo que se considera imprescindible incorporar una fase de simulación para ajustar los parámetros y temporizadores del protocolo e identificar los límites en sus valores.

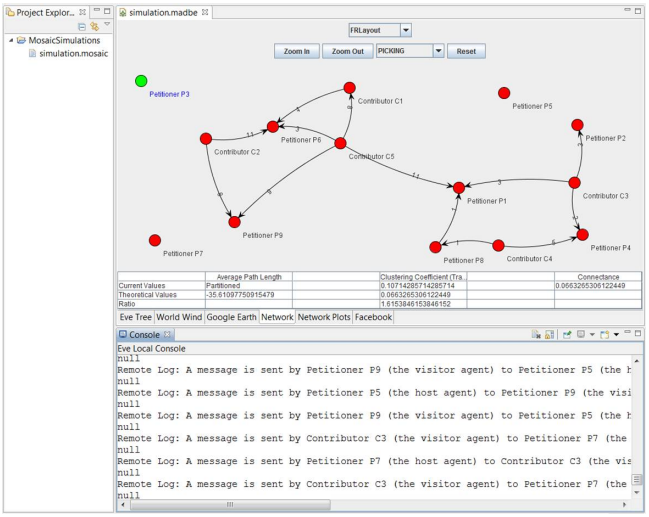


Fig. 7. Simulación de MOSAIC utilizando EveSim/MADBE

Para realizar las simulaciones de MOSAIC (ver Fig. 7) se está adaptando el código de la herramienta EveSimulator [25], un entorno diseñado para plataformas de Ecosistemas Digitales y la simulación de sistemas Multi-Agentes. EveSim está disponible bajo licencia de código libre y aporta un interfaz gráfico que resulta útil y muy atractivo para visualizar los resultados de la ejecución de MOSAIC. Esta implementación está basada en el entorno de Eclipse Modeling Framework (EMF) y desarrollada en lenguaje Java.

EveSim permite al usuario definir un conjunto de nodos de una red y para cada uno de ellos se le asigna un conjunto de propiedades. Para simular redes con una gran cantidad de nodos, éstos se crearán de forma automática adaptando la implementación del simulador. El protocolo está implementado dentro de una clase java (no se puede diseñar el protocolo desde la interfaz de usuario del simulador) y al ejecutarse se visualiza el conjunto de mensajes que se envían los nodos y de forma dinámica se establecen los enlaces entre ellos en la visualización gráfica de la red que presenta el simulador. Cada enlace tiene la forma de una flecha, indicado la dirección en la que fluyen los datos (de un Contributor Agent a un Petitioner Agent).

Una vez desarrollado el simulador de MOSAIC y en base a las simulaciones realizadas se hayan establecido los valores óptimos para sus parámetros y temporizadores, la implementación del sistema en un entorno real pasará por la integración de la plataforma Jade dentro del simulador, de tal forma que el simulador se convertirá en la interfaz de usuario del sistema, y facilitará tanto la activación de los Contributor y Petitioner Agents, como la visualización de los resultados de ejecución del protocolo y de los acuerdos multilaterales sugeridos. Trabajos futuros deberán estar dirigidos a la optimización del proceso de obtención de los acuerdos multilaterales.

VII. Conclusiones

El nivel de desarrollo de estándares y protocolos para la transferencia de datos clínicos, la disponibilidad de centenares de miles de repositorios distribuidos alrededor del mundo y la necesidad de compartir conocimiento entre los médicos y profesionales sanitarios para proveer mejores y más rápidos diagnósticos a los pacientes, inspira el desarrollo del sistema MOSAIC.

Las condiciones para acceder a un cierto conjunto de datos de la red puede, probablemente, incluir el requisito de intercambio de datos que, ocasionalmente, puede ser resuelto con acuerdos bilaterales entre los nodos, pero también puede necesitar de acuerdos multilaterales en los que participen un conjunto de nodos.

Se ha mostrado cómo Sistemas Multi-Agentes son un buen entorno para construir un sistema para compartir datos en un almacén de datos federado. Finalmente, la disponibilidad de los protocolos de interacción de IEEE-FIPA permiten proponer un protocolo de negociación a nivel aplicación para cubrir estas necesidades.

Agradecimientos

Damos las gracias a nuestros colegas del departamento ENTEL de la UPC, a la empresa MicroArt, y a los investigadores de los consorcios de los proyectos europeos HealthAgents, EcoBusiness y OPAALS, por las contribuciones científicas realizadas y las interesantes conversaciones mantenidas con algunos de ellos en relación al contenido de este artículo. Esta investigación ha sido co-financiada por el proyecto P2PSEC (TEC2008-06663-C03-01).

Referencias

[1] H. Oosterwijk. *DICOM Basics*. O Tech, third edition, 2008.

[2] Oleg S. Pianykh. Digital imaging and communications in medicine (dicom), 2008.

[3] ISO/TR 20514. Electronic health record – definition, scope and context, 2005.

[4] ISO/HL7 27931. HL7 version 3 - Reference information model, 2006.

[5] ISO 13606. Electronic health record communication – part 1: Reference model, 2008.

[6] George A. Komatsoulis, Denise B. Warzela, Francis W. Hartela, Krishnakant Shanbhaga, Ram Chilukuric, Gilberto Fragosoa, Sherri de Coronadoa, Dianne M. Reevesa, Jillaine B. Hadfielda, Christophe Ludetb, and Peter A. Covitza. cacore version 3: Implementation of a model driven, service-oriented architecture for semantic interoperability. *Journal of Biomedical Informatics*, 41:106–123, 2008. Article in Press. DOI: 10.1016/j.jbi.2007.03.009.

[7] National Cancer Institute and National Human Genome Research Institute. The cancer genome atlas. web site, 2005-2010. http://cancergenome.nih.gov/ (Último acceso: 30-06-2011).

[8] M. J. Boniface, T. A. Leonard, M. Surridge, S. J. Taylor, L. Finlay, and D. McCorry. Accessing patient records in virtual healthcare organisations. In *eChallenges 2005*, 2005.

[9] HOPE. Hospital platform for e-health. web site. http://sourceforge.net/projects/telemed/ (Último acceso: 30-06-2011).

[10] M. Lluch-Ariet, F. Estanyol, M. Mier, C. Delgado, H. González-Vélez, T. Dalmas, M. Robles, C. Sáez, J. Vicente, S. Van Huffel, J. Luts, C. Arús, A. P. Candiota Silveira, M. Julià-Sapé, A. Peet, A. Gibb, Y. Sun, B. Celda, M. C. Martínez Bisbal, G. Valsecchi, D. Dupplaw, B. Hu, and P. Lewis. *On the Implementation of HealthAgents : Agent-Based Brain Tumour Diagnosis*. Whitestein Series in Software Agent Technologies and Autonomic Computing. Birkhuser Basel, first edition, 2008.

[11] H. González-Vélez, M. Mier, M. Julià-Sapé, T. Arvanitis, J. García-Gómez, M. Robles, P. Lewis, S. Dasmahapatra, D. Dupplaw, A. Peet, C. Arús, B. Celda, S. Van Huffel, and M. Lluch-Ariet. Healthagents: distributed multi-agent brain tumor diagnosis and prognosis. *Applied Intelligence*, 30:191–202, 2009. 10.1007/s10489-007-0085-8.

[12] D. Isern, D. SÃ¡nchez, and A. Moreno. Agents applied in health care: A review. *International Journal of Medical Informatics*, 79(3):145 – 166, 2010.

[13] R. Annicchiarico, U. Cortés, and C. Urdiales. *Agent Technology and e-Health*. Whitestein Series in Software Agent Technologies and Autonomic Computing. Birkhuser Basel, first edition, 2008.

[14] E. Merelli, G. Armano, N. Cannata, F. Corradini, M. d'Inverno, A. Doms, P. Lord, A. Martin, L. Milanesi, S. Moller, M. Schroeder, and M. Luck. Agents in bioinformatics, computational and systems biology. *Brief Bioinform*, 8(1):45–59, 2007.

[15] J. Huang, N. R. Jennings, and J. Fox. An agent-based approach to health care management. *Int. Journal of Applied Artificial Intelligence*, 9(4):401–420, 1995.

[16] IEEE-FIPA SC00026H. Request interaction protocol specification, 2002. www.fipa.org/specs/fipa00026 (Último acceso: 30-06-2011).

[17] IEEE-FIPA SC00027H. Query interaction protocol specification, 2002. www.fipa.org/specs/fipa00027 (Último acceso: 30-06-2011).

[18] IEEE-FIPA SC00028H. Request when interaction protocol specification, 2002. www.fipa.org/specs/fipa00028 (Último acceso: 30-06-2011).

[19] IEEE-FIPA SC00029H. Contract net interaction protocol specification, 2002. www.fipa.org/specs/fipa00029 (Último acceso: 30-06-2011).

[20] IEEE-FIPA SC00030H. Iterated contract net interaction protocol specification, 2002. www.fipa.org/specs/fipa00030 (Último acceso: 30-06-2011).

[21] IEEE-FIPA SC00033H. Brokering interaction protocol specification, 2002. www.fipa.org/specs/fipa00033 (Último acceso: 30-06-2011).

[22] IEEE-FIPA SC00034H. Recruiting interaction protocol specification, 2002. www.fipa.org/specs/fipa00034 (Último acceso: 30-06-2011).

[23] IEEE-FIPA SC00035H. Subscribe interaction protocol specification, 2002. www.fipa.org/specs/fipa00035 (Último acceso: 30-06-2011).

[24] IEEE-FIPA SC00036H. Propose interaction protocol specification, 2002. www.fipa.org/specs/fipa00036 (Último acceso: 30-06-2011).

[25] T. Kurz, R. Eder, C. Ruecker, T.J. Heistracher, and F.A.B. Colugnati. A distributed agent-based approach for interdisciplinary collaboration. In *Digital Ecosystems and Technologies (DEST), 2010 4th IEEE International Conference on*, pages 203–208, Salzburg Univ. of Appl. Sci., Salzburg, Austria, April 2010. IEEE.

S^3OiA: propuesta de arquitectura para la interoperabilidad en la Internet de las Cosas

Mario Vega Barbas, Diego Casado Mansilla, Juan R. Velasco
Departamento de Automática, Grupo de Ingeniería en Servicios Telemáticos
Universidad de Alcalá
Campus Universitario, 28805, Alcalá de Henares, España.
{m.vega, diego.casadom, juanramon.velasco}@uah.es

***Resumen*—El trabajo presentado pretende realizar una aportación hacia la estandarización y la interoperabilidad, a través de sistemas abiertos y accesibles, de los diferentes actores y tecnologías en el Internet del Futuro. Para ello se presenta *S^3OiA*, una arquitectura orientada a servicios que permite integrar cualquier tipo de objeto o dispositivo en la Internet de las Cosas, y utilizar los recursos que ofrecen como sustrato para la generación automática de aplicaciones dinámicas. Éstas serán capaces de evolucionar y de adaptarse al contexto de ejecución mediante un sistema de gestión de dependencias, basado en eventos, entre dominios remotos. El contexto en el que se pretende aplicar el trabajo cubre todos los ámbitos susceptibles de involucrar al ser humano como la domótica, la inmótica o los *smart-grid*.**

***Palabras Clave*—Internet de las cosas, Internet del Futuro, arquitectura software, SOA, interoperabilidad, espacio inteligente, objeto cotidiano inteligente.**

I. Estado actual

La Internet de las Cosas (IdlC) presenta un escenario novedoso donde, la capacidad de cómputo y generación de información se incrementa considerablemente gracias a la inclusión de un número elevado de actores computacionales. Además, en este futuro modelo se pretende ofrecer un canal de comunicación común, Internet, a través del cual los citados actores puedan comunicarse y ofrecer recursos de forma transparente y uniforme.

Resulta casi imposible determinar la inmensidad de actores que formarán parte de la Internet del Futuro (IdF). Un informe reciente [1] augura más de quince mil millones de objetos intercomunicados en los próximos diez años, con una media de seis dispositivos por habitante. Esto es posible gracias a los avances en la industria de circuitos integrados, que permiten que dichos dispositivos tengan un menor tamaño gracias a la miniaturización de sus componentes [2] (p. ej. nanotecnología, microcomputadores y sistemas embebidos) con un menor coste y sin decrementar su capacidad computacional. Además, dichos dispositivos disponen, en la mayoría de los casos, de interfaces de comunicación que permiten enviar y recibir información, y realizar tareas más complejas mediante su asociación en sistemas distribuidos, sistemas de rejilla, redes malladas, *clusters*, etc. Un ejemplo tipo son las redes de sensores (WSN).

Debido a la naturaleza heterogénea de estos actores, las capacidades y recursos que ofrecen serán diferentes. Dicha heterogeneidad debe ser entendida desde el punto de vista computacional de cada objeto, de los protocolos empleados para la comunicación, de la representación de los recursos que ofrecen, y de los mecanismos usados para la interpretación de la información.

En el ámbito y literatura sobre la IdlC aparecen gran número de conceptos y términos de uso común. Por tanto, uno de los objetivos del este trabajo es presentarlos, y realizar un posicionamiento claro de los autores sobre dicha terminología, definiendo además, las necesidades y retos que deben ser abordados en el insólito modelo de comunicación. Realizado el análisis de los nuevos espacios de interacción, este trabajo contribuye a la comunidad investigadora presentando una aproximación de arquitectura software, que permita la integración, la comunicación y la interoperabilidad de esta cantidad ingente de nuevos elementos involucrados.

El trabajo se organiza de la siguiente manera: la sección II representa un primer posicionamiento sobre conceptos tales como *espacio inteligente* y *objeto cotidiano inteligente*; la sección III, muestra la visión del nuevo modelo de interacción que los autores han definido para el ámbito de aplicación de la IdlC; y la sección IV describe los retos que deben se extraen y deben ser abordados en la Internet del Futuro (IdF). En las secciones V y VI, se presenta y analiza la arquitectura S^3OiA comparándola con algunos trabajos relacionados. Finalmente, en la sección VII se realiza una conclusión sobre el texto y se tratan las líneas de trabajo que los autores consideran de interés para el futuro.

II. Posicionamiento

El análisis de la literatura en el ámbito de la computación ubicua, la inteligencia ambiental, y ahora en la IdlC, desprende un elevado uso de dos conceptos: *objeto inteligente* y *espacio inteligente*. En general, estas definiciones resultan vagas y dispersas, sin existir una convergencia clara hacia una interpretación común [3], [4], [5]. Por tanto, consideramos fundamental ofrecer nuestra visión sobre dichos términos con el fin de facilitar la comprensión del texto y de la arquitectura que se expone en la sección V.

II-A. Objeto cotidiano inteligente

El concepto de *objeto inteligente* debe ser extendido para abarcar a todos los objetos que son susceptibles de ofrecer cierta información (p. ej. localización física, origen, estado, utilización, etc.), llevando este concepto más allá del dispositivo electrónico como electrodomésticos, dispositivos móviles, o incluso aquellos productos de alto desarrollo tecnológico e industrial como los vehículos [6]. La idea que se persigue

ISBN: 978-84-694-5948-5

es dotar a elementos tales como mobiliario, ropa, comida, o materiales de construcción, de una identificación virtual única, capacidad de comunicación (mediante RFID, NFC, etc.), y/o de procesado de información (mediante transductores y microcontroladores embebidos). Con ello se consigue aumentar virtualmente objetos físicos mediante tecnologías de la información, e integrarlos así dentro del ecosistema de la IdlC con el fin obtener el máximo beneficio al unir el mundo físico con el mundo digital [7]. Se puede, por tanto, hablar de *objetos cotidianos inteligentes*, y es posible hacer un símil con el término *spime*, definido por B. Sterling [8]. De hecho, dichos objetos inteligentes son los actores principales para realizar el sueño que ya perseguía M. Weisser [9] y que ahora florece con la IdlC.

II-B. Espacio inteligente

La inclusión de cierta capacidad de cómputo dentro de los objetos de la vida cotidiana, no ofrece el objetivo final de obtener un sistema holístico e inteligente con capacidad de procesar y razonaar sobre las diferentes fuentes de información en pro del beneficio humano. Para conseguirlo, los objetos deberían integrarse dentro del entorno físico donde residen, conformar un entorno virtual, ser capaces de ofrecer sus recursos o funcionalidades, e interoperar de forma dinámica con otros objetos inteligentes. Siguiendo este patrón, se crearía un nuevo concepto de espacio de interacción virtual, donde confluyen átomos y bits, al que denominamos *espacio inteligente*.

La integración de estos espacios en Internet, sumado a las técnicas de agregación de información, amplían enormemente las fuentes de datos que son generadas en un dominio físico aislado por los objetos que lo conforman. Por tanto, se define un nuevo tipo de objeto al que denominamos *virtual*, que no posee presencia física en dichos espacios, pero que es capaz de comportarse como si la tuviera. Como ejemplos se pueden citar: los sensores virtuales [10], la agregación de funcionalidades ofrecidas por objetos físicos en una única virtual, fuentes externas de información (servicios Web), etc. Estos objetos virtuales no deben confundirse o ser incluidos en la definición de objeto cotidiano inteligente, debido a que su naturaleza es diferente y están únicamente orientados a extender la funcionalidad de los objetos físicos, mediante técnicas de composición, agregación o fusión de información.

En definitiva, los espacios inteligentes deben brindar entornos mínimos de interacción, que extiendan el principio de localidad, que sean dinámicos, en continua evolución y adaptación, y que sean proactivos en si mismos.

III. Sistemas centrados en el usuario

La evolución de Internet, desde el punto de vista del usuario, ha transformado la Web de ser un simple medio, donde entes especializados ofrecían información de forma unidireccional, a convertirse en una herramienta para la gestión directa de la información. De hecho, actualmente son los usuarios los encargados de generar y consumir la información, de forma cooperativa, y desempeñan un papel director. El principal problema que encontramos con este enfoque se concreta en la regla 90-9-1 [11], es decir, el 90 % de las personas consumen, un 9 % modifica y sólo un 1 % genera. En contraposición a esta regla, se sitúa el siguiente salto evolutivo que persigue la IdlC, donde se definen sistemas de interacción e información centrados en el usuario [12], y donde los usuarios no son los únicos consumidores y generadores de información sino cualquier elemento que sea capaz de hacerlo.

Por tanto, un aspecto importante que se debe afrontar al tratar la IdF, es el rol que debe desempeñar el usuario. La cantidad de elementos involucrados en los espacios inteligentes hace realmente complicado que el usuario cree una imagen mental de ellos. Esto, unido a su característica de ubicuidad impide que el ser humano sea capaz de organizar dichos sistemas sin limitar y degradar su potencial. En cambio, resulta sencillo definir una visión global de los mismos, mediante abstracciones, y controlar si el funcionamiento y los resultados obtenidos, son los esperados. De esta forma se puede definir un nuevo cometido para los usuarios, el rol de supervisión [13], que permita seguir el ciclo de proceso de estos sistemas sin condicionarlos y por consiguiente, maximizando sus capacidades. La IdlC persigue integrar esta nueva forma de procesar y generar la información en la vida diaria de las personas, con el objetivo de conseguir el máximo beneficio [14].

Los autores del trabajo presentado han definido un modelo de interacción no intrusivo desde el punto de vista de relación entre los usuarios y los sistema inteligentes: la *intención*. Una intención representa lo que un usuario desea realizar, define qué es lo que se quiere hacer y una serie de requisitos que deben ser cumplidos, en la medida de lo posible. El usuario no decide cómo llevar a cabo la acción o qué mecanismos son necesarios para su consecución, será el sistema, el entorno inteligente, el que decida esto. Los usuarios deben entender los espacios inteligentes como cajas negras con las que pueden interactuar, y al mismo tiempo, estos sistemas deben hacer partícipes a los usuarios mediante retroacciones.

El diagrama de la Figura 1 muestra este proceso de interacción. Los entornos inteligentes adaptan la información generada para construir reportes semánticamente comprensibles, ya que estos serán ofrecidos al usuario, con el fin de mantenerle informado de cómo se está desarrollando la actividad. Esta acción permite por tanto la supervisión activa de de los usuarios.

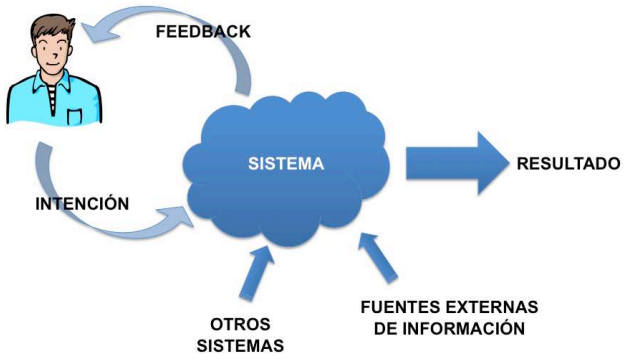


Figura 1. Diagrama de interacción basado en intenciones.

Toda la funcionalidad del sistema enfocada a resolver los problemas y restricciones presentadas en el proceso de interacción *persona - entorno inteligente* está contemplada y definida en un módulo concreto de la arquitectura propuesta en la sección V. El proceso de retroalimentación del sistema hacia

el usuario puede resultar complejo debido, principalmente, a las capacidades físicas y psíquicas de cada persona. Para solventar los problemas que puedan desprenderse de ello se propone el uso de interfaces de usuario inteligentes multi-dispositivo [15]. Además, mediante técnicas de combinación de dispositivos de entrada y salida, es posible lograr un nivel óptimo de interacción [16].

De igual forma que se tiene en cuenta el uso de objetos cotidianos para la composición de los entornos inteligentes, es necesario extender el concepto de interfaz de usuario inteligente a todos los objetos de la vida cotidiana susceptibles a reportar información: televisores, espejos, luces, timbres, teléfonos, audífonos, etc.

IV. Retos en la IdlC y la Interoperabilidad en Espacios Inteligentes

El nuevo modelo de comunicación e interacción de la IdF presenta múltiples retos y concepciones que han de ser abordadas por las futuras tecnologías. La introducción de un número incalculable de objetos con capacidades dispares dentro de las redes de comunicación requiere el plantearse una serie de necesidades y requisitos funcionales en las nuevas propuestas de arquitectura. Por tanto, es obvio que tanto fabricantes como alianzas y grupos de estandarización, se estén replanteando la modificación y adaptación de los protocolos actuales que permiten la integración, la comunicación y la interoperabilidad entre dispositivos.

A continuación se definen los principales retos de la IdlC. De cada uno de ellos se enumeran algunas problemáticas a abordar, y por último se presentan una serie de propuestas y pistas, que no harán sino conducir la definición y diseño de la arquitectura presentada en este trabajo de investigación.

IV-A. Integración en Internet

El principal objetivo que se persigue en el ámbito de la IdlC es doble, al igual que lo es el nombre del concepto. Por un lado la visión centrada en las Cosas, focaliza todo el esfuerzo en dotar al mayor número de objetos físicos de la capacidad de ser unívocamente identificados. Además, en el mejor de los casos, se proveería a los objetos de capacidad de procesamiento y de reportar información del entorno donde se encuentran.

La otra visión, centrada en Internet, argumenta que no es suficiente tener objetos identificados de forma única, si estos se muestran aislados y sin capacidad de relacionarse con sistemas. Por tanto esta visión, que está respaldada por la ITU [17], persigue ofrecer un canal de comunicación global y común, que integre y permita la comunicación entre los diferentes dispositivos, objetos, servicios.

De estas dos visiones se extráen una serie de retos y necesidades a abordar:

- Para conseguir que los objetos se integren en Internet, estos deberían, por si mismos, ser capaces de incorporar la pila de protocolos IP. Sin embargo, el inconveniente principal radica en que en la mayoría de los casos, los objetos presentan bajos recursos computacionales donde las memorias, el procesador y la batería son los factores de restricción.
- Actualmente, los dispositivos que interaccionan dentro de un entorno son, por lo general, homogéneos y comparten un canal de comunicación y de identificación común. Las nuevas arquitecturas han de tener en cuenta que dicho supuesto no será válido y, por tanto, deberán proponerse mecanismos de unificación y traducción de los diferentes protocolos de comunicación (p. ej. *ZigBee*, *BlueTooth*, *RFID*, *NFC*, *Z-Wave*, *DASH7*, etc.).
- Para que los objetos puedan integrarse fácilmente dentro de un dominio de red, son necesarios una serie de servicios a nivel de aplicación (p. ej. *DHCP*) y requieren de gestión experta. En ocasiones dichos mecanismos no están presentes.
- Si se asume que dispositivos heterogéneos son capaces de cooperar, su interacción se suele limitar un dominio o ámbito específico. Resulta pues compleja la fusión de diferentes dominios a través de mecanismos como la encapsulación, o las redes virtuales privadas, ya se que se requiere igualmente de un administrador de red.

Desde el IETF[1] hay un gran número de grupos en activo para conseguir adaptar la pila IP a dichas limitaciones. Algunos ejemplos son 6LoWPAN[2] y Roll[3]. Sin embargo, dichos protocolos sólo se centran en dispositivos programables con capacidades limitadas, olvidando una cantidad enorme de objetos, que son firmes candidatos a integrarse en los espacios de interacción (p. ej. dispositivos de protocolos propietarios).

Para dar solución a dichos requisitos, se requerirían mecanismos de tipo *Plug and Play* que permitiesen la integración de cualquier tipo de objeto en Internet, sin necesidad de gestión o administración por parte del usuario. Para ello, se han de usar objetos de mayor capacidad de cómputo, *gateways*, que han de formar parte de las arquitecturas de red como elementos integradores. Dichos *gateways* deben abstraer de las diferentes heterogeneidades que los objetos subyacentes presentan. Los *gateways* crearán instancias virtuales de objetos que no tienen capacidad de conectarse por sí mismos a Internet, dotandoles de direcciones IPv6 virtuales, un identificador único, y expondrán globalmente sus funcionalidades y recursos de manera uniforme y estandarizada. De igual forma, estos nodos de control han de implementar mecanismos de balanceo de carga, gestión de acceso concurrente y *caching* en favor de los objetos de pocos recursos.

IV-B. Interoperabilidad

Si se parte de la asunción de que millones objetos heterogéneos están integrados, y poseen capacidad de identificación y direccionamiento único en Internet, se daría un paso importantísimo hacia la homogeneización y la interoperabilidad, pero no suficiente. En ese supuesto, se conseguiría disponer *backbone*-IP donde, como mucho, se podrían desarrollar aplicaciones orientadas a la gestión de red mediante protocolos de tipo ICMP o SNMP.

Sin embargo, uno de los principales retos hacia la interoperabilidad en la IdlC, es que dos o más objetos puedan comunicarse de forma autónoma y cooperar (*Machine-to-Machine* - M2M).

[1] http://www.ietf.org/
[2] http://datatracker.ietf.org/wg/6lowpan/charter/
[3] http://datatracker.ietf.org/wg/roll/

Por tanto, se requieren tecnologías y nuevas arquitecturas de comunicación orientadas a servicios ya que, en la mayoría de los casos, lo que se busca es la información, los datos o recursos, en lugar del *host u objeto* que los ofrece.

IV-B1. Arquitecturas Orientadas a Servicios Dinámicos: Las arquitecturas orientadas a servicios (SOA) suelen estar restringidas a dominios únicos y/o privados. En éstas, las búsquedas de recursos para crear aplicaciones, suelen ser bajo demanda, y además dirigidas hacia un *broker* central siguiendo el paradigma *Publicación, Búsqueda, Binding* [18].

El reto de la IdlC es que dichos mecanismos sean descentralizados, autónomos, dinámicos y escalables hacia diferentes espacios de interacción. Una arquitectura de este tipo requiere, que en el momento que un nuevo actor se introduce en un espacio inteligente se ofrezcan: *i)* mecanismos de abstracción que gestionen las funcionalidades descubiertas, *ii)* métodos de especificación y exposición uniforme y *iii)* notificación distribuida a los objetos potencialmente interesados que conforman dicho espacio u otros remotos.

La principal problemática en el desarrollo de estas arquitecturas radica en la falta de generalización y de flexibilidad, pues las soluciones propuestas se centran en un contexto específico. A continuación se detallan algunos de las limitaciones y retos que se plantean en el marco de la IdlC:

- Los mecanismos de Descubrimiento de Servicios (DdS) suelen estar limitados al ámbito local. Además, encontramos dos metodologías que, o bien no son escalables, como aquellas que usan direcciones *multicast* para la notificación de eventos, ZeroConf[4] o SLP [19]; o bien presentan un servidor central, *broker*, que representa un único punto de fallo y de fácil sobrecarga - UDDI[5].
- Los objetos de capacidad limitada no ofrecen ni soportan mecanismos de DdS de alto coste computacional como pueden ser los utilizados por el protocolo UPnP[6].
- Al existir un gran número de dispositivos dentro de un entorno, se pueden ofrecer mecanismos diversos de DdS, y por tanto no existe la posibilidad de comunicarse ni cooperar.
- Algunas especificaciones ofrecen *proxies* de notificación hacia el exterior [20], pero en su mayoría sólo son válidos para alcanzar repositorios centralizados y previamente conocidos.

Estas restricciones hacen evidente la necesidad de diseñar una arquitectura SOA de notificación basada en eventos descentralizada, distribuida entre dominios, y que permita el desarrollo, despliegue y la evolución dinámica de aplicaciones.

IV-B2. Formato, Modelo de Datos y Semántica: La tecnología de mayor impacto en los últimos 20 años ha sido sin duda la Web. Ésta generó una aceptación generalizada debido a que permite la distribución de información, contenidos y recursos a través de un lenguaje común basado en hipertexto. Aún más con la posterior llegada de la Web 2.0, y su sencillo diseño centrado en la interoperabilidad, la cooperación y reutilización de los contenidos que la forman.

El objetivo y la necesidad fundamental para hacer realidad la IdlC se centra en la estandarización de protocolos. De ahí que los futuros diseños y soluciones planteadas no sean aislados, ni de aplicabilidad específica o reducida, ya que, a pesar de ser elegantes o presentar un buen rendimiento, no jugarían un papel integrador con los recursos y las fuentes de información que ahora conocemos.

A continuación se presentan algunas de las barreras que dificultan el poder llevar a cabo los objetivos y argumentos citados:

- Fabricantes y grupos de estandarización no llegan a un acuerdo para normalizar ni el formato de datos, ni los formalismos de representación de objetos, recursos y servicios ofrecidos.
- Existe un amplio número de plataformas implementadas para la IdlC, pero la mayoría de ellas ofrece soluciones propietarias o no sujetas a estándares.
- Los actuales servicios y recursos se crean por el usuario con el objetivo de ser usados e interpretados por éste. Sin embargo, el salto evolutivo en el paradigma de interacción, nos lleva a un aumento de las relaciones que no tienen a las personas como mediadores (comunicaciones M2M, Servicio-Máquina o Sistema-Máquina). Se necesitan pues, nuevos modelos de interacción. Se ilustra dicha problemática mediante un ejemplo: Imaginemos dos objetos, una lámpara con tecnología *Zigbee* y un interruptor *Bluetooth*, que desean interoperar. Suponiendo que pudiesen comunicarse mediante abstracciones (p. ej.: *IP+WADL*[7]*+HTTP*). La pregunta es obvia, ¿Cómo el interruptor podría interpretar que el tipo de datos a enviar con valor *"1"*, significa encendido? Desde el punto de vista del ser humano, parece obvio, pero ¿y entre máquinas?.
- Las ontologías y semántica generadas en diferentes proyectos de investigación son de dominio específico. Habría que plantear el tipo de formalismo semántico para definir los objetos y el entorno inteligente donde se relacionan. Además, no existe una taxonomía, ni un modelo relacional definido para los objetos inteligentes. De hecho, si este se realizase, se encontrarían conceptos comunes entre diferentes entornos de interacción y los objetos. Esto facilitaría sustancialmente la búsqueda y el encaminamiento de fuentes de información a cualquier dominio.

Por consiguiente, se requieren lenguajes estándar para el formato de datos y de recursos ofrecidos (por ejemplo: *EMML, IDL, WSDL o WADL*), así como mecanismos de representación e interpretación semántica de dicha funcionalidad (*RDFS, OWL, RIF*) adaptados a los nuevos actores. De igual forma se precisa un protocolo a nivel de aplicación, que sea estándar para el acceso a estos (e.g. *HTTP, XMPP*). Desde el IETF se ha propuesto un protocolo de aplicación para dispositivos con recursos limitados llamado CoAP[8]. Éste está basado en HTTP y que sigue el modelo arquitectónico REST.

IV-B3. Aplicaciones en la IdlC: Las aplicaciones desarrolladas en de la IdlC pueden abarcar campos de aplicación muy

[4]www.zeroconf.org
[5]http://uddi.xml.org/
[6]http://www.upnp.org/
[7]http://www.w3.org/Submission/wadl/
[8]http://datatracker.ietf.org/doc/draft-ietf-core-coap/

dispares y diversos. Estas han de tener en cuenta factores tales como la dinamicidad del entorno y las limitaciones computacionales y de comunicación de los diferentes actores. Los principales campos de aplicación tecnológica son: medidores de energía inteligentes y *smart grid*, eficiencia energética en infraestructuras, redes domóticas e inmoticas, trazabilidad de bienes físicos, automatización de procesos industriales, etc.

Algunos de los requisitos para conseguir dicho nivel de generación de aplicaciones flexibles se listan a continuación:

- Composición y agregación de funcionalidades, recursos y servicios independientemente de su localización, fuente o naturaleza, y del espacio inteligente desde el cual las aplicaciones sean creadas.
- Mecanismos de persistencia y tolerancia a fallos entre los diferentes registros de servicios, con el fin de asegurar un grado de disponibilidad aceptable de los recursos.
- Resolución de dependencias a nivel de servicio a través de los diferentes dominios de interacción. Es más, en caso de no poder resolver las dependencias necesarias, se deben ofrecer mecanismos de composición dinámicos que ofrezcan funcionalidades similares a las requeridas.

V. Arquitectura para la interoperabilidad en la IdlC

Los retos presentados y analizados en la sección previa, marcan la definición y diseño de la arquitectura presentada, bautizada como S^{3}OiA, la cual debe permitir la interoperabilidad de todos los actores de los espacios inteligentes en el contexto de la IdlC. Esta arquitectura cubre desde las necesidades inmediatas marcadas por la integración de elementos heterogéneos y de capacidades reducidas en Internet, hasta futuros requisitos caracterizados por el uso de mecanismos de interpretación de la información generada por ellos mismos u otros diferentes.

Actualmente no existen estándares para la interoperabilidad en la IdF, por lo que se ha tomado como interfaz de abstracción entre sistemas heterogéneos y base de la comunicación los servicios Web. Sin embargo, la arquitectura ofrece la posibilidad de extender las interfaces sintácticas, permitiendo mecanismos de interacción centrados en la comprensión y el razonamiento de la información y los recursos ofrecidos. Aún más, el posicionamiento realizado sobre el concepto de espacio inteligente requiere que se tenga en cuenta el contexto donde la información es generada, lo que hace que la propuesta se distancie sustancialmente de una *simple* interoperabilidad a nivel de servicio Web.

Debido a la capacidad limitada de la mayoría de los actores de la IdlC, se considera vital disponer de nodos de recursos computacionales mayores, *gateways*, que puedan realizar funciones de control y gestión de los recursos, a favor de los dispositivos subyacentes. Por todo esto, la propuesta se fundamenta en tres niveles de abstracción, como se observa en la Figura 2. Un nivel de comunicación físico, un nivel de exposición de servicios localizados en un mismo entorno y una capa virtual de interacción, donde los servicios son agrupados en espacios inteligentes.

La arquitectura propuesta se ha definido mediante un enfoque modular, donde la funcionalidad se agrupa en conjuntos mínimos, módulos, y se establecen interfaces de comunicación entre estos. Una representación gráfica de la arquitectura se

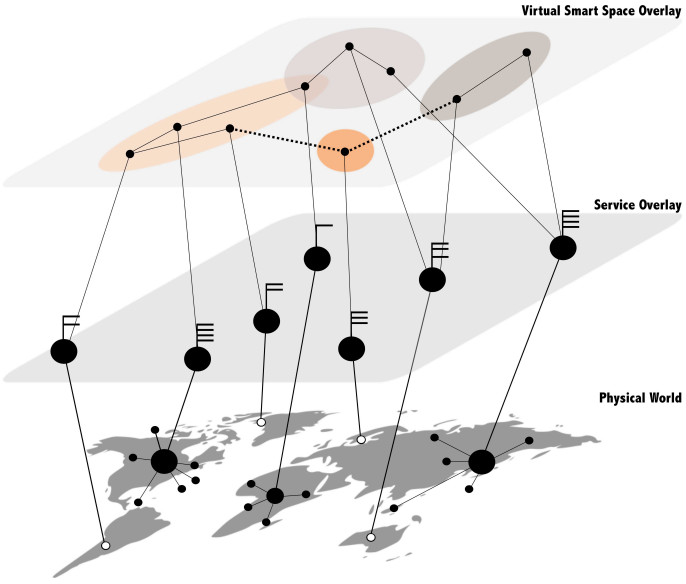


Figura 2. Diagrama de capas.

puede analizar en la Figura 3. Este diseño modular permite la evolución arquitectónica a través del desarrollo paralelo y la reutilización de sus elementos. S^{3}OiA es una arquitectura orientada a servicios dinámicos, lo que significa: *i)* un acoplamiento débil entre los servicios que se pueden llegar a ofrecer por medio de ésta, *ii)* un enlazado de dependencias en tiempo de ejecución (*late-binding*) y *iii)* reutilización de servicios y de recambio por otros de funcionalidad similar. Si además se contempla el uso de súper-nodos, *gateways*, que gestionen uno o más dominios de interacción, se podría diseñar una red distribuida de tipo *peer-to-peer* o federada, la cual permitiese la búsqueda de servicios automática, y la comunicación de eventos a través de dominios remotos en favor de las aplicaciones creadas. Los nodos involucrados deberían cooperar y comunicarse bajo el paradigma de paso de mensajes asíncrono publicación-subscripción.

Es posible concentrar los módulos que definen la arquitectura en cinco grupos funcionales:

- **Descubrimiento de dispositivos y servicios**. La arquitectura dispone de un conjunto de módulos centrados en realizar el descubrimiento de elementos que ofrecen y/o consumen algún tipo de servicio. Este grupo funcional contempla dos casos: comunicación entre dispositivos capaces de implementar protocolos como DPWS y UPnP; y comunicación entre estos y otros sin dicha capacidad, que han de ser abstraídos por *gateways*. De esta forma se permite que en los espacios locales, la comunicación sea lo más descentralizada posible, a través de una interacción máquina-máquina. Por tanto, se limita la funcionalidad de los nodos principales a tareas de abstracción y traducción de protocolos.
- **Semántica de primer nivel y acceso a servicios**. Conjunto de módulos encargados de completar las descripciones de las interfaces de acceso a los servicios descubiertos. Los metadatos utilizados para completar dicha descripción se extraen de la información de contexto local. Se generará pues una descripción formal de los servicios no estandarizados permitiendo su almace-

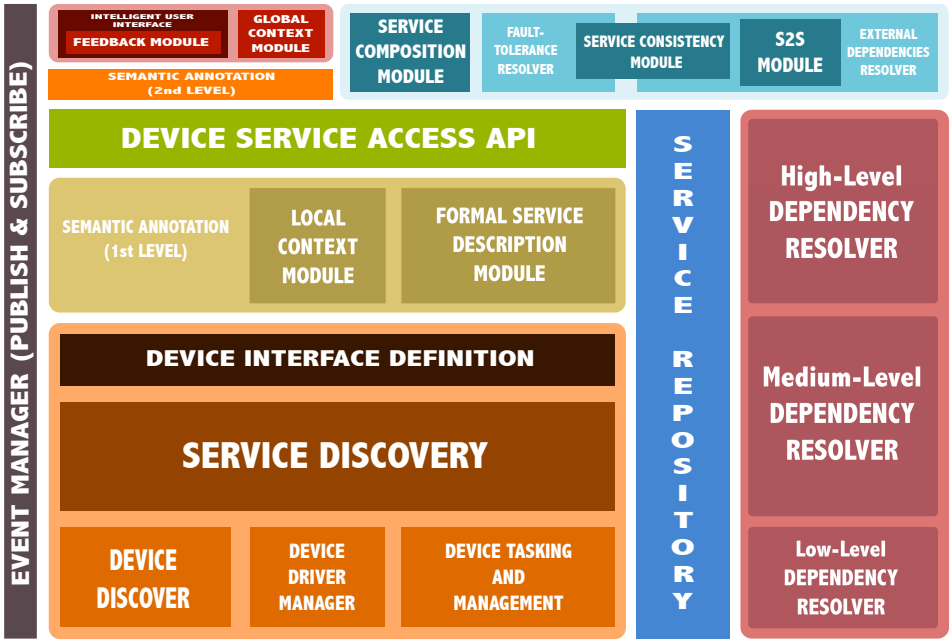


Figura 3. Arquitectura compleja para la interoperabilidad en la IdlC.

namiento y su traducción a servicio Web.

- **Repositorio de servicios y resolución de dependencias**. Este grupo se encarga principalmente de gestionar los servicios disponibles y de solventar las dependencias extraídas de la composición de aplicaciones en el entorno físico local (ver Figura 4, ejemplo *a)*). Además, como toda arquitectura orientada a servicios, se contempla la definición de un módulo de gestión de eventos. Por tanto se facilita la creación de aplicaciones y la composición de servicios por medio del paradigma de publicación y subscripción. Por su parte, el repositorio de servicios representa una instancia de almacenamiento de información del conjunto global distribuido.

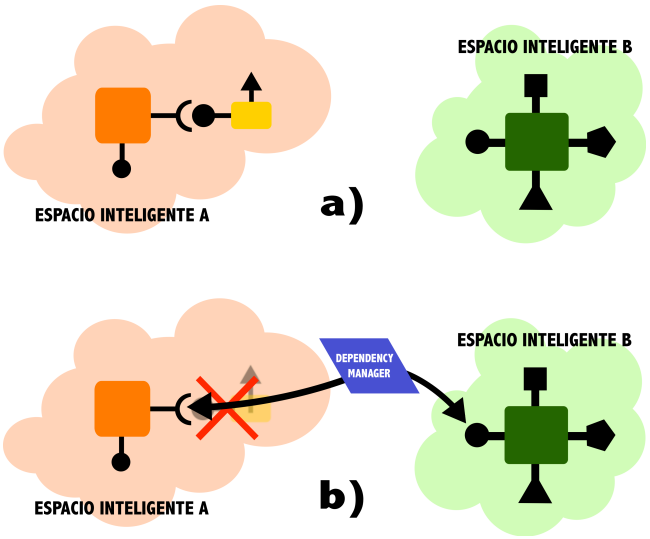


Figura 4. *a*)Resolución de dependecias en local y *b*)remoto.

- **Semántica de segundo nivel e interacción**. Los módulos agrupados en este conjunto se centran en la resolución de los problemas y necesidades de alto nivel cercanas al usuario. En concreto se definen aquí los módulos de *feedback* y contexto global. Estos módulos serán los encargados de gestionar el nuevo modelo de interacción descrito en la sección III. La semántica de segundo nivel es la introducida por el usuario en el proceso de retroalimentación y que resulta sumamente útil en el proceso de adaptación de la información en función del contexto global.
- **Composición, tolerancia a fallos y dependencias externas**. Finalmente, al mismo nivel que el grupo anterior se sitúan los módulos encargados de llevar a cabo la composición de servicios y la resolución de conflictos a nivel de entorno inteligente (*Virtual Smart Space overlay*, Figura 2). La arquitectura no es centralizada y por tanto, todas las dependencias no resueltas a nivel local deben ser elevadas a un espacio virtual, mediante un mecanismo de resolución distribuido (ver Figura 4, ejemplo *b)*). Por último, a dicho nivel, se presentan dos módulos encargados de hacer frente a las inconsistencias, desconexiones o fallos que otros nodos de la red distribuida pudieran generar.

La evolución de las TIC nos llevará a un escenario futuro donde todos los actores tendrán capacidad computacional suficiente para realizar descubrimiento y ofrecimiento de servicios a través de Internet. Por ello, la arquitectura presentada debe adaptarse, sin grandes costes, a dicha situación. Como se puede observar en la Figura 5, los módulos de descubrimiento pueden ser obviados sin perjudicar el funcionamiento global, así como los nodos de funcionalidad superior que hacen dichas funciones en favor de aquellos de pocos recursos. Las funcionalidades de alto nivel podrían ser ofrecidas por entidades software mediante mecanismos de *cloud-computing*. El diseño modular en el que se ha basado la definición de la arquitectura ha de facilitar dicha transición.

Figura 5. Arquitectura futura para la interoperabilidad en la IdlC.

VI. Otras propuestas hacia la Interoperabilidad

La arquitectura presentada no es la única que persigue objetivos similares y de hecho, en la literatura, encontramos un número amplio de plataformas que ofrecen interoperabilidad entre sistemas heterogeneos. Sin embargo, S^{3}OiA aborda ciertas problemáticas necesarias para la IdF como: dinamicidad de los recursos expuestos, evolución y adaptabilidad de las aplicaciones creadas, escalabilidad gracias a la integración de cualquier tipo de objeto, cooperación intra y entre dominios remotos (espacios inteligentes), etc. Por tanto, la propuesta es mucho más ambiciosa que una simple arquitectura orientada a servicios y ofrece características que otras soluciones no presentan.

Las plataformas y arquitecturas de mayor similitud a la propuesta, son aquellos sistemas basados en *middlewares* [21] de abstracción que facilitan el intercambio de información, de forma distribuida, entre sistemas heterogeneos y ubicuos. Sin embargo, dichas soluciones presentan una clara limitación a nivel de escalabilidad y aceptación, pues sólo están orientadas a dispositivos programables con capacidad de almacenar parte del software de abstracción (por ejemplo aquellas basadas en CORBA [22]). Otra limitación se desprende de su propósito específico, por ejemplo el ámbito del hogar con Jini y OSGi [23], [24] o el ámbito empresarial-industrial con aquellas soluciones basadas en SOAP-WS [25], [26]. Aún más, la mayoría de los *middlewares* restringen su aplicabilidad a plataformas de características *hardware* o *software* únicas - dispositivos de grandes recursos [27], sensores con sistemas operativos específicos [28], máquinas virtuales [29], [30], etc. Igualmente encontramos soluciones orientadas únicamente a la monitorización y gestión de redes de sensores utilizando APIs específicas[31].

Al no haber un estándar común se hace necesario crear mecanismos de traducción entre *middleware* diferentes si se desea que estos cooperen [32]. Se encuentran pocos trabajos prácticos que provean un traductor común y unificado entre las diferentes soluciones. Por tanto, S^{3}OiA pretende ofrecer una arquitectura estándar, abierta y de propósito general, que integre y reutilice todas las capacidades de Internet y los recursos y fuentes de información del entorno Web. En este sentido encontramos una propuesta centrada en interoperabilidad entre servicios Web, DPWS[9]. Sin embargo al igual que UPnP, utiliza el paradigma SOAP-WS, que además de no ser ligero, ni fácil de integrar en dispositivos de capacidades limitadas, está orientado hacia servicios empresariales. Al igual que DPWS, pero orientado a automatización de infraestructuras, está la especificación oBIX[10] que está considerada como nueva generación de *middleware* para el hogar, pero que no muestra mecanismos de búsqueda y cooperación dinámica entre servicios de espacios inteligentes remotos. S^{3}OiA permitirá la creación de aplicaciones genéricas y evolutivas, sobre servicios Web basados en el patrón REST, para cualquier tipo de ámbito (domótica, inmótica, automatización o *smart grid*) donde los usuarios puedan jugar un rol de supervisión activa.

[9]http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01
[10]http://www.obix.org/

VII. Conclusiones y trabajo futuro

El desarrollo de la Internet del Futuro y la evolución de las TIC hacia un paradigma más proactivo y ubicuo, exige enfocar los trabajos de investigación hacia un nuevo modelo de interacción entre dispositivos, cada vez más numerosos, y personas. En este sentido la estandarización es primordial, ya que es necesario especificar uniformemente qué define un espacio inteligente, cómo debe realizarse la comunicación dentro de dicho espacio, cómo se realiza la comunicación entre espacios remotos, y cómo se puede, en definitiva, dar soporte a todos los niveles de interoperabilidad. Para poder realizar dichos objetivos, se ha presentado el diseño de una arquitectura, dinámica, flexible y capaz de evolucionar y adaptarse a los avances de la IdlC. S^{3}OiA es una arquitectura de tres niveles orientada a la composición y cooperación ad-hoc de servicios distribuidos y dinámicos, que están uniformemente descritos. Para ello se han definido una serie de módulos de gestión de dependencias y *tracking* de servicios que permiten que las aplicaciones creadas continúen ejecutándose a pesar de cambios de contexto, o de cambios en los recursos utilizados. Las aplicaciones resultantes podrán ser creadas por y/o para el usuario haciendo uso de una serie de dispositivos con mayor capacidad de cómputo, *gateways*, que permiten la integración, la exposición uniforme y uso de cualquier tipo de objeto y sus recursos entre dominios remotos. Además los autores analizados en han ofrecido su propia visión y comprensión de términos y conceptos de uso común, como son los espacios inteligentes y los objetos cotidianos inteligentes.

La IdlC avanza a gran velocidad, y son muchos los grupos de interés, tanto académicos como industriales, quienes han tomado la iniciativa para realizar una verdadera interoperabilidad a escala global. A continuación se detallan algunos de los temas que los autores consideran de interés para futuras investigaciones:

- Se necesita un *esperanto* en la IdlC para soportar interoperabilidad entre diferentes dominios de interacción. Se deben definir modelos de datos comunes a múltiples aplicaciones y ámbitos.
- La semántica juega un papel vital para representar el mundo físico. Se debe estudiar que tipo de semántica ha de aplicarse a ciertos campos e infraestructuras de interacción. Será fundamental investigar en la relación semántica entre las aplicaciones creadas, los ámbitos donde se generan, y los recursos que las conforman.

- La movilidad de los objetos es casi total en los escenarios y dominios de la IdlC y por tanto ha de gestionarse la presencia de dispositivos y recursos ofrecidos de forma global.
- Estandarización en el diseño de *proxies* y *gateways* de propósito general lo que ayudaría a la interoperabilidad entre dominios de interacción globales.
- Los sistemas de la IdF requieren un componente de seguridad y privacidad altísimo. Estos deben ser compatibles entre los diferentes actores sin importar su naturaleza.

Agradecimientos

Este trabajo ha sido realizado al amparo del proyecto ITEA-2 No 2008005, Do-it-yourself Smart Experiences, financiado por el Ministerio Español de Industria y Comercio -Avanza-TSI-020400-2009-124.

Referencias

[1] J. Morrish, "Internet 3.0: the internet of things," Analysys Mason, Tech. Rep., 2010.

[2] G. E. Moore, *Cramming more components onto integrated circuits*. Morgan Kaufmann, 2000, pp. 56–59.

[3] G. Dalton, A. McDonna, and J. Bowskill, "The design of smart space: A personal working environment," *Personal and Ubiquitous Computing*, vol. 2, no. 1, pp. 37–42, 1998.

[4] I. Essa, "Ubiquitous sensing for smart and aware environments," *IEEE Personal Communications*, vol. 7, no. 5, pp. 47–49, 2000.

[5] R. Singh, P. Bhargava, and S. Kain, "State of the art smart spaces: Application models and software infrastrucutre," *ACM Ubiquity*, vol. 7, no. 37, pp. 2–9, 2006.

[6] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A surrey," *Comput. Netw.*, vol. 50, no. 15, pp. 2787–2805, 2010.

[7] D. Casado, M. Vega, and M. López, "Infraestructuras inteligentes en internet del futuro," in *1st Encuentro de Investigadores en Infraestructuras Inteligentes (EI3)*, Guadalajara, Spain, 2011.

[8] *Shaping Things*. MIT Press., 2005.

[9] M. Weiser, *The computer for the 21st century*. Morgan Kaufmann, 1995, pp. 933–940.

[10] S. Kabadayi, A. Pridgen, and C. Julien, "Virtual sensors: Abstraction data from physical sensors," University of Texas, Tech. Rep., 2006.

[11] J. Nielsen. (2006) Participation inequality: Encouraging more users to contribute. [Online]. Available: http://www.useit.com/alertbox/participation_inequality.html

[12] *Inmates Are Running the Asylum*. Sams Publishing, 2004.

[13] D. Tennnenhouse, "Proactive computing," *Communications of the ACM*, vol. 45, no. 5, pp. 43–50, 2000.

[14] M. Kuniavsky, *Smart Things: Ubiquitous Computing User Experience Design*. Morgan Kaufmann, 2010.

[15] M. Vega and J. R. Velasco, "Intelligent multi-device user interfaces," in *PECCS 2011 - 1st International Conference on Pervasive and Embedded Computing and Comunications Systems*, Vilamoura, Portugal, 2011.

[16] M. Maybury and W. Wahlster, "Readings in intelligent user interface," 1998.

[17] I. T. Unit, "Itu internet reports 2005: The internet of things," ITU, Tech. Rep., 2005.

[18] M. Papazoglou and W. van den Heuvel, "Service-oriented design and development methodology," *Int. J. of Web Engineering and Technology (IJWET)*, 2006.

[19] E. Guttman, C. Perkins, J. Veizades, and M. Day, "Service location protocol, version 2," United States, 1999.

[20] G. Moritz, E. Zeeb, S. Prüter, F. Golatowski, D. Timmermann, and R. Stoll, "Devices profile for web services in wireless sensor networks: adaptations and enhancements," in *Proceedings of the 14th IEEE international conference on Emerging technologies & factory automation*, ser. ETFA'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 43–50. [Online]. Available: http://portal.acm.org/citation.cfm?id=1740954.1740961

[21] T. Nakajima, K. Fujinami, E. Tokunaga, and H. Ishikawa, "Middleware design issues for ubiquitous computing," in *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*, ser. MUM '04. New York, NY, USA: ACM, 2004, pp. 55–62. [Online]. Available: http://doi.acm.org/10.1145/1052380.1052389

[22] F. M. Fernández, D. V. Alises, F. V. Molina, J. B. Romero, F. R. Calle, and J. L. López, "Embedding standard distributed object-oriented middlewares in wireless sensor networks," *Journal on Wireless Communications and Mobile Computing*, 2009.

[23] S. Microsystem, "Jini specifications v1.2. - jini network technology," 2010. [Online]. Available: http://www.sun.com/software/jini/specs/

[24] O. Alliance. (2010) Osgi service platform. [Online]. Available: http://www2.osgi.org/Specifications/HomePage

[25] C. Mauro, J. Leimeister, and H. Krcmar, "Service oriented device integration: An analysis of soa design patterns," in *43rd International Conference on System Sciences*, Hawaii, USA, 2010.

[26] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, "Interacting with the soa-based internet of things: Discovery, query, selection, and on-demand provisioning of web services," *IEEE Transactions on Services Computing*, vol. 3, no. 3, pp. 223–235, 2010.

[27] V. Dhingra, "How to connect non ip devices into upnp.v1 fabric," 2010. [Online]. Available: http://www.upnp.org/download/summitslides2003/01-13Echelon.ppt

[28] E. Souto, G. Guimarães, G. Vasconcelos, M. Vieira, N. Rosa, and C. Ferraz, "A message-oriented middleware for sensor networks," in *Proceedings of the 2nd workshop on Middleware for pervasive and ad-hoc computing*, ser. MPAC '04. New York, NY, USA: ACM, 2004, pp. 127–134. [Online]. Available: http://doi.acm.org/10.1145/1028509.1028514

[29] S. Michaelis, A. Wolff, and J. Shumutzler, "More-architecture and services," EU Project MORE, Dortmund, Germany, Tech. Rep., 2007.

[30] I. Marsá, "Seth: A hierarchical, agent-based architecture for smart spaces," University of Alcala, Tech. Rep., 2006.

[31] S. Kunz, T. Uslander, and K. Watson, "A testbed for sensor service networks and the fusion sos: towards plug and measure in sensor networks for environmental monitoring with ogc standars," in *18th World IMACS / MODSIM Congress*, Cairns, Autralia, 2009.

[32] C. Lee and K.-D. Moon, "Design of a universal middleware bridge for device interoperability in heterogeneous home network middleware," in *Consumer Electronics, 2005. ICCE. 2005 Digest of Technical Papers. International Conference on*, 2005, pp. 371–372.

Utilización de datos geográficos auxiliares para la optimización de caches espaciales

P. López Escobés, R. García Martín, J. P. de Castro Fernández, M. J. Verdú Pérez,
L. M. Regueras Santos, E. Verdú Pérez.
Laboratorio de Infraestructuras de Datos Espaciales (IDELab),
Universidad de Valladolid
Camino del Cementerio s/n, 47011 Valladolid.
plopesc@ribera.tel.uva.es, {ricgar, juacas, marver, luireg, elever}@tel.uva.es.

***Resumen*—Dados los problemas de escalabilidad de los servicios de mapas tradicionales, una de las alternativas más recurrente es la utilización de *caches* espaciales. El reto actual reside en sus técnicas de mantenimiento, que deberían tener en cuenta la correlación espacial existente entre las diferentes zonas geográficas, así como la relevancia de cada zona para ofrecer un mejor servicio. Una de las problemáticas existentes reside en la selección de qué zonas geográficas son las más relevantes, y de esta manera precargarlas para ofrecer un mejor servicio. Hasta ahora esta decisión recaía de forma directa en el administrador del sistema, sin embargo, a partir de información geográfica auxiliar se podrían definir ciertos parámetros de representatividad que pueden resultar de gran utilidad a la hora de decidir que zonas son más sensibles de ser solicitadas.**

***Palabras Clave*—Cache, GIS, WMS, WMS-C, WMTS, WFS, QoS**

I. Introducción

Los servicios web de mapas han proliferado de forma masiva en los últimos tiempos gracias a la aparición de numerosos proveedores. Gran cantidad de iniciativas tanto públicas como privadas han competido y aún lo siguen haciendo, para poder hacerse un hueco dentro de este campo. La popularidad de estos servicios, así como su acercamiento al público de forma masiva ha dado lugar incluso a un nuevo término, la neogeografía [1]. Para tratar de acotar la heterogeneidad de servicios y fomentar la interoperabilidad entre los mismos, distintos organismos como la OGC[1] (*Open Geospatial Consortium*) o la OSGeo[2] (*Open Source Geospatial Foundation*) han desarrollado distintos estándares y especificaciones. Gracias a ellos es posible conjugar datos de diferentes servicios de forma sencilla y transparente.

Uno de los estándares más extendido es el WMS (*Web Map Service*) [2], que permite la obtención de representaciones cartográficas a través de peticiones HTTP. Este servicio ofrece una gran sencillez y flexibilidad, dado que la cantidad de parámetros con los que se puede definir la visualización es grande y que los valores posibles para estos parámetros no están acotados. Esta flexibilidad es al mismo tiempo una de sus ventajas, así como un gran inconveniente, ya que las posibles peticiones se hacen infinitas y son difícilmente reproducibles. Por ello es necesario procesar todas y cada una de ellas al vuelo, con el consiguiente coste operacional y de tiempo que representa. Ante un gran número de peticiones, estos servidores ven mermada su QoS (*Quality of Service*), por lo que su escalabilidad queda en entredicho [3].

[1]http://www.opengeospatial.org
[2]http://www.osgeo.org

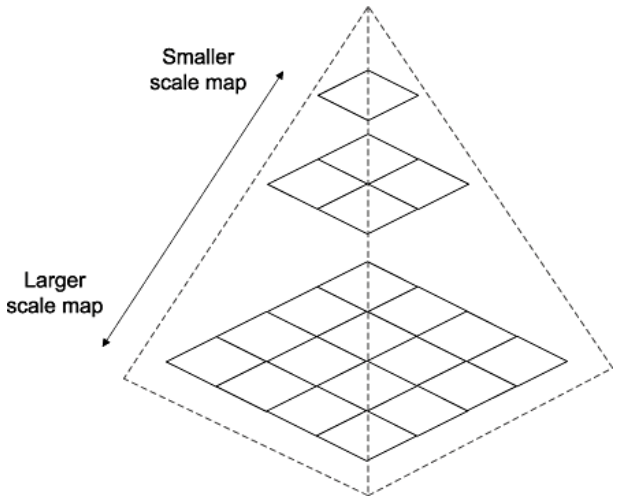


Fig. 1. Pirámide de escalas resultante de la aplicación de la rejilla para dividir el espacio geográfico en un conjunto discreto de elementos

Sin embargo, algunos servicios comerciales, como los de Google o Microsoft, han optado por ofrecer sus propios servicios e interfaces, alejándose de los estándares para tratar de ofrecer mejores servicios. Estas aproximaciones se basan en acotar tanto los parámetros de la petición, así como los valores que pueden tomar cada uno de ellos. El espacio geográfico está delimitado por diferentes rejillas predefinidas para cada uno de los niveles de zoom admitidos [4], [5], asemejándose a una estructura piramidal como la que se puede observar en la Figura 1. De esta forma es posible determinar todas las posibles peticiones que se pueden realizar al servicio. Al restringir las zonas geográficas a estas rejillas es posible pregenerar las visualizaciones correspondientes, denominadas teselas. De esta forma el sistema resulta más escalable y su rendimiento no se ve tan penalizada ante un gran número de peticiones.

La popularidad y los buenos resultados obtenidos por estos servicios teselados han fomentado la creación de nuevos estándares y recomendaciones, como WMTS (*Web Map Tiled Service*) [6] de la OGC o WMS-C (*Web Map Service Tile Caching*) [7] de OSGeo. Sin embargo, pregenerar y tener disponibles todas y cada una de las teselas de las distintas representaciones cartográficas sólo está al alcance de las grandes

empresas que disponen de ingentes cantidades de espacio de almacenamiento. El resto de proveedores, más modestos, deben enfrentarse a complejas decisiones relacionadas con el diseño y el mantenimiento de estos servicios.

En estos casos se suele optar por un diseño denominado *proxy web cache*, que se basa en colocar un dispositivo de *cache* entre el servicio final, en este caso un servidor WMS, y el cliente [8]. De esta forma, si la tesela solicitada se encuentra almacenada en la *cache*, no es necesario solicitarla al servidor WMS, por lo que la experiencia de navegación resulta más satisfactoria. Al mismo tiempo, se evitan congestiones en el servidor WMS, puede resolver otras peticiones más rápidamente.

Estas *caches* parciales presentan un mejor rendimiento a medida que se van poblando, por lo que resultaría interesante llevar a cabo un proceso de poblado automático de la *cache* previo a la publicación del servicio. El proceso mediante el cual las teselas se generan y *cachean* de forma automática se conoce como *seeding*. La ventaja de este proceso es que mejora en gran medida la experiencia de usuario. El inconveniente es que se trata de un proceso que consume más tiempo y recursos de almacenamiento de los estrictamente necesarios, ya que no se garantiza que todas las teselas pregeneradas vayan a ser requeridas posteriormente.

Este artículo pretende explorar nuevas alternativas que tratan de mejorar este proceso de *seeding*, haciéndolo más rápido y evitando precargar más teselas de las necesarias. Una de las más importantes es el autodescubrimiento de las principales zonas de interés de un mapa. A lo largo del documento se presenta la utilización de *features* o fenómenos geográficos directores de las consultas como método de descubrimiento. Estos elementos vectoriales puedan dar pistas sobre las zonas más populares de un determinado mapa, sin embargo no todas son igual de útiles dependiendo del servicio o del momento. Se realizará un estudio que tratará de definir distintos parámetros de calidad de estas fuentes de datos auxiliares que puedan ayudar a los administradores a seleccionar que capas deben utilizar como directoras para cada servicio.

El resto del documento se organiza de la siguiente manera: en primer lugar se describe más a fondo lo que representa una *cache* de teselas espaciales, así como las principales problemáticas que presentan en estos momentos. A continuación se presenta el método de autodescubrimiento basado en capas de *features* directoras para precargar las zonas potencialmente más interesantes. En la sección IV se lleva a cabo un estudio entre diferentes capas para demostrar que se pueden definir distintos parámetros de calidad que pueden resultar útiles a la hora de seleccionar las capas directoras. Por ultimo, las principales conclusiones de este trabajo se recogen en la sección V.

II. La gestión de los sistemas de cache espacial

Debido a la juventud del campo, las aproximaciones de este tipo existentes, como Tilecache y GeoWebcache [9], [10] están basadas en principios básicos de *cacheo* extraídos de otros ámbitos más genéricos. Sin embargo, el hecho de que estas *caches* almacenen elementos espaciales hace que tengan ciertas peculiaridades que se pueden aprovechar para obtener mejores resultados. Aspectos como la correlación espacial entre las diferentes teselas o las relaciones existentes entre las distintas escalas de los mapas deberían tenerse en cuenta a la hora de diseñar los algoritmos de mantenimiento de la *cache*.

A continuación se presentan los mecanismos de mantenimiento de una *cache* espacial que pretenden mejorar el tiempo de respuesta inicial del servicio partiendo de una *cache* vacía.

A. Mecanismos de Seeding y de Carga dinámica

En un servicio de mapas teselado, las teselas pueden ser generadas la primera vez que son pedidas (*carga dinámica*), *cacheándose* en este mismo proceso. De esta forma, la primera vez que se pide una tesela se produce un fallo (*miss*) de *cache* y la petición se resuelve aproximadamente a la misma velocidad que si se tratase de un servicio WMS tradicional. Peticiones subsiguientes de este mismo objeto son aceleradas en gran medida dado que el objeto ya ha sido generado previamente, produciéndose sucesivos aciertos (*hit*) de *cache*.

La principal ventaja de este método es que no requiere preprocesamiento y se produce una rudimentaria autogestión de la *cache*, puesto que las propias peticiones acumuladas en la memoria del sistema equivalen a una evaluación implícita de la probabilidad existente de que una tesela sea pedida por un usuario. El resultado indirecto es que sólo los datos pedidos son *cacheados*, ahorrando por tanto recursos de almacenamiento. El inconveniente de este método es que la devolución de teselas no experimenta ninguna mejora de rendimiento en la primera petición, reduciendo la calidad en la experiencia del usuario en las teselas poco demandadas o ya expiradas.

La otra posibilidad es que las teselas se generen mediante técnicas de *seeding*. Mediante estas técnicas las teselas se generan y *cachean* de forma automática. La ventaja de este proceso es que mejora en gran medida la experiencia percibida por el usuario. El inconveniente es que se trata de un proceso que consume mucho tiempo y recursos de almacenamiento.

Obviamente, conviene reducir este tiempo de puesta en servicio para conseguir alcanzar cuanto antes la QoS esperada respecto al tiempo de respuesta (fase de QoS estacionaria). Puede reducirse la penalización experimentada hasta que se completa el proceso de precarga cargando primero las teselas con mayor probabilidad de ser pedidas, dejando para el final aquellas cuya probabilidad de ser pedidas sea menor.

Sin embargo, dado que se trata de la puesta en marcha del servicio y no se tiene conocimiento previo sobre cómo será la distribución de peticiones, cabe preguntarse en base a qué pueden realizarse las predicciones necesarias. Algunos de los sistemas de *cache* estudiados [9][10] se basan para ello en la naturaleza espacial de los datos que contienen. Una opción razonable es generar primero las teselas para los niveles de resolución más altos (los que muestran zonas más genéricas como países o regiones) que, al menos intuitivamente, recibirán una mayor densidad de peticiones, y generar posteriormente las teselas pertenecientes a escalas más bajas (zonas más específicas como ciudades o barrios) o dejar que éstas vayan poblando la *cache* a medida que son pedidas.

Otra posibilidad interesante que ofrecen algunos de los sistemas de *cache* existentes es la de indicar el *bounding box* o encuadre geográfico del conjunto de teselas que se quiere

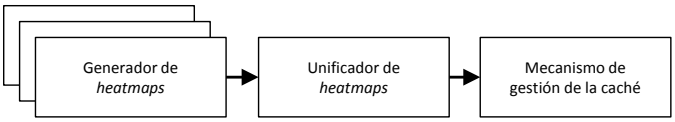


Fig. 2. Generación y unificación de *heatmaps*, y su posterior "'consumo"' por los mecanismos de gestión de la *cache*

cachear para las escalas indicadas. Sin embargo, estas posibilidades manuales se basan en la intuición del administrador. Durante el estudio del arte llevado a cabo no se detectaron indicios acerca de la exploración de mecanismos automáticos o semi-automáticos para la identificación avanzada de regiones potencialmente candidatas para ser generadas durante el proceso de *seeding*. En la Sección II-B se proponen algunos mecanismos.

B. Mapas de estimación probabilística

Los mecanismos de gestión de la *cache* anteriores comparten un mismo objetivo: maximizar la probabilidad de acierto en las peticiones de los usuarios, manteniendo el consumo de recursos por debajo de un nivel dado. Dado que la secuencia de peticiones de los usuarios no es determinista, esta probabilidad debe obtenerse en base a estimaciones. Estas estimaciones pueden plasmarse en un "mapa de estimación probabilística" o *heatmap*, como una representación de la estimación de la probabilidad con la que se pide una tesela de coordenadas x e y en un nivel de resolución n.

Tal y como se muestra en la Fig. 2, pueden desarrollarse diversos generadores de mapas de estimación probabilística, en base a distintas fuentes de conocimiento. Para intentar obtener una estimación más precisa puede combinarse la información de múltiples *heatmaps*, pudiéndose asignar distintos pesos a cada uno de ellos. Cómo determinar la asignación de estos pesos es una tarea compleja y que constituye por sí misma una línea de investigación, en la que sería deseable que esta asignación fuese adaptativa, pudiendo aplicarse para ello técnicas de aprendizaje automático o *Machine Learning*.

La generación de estos mapas de probabilidad es una tarea compleja y para la que pueden existir numerosas alternativas en función de la información disponible y de los objetivos que se pretendan lograr. A continuación se presentan dos de estos métodos:

a) Conocimiento de accesos pasados: Se parte de la premisa de que puede realizarse una estimación de la probabilidad de acceso futuro a las teselas atendiendo a la información disponible en accesos pasados. Para ello, una buena fuente de información son los registros de acceso o *logs* de los servidores de mapas. De estos registros puede extraerse la información de interés, como los *bounding boxes* de las peticiones y la capa (*layer*). También es posible extraer otra información adicional como la identidad del usuario o si la petición fue exitosa, en función de la complejidad del estudio a realizar se tendrán en cuenta unos parámetros u otros.

b) Features directoras de las peticiones de los usuarios: A partir de esta información vectorial se puede obtener información valiosa relativa a las zonas geográficas que pueden resultar más atractivas para los usuarios. Considérese, por ejemplo, un servicio WMS de una IDE con contenidos sobre el patrimonio de una determinada región. En este caso es de esperar que las *features* que identifican monumentos o puntos de interés sean directoras de las peticiones de los usuarios. Estas *features* pueden extraerse, por ejemplo, como información vectorial a partir de un servicio WFS (*Web Feature Service*) [11]. Una importante línea de investigación abierta como consecuencia consiste en la identificación automática de estas relaciones topológicas.

Dado que en la actualidad estos sistemas no están automatizados, sigue siendo responsabilidad del administrador del sistema decidir qué partes del mapa deben ser pregeneradas. Una mala decisión en este aspecto puede hacer totalmente ineficiente el proceso de precarga, si posteriormente se demuestra que las zonas pregeneradas no son realmente del interés de los usuarios. Es por ello que surge la necesidad de diseñar sistemas capaces de llevar a cabo esta tarea de forma automática.

III. Estudio de las capas de features como directoras de las peticiones de los usuarios

La determinación de posibles áreas de interés es un buen punto de partida para las técnicas de optimización basadas en la probabilidad de que los usuarios se muevan por zonas adyacentes. Sirva como ejemplo el caso de un usuario que pretende buscar un hotel cercano a un determinado monumento, si ese monumento ha sido previamente considerado zona de interés y han sido precargadas las teselas correspondientes a las zonas colindantes, la experiencia del usuario será mejor ya que los tiempos de carga del mapa se verán notablemente reducidos.

Los modelos predictivos pretenden crear mapas de estimación probabilística de acceso capaces de ofrecer información fiable sobre las peticiones futuras que realizarán los usuarios sin necesidad de conocer los accesos pasados al servicio. Estos modelos presentan algunas ventajas sobre los modelos descriptivos, basados en el estudio de los accesos pasados. Ofrecen la posibilidad de tener en cuenta determinadas zonas de interés dentro del mapa, tanto permanentes como temporales. También tienen la ventaja de que pueden ser aplicados de forma previa a la publicación de la información geográfica, ya que no es necesario tener en cuenta los accesos pasados de los usuarios [12].

Sin embargo, surge la problemática de decidir qué zonas son las más propensas a recibir las peticiones de los usuarios. En [12] se presenta un modelo predictivo básico que pretende determinar ciertas zonas prioritarias para la labor de *cacheo*. El modelo considera como datos de entrada los datos de poblaciones, zonas costeras, principales vías de comunicación y puntos de específico interés turístico. Esto se corresponde con las zonas que se han considerado más visitadas con Microsoft Web HotMap [13], [14]. A partir de esta información geográfica compone un mapa vectorial que indica que zonas deben ser pregeneradas.

Sin embargo, las capas de *features* que se eligieron en este trabajo son apropiadas y ofrecen buenos resultados para ese caso en concreto, pero es posible que para otros servicios de mapas esas *features* no sean directoras de las peticiones de los usuarios. Téngase en cuenta como ejemplo un servicio

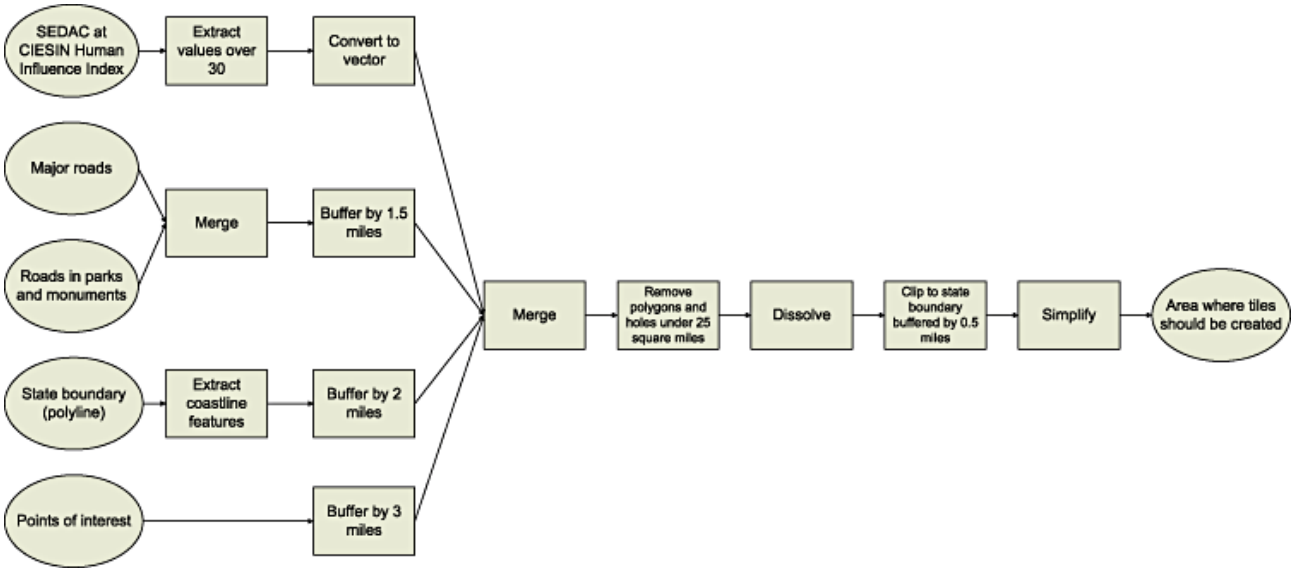


Fig. 3. Diagrama de flujo del proceso seguido en [12] para obtener las áreas a precargar. Extraído de [12]

de mapas que desea publicar información orientada a destacar puntos de interés para montañeros. En ese caso, estos *datasets* deberían incluir información relativa a los principales sistemas montañosos para que el sistema sea capaz de detectarlos y priorizarlos sobre otras zonas de menor interés. En este caso la precarga de las teselas correspondientes a las zonas costeras no tendría ningún sentido y supondría un desperdicio de espacio en disco, además, los usuarios no verían mejorada su experiencia de navegación.

Otra deficiencia detectada en este trabajo es que se da igual importancia a todas las capas seleccionadas. Cabe la posibilidad de que alguna de las capas tenga mayor importancia como directora de las peticiones de los usuarios y sea necesario priorizarla sobre el resto. Esta priorización podría consistir en precargar en primer lugar las teselas que abarcan las capas con más probabilidad de acierto o en desarrollar algoritmos que sean capaces de detectar qué capas deben ser precargadas y cuáles no de entre un catálogo de *features*.

En la Sección IV se presentan varios parámetros de representatividad que podrían ayudar a los administradores a tener una visión más real de las zonas más solicitadas. Estos valores se basan en técnicas mixtas, ya que pretenden generar un mapa probabilístico no sólo a partir de los accesos pasados, sino también a partir de diversos grupos de *features*.

Estos parámetros pretenden, a partir de las estadísticas de acceso a un determinado servicio, extraer conclusiones y determinar si las *features* serían buenas directoras del servicio o no. Asimismo, también se tratan aspectos como la correlación entre diversos *datasets* de datos vectoriales para comparar su bonanza. Con estos cálculos se pretende dotar a los administradores de una información filtrada y de calidad a partir de la cual la toma de decisiones se puede llevar a cabo de una forma más sencilla.

IV. Medidas de representatividad de las capas directoras

En esta sección se pretende presentar los distintos parámetros de representatividad que se han definido para determinar el peso que tienen las distintas capas vectoriales de un determinado catálogo respecto de las peticiones realizadas al servicio por parte de los usuarios. Gracias a estos parámetros se pueden obtener datos objetivos útiles a la hora de seleccionar las *features* a tener en cuenta y que también dan la posibilidad de comparar fácilmente la importancia de cada capa para servicios diferentes.

A continuación se exponen los resultados prácticos obtenidos tras varios estudios. Todos ellos se han realizado sobre el mismo escenario. A partir de los *logs* obtenidos del servicio PNOA[3] (Plan Nacional de Ortofotografía Aérea), se ha estudiado la importancia de tres *datasets* de información vectorial. Los *datasets* elegidos han sido los siguientes: La capa de ríos de la cuenca del Duero, obtenida de la Confederación Hidrográfica del Duero[4], la de poblaciones de España, obtenida de la IDEE[5] (Infraestructuras de Datos Espaciales de España) y la de la red de carreteras de Castilla y León, proporcionada por el Servicio de información Territorial de la Junta de Castilla y León[6]. Considerando que la mayoría de los *datasets* tenidos en cuenta sólo tienen información relativa a Castilla y León, se ha reducido el ámbito geográfico del estudio a un *bounding box* que abarca parte de Castilla y León definido por las coordenadas [-6.25,40.75,-3.9,42.7].

Para los cuatro primeros niveles de resolución, toda la zona geográfica que forma parte de este estudio está contenida en una única tesela, por lo que no tiene sentido tener en cuenta esos niveles. Por lo tanto, el estudio trabajará con datos referidos a los niveles comprendidos entre el cinco y el dieciocho. Pese a esta simplificación, los primero niveles tenidos en cuenta no son excesivamente significativos, ya que el número de teselas que abarca la zona de estudio es muy escaso.

Puesto que el volumen de datos a analizar era muy extenso, se han tenido en cuenta solamente las primeras 20.000 peticiones realizadas al servicio sobre la zona seleccionada.

[3]http://www.ign.es/PNOA
[4]http://www.chduero.es
[5]http://www.idee.es
[6]http://www.idecyl.jcyl.es

Puesto que el comportamiento de los usuarios puede variar para cada nivel de resolución, los resultados están disgregados por niveles de resolución ya que el comportamiento pude ser muy diferente entre unos niveles y otros. Por ejemplo, el callejero de una ciudad se suele consultar con un detalle muy superior al que se utiliza para visualizar una ruta interurbana.

En el caso de líneas y puntos, dado que la superficie de estas geometrías resulta poco representativa, se ha optado por aplicar un *buffer* de 0,001 grados geográficos, que equivalen aproximadamente a un radio de 85 metros. De esta forma se aumenta su superficie para adaptarla a la superficie real que representan las geometrías.

En primer lugar se presentan cuatro parámetros relacionados con la representatividad de cada una de las capas por separado, la representatividad absoluta y relativa, la incidencia de almacenamiento y el índice de acierto por tesela. A continuación se definen dos parámetros que tratan de obtener información conjunta de más de una fuente de datos, como son la unión o la intersección entre capas o la correlación espacio-probabilística.

A. *Representatividad absoluta y relativa de la capa*

Estos parámetros pretenden mostrar la representatividad de una determinada capa dentro de las peticiones que se han realizado al servicio. Con este parámetro se pretende obtener una primera visión general del peso que tiene una capa como directora de las peticiones de los usuarios de un determinado servicio. Por un lado se puede obtener la representatividad absoluta, que indica el número de peticiones realizadas sobre la capa respecto del total de peticiones realizadas. El segundo de los parámetros indica el peso relativo respecto de las peticiones realizadas sobre todas las capas del catálogo.

Como se puede observar en las Figuras 4 y 5, los valores son bastante similares entre unas capas y otras, exceptuando un significativo pico para la capa de carreteras en el nivel 15, ya que es la única que recibe alguna petición. Cabe destacar que en los primeros niveles, los valores son muy similares porque todas las capas cubren prácticamente la totalidad de la zona a estudiar. A medida que aumenta el nivel de detalle, la superficie que cubre cada una de ellas va disminuyendo significativamente, como se puede observar en la Figura 7. Los datos más significativos de estas gráficas se encuentran en el nivel 18, donde las diferencias entre unas capas y otras aumentan. En este nivel, la capa de poblaciones obtiene los mejores valores, indicador de que es la capa con la que se obtendría un mayor número de aciertos de *cache* si se mantuviera la distribución espacial de las peticiones de los usuarios.

Con este parámetro se puede llevar a cabo una primera valoración que puede ayudar a descartar aquellos *datasets* cuyos resultados sean residuales respecto al resto, aunque tampoco nos da información suficiente para determinar la calidad de una capa vectorial como directora de las peticiones de los usuarios. En este primer análisis se obvia un aspecto muy importante y que se debe tener en cuenta en cualquier dispositivo de *cache*, el espacio de almacenamiento. El objetivo final es obtener el mayor número de aciertos con el mínimo número de teselas precargadas. En el caso de una de las capas vectoriales se consiguiera un gran nivel de acierto, pero dada la superficie que abarca, fuera necesario precargar

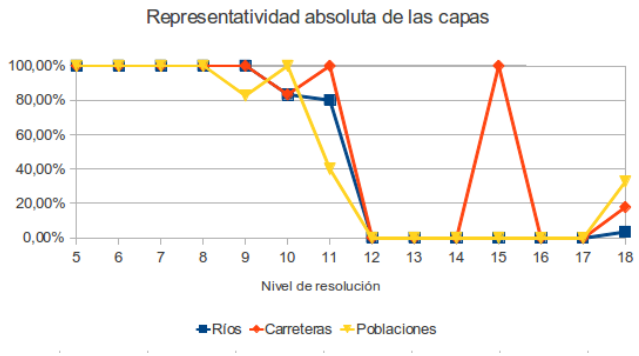


Fig. 4. Estudio de la representatividad absoluta para el servicio PNOA

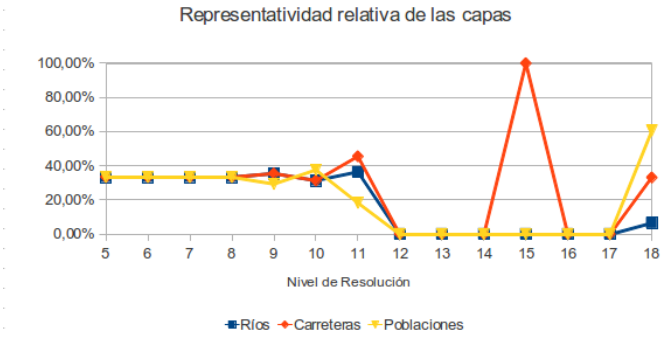


Fig. 5. Estudio de la representatividad relativa para el servicio PNOA

un número muy grande de teselas, el proceso seguido no resultaría óptimo. Podría resultar más útil precargar una capa cuyo peso fuera menor, pero el número de teselas a precargar fuera significativamente menor. Ese razonamiento lleva al siguiente de los parámetros estudiados.

B. *Incidencia de almacenamiento de la capa*

Este parámetro pretende corregir alguna de las deficiencias indicadas para la representatividad absoluta y relativa. Teniendo como parámetro de entrada una determinada zona geográfica de influencia, este parámetro pretende indicar el coste en teselas que representaría precargar la capa. En primer lugar se ha de calcular el número de teselas que abarca el *bounding box* de referencia, y a partir de ahí obtener el número de teselas a precargar para contener cada uno de los *datasets* del catálogo. Finalmente se realiza un cociente entre ambos valores para obtener un índice que sirve de ayuda para estimar el coste de almacenamiento derivado de utilizar como base de la precarga una capa u otra.

Observando la Figura 6 se puede observar que las gráficas para los tres *datasets* tienen una forma similar. Esto se da porque para los niveles altos, las *features* abarcan todas las teselas, pero a medida que se trabaja con un nivel mayor, el número de teselas aumenta por cuatro, mientras que la superficie de las *features* se mantiene, por lo que el número de teselas contenidas disminuye de forma exponencial, como se puede ver en el ejemplo de la Figura 7. Sin embargo, a partir del nivel 8, la diferencia entre unas capas y otras se hace significativa, en favor de la capa de poblaciones,ya que siempre se mantiene por debajo del resto de capas, teniendo valores insignificantes para los niveles más altos.

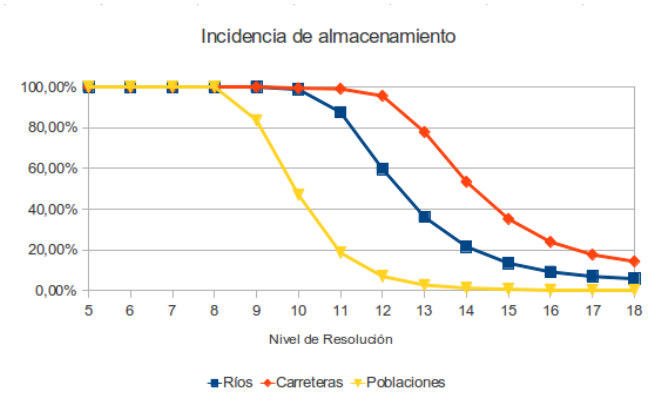


Fig. 6. Valores de la incidencia de almacenamiento para el servicio PNOA

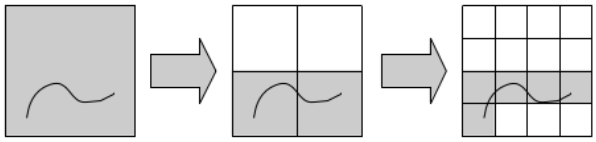


Fig. 7. Ejemplo del número de teselas que abarca una *feature* según aumenta el nivel de resolución

C. Índice de acierto por tesela

Este parámetro pretende indicar el coste en teselas por cada acierto de *cache* que se produciría si las peticiones futuras siguieran la misma distribución probabilística que las analizadas. Consiste en el cociente entre el número de peticiones coincidentes con la capa vectorial entre el número total de teselas a precargar.

A la luz de la Figura 8 se observa que este índice obtiene valores grandes para los primeros niveles, pero según aumenta el nivel de resolución, disminuye. Esto es debido tanto a que el número de teselas que ocupan las *features* es cada vez menor (Figura 7), así como a que hay niveles para los que no se ha recogido ninguna petición. Sin embargo, en el nivel 18, que es el que recoge la mayoría de las solicitudes, se puede observar que el valor para la capa de poblaciones es mayor que para el resto, lo que nos hace suponer que será mejor directora de las peticiones que el resto de las capas.

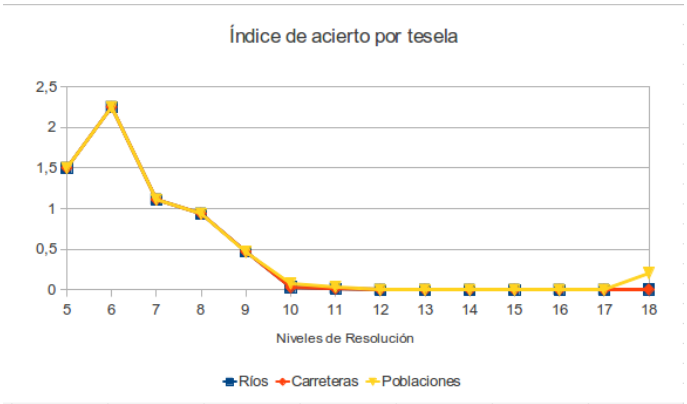


Fig. 8. Valores del índice de acierto por tesela para el servicio PNOA

D. Correlación entre capas

Otro parámetro que se debe tener en cuenta a la hora de decidir qué capas escoger para precargar la *cache* es la correlación existente entre las mismas. Con este parámetro se le puede indicar al administrador la semejanza existente entre varias capas. La inclusión de dos capas muy diferentes puede hacer que el número de teselas a precargar sea muy grande. Sin embargo, si las capas son muy similares, las teselas que se precargarán serán comunes a las dos y es posible que no resulte útil utilizar ambas, sino tan sólo una de ellas para ahorrar costes de procesamiento. En este segundo caso, al descartar una de ellas, se podría tener en cuenta una tercera capa que pudiera ayudar a ofrecer un mejor servicio. En el contexto de la correlación entre las capas se han tenido en cuenta varios casos en función de la información a extraer.

1) Unión de las capas: Con este parámetro se calcula la unión de las teselas que se deberían precargar teniendo en cuenta ambas capas. Con este parámetro se pretende obtener una medida real del espacio de almacenamiento de *cache* necesario para almacenar las teselas coincidentes con ambas capas. Este parámetro unido a los anteriores puede hacer declinar la utilización de alguna de las capas por la posibilidad de exceder el espacio de almacenamiento que se está dispuesto a utilizar.

Nivel de Resolución	Carreteras	Poblaciones	Unión
5	2	2	2
6	4	4	4
7	9	9	9
8	16	16	16
9	49	49	49
10	165	167	167
11	558	638	640
12	1453	2359	2400
13	3348	7380	7958
14	7449	19177	21988
15	17189	47217	57310
16	42122	116351	147517
17	112712	306001	399421
18	333184	871854	1164726

Tabla I
VALORES DE LA UNIÓN ENTRE LAS CAPAS DE CARRETERAS Y POBLACIONES

Según los resultados de la Tabla I, se puede observar que para los primeros niveles, los valores de la unión son muy similares al mayor de los valores de las capas por separado, lo que indica que la información está concentrada en esos niveles, pero a medida que aumenta, se va haciendo más dispersa. Esta información indica que si atendiendo a sus valores individuales (IV-A, IV-B, IV-C), ambas capas son significativas, seleccionarlas podría ser una buena opción. Por contra, en el caso de que una de ellas tenga un peso significativo sobre la otra, se podría descartar la segunda porque implicaría la precarga de un gran número de teselas para obtener una cantidad poco significativa de aciertos de *cache*.

2) Intersección de las capas: Este parámetro que se obtiene de forma directa a partir del anterior indica el número de teselas comunes a varias capas. Indica el porcentaje de teselas comunes a dos fuentes de datos vectoriales diferentes.

Contrastando este parámetro con la Incidencia de Almacenamiento de ambas capas se puede tomar la decisión de cargar una u otra si la intersección es muy grande, pero sin embargo, los grados de incidencia muy diferentes.

Nivel de Resolución	C∩P	R∩C	P∩R
5	2 (100%)	2 (100%)	2 (100%)
6	4 (100%)	4 (100%)	4 (100%)
7	9 (100%)	9 (100%)	9 (100%)
8	16 (100%)	16 (100%)	16 (100%)
9	41 (100%)	49 (100%)	41 (100%)
10	78 (98%)	165 (100%)	78 (98%)
11	116 (96%)	551 (99%)	103 (85%)
12	171 (98%)	1379 (96%)	120 (68%)
13	267 (98%)	2694 (82%)	137 (50%)
14	486 (95%)	4415 (61%)	188 (37%)
15	920 (82%)	6573 (39%)	268 (24%)
16	1948 (65%)	9748 (24%)	448 (14%)
17	4560 (49%)	16427 (15%)	864 (9%)
18	12027 (37%)	34100 (10%)	2077 (6%)

Tabla II
VALORES DE LA INTERSECCIÓN ENTRE LAS CAPAS DE CARRETERAS, POBLACIONES Y RÍOS

La información que se obtiene con este segundo tipo de correlación está directamente relacionado con el anterior. Al conocer el porcentaje de intersección entre dos capas, como se puede observar en la Tabla II, es posible conocer el número de teselas en común que tienen entre ellas y decidir si puede resultar rentable seleccionar ambas o no. Si por ejemplo dos capas tienen una alta intersección, y al mismo tiempo valores similares de representatividad, significa que la mayoría de las peticiones están contenidas en la parte común a ambas, por lo que una buena opción sería utilizar la que menos teselas abarque.

3) Correlación espacio-probabilística de las capas: En los dos apartados anteriores sólo se tenía en cuenta la componente espacial de las fuentes de datos, obviando la probabilidad de acierto de *cache* existente una vez que las capas hayan sido seleccionadas para ser utilizadas en el proceso de precarga. Esta función correlación entre las capas uno y dos $R\left(C_1, C_2\right)$ se define para cada nivel n como el cociente del producto de la función Presencia $P\left(x, y\right)$ de cada una de las capas dentro del espacio de teselas seleccionado por el valor de la función Hits $H\left(x, y\right)$ que define la distribución espacial de las peticiones de los usuarios entre el cuadrado del número de peticiones realizadas N.

$$R_n\left(C_1, C_2\right) = \frac{1}{N^2} \sum_{x,y} P_1(x,y)H(x,y)P_2(x,y)H(x,y) \tag{1}$$

Teniendo en cuenta que la función H es común a ambas capas y que el producto de P_1 y P_2 es igual a la intersección de ambas $P_{1\cap 2}$, (1) se puede simplificar $R_n\left(C_1, C_2\right)$como:

$$R_n\left(C_1, C_2\right) = \frac{1}{N^2} \sum_{x,y} P_{1\cap 2}(x,y)H(x,y)^2 \tag{2}$$

Con este valor de correlación, el administrador del sistema tiene una visión más real de la correlación entre dos fuentes de datos vectoriales. No sólo dispone de datos sobre la relación espacial entre ellas, sino que también tiene en cuenta la información probabilística de las peticiones que se realizan sobre ellas. De esta forma será capaz de determinar si las peticiones se realizan en las teselas comunes a ambas capas o en las teselas que sólo forman parte de una de las capas. A partir de esta información ya preprocesada, resulta notablemente más sencillo tomar una decisión al respecto

Nivel de Resolución	$P_{1\cap 2}(x,y)H\left(x,y\right)^2$	Total peticiones	R_n
5	5	3	0,56
6	21	9	0,26
7	16	10	0,16
8	19	15	0,08
9	25	23	0,05
10	5	6	0,14
11	4	10	0.04
12	0	0	N/D
13	0	0	N/D
14	0	1	0
15	0	1	0
16	0	0	N/D
17	0	0	N/D
18	4958	19916	$1{,}25{\cdot}10^{-5}$

Tabla III
CORRELACIÓN ENTRE LAS CAPAS DE CARRETERAS Y POBLACIONES PARA EL SERVICIO PNOA

Nivel de Resolución	$P_{1\cap 2}(x,y)H\left(x,y\right)^2$	Total peticiones	R_n
5	5	3	0,56
6	21	9	0,26
7	16	10	0,16
8	19	15	0,08
9	29	23	0,05
10	5	6	0,14
11	8	10	0,08
12	0	0	N/D
13	0	0	N/D
14	0	1	0
15	0	1	0
16	0	0	N/D
17	0	0	N/D
18	652	19916	$1{,}64{\cdot}10^{-6}$

Tabla IV
CORRELACIÓN ENTRE LAS CAPAS DE CARRETERAS Y RÍOS PARA EL SERVICIO PNOA

Analizando los resultados de las Tablas III y IV se puede observar que los valores de la correlación de la capa de carreteras es 10 veces mayor con la capa de poblaciones que con la de ríos para el último de los niveles. La explicación es sencilla atendiendo a los parámetros anteriores. Por un lado, las poblaciones concentran un mayor número de peticiones que los ríos y las carreteras, y al mismo tiempo, la intersección entre las carreteras y las poblaciones es mayor que la existente entre las carreteras y los ríos, por lo tanto la correlación espacio-probabilística es mucho mayor entre las carreteras y las poblaciones que entre las poblaciones y los ríos. A la luz de estos resultados, la capa que se debería seleccionar como base para el "mapa de estimación probabilística" de este servicio debería ser la de poblaciones, descartando las otras dos, ya que ambas presentan un mayor número de teselas a precargar, pero sin embargo, una menor probabilidad de que los usuarios

accedan a las teselas que abarcan.

V. Conclusiones

Tras detectarse ciertas deficiencias de escalabilidad en los servicios WMS tradicionales, se ha comprobado que los servicios teselados, como WMS-C o WMTS ofrecen mejor rendimiento. Sin embargo, no resulta práctico tener almacenadas todas las posibles teselas, por lo que se tiende a la utilización de *caches* parciales.

Para mejorar el rendimiento inicial de estos sistemas se utilizan técnicas de *seeding*. Pretenden generar y almacenar teselas de forma automática, tratando de que se minimicen los fallos de *cache* y aumentando la QoS percibida por los usuarios, manteniendo el consumo de recursos dentro de unos límites definidos. Sin embargo, esta técnica necesita como parámetro de entrada un "mapa de estimación probabilística" que le indique qué zonas del mapa son más propensas a recibir peticiones de los usuarios.

Como ayuda para generar estos mapas, en este documento se han definido varios parámetros de representatividad, que a partir de un catálogo de *features* candidatas y de un registro de los accesos pasados al servicio es capaz de dar información sobre la bondad de cada una de las capas como directora de las peticiones de los usuarios.

Como se ha visto, con estos parámetros se puede determinar de forma sencilla qué datos ofrecerán un mejor rendimiento para el sistema. Una vez procesados los datos se obtienen datos numéricos que son fácilmente comparables para poder tomar la decisión óptima a partir de datos objetivos. Estos parámetros dan información de cada una de las capas por separado, así como valores de correlación entre ellas para poder discernir cuál ofrecería un mejor rendimiento.

Con la utilización de estos parámetros, los administradores de este tipo de sistemas verán simplificada su misión de forma notable, ya que hasta ahora, estas labores se llevaban a cabo de forma manual.

Además, estos mapas de estimaciones no sólo podrán ser usados para los procesos de mantenimiento de *caches* de mapas, sino también para otros objetivos muy diversos, como puede ser la determinación de qué zonas debe ofrecer con mayor detalle el servicio, o indicar qué zonas deben ser actualizadas con mayor rapidez para mostrar los datos con la máxima exactitud posible.

Agradecimientos

Este trabajo ha sido realizado como parte del proyecto CENIT España Virtual[7] (ref. CENIT 2008-1030), cofinanciado por el CDTI, dentro del programa Ingenio 2010 y por el CNIG.

Referencias

[1] A. Turner, *Introduction to neogeography*. O'Reilly, 2006.

[2] OGC, "OpenGIS web map service (WMS) implementation specification," http://www.opengeospatial.org/standards/wms, 2009.

[3] L. Plesea, "The design, implementation and operation of the JPL OnEarth WMS server," in *Geospatial Services and Applications for the Internet*, J. Sample, K. Shaw, S. Tu, and M. Abdelguerfi, Eds. Berlin: Springer, 2008, pp. 93–109.

[4] M. Mills, "NASA world wind tile structure," http://www.ceteranet.com/nww-tile-struct.pdf, 2005. [Online]. Available: http://www.ceteranet.com/nww-tile-struct.pdf

[5] L. D. Cola and N. Montagne, "The pyramid system for multiscale raster analysis," *Computers & Geosciences*, vol. 19, no. 10, pp. 1393–1404, Nov. 1993. [Online]. Available: http://www.sciencedirect.com/science/article/B6V7D-48C78W0-2D/2/24c399eec9cbd9eb8311285bdbf53901

[6] K. P. Joan Masó and N. Julià, Eds., *Web Map Tile Service Implementation Standard*, ser. OpenGIS Implementation Standard. Open GIS Consortium Inc., April 2010, no. OGC 07-057r7.

[7] OSGeo, "WMS-C wms tile caching - OSGeo wiki," http://wiki.osgeo.org/wiki/WMS_Tile_Caching, 2008. [Online]. Available: http://wiki.osgeo.org/wiki/WMS_Tile_Caching

[8] K. Cheng, Y. Kambayashi, and M. Mohania, "Efficient management of data in proxy cache," in *Database and Expert Systems Applications, 2001. Proceedings. 12th International Workshop on*, 2001, pp. 479–483. [Online]. Available: 10.1109/DEXA.2001.953107

[9] MetaCarta, "TileCache, from MetaCarta labs," 2008. [Online]. Available: http://tilecache.org/

[10] OpenGeo, "GeoWebCache," 2008. [Online]. Available: http://geowebcache.org

[11] P. A. Vretanos, Ed., *Web Feature Service Implementation Specification*. OGC 04-094: Open Geospatial Consortium Inc, 2005.

[12] S. Quinn and M. Gahegan, "A predictive model for frequently viewed tiles in a web map," *Transactions in GIS*, vol. 14, no. 2, p. 193–216, 2010.

[13] D. Fisher, "Hotmap: Looking at geographic attention," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 6, p. 1184–1191, 2007.

[14] D. Fisher, "The impact of hotmap," 2009. [Online]. Available: http://research.microsoft.com/pubs/81244/Fisher_2008_Hotmap.pdf

[7] http://www.españavirtual.org/

Estrategias de Metatiling para la Aceleración de Servicios de Mapas Teselados en las Infraestructuras de Datos Espaciales

R. García Martín, J. P. de Castro Fernández, M. J. Verdú Pérez
E. Verdú Pérez, L. M. Regueras Santos, P. López Escobés, D. García Martín
Laboratorio de Infraestructuras de Datos Espaciales (IDELab)
Departamento de Teoría de la Señal, Comunicaciones e Ingeniería Telemática
Escuela Técnica Superior de Ingenieros de Telecomunicación
Universidad de Valladolid
Camino del Cementerio s/n, 47011 Valladolid.
{ricgar, juacas, marver, elever,luireg}@tel.uva.es, {plopesc,dgarmar}@ribera.tel.uva.es

***Resumen*—La gran proliferación de los servicios SIG (Sistemas de Información Geográfica) durante los últimos años ha motivado la necesidad de disponer de servicios cada vez más escalables en las Infraestructuras de Datos Espaciales. En el ámbito de los servicios de mapas esto se ha traducido en la aparición de nuevas especificaciones basadas en modelos teselados, como la recomendación WMS-C de OSGeo o el estándar WMTS de OGC. Tales servicios pueden beneficiarse de un sistema de caché, sirviendo imágenes pregeneradas. En este escenario, cuando se produce un fallo de caché, la petición es redirigida al servicio de mapas original, que genera al vuelo la imagen. Uno de los principales cuellos de botella en el proceso de generación de teselas es el acceso a los almacenes externos de información geográfica. Ya que las consultas suelen estar optimizadas mediante índices espaciales, puede obtenerse una mejoría ampliando la región geográfica del mapa a generar, reduciendo por tanto el número de consultas. En este trabajo se estudia el impacto de utilizar esta estrategia, conocida como *metatiling*, durante los procesos de precarga y de carga dinámica. Para ello se utiliza el prototipo de caché de teselas *WMSC-Wrapper* como banco de pruebas, haciendo uso de los registros de acceso de diversos servicios públicos de mapas para simular un escenario real. Los resultados demuestran que estas estrategias permiten poblar la caché de forma más eficiente y conseguir mayores tasas de acierto.**

***Palabras Clave*—Cache, GIS, Metatiling, WMS, WMS-C, WMTS, QoS**

I. Introducción

La solución a la escalabilidad de los servicios en Internet ha pasado históricamente por la utilización de sistemas de caché en diversos niveles [1] del sistema completo cliente, red y dentro de los servidores en las distintas interfaces de intercambio de entidades, según recomiendan las distintas arquitecturas de diseño y de despliegue. En [2] se muestra un interesante diagrama taxonómico (ver Fig. 1) de los distintos patrones de caché clasificado por capas de arquitectura y por tecnologías típicas de aplicación.

En general las operaciones de caché se basan en la existencia de un conjunto finito, o al menos numerable, de elementos identificables en un almacén de datos o sistema de recuperación. Esta característica permite establecer sistemas de almacenamiento rápido para albergar los elementos con más probabilidad de ser accedidos en un futuro próximo.

En el caso de los servicios de información geográfica, las recomendaciones del OGC (*Open Geospatial Consortium*) [1] establecen los interfaces de comunicación de los diferentes servicios buscando cubrir un amplio conjunto de casos de uso. La flexibilidad y modularidad de estos interfaces hacen que los sistemas implantados deban responder en ocasiones a peticiones de servicio en tiempo real muy exigentes que afectan negativamente a la QoS (*Quality of Service*) percibida por el usuario. Un ejemplo prototípico de esta situación es el servicio WMS (*Web Map Service*) [3], al que se le pueden realizar peticiones de mapas con los siguientes grados de libertad:

- Capa: Todas las capas ofrecidas por el servicio pueden ser solicitadas en cualquier orden de superposición.
- Estilo: Cada capa puede solicitarse con cualquiera de los estilos con nombre (*named style*) y, opcionalmente, con cualquier estilo proporcionado por el usuario en cada petición.
- Tamaño de imagen: Cualquier tamaño expresado en pixels que el usuario necesite para su aplicación.
- Sistema de proyección: El mapa se puede pedir proyectado en cualquiera de los sistemas de proyección ofertados por el servidor.
- Área geográfica: Cualquier zona geográfica expresada mediante dos pares de coordenadas decimales de precisión arbitraria sin ninguna restricción en su dominio de definición.

Como al menos dos de los elementos anteriores (tamaño y área geográfica) tienen un dominio de definición continuo, el espacio de claves generado contiene un número infinito de elementos. Tal conjunto no puede beneficiarse de un sistema caché dado que la probabilidad efectiva de que una petición se repita es virtualmente nula.

Para permitir el funcionamiento de los *proxy* o los *Web-cache* se ha propuesto una recomendación para los clientes de mapas denominada WMS-C (*WMS Cached*) [4], que consiste en la acotación de los dominios de las variables de las peticiones. Esta es también la filosofía del reciente estándar WMTS (*Web Map Tile Service*) [5] de OGC, inspirado en la

[1] http://www.opengeospatial.org

ISBN: 978-84-694-5948-5

recomendación anterior, así como en las iniciativas de Google Maps y Nasa OnEarth [6].

Mediante estas especificaciones, todos los valores pertenecen a conjuntos discretos y la caché generará aciertos con probabilidad no nula.

En general puede decirse que los sistemas de caché espacial son de aplicación para todas aquellas fuentes de información primarias cuyas primitivas de servicio puedan normalizarse y discretizarse al menos para un porcentaje significativo de peticiones de servicio.

Además, cuando los servicios incorporan parámetros espaciales, es posible que existan correlaciones espaciales significativas entre las peticiones que puedan hacer más eficaces las técnicas de caché al permitir definir zonas de características o requisitos similares y adoptar en esas zonas técnicas específicas de *cacheo*.

Potencialmente, el campo de aplicación de estas tecnologías abarca a todos los servicios OGC dado que todos tienen una componente geográfica significativa. De hecho las conclusiones obtenidas en este estudio podrán ser de aplicación al resto de servicios de base espacial, que también podrían beneficiarse de un sistema de caché, como se ilustra en la Fig. 2.

Cuando se produce un fallo de caché, la petición de mapa se redirige al servidor de mapas remoto, que genera la imagen al vuelo, a través de un procedimiento generalmente costoso que implica acceso a datos de origen, aplicación de estilos, composición de capas y codificación de la imagen comprimida. El objeto se introduce entonces en la caché para servir peticiones futuras del mismo más rápidamente.

Uno de los principales cuellos de botella en el proceso de generación de teselas es el acceso a los almacenes externos de información geográfica. Ya que las consultas suelen estar optimizadas mediante índices espaciales, puede obtenerse una mejoría ampliando la región geográfica del mapa a generar. Esta estrategia se conoce como *metatiling*. Según esta estrategia, cuando se produce un fallo de caché se pide al servidor de mapas una tesela de mayor tamaño (*metatile*) que el objeto solicitado. Este *metatile* contiene la tesela original y también otras teselas adyacentes. El *proxy* recorta esta macro-imagen en teselas individuales que son introducidas en la caché.

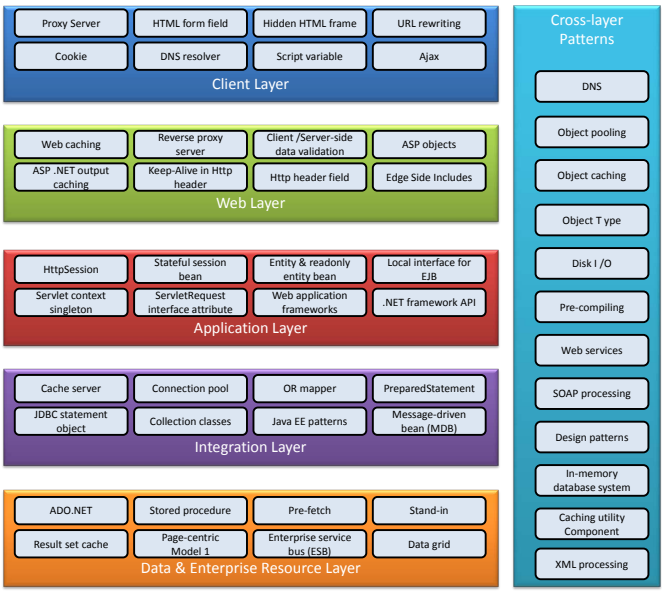


Fig. 1. Taxonomía de los sistemas y puntos de actuación de las caché (Figura adaptada de [2]).

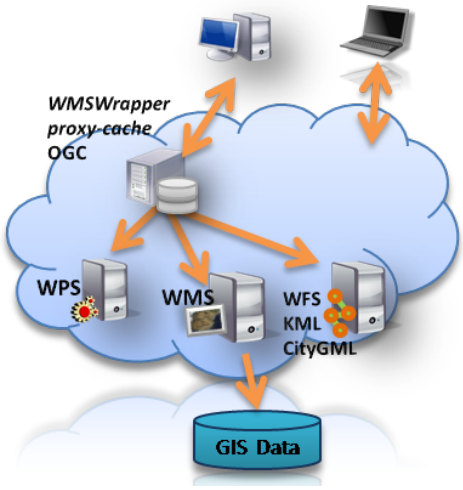


Fig. 2. Relación general de un filtro tipo *Web Cache* con los servicios OGC generadores de objetos a partir de parámetros espaciales discretizables.

En este trabajo se analizan distintas estrategias de generación de *metatiles*. Se pretende demostrar que, mediante el uso de estas estrategias, puede optimizarse el proceso de carga de objetos en las cachés de teselas.

El resto de este documento está estructurado de la forma siguiente: En la Sección II se realiza una caracterización formal de los conceptos y parámetros específicos de un sistema de caché espacial. En la Sección III se describen distintos mecanismos para la población de estas cachés. El prototipo de caché de teselas *WMSCWrapper*, utilizado como banco de pruebas para la experimentación con diversos mecanismos de caché, se presenta en la Sección IV. En la Sección V se introduce la técnica de *metatiling* y se evalúa el rendimiento de la misma utilizando el prototipo *WMSCWrapper*. Por último, las principales conclusiones de este trabajo se recogen en la Sección VI.

II. Sistemas de cache espacial

Durante el estudio del arte se constató que las implementaciones comerciales y *Open Source* existentes utilizan estrategias de mejora de la latencia pregenerando zonas de la cartografía de acuerdo a unos estilos de representación fijos. Generalmente, la información incluida en la caché es muy estática, por lo que lo habitual es disponer de una caché completa (conteniendo el 100% de los objetos del dominio). Esta solución implica asumir dos problemas inmediatos relacionados con el consumo de recursos y con la calidad del servicio:

- Grandes requisitos de almacenamiento: Incluso para aplicaciones de escala modesta se consume una gran cantidad de recursos de almacenamiento. En muchos casos hay que recurrir a cachés incompletas que requieren una adecuada política de gestión y de mantenimiento.
- Retardo de la puesta en marcha: El tiempo de puesta en servicio aumenta con el tamaño de la zona geográfica de

servicio y de la colección de escalas de representación. Durante un cierto periodo de tiempo (fase de QoS transitoria) la calidad del servicio (QoS) se ve degradada hasta alcanzar asintóticamente los parámetros definitivos (fase de QoS estacionaria). Si la información debe actualizarse periódicamente, sería posible que no se alcanzase nunca la QoS objetivo.

La solución más general es, por lo tanto, considerar para el análisis un sistema de caché con recursos de almacenamiento y computacionales limitados y con unos objetivos de QoS que se deben obtener cuanto antes y mantener durante la vida del servicio. Los objetos a *cachear* tendrán un tiempo de vida tras el cual se consideran obsoletos y son descartados. Se parte de la suposición de que el sistema inicia su funcionamiento desde un estado vacío.

Estas condiciones de contorno permiten aplicar los resultados a sistemas de todas las escalas de despliegue y a diversas variedades de servicio basado en parámetros espaciales.

A. *Conceptos y nomenclatura*

Con estas premisas puede definirse formalmente una nomenclatura y una serie de conceptos para poder establecer métricas cuantitativas y poder aplicar criterios para discriminar metodologías de gestión y optimización de las cachés espaciales. En el presente estudio se han analizado dos sencillos indicadores de la calidad de un servicio de caché:

- la latencia τ; entendida como el tiempo necesario para la obtención de los objetos resultado e iniciar su transmisión.
- el tiempo de servicio $t_{service}$; percepción por parte del consumidor del servicio de la espera necesaria hasta la finalización de la entrega del objeto solicitado. Este parámetro está compuesto de la latencia del servicio, la transferencia de la información y el procesado en el cliente. Todos los efectos relacionados con las infraestructuras de red y el tamaño de los objetos no se tienen en consideración en este estudio.

Para ofrecer un servicio de mapas teselado, el servidor de mapas renderiza la imagen de mapa en distintas escalas de representación mediante generalización cartográfica. Las imágenes generadas se dividen en teselas, describiendo una pirámide de teselas como se muestra en la Fig. 3. En los niveles más altos de la pirámide el mapa se representa con poco detalle mediante un reducido número de teselas, mientras que en la base de la pirámide la representación es más detallada al utilizar un mayor número de éstas.

A continuación vamos a establecer una nomenclatura y realizar una formulación probabilítica de los objetos gestionados por la caché.

Como los objetos de una caché espacial pueden identificarse por sus coordenadas, puede utilizarse la notación $T(i,j,n)$ para referirse a la tesela de índices i y j, y nivel de resolución n en la pirámide de escalas. Así mismo, se denomina como $P_h\{T(i,j,n)\}$ a la probabilidad de conseguir un acierto de caché. τ_h (*hit*) será el coste en segundos necesario para obtener un objeto de la caché y τ_m (*miss*) el coste incurrido al construir un nuevo objeto a partir de los servicios originales.

La densidad espacial de probabilidad que caracteriza la distribución espacial de los centroides de las peticiones (*requests*)

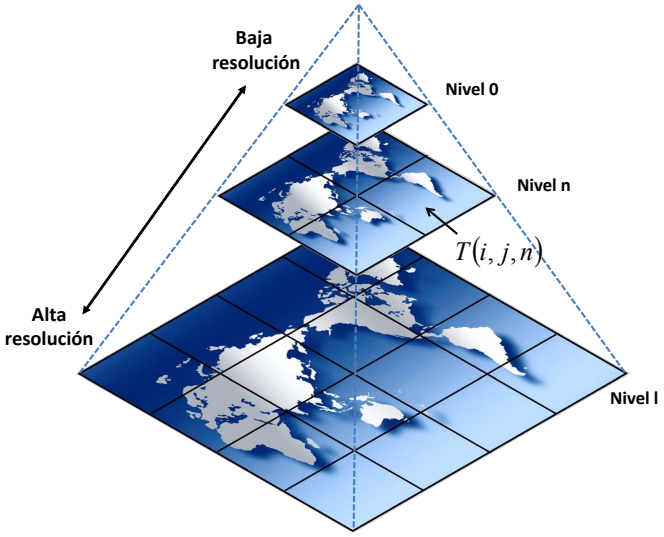


Fig. 3. Representación de la pirámide de escalas.

coincidentes con la escala n se denota como $f_{req}(x,y,n)$. En general, para cualquier distribución de las peticiones, las teselas son solicitadas con probabilidad (1).

$$P_{req}\{T(i,j,n)\} = \int_{y=y_{n,i}}^{y_{n,i+1}} \int_{x=x_{n,i}}^{x_{n,i+1}} f_{req}(x,y,n)\,dxdy \qquad (1)$$

Aunque este resultado general resulta de utilidad a la hora de analizar las peticiones no teseladas dirigidas a un *proxy-cache*, en los análisis iniciales puede simplificarse suponiendo una distribución uniforme de los centroides de las peticiones dentro de cada tesela, o bien que las peticiones están restringidas a la rejilla de referencia (como en el caso de un WMS-C). En estas condiciones la probabilidad de acceso a una tesela de coordenadas $T(i,j,n)$ y de tamaño $\Delta x \Delta y$ es

$$P_{req}\{T(i,j,n),t\} = f_{req}(x,y,n,t)\,\Delta x \Delta y \qquad (2)$$

Para una petición individual de una tesela situada en la coordenada (discretizada) (i,j,n), en un instante t, la latencia observada viene determinada por (3):

$$\begin{aligned}\tau(i,j,n,t) =& P_h\{T(i,j,n)\}(t)\tau_h + \\ &+ (1 - P_h\{T(i,j,n)\}(t))\,\tau_m\end{aligned} \qquad (3)$$

Combinando las definiciones de (3) y (1) se obtiene una expresión probabilística conjunta de las peticiones que permite calcular una latencia media del servicio:

$$\begin{aligned}\tau(t) = &\sum_{\langle ijn\rangle} (\tau_m - P_h\{T(i,j,n)\}(t)(\tau_m - \tau_h)) \cdot \\ &\cdot P_{req}\{T(i,j,n)\}\end{aligned} \qquad (4)$$

O equivalentemente

$$\begin{aligned}\tau(t) = \tau_h \sum_{\langle ijn\rangle} &\left(\frac{\tau_m}{\tau_h} - P_h\{T(i,j,n)\}(t)\left(\frac{\tau_m}{\tau_h} - 1\right)\right) \cdot \\ &\cdot P_{req}\{T(i,j,n),t\}\end{aligned} \qquad (5)$$

Suponiendo que el tiempo de acceso a la caché es constante y que también lo es el tiempo necesario para generar nuevos objetos (al menos en término medio), puede definirse la "ganancia de uso de caché" como el incremento porcentual de rendimiento en los aciertos de caché:

$$G_c = \frac{\tau_m}{\tau_h} \tag{6}$$

y podemos reducir (4) a

$$\tau(t) = \tau_h \sum_{\langle ijn \rangle} \left(G_c - P_h\left\{T(i,j,n)\right\}(t)\left(G_c - 1\right)\right) \cdot \cdot P_{req}\left\{T(i,j,n), t\right\} \tag{7}$$

Donde ya pueden identificarse algunos componentes que deben ser caracterizados al menos localizadamente. En este punto no deben asumirse más simplificaciones probabilísticas, que si bien podrían simplificar el diseño y funcionamiento de estos sistemas, eliminarían la información disponible para mejorar la gestión[2]. Por lo tanto, parece necesario estudiar cuáles son las características típicas de estas propiedades y cómo se relacionan con las métricas de calidad.

Algunas características relevantes del modelo considerado son:

- No hay independencia estadística entre $P_h\left\{T(i,j,n)\right\}(t)$ y $P_{req}\left\{T(i,j,n), t\right\}$, ya que resulta evidente que el estado de la caché está vinculado íntimamente a la historia de las peticiones de servicio.
- No hay invarianza temporal durante el régimen transitorio del sistema.
- La función densidad de probabilidad (y por ende la probabilidad expresada en (1)) no es uniforme y es probable que presente vínculos directos con la estructura espacial de la información subyacente.

III. Mecanismos de carga de la cache

En un servicio de mapas teselado, las teselas pueden ser generadas la primera vez que son pedidas (*carga dinámica*), cacheándose en este mismo proceso. De esta forma, la primera vez que se pide una tesela se produce un fallo (*miss*) de caché y la petición se resuelve aproximadamente a la misma velocidad que si se tratase de un servicio WMS tradicional. Peticiones subsiguientes de este mismo objeto son aceleradas en gran medida (según la ganancia expresada en (6)) dado que el objeto ya ha sido generado previamente, produciéndose sucesivos aciertos (*hit*) de caché.

La principal ventaja de este método es que no requiere preprocesamiento y se produce una rudimentaria autogestión de la caché, puesto que las propias peticiones acumuladas en la inherente memoria del sistema equivalen a una evaluación implícita de la probabilidad $P_h\left\{T(i,j,n)\right\}$ expuesta en (1). El resultado indirecto es que sólo los datos pedidos son cacheados (los que efectivamente tienen $P_{req}\left\{i,j,n\right\} \neq 0$), ahorrando por tanto recursos de almacenamiento. El inconveniente de este método es que la devolución de teselas no experimenta ninguna mejora de QoS en la primera petición, reduciendo la calidad en la experiencia del usuario en las teselas poco demandadas o ya expiradas.

La otra posibilidad es que las teselas se generen mediante *seeding*. *Seeding* es el proceso mediante el cuál las teselas se generan y cachean de forma automática anticipándose a las peticiones de los usuarios. La ventaja de este proceso es que mejora en gran medida la experiencia percibida por el usuario. El inconveniente es que se trata de un proceso que consume mucho tiempo y recursos de almacenamiento.

Obviamente, conviene reducir este tiempo de puesta en servicio para conseguir alcanzar cuanto antes la QoS esperada respecto al tiempo de respuesta (fase de QoS estacionaria). Puede reducirse la penalización experimentada hasta que se completa el proceso de precarga cargando primero las teselas con mayor probabilidad de ser pedidas, dejando para el final aquellas cuya probabilidad de ser pedidas sea menor.

Sin embargo, dado que se trata de la puesta en marcha del servicio y no se tiene conocimiento previo sobre cómo será la distribución de peticiones, cabe preguntarse en base a qué pueden realizarse las predicciones necesarias. Algunos de los sistemas de caché estudiados [7][8] se basan para ello en la naturaleza espacial de los datos que contienen. Una opción razonable es generar primero las teselas para los niveles de resolución más altos que, al menos intuitivamente, recibirán una mayor densidad de peticiones, y generar posteriormente las teselas pertenecientes a escalas más bajas o dejar que éstas vayan poblando la caché a medida que son pedidas.

Otra posibilidad interesante que ofrecen algunos de los sistemas de caché existentes es la de indicar la región geográfica del conjunto de teselas que se quiere cachear para las escalas indicadas. Sin embargo, estas posibilidades manuales se basan en la intuición del administrador. Durante el estudio del arte llevado a cabo no se han detectado indicios acerca de la exploración de mecanismos automáticos o semi-automáticos para la identificación avanzada de regiones potencialmente candidatas para ser generadas durante el proceso de *seeding*.

IV. WMSCWrapper: Prototipo de caché WMS-C

Para la experimentación con distintas estrategias de cache se ha implementado el prototipo *WMSCWrapper*, desarrollado en el IDELab (*Laboratorio de Infraestructuras Espaciales*) de la Universidad de Valladolid[3]. Está disponible como proyecto *Open Source* y tiene una arquitectura adecuada para la inclusión de componentes y sondas experimentales. Otro motivo de tal elección es que el resto de implementaciones de filtros WMS-C examinadas están fuertemente orientadas hacia el exclusivo soporte del servicio teselado de mapas para obtener la mencionada ganancia de caché.

La implementación elegida como banco de pruebas para los experimentos es un *proxy web* que se enmarca dentro del diagrama taxonómico de [2] (ver Fig. 1) en la *Web Layer* como un *Web caching* y en la taxonomía vertical como una *Object Cache* en la que además se implementarán funcionalidades de procesado típicas de la capa de aplicación.

El *WMSCWrapper* es un servicio implementado como un conjunto de *servlets* que exponen los métodos de la recomendación WMS y la extensión WMS-C en un interfaz OGC. Asimismo, permite también el acceso a la información

[2]Por ejemplo, suponer una caché completa reduciría (7) a $\tau(t) = \tau_m G_c = \tau_h$

[3]http://www.idelab.uva.es

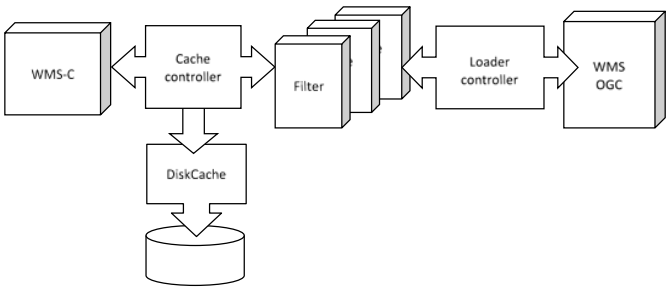


Fig. 4. Diagrama general del flujo de información en el *proxy cache WMSCWrapper*.

cacheada por medio de otros interfaces, como los métodos REST utilizados por *Google Earth* al solicitar teselas en KML, o las interfaces propietarias de *Google Maps* y *Microsoft Bing Maps*. Se configura de forma sencilla mediante la especificación de una serie de capas disponibles en otro servicio WMS y unas características de tratamiento de los objetos en la caché. Cada petición es analizada en busca de los parámetros obligatorios y opcionales de la recomendación y después transferida a una serie de componentes intercambiables que pueden pre-procesar o post-procesar la información según las necesidades (Fig. 4 y 5).

Existen componentes que realizan las siguientes operaciones:

- Validación de parámetros de la petición HTTP recibida.
- Obtención de teselas a partir del servicio del *backend* (componente *Loader*, Fig. 5).
- Ejecución de algoritmos de *metatiling* con procesamiento multi-hilo.
- Inclusión de marcas de agua.
- Etiquetado para depuración y supervisión del funcionamiento del servicio.
- Mantenimiento de la caché (componente *Cache Manager*, Fig. 5).
- Recopilación de estadísticas en índices espaciales (componente *Spatial Index*, Fig. 5).
- Almacenamiento de teselas tanto en el sistema de ficheros como en base de datos (componente *Storage*, Fig. 5).

V. Metatiling

Un importante mecanismo que puede ayudar a mejorar el rendimiento de los servicios WMS-C es la carga dinámica mediante heurísticas. Ante una petición de un usuario, el sistema puede predecir cuál puede ser la siguiente petición o grupo de peticiones. Generando el resultado de estas predicciones dentro del intervalo de tiempo entre dos peticiones sucesivas se puede conseguir una mejora sustancial en la experiencia de usuario, a condición de que las predicciones sean acertadas.

El prototipo *WMSCWrapper* incorpora una sencilla implementación de esta idea; al recibir una petición se pregeneran y *cachean* las teselas adyacentes a la misma, puesto que existe una probabilidad considerable de que el usuario se vaya desplazando por el mapa de forma continua. Se ha detectado la necesidad de traducir esta idea intuitiva en algo riguroso y sistemático que permita cuantificar el grado de interdependencia entre teselas vecinas y obtener una predicción de los vectores de desplazamiento típicos de las zonas solicitadas.

Fig. 6. Problema del etiquetado redundante (Figura extraída de [9]).

Una línea de investigación insinuada por las expresiones (5) y (7) es la mejora, mediante estrategias de racionalización de la carga ejecutadas en el *proxy*, del parámetro τ_m. En el proceso de generación de teselas hay diversos cuellos de botella que se pueden mejorar. Una de las partes del proceso de dibujado es el acceso a los almacenes externos de información geográfica. Ya que las consultas suelen estar optimizadas mediante índices espaciales, resulta más eficiente realizar una petición de una tesela de tamaño $n \times n$ que n^2 peticiones por separado.

Otro beneficio obtenido indirectamente es la reducción del problema del etiquetado fraccionado o redundante provocado por el teselado de las peticiones. Se produce cuando un determinado fenómeno, como por ejemplo un río, un lago o una carretera, se extiende a lo largo de múltiples teselas, de forma que el servidor de mapas añade una etiqueta para el mismo fenómeno que se repite en cada una de ellas, tal y como se aprecia en la Fig. 6.

Es por tanto una práctica habitual el generar peticiones de mayor tamaño que la tesela a *cachear* (esta super-tesela se denomina *metatile*) y posteriormente postprocesarlas para aprovechar la información disponible y generar nuevas teselas.

Las implementaciones investigadas permiten especificar el número adicional de teselas alrededor de la realmente solicitada (*buffer de B teselas*) que se van a pedir en una sola petición al servidor WMS. De esta manera se le pide al servidor de mapas una *metatile* de tamaño $(2B+1)^2$ teselas centrada en el elemento realmente solicitado.

En un escenario de caché no completa (pero no vacía) esta elección del área a generar no resulta muy eficiente, puesto que es muy probable que algunas de las teselas próximas a la solicitada ya estén disponibles en la caché.

Partiendo de la suposición de que la zona que engloba la tesela solicitada no está homogéneamente cargada, se ha desarrollado un algoritmo para la elección óptima de las *metatiles* a generar. El procedimiento ilustrado en la Fig. 7 busca obtener, en función del estado de la caché, la *metatile* que, conteniendo la tesela solicitada (no necesariamente centrado en la misma), minimice la correlación espacial:

$$R_n(\Delta i, \Delta j) = \sum_{l=i-N}^{j+N} \sum_{m=j-N}^{j+N} h\left[l + \Delta i, m + \Delta j, n\right] \quad (8)$$

donde la función *booleana* definida en el dominio discreto $h[i, j, n]$ se define según (9):

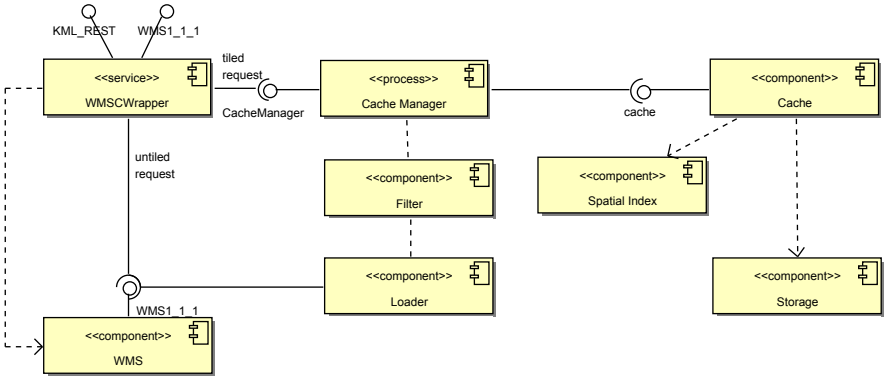


Fig. 5. Arquitectura de componentes del prototipo *WMSCWrapper*

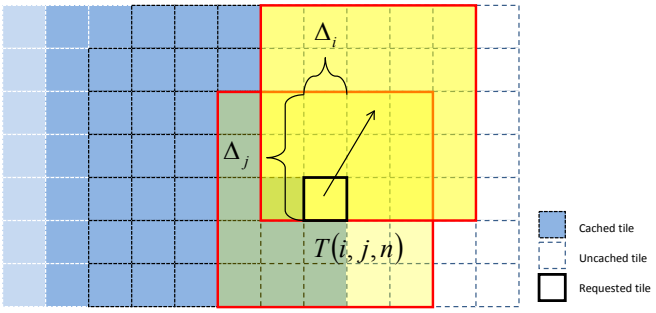


Fig. 7. Estrategia de mínima correlación con la caché para la generación de *metatiles*

Tabla I
LATENCIAS MEDIAS PARA LA OBTENCIÓN DE UN OBJETO A PARTIR DEL SERVICIO WMS ORIGINAL

buffer (B)	$\tau_{m,metatile}$	$\tau_{m,metatile\_n}$	$G_{metatiling}$
0 (sin metatiling)	1454,10 ms	1454,10 ms	1
1 (metatile 3x3)	2933,94 ms	325,99 ms	4,46
2 (metatile 5x5)	5660,63 ms	226,42 ms	6,42
3 (metatile 7x7)	9561,54 ms	195,13 ms	7,45

$$h[i,j] = \begin{cases} 1 & \textit{si la tesela T(i,j,n) está en la caché} \\ 0 & \textit{en caso contrario} \end{cases} \quad (9)$$

Interpretando la correlación en (8) como una medida de la semejanza de la información contenida en ambos objetos (*metatile* y caché), parece evidente que la *metatile* que tenga una menor correlación espacial con el estado de la caché es aquella que proporciona la mayor información al sistema, puesto que la información que contiene es complementaria en mayor grado a la ya disponible. En la Fig. 7 se ilustra la configuración con la que se consigue un mínimo en la redundancia o en la información mutua.

La implementación realizada de esta técnica incluye además un procedimiento en segundo plano para realizar el post-proceso de las teselas adicionales. De esta forma se busca minimizar también el tiempo de servicio del *proxy* (τ_h).

Para validar la hipótesis de que puede obtenerse una mejora mediante el uso del *metatiling*, se ha realizado un experimento utilizando el prototipo *WMSCWrapper* como caché de teselas y el servicio de Ocupación del Suelo (CORINE) del Instituto Geográfico Nacional (IGN) como servidor de mapas remoto. Se han realizado 2000 peticiones de mapas distintas al *proxy*, partiendo de una caché inicialmente vacía, para distintos tamaños de *metatile*, y se han analizado las latencias experimentadas en la devolución de las teselas.

Los resultados del experimento se recogen en la Tabla I. La columna $\tau_{m,metatile}$ corresponde a la latencia media de un fallo de caché para los distintos tamaños de *metatile*, que incluye los retardos de transmisión y propagación en la red, el tiempo de generación de la imagen en el servidor de mapas remoto y los tiempos de procesamiento en el *proxy cache* de teselas. Los valores $\tau_{m,metatile\_n}$ se obtienen normalizando aquellos de la columna anterior por el número de teselas que componen el *metatile* ($[2B+1]^2$). En la última columna se calcula la ganancia por el uso de *metatiling*, medida como la aceleración media en la devolución de una tesela frente a no utilizar *metatiling*, tal y como se refleja en la expresión (10).

$$G_{metatiling}(B) = \frac{\tau_{m,metatile\_n}(0)}{\tau_{m,metatile\_n}(B)} \quad (10)$$

Los resultados reflejan que la latencia involucrada en la petición de un *metatile* aumenta al hacerlo el tamaño del *buffer* utilizado. Sin embargo, aumenta en menor proporción que el número de teselas de que se compone el *metatile*. Por tanto, la latencia media para la obtención de cada tesela individual decrece al aumentar el tamaño del *metatile* solicitado al servidor de mapas remoto.

Un factor limitante a la hora de decidir el tamaño de metatile a utilizar es la cantidad de memoria que requiere la generación de la imagen. Nótese que para un factor de metatile de 5x5 (*buffer* de dos unidades), para una petición de tesela de 256x256 pixels la imagen generada es de 1280x1280 pixels.

Considérese, por ejemplo, que en el servidor de mapas Geoserver [10], la cantidad máxima de memoria permitida para una única petición WMS (parámetro *maxRequestMemory*) viene configurada a 16MB. Esta cantidad de memoria permite generar una imagen de 2048x2048 pixels con 4 bytes/pixel, lo que es equivalente a un *metatile* de 8x8. Si el SLD (*Styled Layer Descriptor*) contiene dos elementos *FeatureTypeStyle* para el dibujado de distintos tipos de línea, el tamaño máximo de la imagen se limita a 1448x1448 pixels (*metatile* máximo de 5x5).

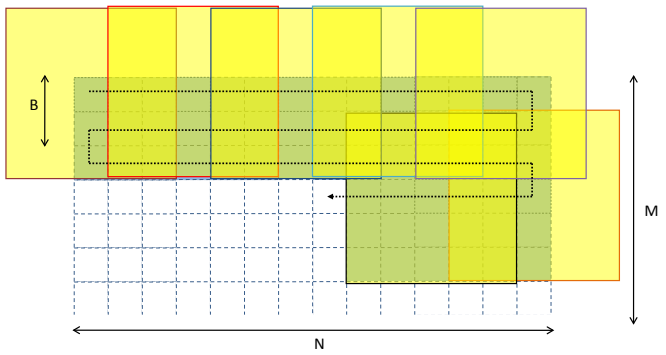


Fig. 8. *Metatiles* pedidos al servidor de mapas remoto durante una tarea de *seeding*, utilizando *metatiles* centradas en la tesela a *cachear*, con $B = 2$.

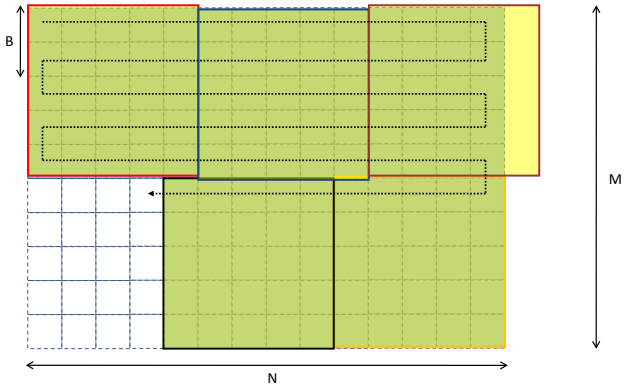


Fig. 9. *Metatiles* pedidos al servidor de mapas remoto durante una tarea de *seeding*, utilizando la estrategia de *metatiling* de mínima correlación con la caché, con $B = 2$.

A. Rendimiento del metatiling en la tarea de seeding

Como se comentó en la Sección III, las técnicas de *seeding* permiten realizar una precarga de teselas, anticipándose a las peticiones de los usuarios, con el objetivo de mejorar la experiencia de los mismos. Una práctica habitual consiste en ir recorriendo en *zig-zag* la región geográfica para la generación de teselas. En este caso, es sencillo analizar las mejoras que pueden conseguirse mediante la aplicación del *metatiling*.

Suponiendo una rejilla rectangular de $M \times N$ teselas, para pregenerar todo el contenido mediante la técnica de *metatile* con tamaño de *buffer* B y centrada en la tesela solicitada (ver Fig. 8), se requieren aproximadamente $\frac{M \times N}{(B+1)^2}$ peticiones al servidor de mapas remoto. Como se puede observar, i.e. el último *metatile* pedido contiene 16 teselas que ya estaban previamente en la caché, con lo que tan sólo un 36% de la imagen añade información nueva al sistema.

Mediante la estrategia de mínima correlación (ver Fig. 9), el número de peticiones se reduce a $\frac{M \times N}{(2B+1)^2}$ aproximadamente, obteniendo una ganancia de $\frac{(B+1)^2}{(2B+1)^2}$. En la figura se muestra cómo los *metatiles* se desplazan de forma que no se producen solapamientos con aquellos previamente solicitados, maximizándose por tanto la información añadida al sistema.

Así, utilizando un tamaño de *buffer* de 3 unidades ($B = 3$), se reduce en un factor de 16 el número de *metatiles* solicitados al *backend*.

B. Rendimiento del metatiling durante la carga dinámica

Para evaluar el rendimiento de las estrategias de *metatiling* durante el proceso de carga ante las peticiones de los usuarios, se ha utilizado el prototipo de caché de teselas *WMSCWrapper*, descrito en la Sección IV, como banco de pruebas.

Para simular las peticiones de los usuarios, se ha hecho uso de los registros de acceso o *logs* del servicio de mapas WMS-C de Cartociudad[4] facilitados por el IGN. De esta forma se consigue un patrón de acceso que refleja fielmente el comportamiento real de los usuarios, y los resultados son más representativos que en el caso de utilizar un patrón de peticiones "sintético".

Como servidor de mapas remoto se ha utilizado el servicio de CORINE. No se ha podido utilizar el propio servicio de Cartociudad, cuyos registros de peticiones se han analizado, por problemas en la disponibilidad del servicio.

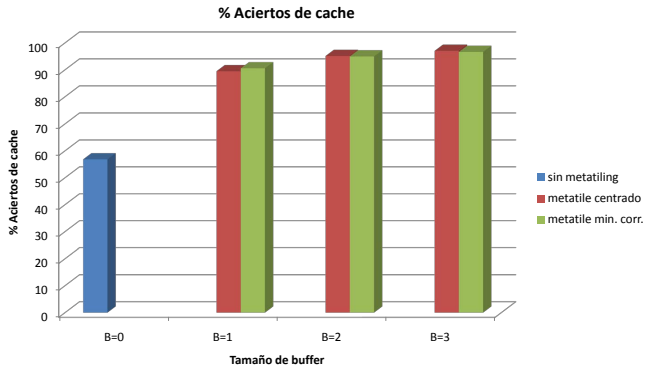


Fig. 10. Comparativa de aciertos de caché, para distintos tamaños de *buffer* y configuraciones de *metatile*.

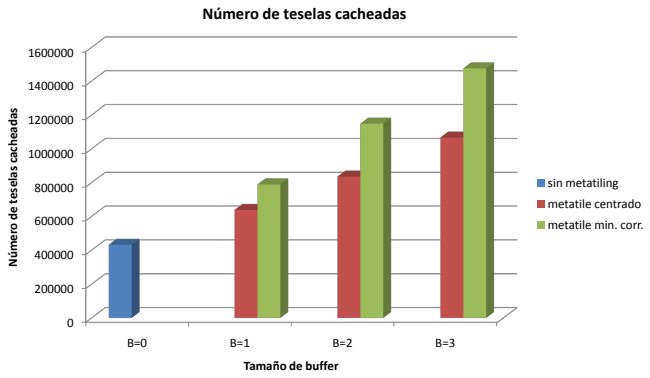


Fig. 11. Comparativa de objetos cacheados al finalizar la tarea, para distintos tamaños de *buffer* y configuraciones de *metatile*.

Un total de 1.000.000 peticiones se han realizado a la caché, utilizando distintas configuraciones de *metatiling*. Para cada una de estas configuraciones se ha recogido la tasa de aciertos de caché conseguida, y el número de objetos almacenados en la misma una vez completada la tarea. En todas las pruebas se parte de una caché vacía. Los resultados de los experimentos se muestran en las Figuras 10 y 11.

[4]http://www.cartociudad.es

Como se puede observar, tanto la tasa de acierto como el número de objetos *cacheados* aumentan al hacerlo el tamaño del *buffer* utilizado. A igualdad de éste, ambas estrategias de *metatiling* obtienen unas tasas de acierto muy próximas. Sin embargo, mediante la configuración de mínima correlación, el número de objetos introducidos en la caché es notablemente superior que con la configuración de *metatile* centrado en la tesela solicitada. Esta diferencia aumenta al hacerlo el tamaño del *metatile*.

La ventaja conseguida con la configuración de *metatiling* de mínima correlación es que, manteniendo la tasa de fallos de caché, y por ello manteniendo el número de peticiones al servidor WMS remoto, se consigue una mayor cantidad de imágenes de mapa pregeneradas almacenadas en la caché para satisfacer más rápidamente posibles peticiones futuras.

VI. Conclusiones

En el actual servicio WMS los mapas necesitan generarse al vuelo. Este hecho dificulta que el servicio pueda responder adecuadamente a un número elevado de peticiones simultáneas. Puede afrontarse este problema reduciendo los dominios de los parámetros a un conjunto discreto de valores. La zona geográfica queda dividida en una rejilla compuesta por elementos de geometría predefinida (denominados teselas) e identificables mediante índices enteros, posibilitando la actuación de mecanismos de caché.

Las características particulares de la información geográfica de una caché espacial permiten que los algoritmos de reemplazo y carga inicial en estos sistemas se beneficien de las características espacio temporales del comportamiento de los usuarios y la correlación multidimensional de las teselas.

Mediante las técnicas de *seeding* las teselas se generan y cachean de forma automática, anticipándose a las peticiones de los usuarios y mejorando la QoS percibida por los mismos.

Habitualmente, los usuarios se desplazan de forma continua por el mapa, por lo que es probable que al recibir la petición de una tesela de mapa, las teselas adyacentes sean solicitadas a continuación.

En este trabajo se ha demostrado que puede obtenerse una mejora de rendimiento al solicitar al servidor de mapas remoto una tesela de gran tamaño (*metatile*), que cubra también otras teselas adyacentes a la solicitada, y luego "recortarla" e introducir los fragmentos en la caché, frente a realizar las peticiones de cada tesela individual cubriendo el mismo área.

Se ha diseñado y evaluado una estrategia de mínima correlación con la caché para la generación predictiva de teselas basada en *metatiles*, de forma que la información solicitada al servidor de mapas complemente en mayor medida a la ya almacenada en la caché.

El prototipo *WMSCWrapper* desarrollado posibilita la experimentación con los mecanismos de gestión anteriores, así como el funcionamiento como banco de trabajo para el estudio de nuevas estrategias.

Agradecimientos

Este trabajo ha sido realizado como parte del proyecto CENIT España Virtual[5] (ref. CENIT 2008-1030), cofinanciado por el CDTI, dentro del programa Ingenio 2010 y por el CNIG.

[5]http://www.españavirtual.org/

Referencias

[1] H. Hassanein, Z. Liang, and P. Martin. Performance comparison of alternative web caching techniques. In *Proceedings of the Seventh International Symposium on Computer and Communications*, 2002.

[2] Tony C. Shan and Winnie W. Hua. *Encyclopedia of Information Communication Technology*, chapter Data Caching Patterns, pages 139–149. IGI global, 2009 edition, 2009.

[3] OGC. OpenGIS web map service (WMS) implementation specification. http://www.opengeospatial.org/standards/wms, 2009.

[4] OSGeo. WMS tiling client recommendation - OSGeo wiki [online]. 2008. Disponible en: {http://wiki.osgeo.org/wiki/WMS_Tiling_Client_Recommendation} [última consulta: 28 de Junio de 2011].

[5] Keith Pomakis Joan Masó and Núria Julià, editors. *Web Map Tile Service Implementation Standard*. Number OGC 07-057r7 in OpenGIS Implementation Standard. Open GIS Consortium Inc., April 2010.

[6] Lucian Plesea. The design, implementation and operation of the JPL OnEarth WMS server. In J.T. Sample, K. Shaw, S. Tu, and M. Abdelguerfi, editors, *Geospatial Services and Applications for the Internet*, pages 93–109. Springer, Berlin, 2008.

[7] MetaCarta. TileCache, from MetaCarta labs [online]. Disponible en: http://tilecache.org/ [última consulta: 28 de Junio de 2011].

[8] OpenGeo. GeoWebCache [online]. Disponible en: http://geowebcache.org [última consulta: 28 de Junio de 2011].

[9] Metatiles — GeoWebCache user manual [online]. Disponible en: http://geowebcache.org/docs/current/concepts/metatiles.html [última consulta: 28 de Junio de 2011].

[10] Sito web de GeoServer [online]. Disponible en: http://geoserver.org [última consulta: 28 de Junio de 2011].

Sesión 1.B
Servicios multimedia y web semántica

u-Bcast: geolocalización de contenidos multimedia

Sergio Machado, José M. Yúfera
Departamento de Ingeniería Telemática
Universitat Politècnica de Catalunya
Escola d'Enginyeria de Telec. i Aero. De Castelldefels, Esteve Terradas, 7. 08860 Castelldefels
smachado@entel.upc.edu, yufera@entel.upc.edu

***Resumen*- Este documento detalla la arquitectura de un servicio de acceso y reproducción de información multimedia georeferenciada basado en relaciones de tipo red social al que hemos denominado u-Bcast. Dicha arquitectura combina cuatro servicios diferentes: red social, *streaming*, geolocalización y notificaciones. Se han utilizado soluciones estándar en el diseño del servicio: una solución LAMP para implementar la red social y el almacenamiento de la información multimedia, RTSP/RTP para el *streaming* de la información y MQTT para el servicio de notificaciones. Los mapas están proporcionados por OpenStreetMap, un proyecto colaborativo para crear mapas de libre acceso y editables. La parte cliente está implementada para ser accesible mediante una web y desde dispositivos con sistema operativo Android.**

***Palabras Clave*-Android, geolocalización, multimedia.**

I. INTRODUCCIÓN

Desde 2009 los servicios basados en geolocalización presentan un notable crecimiento, como demuestra el aumento del número de usuarios de aplicaciones como Foursquare o Facebook Places. Las grandes motivaciones para usar esos servicios son básicamente dos: a) obtener información sobre puntos de interés o usuarios, como recomendaciones, por ejemplo; b) encontrar descuentos o promociones especiales. Los servicios basados en geolocalización obviamente tienen sentido en dispositivos móviles tales como teléfonos inteligentes o tabletas ya que acompañan al usuario allí dónde se desplaza y, por lo general, disponen de GPS. Recientes estudios sitúan el índice de penetración de tales dispositivos en España en el orden del 25% [1].

Por otro lado, las redes sociales en Internet se han convertido en un fenómeno social que revoluciona la forma de comunicarse e interactuar de sus usuarios. Redes sociales hay de varios tipos: Facebook, basada en las relaciones de amistad, Linkedin para relaciones de tipo profesional, Twitter para *microblogging*, etc.

Asimismo, la banda ancha en el hogar y las redes 3G en los terminales móviles han permitido la proliferación de servicios de *streaming*, incluso con calidad HD, tales como Youtube, Spotify o Netflix entre otros.

En este artículo se describe u-Bcast, un servicio que aúna los conceptos de geolocalización, red social y *streaming* ofreciendo a sus usuarios el poder compartir información multimedia asociada a puntos geográficos. Este servicio puede tener aplicación, por ejemplo, en la implementación de nuevas actividades de *geocatching*, publicidad personalizada o rutas turísticas.

Existen diferentes trabajos que permiten servicios similares [2]-[5], pero u-Bcast se diferencia de ellos, entre otras cosas, en que integra redes sociales, geolocalización y *streaming* en un único sistema; en que está diseñado según una arquitectura robusta, conocida y escalable; y en que ha sido implementado y probado en Android, una plataforma para sistemas móviles en rápida expansión. Además, y a diferencia del objetivo de este artículo, las diferentes referencias centran su estudio únicamente en la descripción del servicio y no en su arquitectura e implementación. Asimismo, u-Bcast utiliza OpenStreetMap como servicio de mapas, apostando por este proyecto colaborativo que permite editar mapas y, por lo tanto, alcanzando un nivel de detalle poco común en el más ampliamente utilizado servicio de Google Maps y que, además, no tiene las restricciones de licencia de este último. La implementación del servicio de notificaciones utilizando el protocolo MQTT es también novedosa.

El artículo se organiza de la siguiente manera: en la sección II se describe el servicio ofrecido por u-Bcast, explicando su arquitectura, el modelo de datos, la API y el sistema de notificaciones utilizados; en la sección III se muestran los elementos que permiten reproducir la información multimedia mediante *streaming*; en la sección IV se describe la implementación del cliente u-Bcast, tanto el caso web como el de la aplicación utilizada en Andoid; y finalmente, en la sección V se comentan las conclusiones del trabajo.

II. DESCRIPCIÓN DEL SERVICIO

u-Bcast forma una red social que permite compartir información multimedia asociada a puntos geolocalizados y parametrizada por tiempo, permisos y etiquetas. La información multimedia está constituida por vídeos y audios subidos por los usuarios y reproducibles vía *streaming* por otros usuarios de la red social únicamente cuando se encuentren en el entorno del punto geográfico al que se ha asociado la información.

Se han desarrollado dos aplicaciones que permiten el acceso y uso de la red social. Por un lado, una aplicación web que, básicamente, permite a los usuarios la gestión de sus relaciones, la búsqueda de la información que tienen disponible en una zona geográfica determinada y la posibilidad de subir información multimedia al servidor de u-Bcast. Por otro lado, y dado que la información es accesible sólo en el entorno del punto geográfico, se ha desarrollado una aplicación cliente para teléfonos móviles con sistema

ISBN: 978-84-694-5948-5

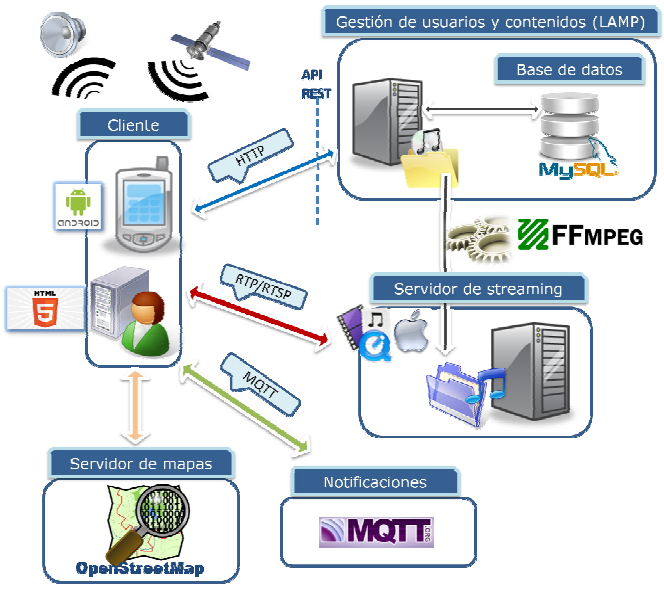


Fig. 1. Arquitectura u-Bcast.

Android, que a las capacidades anteriores comentadas añade la reproducción en *streaming* de la información a la que tenga acceso de modo no invasivo, esto es, al usuario se le notifica que tiene información disponible y él decide si reproducirla u obviarla.

A. *Arquitectura del servicio*

El servicio u-Bcast está subdividido en seis componentes fundamentales: las aplicaciones cliente, el servidor de mapas y los subsistemas de gestión de usuarios e información, de *streaming,* de notificaciones y de transcodificación. La Fig.1 muestra cada uno de estos componentes así como los protocolos de comunicaciones usados en cada caso.

El subsistema de gestión se ha implementado sobre una solución LAMP: Linux como sistema operativo; Apache como servidor web; MySQL como gestor de base de datos; y PHP como lenguaje de programación. Asimismo, toda la operativa de acceso de los clientes al subsistema se realiza a través de una API basada en REST [6].

Para el servicio de mapas hemos optado por el uso de OpenStreetMap [7], un proyecto colaborativo para crear mapas libres y editables. La elección se ha basado en los problemas de licencia que podría acarrear utilizar, por ejemplo, Google Maps [8]. A diferencia de Google Maps, OpenStreetMap no tiene una librería estable para poder representar estos mapas en Android, aunque sí existe una en desarrollo [9]. Sin embargo, y para evitar problemas derivados de la no estabilidad de la librería en desarrollo, nosotros hemos optado por utilizar una librería propia para trabajar con OpenStreetMap desde Android.

Dado que los sistemas Android únicamente aceptan formatos contenedores y *codecs* multimedia muy determinados, y de cara a facilitar la subida de ficheros, hemos implementado un servicio de transcodificación utilizando el software FFmpeg [10], que convierte cualquier formato subido por el usuario a cualquiera de los aceptados por la plataforma Android.

Por último, para el servicio de notificaciones hemos implementado una solución basada en el protocolo MQTT (MQ Telemetry Transport) [11], tal y como se explica en el apartado D de esta sección.

B. *El modelo de datos*

La Fig. 2 muestra el modelo entidad relación simplificado usado en u-Bcast. Se divide en diferentes entidades, lo que permite mantener el sistema particionado en bloques estancos pero, al mismo tiempo, establece relaciones entre ellos, de modo que el sistema escala adecuadamente con la adición de nuevas funcionalidades o la alteración de alguna de sus entidades.

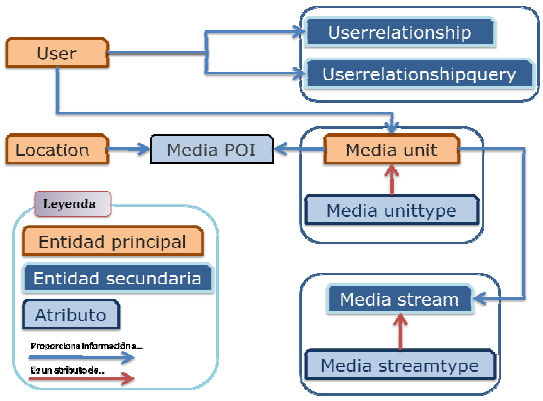


Fig. 2. Modelo de datos de u-Bcast.

En el modelo distinguimos entidades principales de entidades secundarias, además de otras entidades que son atributos de entidades principales o secundarias. Se entiende por entidad principal aquella que mantiene información básica de una parte del servicio. Una entidad secundaria proporciona información extendida o relaciona entidades principales. Finalmente, la funcionalidad de las entidades atributo consiste en añadir información y catalogar a entidades principales o secundarias.

A continuación describimos las diferentes entidades u-Bcast.

Entidades principales:

Las entidades principales manejan la información básica del servicio, esto es, que los usuarios puedan asociar una información multimedia (actualmente audio y vídeo) a un determinado punto geográfico con restricciones sobre quién puede acceder a dicha información, restringirla a un periodo temporal y etiquetarla, de tal modo que otro usuario con permisos, dentro del rango temporal establecido y con interés en el etiquetado pueda acceder a la información vía *streaming*.

La entidad *User* contiene básicamente información de registro del usuario, léase un identificador o alias de usuario, contraseña, información de contacto y relaciones con la información que ha compartido.

La entidad *Location* contiene la información de geolocalización: latitud, longitud, rango temporal y etiquetado. Las relaciones entre las diferentes unidades multimedia permiten a) que una misma unidad esté vinculada a varias localizaciones, b) que varias localizaciones compartan una misma unidad y c) que un conjunto de unidades estén asociadas a un conjunto de localizaciones formando una ruta. Estas relaciones se logran a través de la entidad atributo denominada *Media POI*.

La información multimedia se recoge en la entidad *Media Unit* en la que se recoge el fichero que contiene la información. Los atributos de este fichero se almacenan en la unidad atributo *Media Unit Type* y catalogan y etiquetan a la instancia *Media Unit* a la que hacen referencia.

Entidades secundarias:

La implementación en el servicio de las funcionalidades tanto de red social como de *streaming*, necesitan la definición de entidades secundarias que relacionen por un lado usuarios entre sí, y por otro geolocalizaciones con información multimedia.

Las entidades *Userrelationship* y *Userrelationshipquery*, relacionan a los usuarios entre sí con un vínculo de interés mutuo o unidireccional, esto es, que dos usuarios pueden estar interesados en los contenidos que cada uno aporta, o bien, sólo uno de ellos tenga interés por el otro. Por cuestiones de privacidad la suscripción a los contenidos de un usuario debe ser autorizada por éste. El nivel de privacidad varía entre el público (que no necesita autorización alguna) y la definición de reglas de acceso al nivel de etiquetado del contenido. Por último, esta parte del modelo también permite la categorización de usuarios en grupo de tal modo que colectivamente funcionan como un usuario individual pero manteniendo además su perfil propio.

El proceso de *streaming* descrito en la Sec.III requiere de información adicional que se mantiene en la entidad *Media Stream* y su atributo *Media Stream Unite*. Además, esta entidad permite la abstracción en una única unidad de diferentes *Media Unit*, de tal modo que la misma información puede tener codificaciones distintas estableciendo criterios de calidad de *streaming* en función, por ejemplo, del consumo de datos que pueda hacer el terminal móvil.

C. La API REST del servicio

Los accesos al servicio se realizan vía una API REST. REST [6] es el acrónimo de REpresentational State Transfer, una arquitectura para el diseño de servicios Web utilizando el protocolo HTTP de forma más simple que las tradicionales implementaciones vía CORBA, RPC o SOAP. Las aplicaciones que implementan esta arquitectura utilizan peticiones HTTP para enviar datos (creación y/o actualización), leerlos o eliminarlos. Todas las peticiones llevan consigo toda la información necesaria, es decir, el estado, para que el servidor pueda resolver la petición.

Toda la operativa REST se resuelve utilizando cuatro verbos que no dejan de ser operaciones HTTP: GET, utilizado para leer un recurso; PUT y DELETE, usados para alterar el estado de un recurso atómicamente; y POST, que se utiliza a modo de cajón de sastre para resolver operaciones que no cuadran exactamente con la filosofía de los verbos anteriores.

Hay APIs REST, y en concreto la API de u-Bcast, que requieren autenticación. La forma habitual de implementar la autenticación en esta arquitectura es mediante OAuth [12]. Esta solución consta de tres entidades: el *servicio*, esto es, la aplicación que provee el servicio; el *consumidor*, que es la aplicación que desea acceder a los recursos protegidos; y los *recursos protegidos*, es decir, la información en el servicio que requiere autenticación por parte del usuario. En líneas generales la autenticación se resuelve en tres pasos. Primero el consumidor se registra en el servicio obteniendo una clave y una palabra secreta para identificarse. En un segundo paso el consumidor solicita un *token* (testigo) de acceso que identifica el acceso de una aplicación a los recursos protegidos de un usuario. Por último, si el usuario concede acceso al consumidor, éste dispondrá de un *token* válido para acceder a los recursos protegidos del usuario.

La API REST de u-Bcast ha sido implementada totalmente en PHP utilizando la biblioteca OAuth-PHP [13] para implementar consumidores y servicios. Hemos seguido la filosofía REST para la operativa de lectura (vía GET), de eliminación (vía DELETE), y de la adición de nuevas relaciones (vía PUT). El dilema principal lo tuvimos a la hora de escoger el verbo para la subida de ficheros. Si bien PUT parece el verbo más adecuado según las reglas REST, en nuestro caso una subida de fichero no implicaba el almacenamiento directo de la información si no que ésta podría pasar antes de su almacenamiento por una serie de procesos de transcodificación para adaptarse a los formatos admitidos por la plataforma Android. Así pues, optamos por implementar la carga de ficheros mediante POST aprovechando la petición para parametrizar el proceso de transcodificación y que, en conjunto, toda la operativa se resolviera en una única petición.

Los resultados de todas las operaciones se devuelven en formato XML o JSON, en función de un parámetro común a todas las operaciones que permite a la aplicación que invoca el método elegir el formato de respuesta. Si bien HTML también sería un formato de respuesta aceptable dentro del diseño REST, consideramos que eso sólo sería aprovechable por la aplicación Web que íbamos a desarrollar, ya que no sólo respondería con los datos sino que, además, forzaría un estilo de presentación. Por ello optamos por hacer una aplicación Web que fuese un consumidor más del servicio y, a partir de respuestas en formato JSON, presentase la respuesta en formato HTML.

Tanto la aplicación Web como la aplicación para la plataforma Android desarrolladas están ambas autorizadas, por defecto, como consumidores del servicio.

D. Notificaciones

El problema fundamental de u-Bcast es proporcionar la información geolocalizada al usuario en función de su posición. Para tratar este problema nos basamos en un modelo híbrido *pull/push* que a) realiza una descarga inicial de los datos próximos a la posición del usuario, mediante peticiones *pull*, aprovechando la subdivisión en imágenes del mapa que realiza el motor de renderizado de imágenes Mapnik [14] de OpenStreetMap para los diferentes niveles de zoom; y b) utiliza el protocolo MQTT para las peticiones *push*.

La renderización del mapa de nuestra aplicación permite niveles de zoom entre 15 y 18, que corresponde a una escala 1:200m y 1:20m., respectivamente. Dado que nuestra aplicación tiene sentido en lo que se denomina *nivel de calle*, niveles de zoom más alejados corresponderían a áreas demasiado extensas para lo que consideramos adecuado.

Debemos considerar que a la coordenada geográfica de una posición (determinada por latitud y longitud) le corresponde una imagen de mapa referida por una URL. Ésta, a su vez, está formada por el nombre del servidor, posibles parámetros de renderización según los diversos servidores de imágenes de OpenStreetMap, y una parte final */zoom/xtile/ytile.png*, siendo *zoom* el nivel de zoom, y *xtile*e*ytile* los identificadores de la imagen del mapa relativos a la posición. Para el cálculo de *xtile* e *ytile* se implementa el siguiente algoritmo que relaciona una posición (latitud, longitud) con un área/imagen identificada por (*xtile*,*ytile*):

```
n=2^zoom
xtile=((lon_deg + 180)/360*n
ytile=(1-(ln(tan(lat_rad)+sec(lat_rad)/π))/2*n
```

Inversamente, se puede obtener la latitud y la longitud de la esquina superior izquierda del área/imagen referenciada por la dupla *(xtile, ytile)* del siguiente modo:

```
n = 2^zoom
lon_deg=xtile / n * 360.0 - 180.0
lat_rad=arctan(sinh(π * (1 - 2 * ytile / n)))
lat_deg=lat_rad * 180.0 / π
```

Con estos cálculos es posible obtener las coordenadas geográficas de las cuatro esquinas que delimitan un área/imagen.

Con estos datos, cuando un usuario inicia la aplicación en su terminal móvil y se obtiene el primer posicionamiento válido, la aplicación invoca la API REST del servicio para obtener toda la información geolocalizada a la que tiene acceso y que está en su *área de visión*. Esta área de visión se calcula con el nivel de zoom mínimo, esto es, el más alejado, y contiene la dupla *(xtile, ytile)* que corresponde a sus coordenadas geográficas y a las 8 duplas que la rodean (ver Fig. 3). A medida que el usuario se desplace y se produzca un cambio en su dupla (detectado mediante el cálculo de las esquinas de las áreas/imágenes), se pide la información multimedia asociada a las duplas que la rodean y de las que aún no se dispone. Si el desplazamiento se ha producido a cualquiera de las dos duplas contiguas en sentido horizontal o en sentido vertical, se demandarán 3 duplas nuevas, y si es a alguna de las que intersectan en las esquinas, se demandan 5 duplas nuevas.

Cabe decir que si un usuario obtiene información asociada a una dupla y, posteriormente, esa información multimedia se modifica, este hecho no tendría reflejo en la aplicación. Una posible solución sería reiniciar el proceso periódicamente para ir obteniendo actualizaciones del estado de la dupla, lo que puede resultar innecesario y, además, caro, por la tarificación que hacen los operadores sobre el tráfico de datos.

La plataforma cliente sobre la que trabajamos es Android que no tiene ningún servicio de notificaciones *push* en su SDK. Bien es cierto que a partir de la versión Android 2.2 se ofrece el *Cloud To Device Messaging* (C2DM) muy similar al servicio de notificaciones *push* que ofrece Apple para iPhone. Además del requisito de versión (la penúltima a fecha de redacción de este artículo), el otro gran inconveniente de usar C2DM, es que es necesaria una cuenta de Google para poder trabajar con él ya que aprovecha las conexiones del móvil con los servicios de Google.

Buscando soluciones para implementar servicios *push* en Android encontramos la propuesta de Dale Lane [15] que utiliza el protocolo MQTT desarrollado por IBM y que además tiene la ventaja de que todas las herramientas se pueden descargar libremente y sin coste alguno.

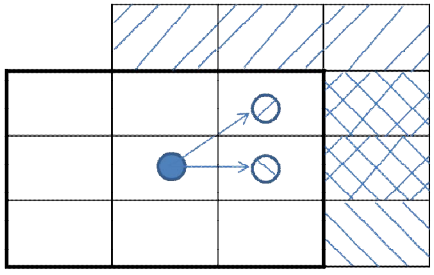


Fig. 3. Área de visión y desplazamiento del usuario.

El elemento central de MQTT es el denominado *MessageBroker* cuya misión es la de recibir mensajes de clientes y reenviarles mensajes a estos según criterios de suscripción. Un mensaje MQTT se compone de dos campos: *payload*, la información que se quiere transmitir, y *topic*, que constituye la dirección de envío. En MQTT un *topic* se construye de manera jerárquica a partir de una cadena con una longitud máxima de 32.767 octetos de cualquier carácter de un octeto a excepción de '/', '#' y '+' que tienen un significado especial: el carácter '/' sirve como delimitador del orden jerárquico; '#' se utiliza como comodín que indica "cualquier cadena a la derecha"; y '+' significa "todos los del mismo nivel jerárquico".

El nivel jerárquico en u-Bcast se organiza con el nombre del usuario en el nivel más alto; el segundo y tercer nivel formados por las coordenadas *xtile* e *ytile*, respectivamente; y el cuarto nivel está constituido por los tipos de visibilidad del contenido: público y de relación. Por debajo de estos últimos cuelgan las etiquetas del contenido y posibles subetiquetas. En la Fig. 4 vemos un ejemplo de esta jerarquía.

La configuración por defecto de las preferencias de usuario suscriben a éste a todos los *topic* públicos de la dupla en la que está (usando la regla +/public/xtile/ytile/#), y a todo el contenido de los usuarios con los que tiene relación (usando para todos ellos la regla [user_id]/relationship/xtile/ytile/#). La aplicación cliente para móviles permite el refinamiento de estas suscripciones. Así pues cuando un usuario entra en una dupla lanza el mensaje de suscripción más un parámetro temporal que indica la última actualización que tiene de la dupla obtenida por el *pull* inicial o por una suscripción previa, manteniendo la suscripción por todo el tiempo que permanezca en la dupla.

Finalmente, comentar que MQTT no define ningún mecanismo de seguridad que, por ejemplo, impida a un usuario suscribirse a los contenidos marcados como *relationship* por otro usuario con el que no tiene relación alguna. A falta de un mecanismo de seguridad mejor, nuestra solución de urgencia ha sido realizar las suscripciones no directamente sino a través de un proxy que valida la suscripción.

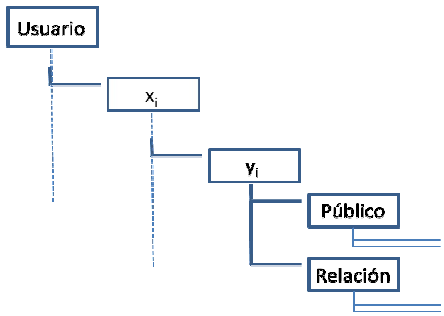


Fig. 4. Nivel jerárquico u-Bcast

III. *STREAMING* DE CONTENIDOS

Los usuarios del servicio pueden decidir reproducir cualquier información siempre y cuando se encuentren dentro del radio de acción de la misma, ya que uno de los parámetros asociados a un contenido es la distancia radial a la que se debe encontrar un usuario para iniciar su reproducción. La reproducción se realiza mediante *streaming*, no descarga, encargándose el protocolo RTSP del control de la reproducción. De hecho se trata de un servidor

multimedia bajo demanda permitiendo el control de la reproducción utilizando el protocolo RTSP.

A. *Apple Darwin Streaming Server*

Apple Darwin Streaming Server (DSS) [16] es un servidor de *streaming* RTP/RTSP de código abierto. Su primera versión pública data del 16 de Marzo de 1999 y puede transmitir varias tipos de formatos y códecs de audio y vídeo, incluyendo H.264/MPEG-4 AVC, MPEG-4 Part 2 y 3GPP.

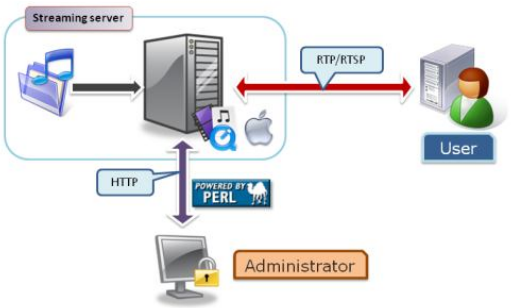


Fig. 5. Esquema del servidor de *streaming*.

Una de las características más interesantes de DSS es la capacidad de entunelar RTP/RTSP utilizando HTTP a través del puerto 80, permitiendo el envío de los flujos saltándose cortafuegos restrictivos. La configuración del servidor DSS se realiza mediante una interfaz web que se arranca ejecutando un script Perl.

En la Fig. 5 se muestran los componentes que forman el subsistema de *streaming* utilizado en u-Bcast.

B. *FFmpeg*

Los usuarios pueden subir cualquier contenido usando un amplio rango de formatos contenedores y códecs, lo cual facilita mucho el uso de la aplicación por parte de usuarios no avanzados, pero implica la posible subida de formatos y/o códecs no aptos para el *streaming*, ya sea por la incompatibilidad de formatos con el terminal móvil o por cuestiones de calidad del contenido que podrían tarificar excesivamente su reproducción. Por esta razón, cuando un usuario sube un contenido debería indicar al menos uno de los formatos y codificaciones soportados, realizándose un proceso de transcodificación en el servidor una vez que se ha completado la subida del contenido original (Fig. 6). Para mantener la posibilidad de poder realizar posteriores ediciones de los formatos disponibles para un mismo contenido, también se almacena el formato original.

Fig. 6. Diagrama de conversión de audio

El proceso de transcodificación se realiza utilizando la herramienta FFmpeg, una colección de software libre que permite grabar, reproducir, convertir e incluso hacer *streaming* de audio y vídeo. Incluye además la librería libavcodec, una biblioteca con un amplísimo catálogo de códecs de video y audio. Tiene licencia GNU LGPL o GNU GPL (dependiendo de qué bibliotecas estén incluidas). No hay distribuciones formales y los desarrolladores recomiendan utilizar el último *snapshot* del repositorio de Subversion.

La Tabla 1 contiene todos los formatos admitidos por la plataforma Android, pero eso no quita que implementaciones particulares según el dispositivo puedan ampliar esta lista.

Tipo	Formato/Códec	Contenedor
Audio	AAC/LC LTP	3GPP (.3gp) y MPEG-4 (.mp4, .m4a). No se admite raw AAC (.aac)
	HE-AACv1 (AAC+)	3GPP (.3gp)
	HE-AACv2 (AAC+ mejorado)	3GPP (.3gp)
	AMR-NB	MP3 (.mp3)
	AMR-WB	Tipo 0 y 1 (.mid, .xmf, .mxmf). También RTTTL/RTX (.rtttl, .rtx), OTA (.ota), e iMelody (.imy)
	MP3	Ogg (.ogg)
	MIDI	WAVE (.wav)
	OggVorbis	
	PCM/WAVE	
Vídeo	H.263	3GPP (.3gp) and MPEG-4 (.mp4)
	H.264 AVC (Android 3.0+)	3GPP (.3gp) and MPEG-4 (.mp4)
	MPEG-4 SP	3GPP (.3gp)
	VP8	WebM (.webm)

Tabla 1. Formatos soportados por Android.

La Tabla 2 muestra los dos perfiles predefinidos para que usuarios poco avanzados definan los parámetros de la transcodificación, recomendándose escoger los dos.

Perfil baja calidad	
Códec de vídeo	H.264 Baseline Profile
Resolución de vídeo	176 x 144 px
Tasa de cuadros	12 fps
Tasa de bits de vídeo	56 Kbit/s
Códec de audio	AAC-LC
Canales de audio	1 (mono)
Tasa de bits de audio	24 Kbit/s
Perfil alta calidad	
Códec de vídeo	H.264 Baseline Profile
Resolución de vídeo	480 x 360 px
Tasa de cuadros	30 fps
Tasa de bits de vídeo	500 Kbit/s
Códec de audio	AAC-LC
Canales de audio	2 (estéreo)
Tasa de bits de audio	128 Kbit/s

Tabla 2. Perfiles u-Bcast.

El formato contenedor escogido es 3GPP. Éste es el formato contenedor estándar definido por el Third Generation Partnership Project (3GPP) para servicios multimedia 3G UMTS [17].

C. *MP4Box*

Después del proceso de conversión de la unidad multimedia es necesario introducir información adicional en el fichero resultante para permitir al servidor DSS servir el contenido mediante *streaming*. Este proceso, denominado *hinting*, se realiza utilizando la herramienta MP4Box [18]. Se trata de una herramienta de línea de comandos. Forma parte de GPAC, un proyecto de código abierto para realizar diversas tareas con archivos multimedia.

El proceso de *hinting* consiste en añadir pistas especiales en el archivo que contienen información específica del

protocolo de transporte, y opcionalmente, información de multiplexado. Esta información es utilizada por el servidor de *streaming* para generar los paquetes que se transmiten por la red.

IV. Aplicaciones cliente

El servicio u-Bcast tiene dos aplicaciones cliente para acceder a él: una interfaz web orientada principalmente a la gestión de relaciones y de contenido aportado por el usuario, y el cliente Android, versión 2.1-update 1 o superior, orientado a la reproducción del contenido.

A. *Interfaz web*

La interfaz web se ha implementado utilizando la combinación de HTML, CSS y JavaScript. La Fig.7 muestra el mapa web simplificado de la aplicación.

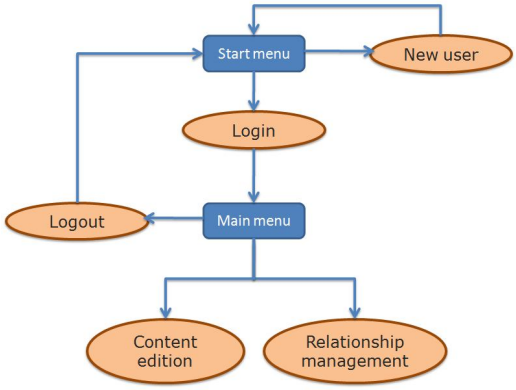


Fig. 7. Esquema de menú de inicio y menú principal web

La página de inicio es la puerta de entrada a través de la identificación del usuario y da acceso a dos opciones: edición de contenido y gestión de las relaciones. Los datos mínimos que debe entrar un usuario en su registro son su identificador (cadena de caracteres sin espacios en blanco), contraseña (mínimo seis caracteres) y correo electrónico para confirmación del proceso de registro. A la hora de identificarse en el servicio puede hacerlo tanto con su identificador de usuario como con la dirección de correo electrónica asociada, más la contraseña.

Edición de contenidos:

La opción de edición permite tanto la subida de contenido nuevo como la edición y eliminación de contenido anteriormente subido. La estructura de la página se muestra en la Fig. 8.

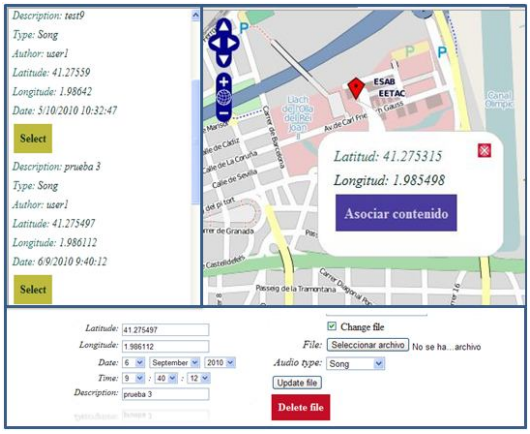


Fig. 8. Cliente web u-Bcast

La parte izquierda es un árbol expandible organizado por etiquetas que permite seleccionar el contenido subido. La parte superior derecha, área de mapa, muestra el mapa de OpenStreetMap centrado en la localización del contenido seleccionado o, en caso de que no haya ninguno, en la localización actual del usuario si es posible obtenerla. La parte inferior, denominada área de edición, detalla las características del contenido seleccionado y permite editarlo.

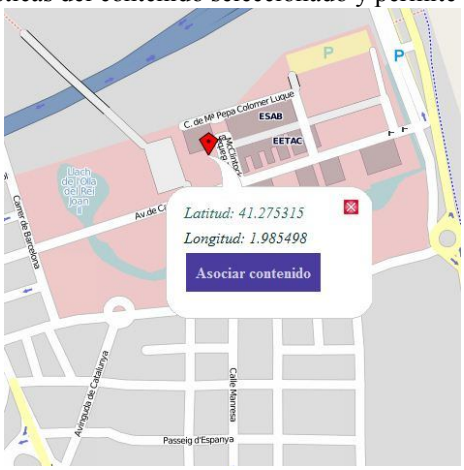


Fig. 9. Captura del área de mapa donde se permite la asociación de nuevo contenido a una coordenada geográfica.

El usuario también puede desplazarse por el mapa e ir haciendo zooms. El mapa señala todas las coordenadas geográficas en las que el usuario ha asociado contenido. La carga de las coordenadas geográficas asociadas a los contenidos subidos por el usuario sigue la misma filosofía del modelo *pull* explicado en la Sec. II-D. Cuando se pincha con el botón izquierdo del ratón sobre la información multimedia aparece una descripción del contenido subido y, en el área de edición, el detalle, pudiendo entonces el usuario actualizar la información. Si el usuario pincha con el botón derecho sobre el mapa se muestran las coordenadas geográficas del punto y un botón que habilita la asociación de un contenido a ese punto introduciendo la información en el área de edición (ver Fig. 9).

Gestión de relaciones:

Todo el servicio está diseñado como una red social en la que los usuarios pueden administrar sus relaciones con otros usuarios. Se basa en un sistema de relaciones unidireccionales, esto es, un usuario puede estar interesado en el contenido aportado por otro usuario pero no necesariamente a la inversa, siguiendo el modelo de seguidores de Twitter. Un usuario puede crear un grupo e invitar a cualquier otro a aportar contenidos al grupo. El grupo funciona como un usuario individual y otros seguidores pueden interesarse por su contenido. El grupo como tal no puede interesarse por el contenido ni de otro usuario ni de otro grupo, al menos en la versión actual.

B. *Cliente Android*

El cliente para la plataforma Android consta de la aplicación y del servicio de notificaciones y actualizaciones de contenido sobre MQTT. Cuando se abre por primera vez la aplicación, se crea el servicio de notificaciones que, cuando la aplicación no esté en primer plano, avisará al usuario de que hay un contenido reproducible en la zona donde se encuentra, a través de una notificación en la barra de notificaciones de Android. Para la comunicación con el

servidor a través de invocaciones a la API REST se ha utilizado el formato de respuesta JSON, ya que resulta más ligero en tamaño de octetos a transferir que XML (ver Fig. 10).

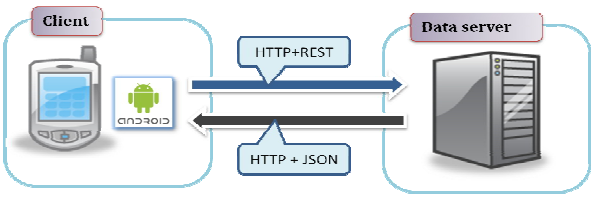


Fig. 10. Modelo de comunicación entre el cliente Android y el servidor.

La aplicación Android es básicamente un navegador de mapas que muestra la posición actual del usuario y los contenidos reproducibles que tiene a su disposición. En la Fig. 11 se puede ver una captura de pantalla de la aplicación para móvil. El punto azul representa la posición del usuario y la sombra alrededor del punto representa la precisión que proporciona el sistema de posicionamiento de Android. También se puede observar que hay un contenido reproducible, en este caso de audio. Cuando el usuario pulsa sobre el icono representativo se le despliega un diálogo que le permite reproducir los contenidos asociados a esa localización. En el caso de la captura de la Fig. 11 corresponde a un mismo contenido de audio a baja y alta calidad. En el caso de que el usuario seleccione uno de los contenidos comenzará su reproducción en *streaming*.

En el caso de no estar activa la aplicación, el usuario irá recibiendo notificaciones en la barra de notificaciones cada vez que tenga un contenido disponible. Si el usuario escoge hacer caso de la notificación, la aplicación se activará y el mapa quedará centrado en la localización del contenido.

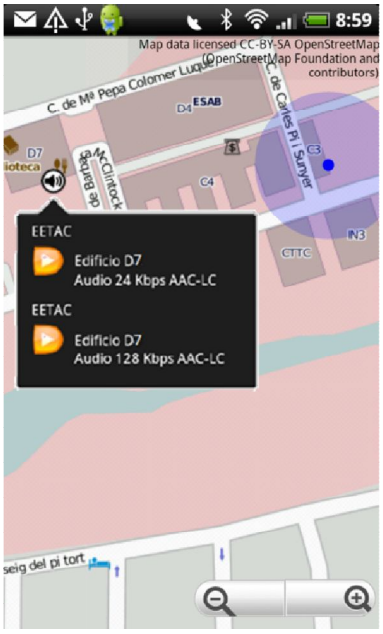


Fig. 11. Captura de la aplicación cliente en Android

V. Conclusiones

En este artículo se explica el desarrollo de un conjunto de aplicaciones que permiten utilizar un servicio sencillo que engloba *streaming*, red social y realidad aumentada, y que hemos denominado u-Bcast.

Dicho servicio permite a un usuario compartir información multimedia geolocalizada y poder reproducirla con un móvil, por ejemplo, si se encuentra cerca de su posición geográfica.

Las diferentes facetas del trabajo se han resuelto utilizando en todo momento *software open source*.

El resultado final es un servicio que contiene un buen número de tecnologías diferentes pero que las utiliza de una manera muy simple y directa, escalable y robusta.

u-Bcast se ha probado en diferentes navegadores y, para la parte móvil, se ha utilizado un HTC Desire con sistema operativo Android.

Referencias

[1] Why Smartphone Adoption May Not Be as Big as You Think, http://mashable.com/2010/08/26/smartphone-adoption-trends/
[2] J.Rozier, K.Karahalios and J.Donath. "Here and there: An augmented reality system of linked audio", in Proceedings of *ICAD'00 (International Conference on Auditory Displays)*, Atlanta, GA:2_5, pp. 63-67, Abril, 2000.
[3] Dimitrios Doiranlis. "Mobile Audio Augmented Reality". Master of Science Thesis. Stockholm, Sweden. 2007
[4] W. Carter, S. Fisher, T. Furmanski, K. Macdonald, T.Millican, "Patholog: Creating Location-based Web Logs," in Proceedings of the *Tenth International Conference on Virtual Systems and Multimedia*, Ogaki City, Japan, pp. 17-19, Nov. 2004.
[5] M. Wozniewski, Z. Settel, and J. Cooperstock, "User-Specific Audio Rendering and Steerable Sound for Distributed Virtual Environments", in Proceedings of *International Conference on Auditory Display*, Montreal, pp. 26-29, June 200.
[6] J.Webber, S.Parastatidis and I.Robinson, *REST in Practice: Hypermedia and Systems Architecture*, O'Reilly Media Inc., 2010.
[7] OpenStreetMap, http://www.openstreetmap.org/
[8] Google Maps, http://maps.google.com/
[9] Osmdroid, http://code.google.com/p/osmdroid/
[10] FFmpeg, http://www.ffmpeg.org/
[11] MQTT, http://www.mqtt.org/
[12] OAuth Community Site, http://oauth.net/
[13] OAuth consumer and server library for PHP, http://code.google.com/p/oauth-php/
[14] Mapnik, http://wiki.openstreetmap.org/wiki/Mapnik
[15] Dale Lane, Push notifications for mobile apps, http://dalelane.co.uk/blog/?p=938/
[16] Darwin Streaming Server, http://dss.macosforge.org/
[17] 3GPP, http://www.3gpp.org/
[18] GPAC, http://gpac.wp.institut-telecom.fr/mp4box/

Current and prospective role of augmented reality in mobile learning

A. Reina Nieves[*], A. Di Serio[*+], C. Delgado Kloos[*]

[*] Departamento de Ingeniería Telemática
Universidad Carlos III de Madrid
Av. Universidad 30, 28911 Madrid SPAIN.

[+]Departamento de Computación y Tecnología de la Información
Universidad Simón Bolívar
Caracas. Venezuela

alvaro.reina@uc3m.es, adiserio@ldc.usb.ve, cdk@it.uc3m.es

***Abstract*-. Augmented reality (AR) has been identified as a ground-breaking technique in education by some recent technological reports. AR overlaps virtual entities on-top-of real video and associates real objects with multimedia data. Therefore, it provides an intuitive way to perceive contextual information. The integration of AR with m-learning is making this technology very popular and well considered for teaching purposes. This paper provides an overview of the state of the art in mobile augmented learning and its enabling technologies. In sight of this study, some considerations about the prospective role of AR in m-learning are also discussed. Finally, the paper also contributes with a description of a collection of mini applications (the eduAR beans) that illustrates the statements previously provided.**

***Key words*- m-learning, e-learning, technology enhanced learning, augmented reality**

Introducción

Along with the evolution of information technologies, many attempts have been made to introduce them within the educational process from k12 to high-education. Although the use of such technologies has reported many benefits for students, installation difficulties and maintenance costs have prevented it from being widely used in educational centers. Rather than sophisticated systems, the educational community demands adaptable non-intrusive and easy-to-use solutions. Mobile learning, as it is usually called the use of mobile technologies with educational purpose, is devoted to bringing those simplicities through mobile devices. Since the recent commercial boom and popularization of smart phones, limitations in mobile computing, communications and context-awareness are being overcome and thus, the horizons of mobile learning have been extended far away, especially in the area of human-computer interaction. Among such innovative lines enabled by the new-generation mobile computing, the augmented reality (AR) must be highlighted.

AR is the result of superimposing virtual and digital information on top of real objects, so that it would be perceived as a unique scenario (blended reality). Despite the augmented reality was conceived and even implemented in the early '90s, the interest of the technological community has not been focused on it until the time being. The reason was the lack of appropriate hardware devices able to provide an adequate environment to jump from labs to society. However, current smart phones fill this gap and thus, open new opportunities in many sectors like marketing, art, culture, tourism and, of course, education.

Along this paper, the role of this novel approach in education will be discussed. The most relevant state-of-the-art experiences, development frameworks and applications will be revised. Thereafter, the eduAR beans collection will be presented with the aim of illustrating some ways to implement educational concepts and subjects in small augmented applications.

State of the Art

Recent reports have situated the augmented reality in the mainstream of Technology Enhanced Learning (TEL). It was mentioned for the first time in the Horizon Report, from the New Media Consortium, in 2010 [1] and it has reappeared in 2011 [2]. First, in 2010, the role of AR in education was recognized as a mean of presenting contextual information and in situ learning experiences. Afterwards, in 2011, it is specially highlighted the evolution of AR towards the interactivity, so that it is able to support active learning driven by manipulation and interaction. Nevertheless, both reports are aligned at the statement that smart phones are providing the required portability and usability that AR needed to become popular.

However, before the emerging of mobile learning, some testings and proofs of concept were published in the literature. A variety of approaches has to do with scholar environments. The magic book [3] introduces a tool to integrate 3D models into traditional textbooks. The project ARISE [4] imports and extends the concept of magic book with the Augmented Reality Teaching Platform (ARTP), a sort of cabin where students might share their learning experience with several mates. It explores the concept of collaborative augmented spaces, which was previously described more in depth in the platform Studierstube [5], and exploded by other projects like Construct3D [6]. It also discussed about the concept of augmented classroom which is also referred as mixed reality classroom in later works [7]. All those projects require expensive installations as well as difficult to use gadgets which hinder their use in real

ISBN: 978-84-694-5948-5

educational environments.

In detriment of immersion and personal experience but in favor of usability, a combination of webcams, laptops and web browser irrupted into AR as a cheap enough device that brought augmentation closer to population. Thus, augmented applications in education also started to be implemented in such environments. Zooburst [8] and Ariux [9], for instance, are frameworks to built augmented books (interactive virtual pop-up books and markerless respectively). Another representative example is learnAR [10] that provides AR resources for science lessons (3D models superimposed over the student's human body).

Nowadays, AR is evolving towards mobile environments. This change is mostly motivated by the advances in smart phones. Some educational experiments demonstrate how AR can be used to implement game based learning activities for language learning [11] or campus discovery via gymkhanas [12].

AR Supporting Technologies

AR is closed linked to a large number of technologies which determine the advances in AR. Among them, the following should be remarked.

Human-machine interfaces

Computer interaction has been dominated for ages by non-natural devices like mouse, keyboard and screen. Such in/out devices do not fulfill the naturally that AR demands. Otherwise, interfaces like head-up displays (HUD) allow users to interact with the environment (even if it is virtual, real or blended) in a more natural way. The main challenge for the adoption of HUDs in real contexts, especially in education, is to achieve ergonomic and non-intrusive devices. Examples of types of HMIs that will support AR in the future are pico-projectors, augmented reality glasses, transparent screens and 3D displays.

However, nowadays the mobile phones and their most recent evolution, the touch-screen tablets, provide the most useful interface for learning and cultural environments (e.g. classroom, museum, etc.). Their features (high resolution displays, touch-screen, camera and autonomy) allow the interaction with augmented environments.

Smart phones

Smart phones are revolutionizing computing. While previous generations of mobile phones were primarily conceived for calling, smart phones are thought to provide ubiquity, full connectivity and context-awareness. The complex set of sensors and communication interfaces such devices are equipped with, capture the environment and link the user with his social networks, multimedia resources, digital libraries and a large list of networked services. The complete technological equipment includes gyroscope, high quality cameras, GPS receptor, RFID readers, touch-screen, fast chip-set and large memories. In addition, the modern embedded operative systems (OS) like Android and iOS provide useful and attractive interfaces for users as well as powerful platform for developers.

AR, in particular, has obtained huge benefits from this evolution: the embedded camera captures the environment graphically, the mobile phone processor runs the tracking algorithms, the graphical processor and the 3D engine integrated within the OS render instantly the 3D models, the GPS locates the user and the display shows in real time the digital information (3D models, video, POIs) overlapped on the live video stream.

Computer vision

The marker-based AR applications consist of superimposing any kind of graphical digital information (video, image or 3D models) on top of a life video stream so that the digital object seems to be glued to a real object (which is usually called fiduciary marker or simply marker). Tracking algorithms, which apply computer vision techniques to the images captured by the camera, make possible such effect.

A tracking algorithm receives an image as input (usually a frame from the camera) and tries to match a pattern trained in advance. As a result, the algorithm returns a transformation matrix defining the translation and rotations (yaw, pitch and roll) that should be applied to a virtual rendering camera that is supposed to be placed in the location of the real one. The desired effect is got after rendering the virtual object with a 3D engine and applying the transformation matrix to its view.

According to the object type they are able to detect, tracking algorithms are classified in: fixed patter tracking (marker), variable pattern tracking (e.g. tracking of QR codes [13]) and natural features tracking (markerless).

For educational purposes, both fixed fiduciary markers and natural tracking objects have been used. The main advantage of the last one is that, in an ideal world, special environments and scenarios setups are unnecessary. Book images or pictures in exhibitions could be used for tracking instead.

Unfortunately, the state-of-the-art implementations of such algorithms are neither fast nor accurate enough and they are also very sensitive to light conditions. That is why many researches in augmented learning chose traditional markers for their AR experiments.

Geolocation

Despite positioning via satellite is a mature technology and nowadays this problem is solved with a high degree of accuracy, indoor positioning is challenging yet. Geolocation-based AR applications are strongly supported by the GPS receptors of mobile phones. Meanwhile, indoor location-based applications are hardly supported. Some approaches like triangulation with WiFi antennas provides partial solutions for this problem.

In the meanwhile, specific approaches for mobile AR have appeared aiming at "enhancing the location accuracy". Latitude Longitude Altitude (LLA) markers [14] is the most remarkable example. It consists of a two dimensional barcode that codifies the geographic coordinates of the place it is going to be situated. The user reads the LLA with his mobile phone camera so that his actual location could be identified. Then, the surrounding POIs are located taking into account this reference and using the compass to infer the right orientation.

Indoor location is especially important in education

insofar as it enables location-based AR activities in classrooms, museums, exhibitions, theatres, cathedrals and so on.

AR frameworks

A variety of frameworks and engines provide development kits and runtimes that support AR applications. Most of them include abstraction from tracking (localization, image based or both) and rendering but only a few are concerned about the necessity of providing easy-to-use tools for people without programming skills.

Table 1 lists a selection of AR frameworks and their features. Junaio and Layar, the most extended AR browsers, have launched mobile AR to the masses. Although AR browsers have been devoted to geolocation-based applications, they are extending their functionality with image-based applications (e.g. the Junaio GLUE channels [15]). AR browsers are suitable tools for educative gymkhanas. Some AR open-source browsers are even included in public repositories (markets) for downloading. LibreGeoSocial, an AR browser integrated with a social network, is a good example.

From the education point of view, such geolocation-based frameworks usually lack from intuitive authoring tools to support the task of developing further educational contents (usually it is necessary to implement and deploy some web services implemented into PHP and integrate with a SQL database, which is usually much more work than teachers could do). Sometimes, easier web frontends (HOPPALA for Layar) are offered from third-party partners, but they are usually restricted to limited functionality and mostly always are general-purpose tools (for marketing, education, culture, entertainment and any other field). Otherwise, some examples of intuitive tools can be found into desktop-oriented AR frameworks like the Authoring Mixed Reality (AMIRE) project [16], which offers a component-based architecture that supports a user interface based on connected block diagrams. Similar approaches can be found for the web (ATOMIC [17]).

Applying AR in Education

Recently, AR has burst in mobile education and it is causing high expectations. Although its benefits for learning are not rigorously demonstrated yet, many results endorse that AR applications increase attention and motivation of students. Concretely, best results have been reported with various objectives: training, explaining processes and developing skills like spatial perception.

It seems that most successful AR learning applications combine such an engaging environment with other techniques of participative and interactive learning like collaborative activities, tangible learning objects [18], serious games, gymkhanas, etc., that have been traditionally exploited in e-learning and m-learning. Then, successful testings should be considered as a result of the implication of students in the learning process. Therefore, an accurate impression about the qualities of AR is that this technology fosters the properties of the underlying educational technique and reinforces the resulting learning outcomes through an immersive environment that does not substitute the real world.

Nowadays, remarkable AR mobile learning developments are magic books, tutorials, overlapped 3D models and gymkhanas. These are only some of the possibilities that this novel technology brings to education. Further applications will exploit in depth interactivity and advance tracking algorithms. Thus, augmented museums, augmented toys and games, augmented blackboards and any object in any cultural and scholar environment are expected to be augmented with digital information. However, a successful integration of AR in education must take into account the relevance of providing appropriate authoring tools for teachers so that they can easily generate augmented contents for their own lectures.

Setting Up The eduAR Beans Collection

On the basis of the review that has been presented along this paper, the authors implemented a series of prototypes with the aim of demonstrating the applicability of AR in different learning situations. This is how the eduAR beans collection was created. EduAR beans are intended to set up a sandbox that would help to validate augmented learning applications in a wide range of learning contexts. Moreover, they are conceived as simple and reusable building blocks that allows for setting up more complex experiences as a result of combining the individuals. Each bean is also devoted to test a particular platform among the large list available in the state-of-the-art.

Four beans have been implemented since now and they will be presented afterwards.

MathAR: An AR game for mathematics

MathAR has been designed to learn basic mathematics. Both operators and operations are represented with their own markers. The objective is to combine them to form a correct mathematic sequence. Then, the AR application assigns a value to each marker and a challenge (operation) is presented to the player in an AR interface (see Figure 1).

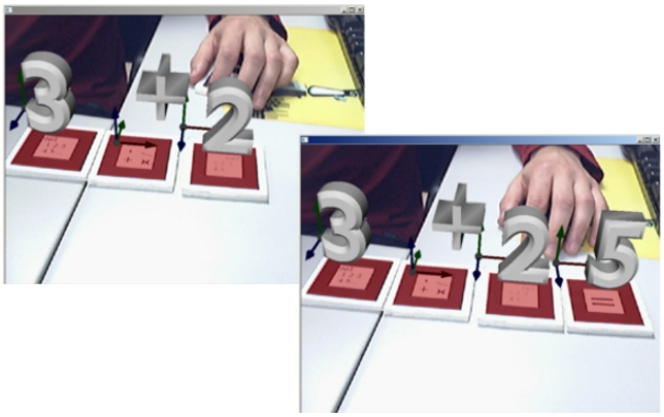


Fig. 1. A screenshot of mathAR

The player may demand the correct response by presenting the marker with the symbol '=' to the webcam. Such a simple game is a good example of how interactivity can be run in an AR environment in order to get learning outcomes. Moreover, an extension of this game including a larger set of operations would provide a funny environment for training mental arithmetic.

SyllbAR: An AR game for vocabulary learning

The purpose of SyllbAR, a marker-based game for children, consists of combining syllables to compose as much words as the player can. Syllables are codified in fiduciary markers so that the player can handle them as they were counters of some traditional game. The AR engine (AMIRE in this case) watches the playing scenario and detects the correct combination when the player puts the right syllables together. Automatically, the application sticks a 3D model that represents the word on top of the markers (see Figure 2).

The role of AR in this game is to provide feedback to the user and enhance the association of concepts. This game does not allow teaching abstract names.

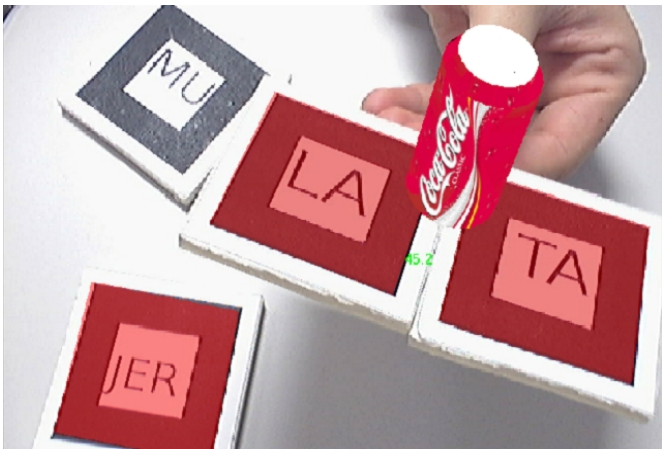


Fig. 2. A screenshot of SyllbAR

FlashAR: Augmented flashcards

Flashcards is a useful resource for studying in many subjects. A flashcard shows a short question on one side and the corresponding answer on the other side. A pack of flashcards can be used, for instance, for studying, personal training and competitions. Instead of providing a written answer, an augmented flashcard presents the answer with AR techniques, so that they might be enhanced with enriched information. In the example given in Figure 3, the answer to the question: "Who was Dali's muse?" is provided with the image of "Gala".

Fig. 3. A screenshot of FlashAR

PuzzAR: Augmented puzzles

Puzzles are activities that either educate or entertain the person. They can be used to teach different aspects such as shape, color, word recognition, spatial reasoning, problem solving, etc. Puzzles are primarily based on pictures, words and numbers. In image puzzles, pictures are reduced into different blocks that will need to be assembled to get the unique image. Puzzles encourage kids to learn effectively. PuzzAR has been split into two experiences. In the first one, once the image-based marker is recognized, the set of letter composing the object's name is presented to the kid. The activity consists on assembling the letters in order to obtain the word associated with the image. If the assembled word is the correct one, a 3D model associated to the word is shown (Figure 4).

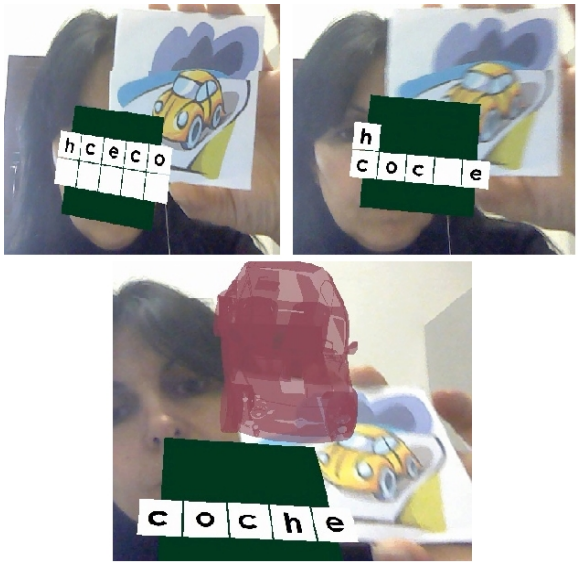


Fig. 4. A screenshot of PuzzAR

The second application, also based on image-based markers, is related to an art experience. Once a painting is recognized by the application, an explanation of the most important aspects is presented through interaction with the painting. Afterwards a puzzle of the painting could be used to reinforce the different components of the master piece (Figure 5).

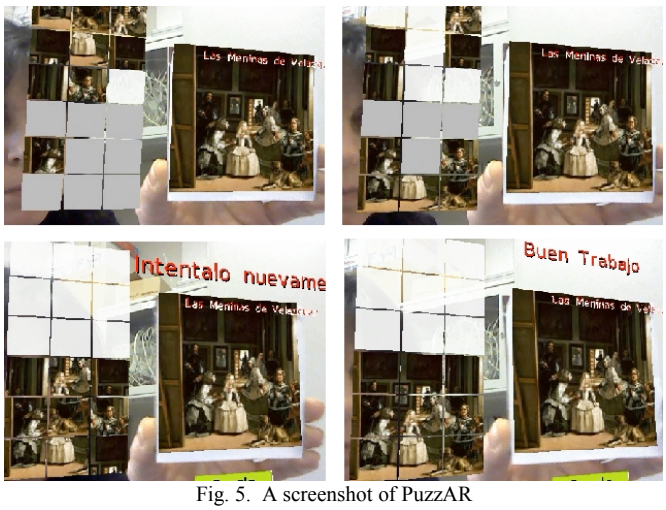


Fig. 5. A screenshot of PuzzAR

	Environment	License	Notes	Type	Interactivity	Authoring tools	SDK
Layar	Mobile	P	-	Geolocation	Low	Yes (not native)	No
Junaio	Mobile	P	Based on Metaio Unifeye	Geolocation Markerless	Medium	No (code examples available)	No
Metaio Unifeye	Mobile/Desk	P	-	Markerless	N/E	Yes	Yes
AMIRE	Desktop	O	Based on ARToolkit	Marker	High	Yes	Yes
ATOMIC	Web	O	Based on ARToolkit	Marker	None	Yes	Yes (API)
ARToolKit	Desktop	O	-	Marker	None	No	Yes (API)
D'Fusion	Mobile/Desk	P	-	Markerless Geolocation	N/E		Yes
NyARTooolkit	Desktop	O	ARToolKit for Java	Marker	None	No	Yes (API)
Popcode	Mobile Desktop	P (1)	-	Markerless	High	Yes (programmatic)	Yes
Qualcomm AR	Mobile	P	-	Markerless	N/E	Yes	Yes
AndAR	Mobile	O	Based on ARToolkit	Marker	None	-	Yes (API)
LibreGeoSocial	Mobile	O	Oriented to education	Geolocation	Low	Yes (web)	Yes
ARmsk	Mobile	O	-	Markerless	N/E	No	Yes (API)
Mixare	Mobile	O	Up to now displays Wikipedia POIs	Geolocation	N/E	No	Yes (API)
FLARToolkit	Web	(2)	Based on NyARToolkit	Marker	None	Yes (FLARTManager)	Yes
SLARToolkit	Web	O	- AR library for Silverlight - Based on NyARTollkit	Marker	None	-	-
ARIS	Mobile	O	- Oriented to education - Not authentic augmented reality yet - Use of QR codes	Geolocation	Low	-	No
Argon	Mobile	O	- Based on standards. - KHARMA Framework	Geolocation	N/E	-	-

Table 1. Augmented reality frameworks

(1) University of Cambridge owns the Copyright of source.
(2) Open for non commercial apps. Commercial license available.
N/E Not evaluated
O Open-source
P Proprietary

Conclusions and Future Work

Education is extremely sensible to changes and thus very demanding and strict. In this context, non-intrusive, easy-to-use technologies have an opportunity of success. Augmented reality offers a blended environment of which main feature is that does not substitute but complete the real world with useful contextual information.

The evolution of a wide range of technologies determines the advances in augmented reality. Althoug AR technology is not new, its potential in education is just beginning to be explored.

This potential might enable a new quality of education environment providing a way to enjoy learning. It is important to evaluate the effectiveness of this technology that could more actively involve students and teachers in the learning process. Therefore, we are currently working on preparing an AR instructional material with augmented reality to be used to reinforce an art class for middle school students.

We firmly believe that AR will open up a world of enhanced learning but educators should work with researchers in the field to explore how these characteristics can best be applied in a school environment.

Acknowledgements

This research has been partially supported by the project "Learn3: Towards Learning of the Third Kind" (TIN2008-05163/TSI) of the Spanish "Plan Nacional de I+D+i" and the Madrid regional project "eMadrid: Investigación y Desarrollo de tecnologías para el e-learning en la Comunidad de Madrid" (S2009/TIC-1650).

References

[1] Johnson, L., Levine, A., Smith, R., & Stone, S. (2010). *The 2010 Horizon Report.* Austin, Texas: The New Media Consortium.

[2] Johnson, L., Smith, R., Willis, H., Levine, A., and Haywood, K., (2011). *The 2011 Horizon Report.* Austin, Texas: The New Media Consortium.

[3] Billinghurst, M.; Kato H.; Poupyrev, I..: The MagicBook: Moving Seamlessly between Reality and Virtuality. In IEEE Computer Graphics and Applications, pp. 2-4, May/June, 2001

[4] W. Nejdl, K. Tochtermann, M. Bogen, J. Wind, and A. Giuliano, "ARiSE - Augmented Reality in School Environments," W. Nejdl and K. Tochtermann, eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 709-714-714.

[5] Schmalstieg, D., Fuhrmann, A., Szalavari, Z., Gervautz, M., (1996) Studierstube - An Environment for Collaboration in Augmented Reality. In CVE '96 Workshop Proceedings, 19-20th September 1996, Nottingham, Great Britain.

[6] H. Kaufmann and D. Schmalstieg, "Mathematics and geometry education with collaborative augmented reality," *Computers & Graphics*, vol. 27, Jun. 20 03, pp. 339-345.

[7] Liu, W., Cheok, A. D., Mei-Ling, C. L., & Theng, Y.-L. (2007). *Mixed reality classroom. Proceedings of the 2nd international conference on Digital interactive media in entertainment and arts -*

DIMEA '07 (p. 65). New York, New York, USA: ACM Press. doi: 10.1145/1306813.1306833.
[8] Zooburst web, 2011. http://www.zooburst.com/index.php
[9] Ariux web, 2011. http://www.ariux.com/AR_indexV1a.html
[10] LearnAR web, 2011. http://www.learnar.org/
[11] Liu, T.-Y., Tan, T.-H., & Chu, Y.-L. (2007). 2D Barcode and Augmented Reality Supported English Learning System. 6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007) (pp. 5-10). IEEE. doi: 10.1109/ICIS.2007.1.
[12] Robles, G., González-Barahona, J.M.; Fernández-González , J. "Implementing Gymkhanas with Android Smartphones" IEEE EDUCON 2011
[13] Kan, T.-W., Teng, C.-H., & Chou, W.-S. (2009). "Applying QR code in augmented reality applications". Virtual Reality Continuum And Its Applications, 253-257. Retrieved October 27, 2010, from http://portal.acm.org/citation.cfm?id=1670252.1670305.
[14] Junaio web. LLA markers, 2011. http://www.junaio.com/publisher/llamarker
[15] Junaio web. GLUE channels, 2011.http://www.junaio.com/publisher/junaioglue
[16] Grimm P., Haller M., Paelke V., Reinhold S., Reimann C., Zauner J., "AMIRE - Authoring Mixed Reality", The First IEEE International Augmented Reality Toolkit Workshop, 29 September, 2002. Darmstadt, GERMANY
[17] ATOMIC web, 2011. http://www.sologicolibre.org/projects/atomicweb/en/
[18] Levelhead web, 2011. http://selectparks.net/~julian/levelhead/

Sistema de distribución de vídeo streaming adaptativo basado en codificación SVC

Laura Pozueco, Xabiel G. Pañeda, Alberto Álvarez, Sergio Cabrero, David Melendi, Roberto García.
Departamento de informática,
Universidad de Oviedo
Campus de Viesques, SN 33204, Gijón, Asturias, España.
laurapozueco@gmail.com, {xabiel, alvarezgalberto, cabrerosergio, melendi, garciaroberto}@uniovi.es

***Resumen*- Hoy en día, el acceso a contenidos multimedia puede realizarse a través de diversos tipos de redes y con una amplia variedad de terminales que presentan diferentes restricciones. Por tanto, la adaptación de los contenidos a un entorno heterogéneo como es Internet será un proceso clave en la mejora de la percepción de la calidad del usuario. En este trabajo presentamos un sistema adaptativo de *streaming* empleando la tecnología *Scalable Video Coding* (SVC), recientemente estandarizada como extensión de H.264/AVC. Empleando información de *feedback* proveniente del cliente acerca del estado de la transmisión, el servidor es capaz de adaptar las capas que se envían del vídeo escalable en función de la congestión del enlace. El sistema se implementa en equipos reales y los resultados muestran el correcto funcionamiento ante diferentes variaciones del ancho de banda disponible, así como la escalabilidad del sistema cuando al servicio acceden clientes de manera simultánea.**

Palabras Clave- Scalable Video Coding (SVC), streaming, estimación, adaptación.

I. INTRODUCCIÓN

El desarrollo de las tecnologías de *video streaming* y el aumento del ancho de banda en las redes de acceso han propiciado la explosión de la transmisión de contenidos multimedia en Internet. Este nuevo servicio hace posible, por ejemplo, la televisión a través de Internet o los servicios de vídeo a la carta, de manera que presenta un gran interés en diversos ámbitos. Los usuarios cada vez hacen un uso más intensivo de este tipo de servicios y los proveedores intentan ofrecer vídeos de mejor calidad para responder a esas necesidades.

La principal limitación de esta tecnología se encuentra en la necesidad de unas condiciones de transmisión estables para poder garantizar una cierta calidad de servicio (QoS). Sin embargo, garantizar la QoS (expresada en términos de mínimo ancho de banda o máximo retardo o pérdidas) para servicios con requisitos de tiempo real en redes como Internet no resulta trivial, ya que no se tiene control sobre el enrutamiento de los datos y pueden darse situaciones de elevada latencia y pérdida de paquetes durante la transmisión. Este hecho tiene especial relevancia en las líneas de acceso del usuario debido, principalmente, a las limitaciones que existen en la tasa binaria. Generalmente, las líneas de acceso presentan un alto porcentaje de ocupación, así que las prestaciones máximas dependerán de las características intrínsecas de los servicios y del número de usuarios compitiendo por los recursos. Factores como la pérdida de paquetes, el retardo y la congestión de red afectan de manera directa a la calidad de presentación del vídeo.

Una manera de que este tipo de aplicaciones sean soportadas en redes *best-effort* es usar mecanismos adicionales de control que adapten los procesos de codificación, transmisión, recepción o decodificación dependiendo del estado de la red. Una posibilidad la brinda SVC (*Scalable Video Coding*), una nueva tecnología que permite codificar un fichero de vídeo en diferentes calidades dentro del mismo *bitstream*. Esto incluye diferentes resoluciones, diferentes frecuencias (fps) y diferentes calidades, (bits por píxel). Es decir, SVC permite escalabilidad espacial, temporal y de calidad. Este concepto permite la adaptación del vídeo transmitido al ancho de banda disponible o a los parámetros del servicio. En caso de congestiones en la red, donde el ancho de banda no es suficiente para transmitir un vídeo con la máxima calidad, podemos reducir la resolución, los *frames* por segundo o la calidad de la imagen y, por tanto, el ancho de banda necesario, con el fin de evitar la pérdida de paquetes y una mayor degradación de la calidad de la experiencia (QoE) del usuario.

El problema de este planteamiento radica en la estimación de ese ancho de banda disponible. Se hace necesario analizar la red durante el proceso de transmisión del vídeo para estudiar el comportamiento del tráfico en diversas situaciones de congestión. De esta manera se podrán extraer los parámetros que están directamente relacionados con la disponibilidad de recursos en la red y, por tanto, podrán ser empleados para realizar un algoritmo que permita calcular el ancho de banda disponible en cada momento, para el posterior ajuste de la calidad de los contenidos a las condiciones de transmisión.

Este trabajo aborda la implementación de un servicio de *streaming* adaptativo, empleando las ideas que se plantearon en [1], pero mejorando el sistema de estimación y empleando la tecnología SVC, capaz de proporcionar contenidos adaptados a las condiciones que la red impone al usuario, mediante el escalado de las capas temporal y de calidad. Los parámetros observados de forma no intrusiva, como consecuencia de la transmisión del propio contenido multimedia, son empleados para realimentar el algoritmo diseñado al objeto de estimar los recursos disponibles para cada usuario. El algoritmo decide y actúa sobre el servidor de contenidos, adaptando la calidad del flujo a los recursos disponibles. El sistema completo es evaluado empleando

ISBN: 978-84-694-5948-5

streams de vídeo reales en escenarios emulados, lo que permite un excelente control de las condiciones de transmisión. Los resultados demuestran que la capacidad de adaptación del sistema es precisa, flexible y escalable, mejorando la QoE del usuario. Este hecho evidencia la viabilidad de un sistema de estas características.

El resto del artículo está organizado como sigue. En la Sección II se resumen los principales trabajos relacionados. En la Sección III se describe la arquitectura e implementación del sistema de *streaming* adaptativo. En la Sección IV se realiza un estudio de las métricas que serán empleadas en el algoritmo de estimación descrito en la Sección V. La evaluación del sistema, experimentos, análisis y resultados se discuten en la Sección VI. Finalmente, la Sección VII recoge las conclusiones y en la Sección VIII se plantean las líneas de trabajo futuro.

II. TRABAJOS RELACIONADOS

Existen dos cuestiones claves en las investigaciones relacionadas con los sistemas de distribución multimedia adaptativos: por un lado el cálculo del ancho de banda disponible en la red y por otro lado la forma en la que se lleva a cabo la adaptación de los contenidos a los recursos libres en la red.

Para el cálculo del ancho de banda disponible en la red podemos recurrir a técnicas activas o técnicas pasivas. Las técnicas activas o intrusivas se basan en la inyección de carga adicional en la red, empleando ese tráfico extra para obtener datos (tales como los tiempos de ida y vuelta o medidas de dispersión de los paquetes) con los que estimar el ancho de banda disponible [2][3]. El problema que presentan estas técnicas reside en la influencia que presentan en las condiciones de transmisión de la red y, por tanto, pueden producir situaciones de congestión a causa del tráfico adicional que se introduce para hacer la estimación. Por el contrario, en las técnicas pasivas o no intrusivas la estimación se hace a partir de tráfico que ya existe en la red, sin necesidad de inyectar tráfico adicional. Las métricas empleadas son muy variadas: se puede utilizar las medidas de la pérdida de paquetes [4], el retardo, o incluso el *buffer* del cliente, ya que puede verse afectado por situaciones anómalas de la red [5][6]. En otros estudios emplean el orden de los números de secuencia recibidos en el cliente, de manera que los paquetes que se descartan o reordenan se emplean como indicador de congestión [7] o una combinación de diferentes métricas como pérdidas y *jitter* [8][9].

La siguiente cuestión que se plantea se refiere a la forma en la que se lleva a cabo la adaptación de los contenidos. Entre las diferentes posibilidades se encuentran la opción de disponer de múltiples versiones de diferentes características para el mismo contenido, las técnicas de *transcoding* o los formatos de codificación escalables, además de otras técnicas más elaboradas en las que se analiza información semántica (donde se detectan eventos en el vídeo) o se estudian las relaciones de los elementos estructurales de las imágenes (identificando los *frames* representativos).

En cualquier caso, todas se pueden utilizar para adaptar la tasa de codificación modificando características de los contenidos multimedia, tales como la resolución (dimensiones del vídeo), el número de *frames* por segundo o la calidad de los *frames*. Las técnicas de multiplicidad de versiones requieren cierta capacidad de almacenamiento en la cabecera del servicio, ya que habrá que tener almacenadas diferentes copias del vídeo, cada una de ellas con unos niveles de calidad distintos. La opción de transcodificar exige cierto coste computacional, ya que es necesario decodificar el contenido y volverlo a codificar de acuerdo a las restricciones del usuario final, lo que supone un gran consumo de recursos por parte del servidor y se traduce en una opción no escalable [1]. Una tercera opción nos permite tener diferentes niveles de escalabilidad en un único *stream* de vídeo, de manera que no será necesario contar con varias versiones de los mismos contenidos con diferentes niveles de calidad, con el consiguiente ahorro de espacio de almacenamiento. La idea es poder servir a un amplio rango de terminales, sobre redes heterogéneas, con una única versión del vídeo que contará con varias capas de calidad. En función de las características del estado de la red del usuario, se accederán a las capas de mayor o menor calidad. Esta tecnología recibe el nombre de SVC (*Scalable Video Coding*) [10], y permite realizar las tareas de adaptación a un coste computacional bajo. La estandarización de SVC finalizó en el año 2007 como una extensión del estándar H.264/AVC, y ya existen algunas plataformas y publicaciones que estudian esta manera de transmisión de los contenidos multimedia [11][12][13] y propuestas de algoritmos para realizar la adaptación de los contenidos [14]. SVC presenta la ventaja de la escalabilidad a un coste computacional abordable si a nuestro servicio accede un gran número de usuarios, ya que los requerimientos de computación para adaptar los formatos son mucho menores. Esto es posible gracias a que este sistema permite una adaptación eficiente directamente a nivel de *bitstream*, sin la necesidad de transcodificar, pudiendo extraer las capas con operaciones de baja complejidad, de manera que la adaptación a las condiciones de red y de los terminales es relativamente sencilla.

Uniendo los dos conceptos de estimación de la congestión y adaptación de los contenidos, podemos mejorar la calidad de un servicio *streaming*. Si nos vamos al campo de la adaptación empleando la tecnología SVC, el terreno aún está sin explorar, si bien es verdad que existen algunas propuestas teóricas que no se han aplicado de manera práctica. Por ejemplo, Kofler et al. [15] desarrollan un *proxy* RTP/RTCP en un *router wifi*, empleando la información RTCP de *feedback* para los propósitos de adaptación en los elementos de red, pero dejando para trabajos futuros la implementación de una aplicación de control que determine los parámetros óptimos de adaptación. En la misma línea de empleo de elementos intermedios que realizan las tareas de adaptación del *bitstream* SVC, otros autores [12][16] proponen y analizan diferentes diseños e implementaciones de adaptación RTP/RTSP, para contenidos SVC, basada en MANEs (*Media Aware Nerwork Elements*).

Siguiendo con la idea de estimación de ancho de banda y adaptación empleando la tecnología SVC, Nguyen et al. [17] proponen un sistema *streaming* de tiempo real usando SVC, empleando un algoritmo de control de la congestión basado en algoritmos *TCP-Friendly*, y empleando información de *feedback* del cliente, como su estado del *buffer*. En este caso, además de adaptar la tasa binaria a las variaciones de ancho de banda, optimizan la calidad visual en el cliente evitando cambios frecuentes en las resoluciones espacial o temporal.

En este sentido, Bianchi et al. [18], también tienen en cuenta la calidad final percibida, priorizando los paquetes en función de su importancia para la calidad del vídeo, de manera que los paquetes que se descartan son los que tienen un impacto menos negativo.

Las investigaciones de los trabajos anteriores sirven como punto de partida al diseño, realización e implementación práctica en entornos reales de un sistema adaptativo empleando la tecnología SVC.

III. Arquitectura del Sistema

En la arquitectura que planteamos para proporcionar un servicio de *video streaming* de calidad es necesario tener en cuenta una serie de componentes adicionales a la arquitectura tradicional, tales como un módulo de estimación del ancho de banda y un módulo de adaptación de los contenidos.

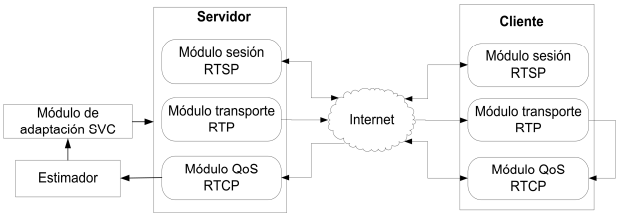


Fig. 1. Arquitectura del sistema

En la Fig. 1 se muestra la arquitectura del sistema y los subsistemas que conforman cada elemento. Un servicio básico de *streaming* únicamente contaría con los sistemas de servidor y cliente (y proveedor de contenidos en aquellas arquitecturas que así lo precisen). Sin embargo, en nuestro caso se añaden dos nuevos sistemas: el sistema de estimación y el sistema de adaptación para SVC, que permiten adecuar la tasa de transmisión de los contenidos (esto es, las capas del vídeo escalable que se envían al cliente) a los recursos libres de la red.

Según lo anterior, la solución propuesta para el sistema *streaming* adaptativo emplea los paquetes de *feedback* de RTCP para enviar información del cliente al servidor. Con esa información el servidor es capaz de identificar los estados de congestión en la red, respondiendo por medio de una adaptación de las capas que se transmiten al cliente. Por lo tanto, el sistema puede resumirse en dos fases:

1. Determinar el estado de la transmisión, basándonos en la información de retroalimentación que proviene del cliente.

2. Determinar la combinación óptima de capas a transmitir en función de dicho estado, de manera que consigamos adaptar los contenidos de vídeo al ancho de banda disponible.

Esta propuesta ha sido desarrollada sobre los protocolos estándar para la transmisión de datos multimedia en una sesión de *streaming*: RTP/RTCP. A medida que los paquetes de datos RTP llegan al cliente, son procesados y analizados para calcular las métricas. Toda la información necesaria para el cálculo de dichas métricas se encuentra en la cabecera de los paquetes RTP (*timestamp*, número de secuencia, instante de recepción y el tamaño del paquete en bytes). Una vez calculadas las métricas se envían al servidor por medio de paquetes de control del protocolo RTCP. En concreto, se empleará el tipo de paquetes RTCP-APP ya que se trata de paquetes de uso experimental para el desarrollo de nuevas aplicaciones y es posible personalizar los campos con las necesidades del sistema. El envío de estos paquetes se hará de forma periódica, cada 5 segundos. Ésta es una solución de compromiso, ya que un envío muy frecuente de paquetes RTCP-APP podría tener un impacto significativo en la ocupación del canal. Además, los procesos de adaptación de tasa en el servidor serían muy frecuentes e innecesarios en la mayoría de los casos, con el consiguiente consumo de recursos extra en el servidor. Por otro lado, un envío muy poco frecuente podría acarrear que la detección de situaciones de congestión se produjese demasiado tarde. Destacar en este punto que la sobrecarga producida en la red a causa de la información de *feedback* es insignificante. En los análisis del estimador sobre las pruebas en maqueta podremos calcular ese porcentaje de sobrecarga y veremos que el coeficiente de utilización de ancho de banda apenas se verá incrementado en un 0.014 %. De ahí que podamos clasificar la técnica empleada para la estimación del ancho de banda disponible como no intrusiva.

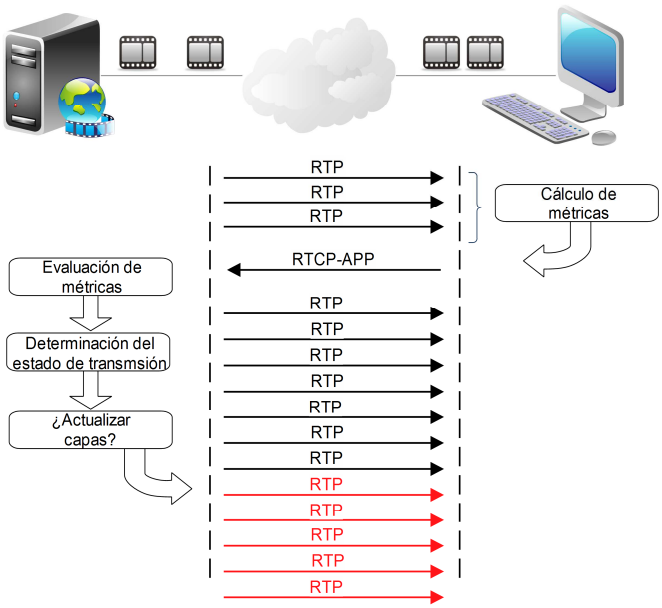


Fig. 2. Esquema de diseño del algoritmo

Una vez que el paquete RTCP-APP llega al servidor, se procesa y, con la información que contiene, se determina el estado del enlace y las capas óptimas a transmitir (Fig.2). La principal ventaja de emplear la tecnología SVC en la arquitectura, además de la reducción en el tamaño de almacenamiento en los servidores, se encuentra en la disminución computacional en el procesado de vídeo para la adaptación en vivo en función de las variaciones en las condiciones de transmisión, ya que SVC permite un manejo del *bitstream* de una manera sencilla gracias a la cabecera extra de 3 bytes que se añade con información adicional que contiene identificadores tales como D (*Dependency ID*), Q (*Quality ID*) y T (*Temporal ID*) [19] que permiten llevar a cabo los procesos de adaptación.

Una vez que hemos estimado el ancho de banda disponible y conociendo la tasa binaria de las capas que forman el vídeo, podemos elegir aquella combinación de capas que cumpla con las condiciones de transmisión actuales.

IV. ANÁLISIS DE TRÁFICO

Para implementar el sistema de adaptación es necesario llevar a cabo un proceso previo de análisis del tráfico bajo diferentes condiciones de ocupación de red. En este punto se define el factor de disponibilidad 'q' como el cociente entre el ancho de banda disponible en la línea y la tasa a la que se sirven los contenidos al cliente:

$$q = \frac{BW\ disponible}{Tasa\ codificación} \quad (1)$$

A efectos prácticos, es inmediato comprobar que un valor de q<1 indicaría una situación de congestión, ya que la tasa a la que se sirven los contenidos es mayor que el ancho de banda disponible.

El objetivo del análisis del tráfico generado es caracterizar los parámetros más significativos para una sesión de *streaming* en función del nivel de ocupación de la red. Para la transmisión de contenidos multimedia a través de una red de conmutación de paquetes (como lo es Internet) es necesario fragmentar los contenidos para posteriormente enviarlos al cliente en paquetes RTP. La transmisión de dichos paquetes por líneas que no implementan calidad de servicio hace que las condiciones de la red tengan una influencia significativa sobre las características del tráfico recibido en el cliente. Por ejemplo, los paquetes se verán afectados de diferente manera por los procesos de encolado de los elementos intermedios de la red. Este hecho hace que los contenidos recibidos en el cliente difieran de los contenidos originales a causa de las pérdidas o los retardos que puedan producirse durante la transmisión.

En este trabajo analizaremos la variación de tres métricas diferentes en función del nivel de ocupación de los recursos de la red. Estas métricas serán la pérdida de paquetes, el *jitter* y la linealidad en los tiempos de recepción de los paquetes. La pérdida de paquetes es una métrica ampliamente utilizada para detectar situaciones de congestión. El *jitter* afecta especialmente al tráfico con requerimientos de tiempo real. Y por último, la curva de los instantes de recepción de los paquetes puede verse afectada por situaciones de congestión en la red a causa de los retardos derivados de este fenómeno. Este hecho provoca que los paquetes no lleguen a ráfagas y que la curva de recepción presente un aspecto lineal. Por lo tanto, podemos analizar la linealidad en los paquetes de llegada para detectar situaciones de congestión en la red y para ello emplearemos el coeficiente de correlación de Pearson, R, según se indica en [1].

Los experimentos se llevarán a cabo en un entorno controlado de pruebas, donde recurriremos a herramientas de emulación (NS-3) para modificar las condiciones de capacidad de la línea sin la necesidad de desplegar físicamente la red. Además, para este trabajo se ha desarrollado un servidor *streaming* que soporta la transmisión de contenido H.264/SVC a través de RTP utilizando las librerías de live555[1]. Por otro lado, se han empleado diferentes tipos de vídeo con diferentes contenidos, codificados en SVC mediante el software de referencia que proporciona la ITU[2], para, de esta manera, obtener un análisis más exhaustivo del tráfico. Se ha optado por incluir en el vídeo codificado dos tipos de escalabilidad: temporal y de calidad. La opción de la escalabilidad espacial no se contempla, ya que los cambios en el tamaño/resolución de la imagen durante el proceso de reproducción no proporcionan una buena experiencia de usuario. En el plano de escalabilidad temporal se dispondrá de 2 capas: la capa T0 y la capa T1, con valores de *frames* por segundo de 15 y 30 fps respectivamente. A su vez, a cada nivel temporal se puede añadir hasta 3 niveles extra de escalabilidad de calidad (desde Q0 a Q3) que permiten reducir el error de cuantificación de codificación y mejoran la PSNR.

En la Fig. 3 se muestra la métrica de la pérdida de paquetes en función del nivel de disponibilidad 'q' para diferentes combinaciones de escalabilidad temporal (T0-T1) y de calidad (Q0-Q3) de un *bitstream* (que suponen diferentes tasas de codificación y por tanto diferentes consumos de ancho de banda). En las situaciones en las que no existe congestión (q>1) las pérdidas de paquetes son inferiores al 1% para todas las calidades de vídeo analizadas. Según los estudios realizados por Wu et al. [20] este porcentaje de pérdidas no afecta en gran medida a la calidad del vídeo recibido, ya que el análisis de la PSNR devuelve valores en torno a 27 dB en el peor de los casos, y concluyen que es un indicador de buena calidad.

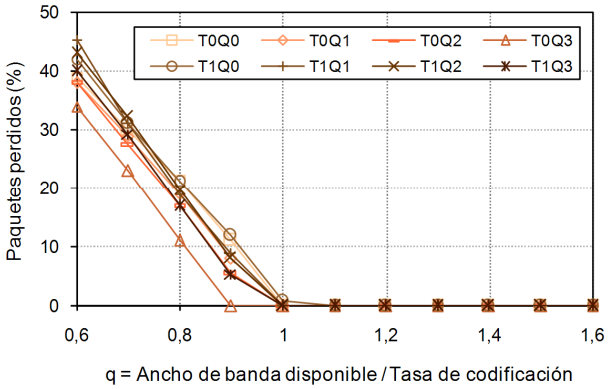


Fig. 3. Paquetes perdidos en función de la congestión

Asimismo, en las Fig. 4 y Fig. 5 se analizan las métricas del *jitter* y el coeficiente de correlación de Pearson, este último como medida de la linealidad de los instantes de recepción, para diferentes niveles de congestión en la red y, de nuevo, para diferentes tasas de codificación.

Para el *jitter* se produce un claro aumento en los valores a medida que la ocupación en la red aumenta, a consecuencia de los incrementos en los tiempos de encolado de los elementos intermedios de la red.

[1] http://www.live555.com

[2] http://ip.hhi.de/imagecom_G1/savce/downloads/SVC-Reference-Software.htm

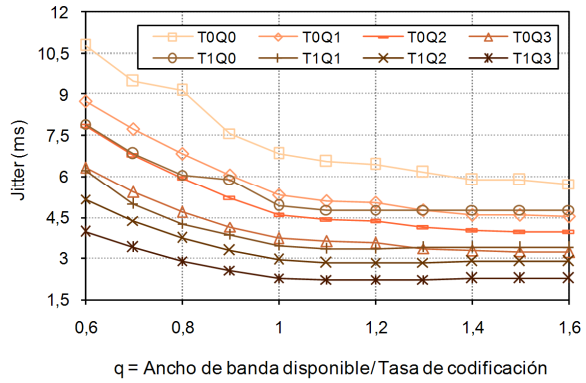


Fig. 4. Jitter en función de la congestión

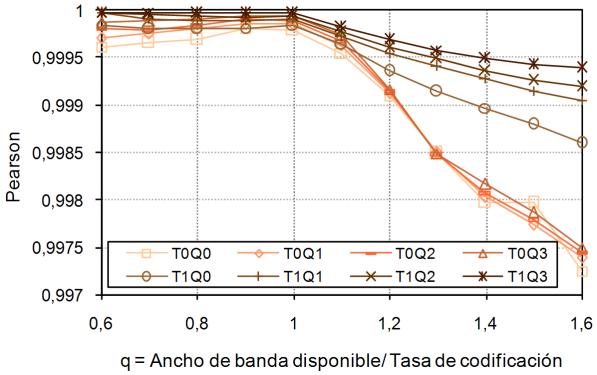


Fig. 5. Linealidad en función de la congestión

V. ALGORITMO DE ESTIMACIÓN

Las métricas descritas presentan una relación directa con el nivel de congestión. Por tanto, podemos emplear esa información para llevar a cabo procesos de adaptación de los contenidos multimedia al ancho de banda disponible, utilizando así los recursos de forma eficiente y mejorando la calidad percibida para las condiciones de transmisión dadas. Con la adaptación se pretende dotar de cierta QoS a los procesos de distribución de audio/vídeo a través de Internet, evitando las situaciones que producen pérdida de paquetes y que, como resultado, reducen la calidad causando un impacto negativo en la experiencia de usuario. Cuando se detecte una situación de congestión deberemos disminuir la tasa binaria de transmisión del vídeo, esto es, habrá que eliminar las capas que conforman los niveles más altos de calidad. Por el contrario, cuando no haya congestión, la tasa binaria podrá incrementarse, aumentando el número de capas que se envían al cliente.

La Fig. 6 describe el algoritmo empleado en el proceso de adaptación en el servidor, que incluye modificación en tiempo real de los niveles temporales y/o de calidad que se transmiten en función del estado del enlace. El servidor analiza el paquete RTCP-APP que recibe y que contiene el valor de las métricas (pérdidas, *jitter*, coeficiente de correlación de Pearson) calculadas por el cliente. El primer paso en la evaluación de las métricas será comprobar el porcentaje de pérdidas. Si éstas superan el umbral establecido del 1%, se procederá a decrementar la tasa binaria de transmisión a base de eliminar capas en el vídeo transmitido. Si por el contrario las pérdidas son menores del 1%, habrá que estudiar el estado de la transmisión para estimar si es posible aumentar la tasa de envío de los contenidos con capas de niveles superiores.

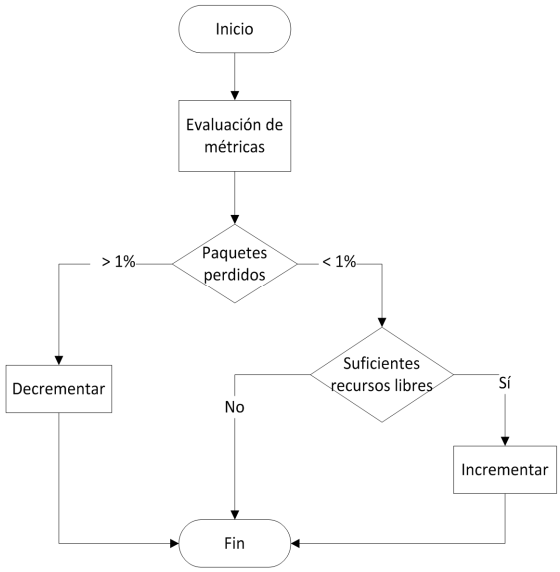


Fig. 6. Diagrama de flujo del algoritmo de estimación

El siguiente diagrama de flujo (Fig.7) explica el proceso de decremento de tasa e incremento. Cuando sea necesario aumentar la tasa de transmisión, se empleará la métrica del coeficiente de correlación de Pearson para estimar el ancho de banda disponible, ya que, para este caso la métrica presenta variaciones cuando q>1 y por tanto, será una medida válida de estimación de los recursos libres en ese tipo de situaciones.

Cuando sea necesario disminuir la tasa binaria (situación en la que las pérdidas presenten un valor superior al 1%), el *jitter* será la métrica elegida para estimar el estado óptimo para la transmisión, puesto que para este caso, la región de interés en la que la métrica del *jitter* depende del nivel de ocupación de la línea será para valores de q<1.

Según las gráficas vistas en las Fig. 4 y Fig. 5, las curvas del *jitter* y la correlación siguen una ecuación exponencial de la forma,

$$A \cdot B^{q} \qquad (2)$$

donde las constantes A y B dependerán de la tasa de codificación y del ID temporal. Una vez estimadas las constantes, se puede obtener el valor del coeficiente de disponibilidad 'q' a partir de la métrica del *jitter* o de la métrica de la correlación de Pearson. Seguidamente, el ancho de banda disponible en el canal se calcula siguiendo la expresión (1). En función de la capacidad libre del enlace, el servidor podrá eliminar capas correspondientes a los niveles de calidad o a los niveles temporales con un ID mayor que el ID óptimo para la transmisión calculado por el algoritmo de adaptación. Asimismo, en los casos en los que sea necesario, se añadirán capas hasta alcanzar el ID temporal o el ID de calidad objetivo.

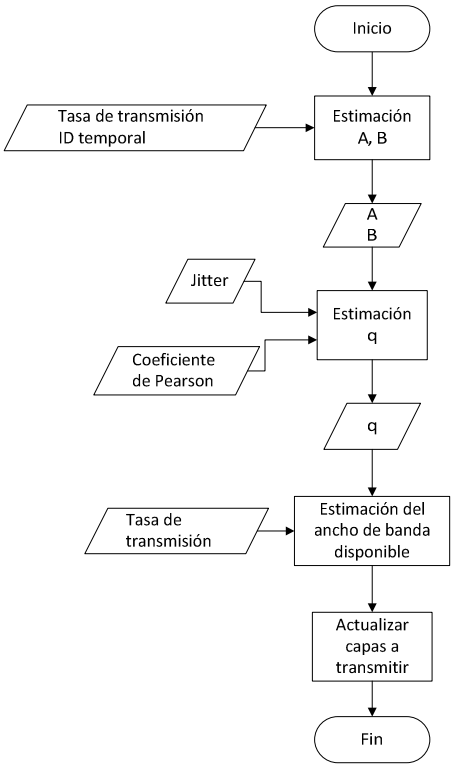


Fig. 7. Diagrama de flujo para el proceso de incremento/decremento

VI. EVALUACIÓN

Para el análisis del comportamiento del estimador se empleará el vídeo "BRIDGE" obtenido de los repositorios de las secuencias de test de vídeo[1], y codificado con SVC siguiendo las indicaciones vistas en el apartado IV y de acuerdo a objetivos de distribución de vídeo de alta calidad (Fig.8).

El equipo servidor cuenta con los módulos de estimación y adaptación desarrollados para la tecnología SVC, y por su parte, el cliente incluye una versión modificada de openRTSP[2] que incluye el módulo de cálculo de métricas a partir de la información de los paquetes RTP.

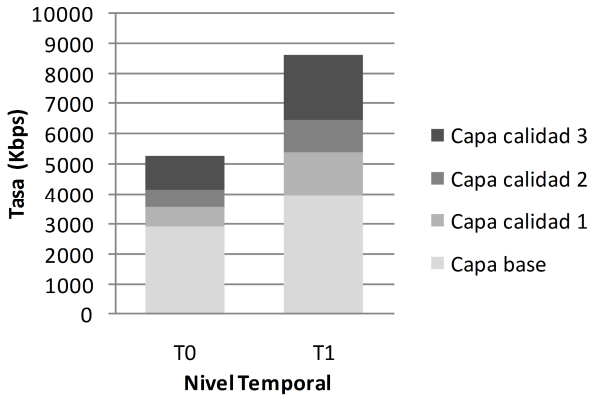


Fig. 8. Características del vídeo empleado en las pruebas

Para modificar las condiciones de variación del ancho de banda del canal en un entorno controlado se empleará la herramienta de emulación NS-3, en la que se inyectará tráfico de fondo CBR con el fin de congestionar el medio. El enlace de la red de acceso contará con una capacidad de 10Mbps y el tráfico CBR seguirá un modelo de rampa ascendente o descendente, con valores máximos de 6Mbps.

Con el fin de obtener resultados estables, se ejecutarán 50 realizaciones de cada experimento.

A. *Resultados de adaptación*

El análisis del estimador se hará en base al porcentaje total de sobreestimaciones que tienen lugar durante el proceso de adaptación. Las sobreestimaciones ocurrirán cuando la capacidad disponible en el canal sea menor que la capacidad disponible estimada. En esos casos, se producirán situaciones de pérdida de paquetes y retardos, efectos nada deseables en las transmisiones de *streaming* de vídeo.

En la tabla se muestra el porcentaje total de sobreestimaciones en dos situaciones: por un lado se presentan las situaciones en las que el ancho de banda disponible disminuye, y por otro lado, en las que los recursos libres en la red siguen una tendencia ascendente.

	% total de sobreestimaciones
Modelo decreciente	9.90
Modelo creciente	1.91

Tabla 1. % total de sobreestimaciones

Como se puede observar en los resultados, se producen más sobreestimaciones en los modelos de canal en los que el ancho de banda disponible disminuye. Esto es debido a que el servidor no actualiza el estado de la transmisión hasta que recibe el siguiente paquete RTCP-APP con las métricas y, durante ese periodo (un máximo de 5 segundos), la congestión en la red puede seguir en aumento.

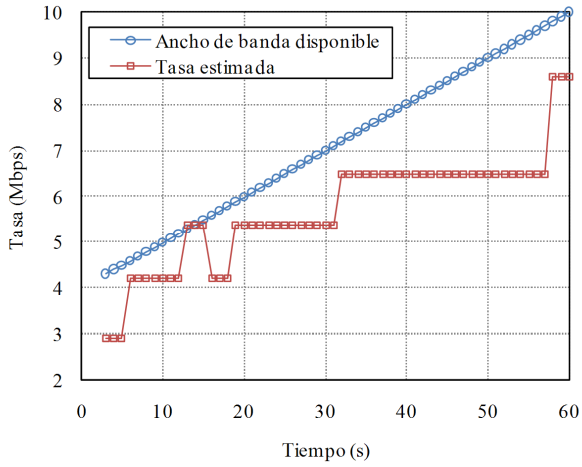


Fig. 9. Modelo de canal ascendente

En las Fig. 9 y Fig. 10 se muestra gráficamente el proceso de adaptación para un modelo de canal ascendente y un modelo descendente respectivamente. En ambos casos, la combinación óptima de niveles temporales y niveles de calidad estimados sigue la tendencia que marca la variación del ancho de banda disponible.

La respuesta ante cambios en las condiciones del enlace es un atributo importante en cualquier algoritmo de control de congestión y, como se observa en las figuras anteriores, nuestro sistema responde adecuadamente ante las variaciones

[1] http://trace.eas.asu.edu
[2] http://www.live555.com

en los recursos libres de la red, modificando los niveles temporales y de calidad del *bitstream* escalable que se transmite, con el fin de evitar la congestión.

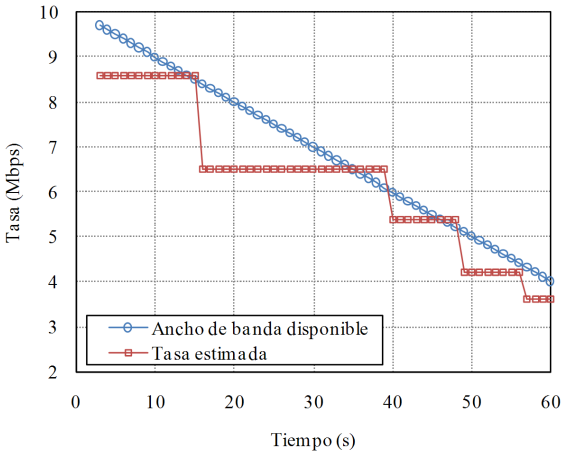


Fig. 10. Modelo de canal descendente

Cabe destacar también que las tasas a las que se puede ajustar el sistema son valores discretos, esto es, cada combinación de niveles temporales y de calidad presenta una determinada tasa binaria (Fig. 8). De ahí que, en algunas situaciones, a pesar de disponer de recursos libres en la red no se realizan cambios en las capas que se transmiten, ya que el siguiente nivel supondría exceder los límites del ancho de banda en la red en ese instante.

B. Análisis de escalabilidad

Otro factor clave y que marca una de las principales diferencias con el trabajo presentado en [1] será la escalabilidad del sistema. En [1], debido a los procesos de transcodificación que tienen lugar para realizar el proceso de adaptación de los contenidos al ancho de banda estimado, el servidor se veía rápidamente sobrecargado, limitando el número de usuarios concurrentes que acceden al servicio.

En el estimador basado en codificación SVC, los procesos de transcodificación para adaptar la tasa de transmisión a los recursos disponibles no son necesarios, quedando las tareas de adaptación reducidas a la decisión del envío de los paquetes en función de la capa a la que pertenezcan. En este sentido, la limitación ahora estará en el tráfico real que es capaz de manejar correctamente el emulador de red NS3. Álvarez et al. [21] realizan dicho estudio, llegando a la conclusión de que, para un escenario sencillo, el tráfico que puede llegar a gestionar NS-3 está en torno a 20 Mbps.

Por todo lo dicho, se recurrirá a una segunda opción de emulación de ancho de banda disponible basada en el conformado de tráfico, empleando TBF (*Token Bucket Filter*). Para este caso, el ancho de banda del canal será de 100 Mbps. Si se tiene en cuenta que la combinación de capas consideradas en el servidor y que producen la menor tasa binaria en el vídeo empleado en las pruebas es de 2.9 Mbps se obtiene que, el número máximo de clientes concurrentes accediendo al servicio para ese ancho de banda disponible total es de 34 clientes. En la siguiente tabla mostramos la evolución del porcentaje de uso de CPU en el servidor por parte del proceso del servidor de *streaming* SVC adaptativo en función del número de clientes concurrentes accediendo al servicio. Para mantener el uso de CPU por debajo de los límites aceptados de 80%, el número máximo de clientes concurrentes en el servidor se encuentra entre 15 y 20. Cuando al sistema acceden múltiples usuarios de manera concurrente, para cada uno de ellos se crea un proceso de estimación de los recursos libres teniendo en cuenta el resto del tráfico presente en la red, tanto si es tráfico de otros usuarios de *video streaming* como si es tráfico de otras aplicaciones. Además, los procesos de subida de la tasa de transmisión siguen una postura conservadora, de manera que se pretende que exista una salvaguarda, evitando consumir el 100% de los recursos libres de red. Por otro lado, en el proceso de evaluación de la escalabilidad que se describe, los clientes acceden al servicio de manera escalonada.

	% de uso de CPU en el servidor
5 clientes	21.74
10 clientes	44.92
15 clientes	68.34
20 clientes	87.95

Tabla 2. % de uso de CPU en el servidor adaptativo

Centrémonos en el caso en el que sean 15 los usuarios concurrentes. Con un sistema adaptativo, el porcentaje de uso de CPU en el servidor es de 68.34%. Podemos realizar el experimento en las mismas condiciones, pero con un sistema no adaptativo, donde se envíe siempre la misma combinación de capas. En concreto se van a estudiar dos casos: que siempre se envíen la combinación de capas T0Q0 (15 fps con el nivel de calidad menor, 2.9 Mbps) o que siempre se envíe la combinación T1Q3 (30 fps con el máximo nivel de calidad, 8.6 Mbps).

	% uso CPU en el servidor (T0Q0)	% uso CPU en el servidor (T1Q3)
15 clientes	63.54	68.96

Tabla 3. % de uso de CPU en el servidor no adaptativo

Como observamos en las tablas, no existen grandes diferencias entre el sistema no adaptativo y el sistema adaptativo en cuanto al consumo de recursos en el servidor. La limitación en este caso viene impuesta por las elevadas tasas de transmisión que se manejan y, por consiguiente, en el número de usuarios concurrentes accediendo al servicio, y no a causa del ancho de banda disponible.

VII. Conclusiones

Los servicios de vídeo en Internet han experimentado una importante evolución en los últimos años y cada vez son más los usuarios que demandan este tipo de recursos. Sin embargo, al tratarse de servicios con requerimientos de tiempo real, son muy sensibles a las variaciones de las condiciones de red, necesitando entornos estables para la transmisión. Por otro lado, las capacidades de las líneas de acceso de los usuarios han aumentado considerablemente. De la misma manera, el tráfico que generan los usuarios también se ha visto incrementado, por lo que las situaciones de congestión, competencia de recursos, pérdidas y errores son probables. En estos casos, cuando la ocupación de la línea es elevada, la transmisión de vídeo tradicional sufre una degradación de la QoE. Es por eso que se hace necesario contar con un sistema que permita adaptar el formato de los contenidos a las condiciones de transmisión. En base a esto

hemos empleado las ideas vistas en [1] de un estimador no intrusivo para desarrollar un sistema *streaming* adaptativo empleando la tecnología SVC.

Mediante las pruebas de emulación, se ha demostrado que el sistema implementado es capaz de realizar una adaptación rápida y precisa ante los cambios que se producen en la capacidad disponible en la red, evitando la proliferación de indeseadas oscilaciones en los ajustes de tasas. La utilización de un esquema SVC ha permitido incrementar la escalabilidad del conjunto del sistema con respecto a trabajos anteriores y las métricas empleadas para la estimación de los recursos disponibles constituyen un aporte novedoso al incluir la linealidad de los instantes de recepción de los paquetes RTP.

VIII. Trabajos Futuros

Entre las líneas de trabajo futuras se encuentra el desarrollo o adaptación de las métricas de evaluación de QoE objetivas a entornos adaptativos SVC, donde la variación dinámica del número de capas transmitidas impone restricciones de sincronización en las métricas de QoE más extendidas. Además, deberán ser analizados más escenarios y patrones de variación de tráfico para un análisis en detalle del sistema de adaptación. El desarrollo de un sistema de predicción puede completar la robustez del mecanismo de estimación del ancho de banda disponible, de forma que, en base a comportamientos previos, se pueda determinar el patrón o tendencia de variación del canal.

Otro gran frente abierto comprende la construcción de un reproductor para el cliente, compatible con la tecnología SVC con sus tres niveles de escalabilidad, así como el diseño de una estructura para servicios masivos de audio/vídeo, con diferentes servidores en distintos emplazamientos y *proxys* para acercar los contenidos al usuario, todos ellos compatibles con el estándar H.264/SVC. De esta manera, con la arquitectura CDN (*Content Distribution Network*) podríamos conseguir multiplicar la escalabilidad del sistema, ya que, como se ha visto en los resultados de la evaluación del sistema (Sección VI.B), un único servidor sería capaz de dar soporte a 20 clientes.

Agradecimientos

Este trabajo está financiado parcialmente por el Vicerrectorado de Investigación de la Universidad de Oviedo a través de su Programa de Promoción de la Investigación (UNOV-11-MA-03).

Referencias

[1] A. Fraga, L. Pozueco, X. Garcia Pañeda, R. García, D. Melendi, and S. Cabrero, "A non-intrusive estimation for high-quality Internet TV services," *Multimedia Tools and Applications*, 2010.

[2] R. L. Carter and M. E. Crovella, "Measuring Bottleneck Link Speed in Packet-Switched Networks," *PERFORMANCE EVALUATION*, vol. 27, p. 297--318, 1996.

[3] M. Jain and C. Dovrolis, "Pathload: A Measurement Tool for End-to-End Available Bandwidth," *IN PROCEEDINGS OF PASSIVE AND ACTIVE MEASUREMENTS (PAM) WORKSHOP*, p. 14--25, 2002.

[4] Cheng Wanxiang and Lei Zhenming, "An modified RTP adaptive algorithm," in *Info-tech and Info-net, 2001. Proceedings. ICII 2001 - Beijing. 2001 International Conferences on*, 2001, vol. 2, pp. 33-38 vol.2.

[5] P. Frojdh, U. Horn, M. Kampmann, A. Nohlgren, and M. Westerlund, "Adaptive streaming within the 3GPP packet-switched streaming service," *Network, IEEE*, vol. 20, no. 2, pp. 34-40, 2006.

[6] J.-C. Bolot and T. Turletti, "Experience with control mechanisms for packet video in the internet," *ACM SIGCOMM Computer Communication Review*, vol. 28, pp. 4–15, Jan. 1998.

[7] P. Papadimitriou and V. Tsaoussidis, "A Rate Control Scheme for Adaptive Video Streaming over the Internet," IEEE International Conference on Communications, ICC'07. Glasgow, Scotland, 2007, pp. 616-621.

[8] S. Ahmad, N. D. Gohar, and A. Kamal, "A Dynamic Congestion Control Mechanism for Real Time Streams over RTP," in *The 9th International Conference on Advanced Communication Technology*, , 2007, vol. 2, pp. 961-966.

[9] L. Tionardi and F. Hartanto, "The use of cumulative inter-frame jitter for adapting video transmission rate," in *TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region*, 2003, vol. 1, pp. 364-368 Vol.1.

[10] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103-1120, 2007.

[11] S. Wenger, Y.-kui Wang, and M. M. Hannuksela, "RTP payload format for H.264/SVC scalable video coding," *Journal of Zhejiang University SCIENCE A*, vol. 7, no. 5, pp. 657-667, 2006.

[12] S. Wenger, Ye-Kui Wang, and T. Schierl, "Transport and Signaling of SVC in IP Networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1164-1173, 2007.

[13] K.-deok Seo, J.-soo Kim, S.-heung Jung, and J. Yoo, "A Practical RTP Packetization Scheme for SVC Video Transport over IP Networks," *ETRI Journal*, vol. 32, no. 2, pp. 281-291, 2010.

[14] Peng Chen et al., "A network-adaptive SVC Streaming Architecture," in *The 9th International Conference on Advanced Communication Technology*, 2007, vol. 2, pp. 955-960.

[15] I. Kofler, M. Prangl, R. Kuschnig, and H. Hellwagner, "An H.264/SVC-based adaptation proxy on a WiFi router," in *Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, New York, NY, USA, 2008, pp. 63–68.

[16] R. Kuschnig, I. Kofler, M. Ransburg, and H. Hellwagner, "Design options and comparison of in-network H.264/SVC adaptation," *Journal of Visual Communication and Image Representation*, vol. 19, pp. 529–542, Dec. 2008.

[17] Dieu Thanh Nguyen and J. Ostermann, "Congestion Control for Scalable Video Streaming Using the Scalability Extension of H.264/AVC," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 2, pp. 246-253, 2007.

[18] G. Bianchi et al., "Application-aware H.264 Scalable Video Coding delivery over Wireless LAN: Experimental assessment," in *Second International Workshop on Cross Layer Design, 2009. IWCLD '09*, 2009, pp. 1-6.

[19] P. Amon, T. Rathgen, and D. Singer, "File Format for Scalable Video Coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 9, pp. 1174-1185, 2007.

[20] Dapeng Wu et al., "On end-to-end architecture for transporting MPEG-4 video over the Internet," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 6, pp. 923-941, 2000.

[21] A. Alvarez, R. Orea, S. Cabrero, X. G. Pañeda, R. García, and D. Melendi, "Limitations of network emulation with single-machine and distributed ns-3," in *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, 2010, pp. 1-9.

Ahorro de ancho de banda en juegos online mediante el uso de técnicas de tunelado, compresión y multiplexión

José Mª Saldaña, Julián Fernández-Navajas, José Ruiz-Mas, José I. Aznar,
Eduardo Viruete Navarro, Luis Casadesus
Grupo de Tecnologías de las Comunicaciones – Instituto de Investigación en Ingeniería de Aragón
Dpt. IEC. Centro Politécnico Superior Universidad de Zaragoza
Edif. Ada Byron, 50018, Zaragoza
{jsaldana, navajas, jruiz, jiaznar, eviruete, luis.casadesus}@unizar.es

***Resumen*- Las empresas desarrolladoras de juegos *online* necesitan recursos de *hardware* y ancho de banda para dar un buen servicio a los usuarios. Estos juegos producen altas tasas de paquetes UDP de pequeño tamaño desde el cliente al servidor, teniendo una baja eficiencia. Las acciones de los jugadores se tienen que propagar al servidor y al resto de jugadores en muy poco tiempo, por lo que los retardos de red son muy críticos. Este trabajo presenta un método que ahorra ancho de banda mediante un agente local que comprime las cabeceras y utiliza un túnel para enviar varios paquetes dentro de uno multiplexado. Se ha estudiado el comportamiento del sistema para IPv4 e IPv6, mostrando que el ahorro de ancho de banda es significativo. Como contrapartida, se añade un retardo que tiene una cota superior modificable. Si el número de jugadores es suficiente, este retardo no empeora la experiencia del usuario.**

***Palabras Clave* - juegos online, retardo, multiplexión, compresión, First Person Shooter**

I. INTRODUCCIÓN

Los juegos *online* son un servicio que crece día a día en Internet. Algunos títulos tienen millones de usuarios, y por eso las empresas desarrolladoras se enfrentan a un difícil problema cada vez que lanzan un nuevo título: necesitan recursos *hardware* y ancho de banda para evitar que su infraestructura se sature. Dado que el éxito de un nuevo título no es muy predecible, en ocasiones deben sobredimensionar los recursos para dar un buen servicio a los usuarios. En [1] se presentó un estudio del comportamiento de los jugadores *online*, y los autores llegaron a la conclusión de que son muy difíciles de satisfacer: si encuentran problemas, suelen abandonar ese servidor, y tienden a variar mucho sus preferencias.

Dos de los géneros más populares son los MMORPG (Juegos Masivos de Rol Multijugador Online, *Massive Multiplayer Online Role Playing Game*) y los FPS (Tirador en Primera Persona, *First Person Shooter*). En [2] se estudió el tráfico de los MMORPG, llegando a la conclusión de que tienen algunas características como la periodicidad y la autosimilitud. Otra conclusión de dicho estudio es que estos juegos presentan unos requerimientos de ancho de banda y tiempo real menores que los FPS.

En los juegos FPS las acciones de los jugadores se deben propagar al servidor y al resto de jugadores en muy poco tiempo, por lo que los retardos de red son muy críticos. Estos juegos producen altas tasas de paquetes UDP de pequeño tamaño (algunas decenas de bytes) desde el cliente al servidor. Por eso el *overhead* causado por las cabeceras IP y UDP es significativo. Los paquetes del servidor al cliente son habitualmente más grandes.

Existen escenarios en los que muchos jugadores comparten la ruta desde la red de acceso hasta el servidor del juego: por ejemplo, los *cibercafés* (Fig. 1a y 1b), muy populares en algunos países, disponen frecuentemente de ordenadores capaces de ejecutar juegos *online* y grupos de jugadores suelen acudir a estos establecimientos. El tráfico entre los servidores de un mismo juego es otro escenario donde se comparte la misma ruta (Fig. 1c).

El tráfico de estos grupos de usuarios se podría comprimir para ahorrar ancho de banda, teniendo en cuenta que la red de acceso suele constituir el cuello de botella más restrictivo. Además, en el caso de conexiones asimétricas como el ADSL, el *uplink* tiene normalmente menos ancho de banda que el *downlink*. Mediante el uso de un agente local encargado de retener los paquetes, comprimir las cabeceras y utilizar multiplexión para su envío, se podría ahorrar ancho de banda, con el coste de incluir un nuevo retardo, que estará causado principalmente por el tiempo de espera en la cola del multiplexor.

Se han presentado varias propuestas [3], [4] para ahorrar carga de trabajo al servidor central del juego, incluyendo algunos elementos *(proxies)* cercanos a la red de acceso. En contraste, el agente local que se propone en este trabajo se podría distribuir con el juego, igual que la aplicación servidor se distribuye con algunos títulos. El túnel desde los clientes hasta el servidor se podría crear bien desde una máquina dedicada (Fig. 1a), o bien desde la máquina de uno de los jugadores (Fig. 1b), evitando así el coste de un nuevo equipo.

En el otro lado de la comunicación, el servidor debería implementar el demultiplexor y el descompresor, lo que conllevaría cierta capacidad de proceso, y espacio para almacenar el *contexto* de cada flujo, es decir, la información necesaria para reconstruir las cabeceras comprimidas (algunas decenas de bytes, como veremos [5]). Esto no conlleva un problema de escalabilidad, ya que el servidor de hecho ya almacena el estado del juego de cada usuario. Por otra parte, el ahorro de ancho de banda y paquetes por segundo sería beneficioso para el servidor.

Si el número de jugadores es lo suficientemente grande, se puede esperar que, añadiendo pequeños retardos, un gran

ISBN: 978-84-694-5948-5

número de paquetes se pueda multiplexar en otro más grande. El ahorro de ancho de banda no sólo afectará al tráfico del juego, sino que también podrá ser beneficioso para el tráfico que comparte el acceso con él. Es más, la multiplexión tiene la ventaja de reducir el número de paquetes por segundo que el *router* debe gestionar.

Existen otras aplicaciones y escenarios en los que varios flujos de información en tiempo real comparten la misma ruta, como por ejemplo el denominado *trunking* usado en aplicaciones de voz sobre IP (VoIP), donde las técnicas de multiplexado y compresión llevan tiempo usándose e incluso se han estandarizado [6]. Muchos juegos presentan patrones de tráfico similares, generando altas tasas de paquetes pequeños y presentando por eso un gran *overhead*. La novedad del presente trabajo es la aplicación de las técnicas usadas para VoIP al caso de juegos *online*, donde también se pueden conseguir ahorros significativos de ancho de banda sin perder calidad.

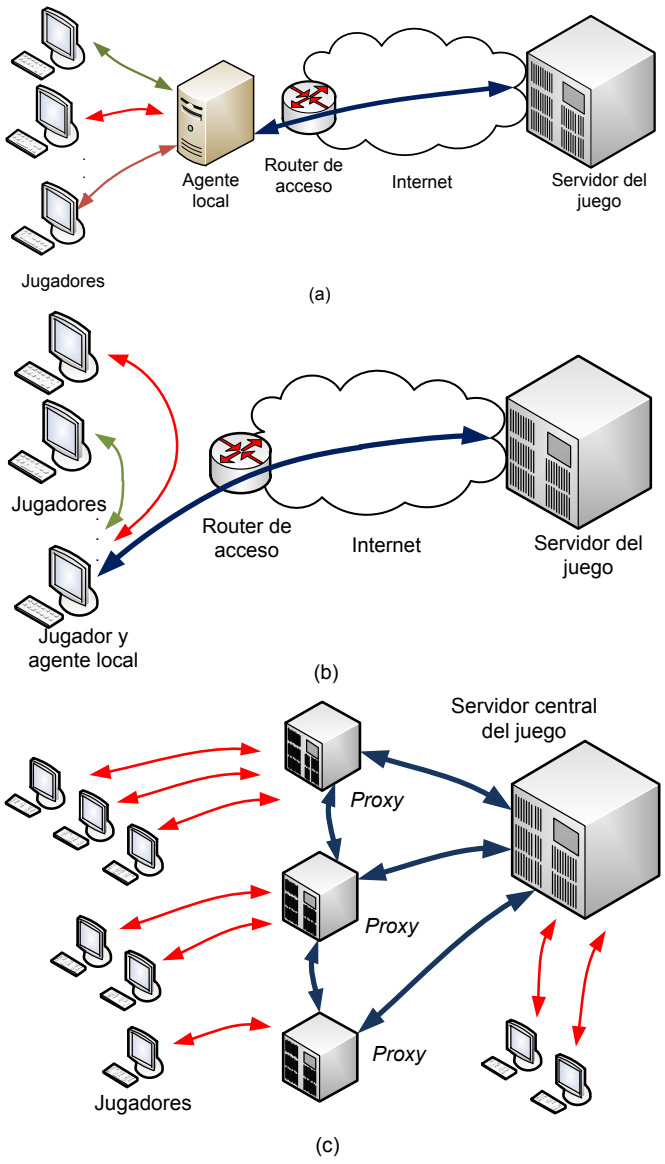


Fig. 1. Escenarios donde varios jugadores comparten la misma ruta: a) jugadores que comparten la red de acceso, usando un agente local. b) jugadores usando la máquina de otro para crear el túnel. c) Tráfico entre *proxies* del mismo juego. Las líneas gruesas representan el tráfico de varios jugadores.

En este trabajo nos centraremos en los escenarios que se presentan en las Fig. 1a y 1b, pero muchas de las conclusiones se pueden aplicar también al escenario de la Fig. 1c. Trataremos de optimizar el tráfico del cliente al servidor, ya que es donde se puede conseguir un mayor ahorro y, en muchos escenarios, como en un acceso ADSL, usa el enlace más restrictivo. Tendremos que medir también la degradación de los parámetros que determinan la calidad experimentada por los usuarios para ajustar correctamente el método, encontrando un compromiso entre el ahorro de ancho de banda y la calidad.

Aunque esta técnica también se puede aplicar a otros géneros, en este trabajo nos centraremos en los juegos FPS, a causa de sus grandes requerimientos de interactividad. La calidad subjetiva depende fundamentalmente del retardo y las pérdidas de paquetes [7]. El tiempo de respuesta del sistema (*System Response Time, SRT*), que se define como el tiempo necesario para detectar un evento del usuario, procesarlo en el servidor actualizando el estado del juego, y presentarlo en el dispositivo de salida correspondiente, debe mantenerse por debajo de unos determinados valores.

El resto del trabajo se ha organizado de la siguiente manera: la siguiente sección presenta los trabajos relacionados. La sección III explica el método de tunelado, compresión y multiplexión propuesto. Los resultados se detallan en la sección IV y, finalmente se exponen las conclusiones.

II. Trabajos Relacionados

A. Tráfico de los juegos online

En la literatura se pueden encontrar muchos trabajos sobre el tráfico que generan los juegos *online*. En este trabajo estudiaremos el tráfico activo del juego, que es el generado una vez comenzada la partida. Este tráfico presenta dos comportamientos diferentes: por un lado, la aplicación cliente se encarga de comunicar las acciones de los jugadores al servidor, usando para ello paquetes pequeños con una frecuencia alta. Por otro lado, el servidor calcula el nuevo estado del juego y se lo envía a todos los jugadores, usando paquetes más grandes, cuyo tamaño depende del número de jugadores. En [8] se presentó un método para extrapolar el tráfico del servidor al cliente, obtenido a partir de medidas empíricas. Obtuvieron las distribuciones para un juego de N jugadores a partir del de 2 o 3. En ese trabajo también se decía que el tráfico del cliente al servidor es independiente del número de jugadores.

En [9] se analizó una traza de 500 millones de paquetes de un servidor de Counter Strike, y a partir de ese análisis se concluyó que el juego está diseñado para saturar el cuello de botella que constituye la red de acceso. También en [10] se analizaron otros juegos en términos de tamaño de paquete y tiempo entre paquetes. En [11] se presentó un resumen de diferentes modelos de tráfico que existen en la literatura para 17 juegos muy populares. Todos estos estudios muestran que estos juegos generan altas tasas de paquetes pequeños. Esto produce una eficiencia muy pobre, por lo que se pueden conseguir ahorros de ancho de banda si se comprimen las cabeceras y se multiplexan paquetes.

En [9] también se dice que el cuello de botella no es sólo el ancho de banda de la red de acceso, sino el número de paquetes por segundo que el *router* puede gestionar. Los *router* están diseñados frecuentemente para paquetes

grandes, y pueden experimentar problemas al gestionar ráfagas con un gran número de paquetes pequeños.

B. *Infraestructura para soporte de juegos online*

El problema de la infraestructura necesaria para dar soporte a estos juegos se ha tratado también en diversos trabajos. Desde el punto de vista del usuario, en [12] se presentó un algoritmo para permitir que el cliente seleccione adaptativamente el mejor servidor para un juego concreto. Esto podría permitir a un grupo de usuarios jugar en el mismo servidor, y así poder usar técnicas de multiplexión.

Desde el punto de vista del servidor, existen dos arquitecturas para soportar este servicio: centralizadas y distribuidas. En las primeras existe un servidor que mantiene el estado del juego y lo distribuye a los jugadores. El problema que presentan es que el servidor constituye un cuello de botella. En las arquitecturas distribuidas [13] no se necesita un servidor central, ya que los jugadores se intercambian directamente la información. Pero esta arquitectura no suele usarse en juegos comerciales.

El problema de la escalabilidad de la infraestructura para soportar estos juegos ha sido estudiado por Mauve y otros [3], y propusieron el uso de *proxies* para conseguir control de congestión, robustez, reducción de los retardos y evitar las trampas de algunos jugadores. Algunos *proxies* podrían situarse cerca de los jugadores, evitando trabajo al servidor central. Así, en la Ref. [4] también se propuso el uso de *booster-boxes,* que se podrían situar cerca del *router*, para así conocer el estado de la red, y ser capaces de dar soporte de red a las aplicaciones. Como se ha dicho en la introducción, la solución propuesta en el presente trabajo podría incluso correr en la máquina de uno de los jugadores.

C. *Algoritmos de compresión*

Existen algunos estándares del IETF para compresión de cabeceras, que se desarrollaron hace varios años: en primer lugar, VJHC [14] presentó un método para comprimir las cabeceras IP/TCP. Algunos años después se presentó IPHC [5], capaz de comprimir también las cabeceras UDP e IPv6. En ese mismo momento se publicó CRTP para comprimir las cabeceras IP/UDP/RTP. Varios años después se mejoró y se denominó ECRTP. Pero estos dos protocolos no son adecuados para comprimir el tráfico de juegos, ya que éste no es RTP. También hubo una propuesta interesante [15] de usar un protocolo similar a RTP para juegos *online*. Una ventaja de esta propuesta es la posibilidad de reutilizar servicios genéricos, evitando así la necesidad de implementarlos para cada juego. Pero hoy en día los juegos comerciales utilizan principalmente paquetes IP/UDP.

Estos algoritmos comprimen la cabecera sólo de nodo a nodo, utilizando la redundancia de los campos de las cabeceras IP, UDP y TCP para evitar enviar algunos de ellos. Se define un *contexto*, que se transmite inicialmente con las primeras cabeceras. Los diferentes campos de las cabeceras se dividen en *estáticos, aleatorios, delta* e *inferidos*. Los *estáticos* se envían en las primeras cabeceras. Los *aleatorios* no se comprimen. Los clasificados como *delta* se codifican con menos bytes que en el campo original. Finalmente, los *inferidos* se deducen de los campos de los niveles inferiores: por ejemplo, la longitud del paquete se puede inferir del campo correspondiente del nivel 2.

ROHCv2 [16] es un estándar más reciente, que puede comprimir las cabeceras IP/UDP/RTP, y también las IP/UDP. Reduce el impacto de la desincronización del contexto mediante un sistema de realimentación que funciona entre el descompresor y el compresor. Utiliza diferentes niveles de compresión, que se corresponden con los modos de operación: *inicialización*, *primer orden* y *segundo orden*. En el último modo, la cabecera se puede comprimir a un solo byte [17]. El uso de estas técnicas avanzadas hace más difícil la implementación [18], añadiendo más retardo de procesado.

III. MÉTODO DE COMPRESIÓN Y MULTIPLEXADO

En esta sección explicaremos el método propuesto para comprimir y multiplexar. Se presentará también un estudio de la eficiencia en ancho de banda que se puede conseguir.

A. *Algoritmo de Tunelado, Compresión y Multiplexión*

En el RFC 4170 [6], el IETF aprobó TCRTP (*Tunneled Compressed* RTP) para comprimir y multiplexar flujos RTP. En primer lugar se utiliza ECRTP para comprimir las cabeceras. Después, varios paquetes se incluyen en uno usando PPPMux. Finalmente, se usa un túnel L2TP para enviar el paquete multiplexado extremo a extremo.

En este trabajo se ha usado un esquema similar, pero en nuestro caso el tráfico no es RTP, por lo que sólo podemos comprimir las cabeceras IP/UDP usando IPHC o ROHCv2. Denominaremos a este método TCM (Tunelado, Compresión, Multiplexión). La Fig. 2 muestra la pila de protocolos y la estructura de un paquete TCM. Se puede dividir en las siguientes partes:

- **Cabecera común (*Common Header,* CH)**: Corresponde a las cabeceras IP, L2TP y PPP.
- **Cabecera PPPMux (MH):** Se incluye al principio de cada paquete comprimido.
- **Cabecera reducida (*Reduced Header,* RH):** Corresponde a la cabecera comprimida IP/UDP del paquete original.
- **Payload (P):** Contenido de los paquetes originales generados por la aplicación.

B. *Análisis teórico del método propuesto*

A continuación expondremos un análisis del ahorro de ancho de banda que se puede lograr con esta técnica. Asumiremos que el tamaño del paquete multiplexado nunca excederá los 1500 bytes, cosa cierta para los tráficos usados.

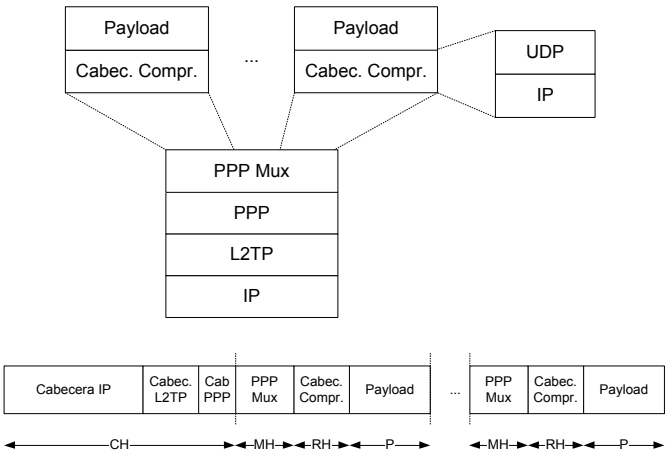


Fig. 2. Pila de protocolos de TCM y esquema de un paquete multiplexado

Como se ha dicho en la introducción, el retardo tiene una gran importancia en este servicio. Por eso, hemos usado una política de multiplexión que mantiene el retardo por debajo de una cota superior, denominada *T,* de manera que un paquete multiplexado se envía al final de cada periodo, incluyendo a todos los que ya hayan llegado (Fig. 3). Hay dos excepciones: si no ha llegado ningún paquete, no se envía nada; y si hay un solo paquete, se enviará en su forma nativa, ya que el túnel lo haría más grande.

Nos referiremos a los paquetes generados por la aplicación como *nativos*, en contraste con los multiplexados (*mux*). Con esta política, el retardo añadido será en media *T/2*, y su cota superior será *T*.

Un parámetro interesante es la relación de anchos de banda (*Bandwidth Relationship, BWR)*, entre las situaciones de usar o no multiplexión. Denominaremos *k* al número de paquetes que llegan en un periodo. *NH* se refiere al tamaño de una cabecera normal IP/UDP. Para obtener *BWR*, calcularemos primero el número de bytes enviado en un periodo cuando se usa multiplexión, distinguiendo los casos de uno o varios paquetes:

$$bytes_{mux} = Pr\,(k=1)(\,NH + E\,[P]) + \\ + Pr(\,k>1)\,[CH+E[k|k>1](MH+E[RH]+E[P])] \quad (1)$$

Y en el caso de los paquetes nativos será:

$$bytes_{nativo} = E[k]\ (NH + E\,[P]) \quad (2)$$

Dividiendo (1) entre (2) obtenemos *BWR*:

$$BWR = \frac{\Pr(k=1)}{E[k]} + \Pr(k>1)\frac{CH}{E[k](NH+E[P])} + \\ + \Pr(k>1)\frac{E[k \mid k>1]}{E[k]}\frac{MH+E[RH]+E[P]}{NH+E[P]} \quad (3)$$

El primer término está causado por la decisión de no multiplexar cuando hay un solo paquete. El segundo expresa cómo la cabecera común es compartida por todo el paquete, y se va haciendo menor según aumenta el número de paquetes multiplexados. El tercer término depende del algoritmo de compresión y del tamaño medio de paquete generado por la aplicación.

Por tanto, si tenemos un gran número de usuarios y un periodo grande, el número de paquetes multiplexados será también grande, y el primer y segundo términos tenderán a ser despreciables. Con respecto al tercero, *Pr(k>1)* será casi la unidad, y lo mismo ocurrirá con *E[k|k>1]/E[k]*. De esta manera, podemos obtener una expresión para la asíntota de *BWR:*

$$BWR_a = \frac{MH+E[RH]+E[P]}{NH+E[P]} \quad (4)$$

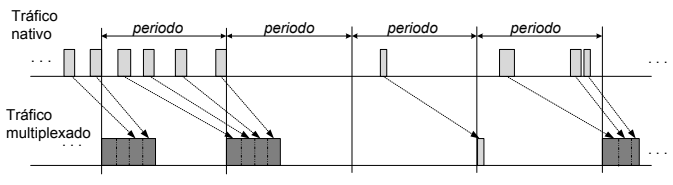


Fig. 3. Comportamiento de la política de multiplexión

Podemos observar que cuanto menor sea *E[P]*, menor será el valor de la asíntota. Por eso, la técnica presentada tendrá un buen comportamiento para aplicaciones que generen un gran número de paquetes pequeños, como hacen los juegos FPS. Lógicamente, lo esperado es que a mayor número de jugadores se consiga un mayor ahorro, ya que el mismo número de paquetes podrá ser multiplexado añadiendo retardos menores. El incremento de *T* será beneficioso para *BWR*, pero no lo podemos aumentar indefinidamente, puesto que los jugadores son muy sensibles al retardo.

Por claridad, no se va a incluir aquí el cálculo de *E[k|k>1], Pr(k=1)* y *Pr(k>1)*, que se puede encontrar en el Apéndice. Como se verá, estos valores varían según el comportamiento de cada juego.

Para poder obtener resultados numéricos y gráficos, utilizaremos parámetros reales de algunos juegos comerciales, y también los de los protocolos usados:

- *NH*: 28 bytes para IPv4 y 48 para IPv6.
- *CH*: 25 bytes para IPv4: 20 corresponden a la cabecera IP, 4 a L2TP y 1 a la cabecera PPP. Para IPv6, *CH* = 45 bytes.
- *MH*: 2 bytes, correspondientes a PPPMux.
- *E[P]:* El valor del *payload* UDP depende de la aplicación usada.
- *E[k]:* El número de paquetes por segundo generados por los *N* jugadores.
- *E[RH]:* En este ejemplo, para situarnos en el peor caso, hemos considerado IPHC comprimiendo las cabeceras UDP a 2 bytes, usando sólo 8 bits para el campo CID, y evitando el *checksum* opcional. Las cabeceras IPv4 e IPv6 se pueden también comprimir a 2 bytes. Así que consideraremos una media de 4 bytes para las cabeceras comprimidas y 28 o 48 bytes para las cabeceras completas, que se envían cada 5 segundos (el parámetro F_MAX_TIME que usa por defecto IPHC).

Usando estos valores, obtenemos los resultados que se muestran en la Tabla 1. Son valores de la asíntota, o sea, el mejor *BWR* que se puede obtener si el número de jugadores y el periodo son lo suficientemente grandes. Se han obtenido para IPv4 e IPv6. Hemos seleccionado algunos juegos populares, y los valores concretos se han obtenido de [10] y [11]. Los valores para Halo2 se refieren a una consola con un solo usuario [19]. El valor de λ (paquetes por segundo) para Quake 3 es el que se obtiene con las tarjetas gráficas más rápidas.

Observamos que los valores de *BWR* obtenidos son significativos: todos los juegos permiten un ahorro de un 30% en ancho de banda para IPv4, y este ahorro puede llegar hasta el 54% si se usa IPv6 con algunos títulos.

Para hacernos una mejor idea de los beneficios del sistema, presentaremos ahora algunas gráficas que ilustren el comportamiento del *BWR* no sólo en la asíntota, sino con diferentes valores de número de usuarios y periodo. Dado que tenemos que representar las gráficas para un juego concreto, hemos seleccionado *Half Life Counter Strike 1* a causa de su popularidad y de la disponibilidad de muchos estudios de su comportamiento [9], [20]. La Fig. 4a

Juego	*Motor gráfico*	*E[P]*	*λ*	*BWR_a IPv4*	*BWR_a IPv6*
Unreal T 2003	Unreal 2.0	29.5	25	62%	46%
Quake III	Id Tech 3	36.15	93	65%	50%
Quake II	Id Tech 2	37	26.38	66%	51%
Counter Strike	GoldSrc	41.09	24.65	68%	53%
Halo 2	Halo2	43.2	25	69%	54%

Tabla 1. Valores de la asíntota de *BWR* para diferentes juegos

representa el *BWR* para IPv4 en función del número de jugadores y el periodo. Si dejamos fijo el número de jugadores, obtenemos la Fig. 4b y fijando el periodo, la Fig. 4c. El comportamiento asintótico se puede observar para ambos parámetros, y por eso consideramos que la zona más interesante se da cuando *BWR* está entre 0,70 y 0,75. Por ejemplo, si observamos la gráfica de 20 jugadores de la Fig. 4b, una vez que se alcanza el valor de 0,75, el incremento del retardo para mejorar el ahorro de ancho de banda supondrá un beneficio muy pequeño.

También el número de jugadores influye. Lógicamente, si hay más jugadores, se podrá conseguir el mismo valor de *E[k]* para valores más pequeños de *T*. Por tanto, confirmamos que el aumento del número de jugadores es siempre beneficioso. De hecho, en el caso de tener solamente 5 jugadores, quizá lo mejor sea mantener el valor de *BWR* en torno a 0,80. Lógicamente, el valor del retardo de red tendrá una cierta influencia en el valor elegido para *T*. Si la red es rápida, podremos permitirnos añadir un retardo mayor para así mejorar el ahorro de ancho de banda.

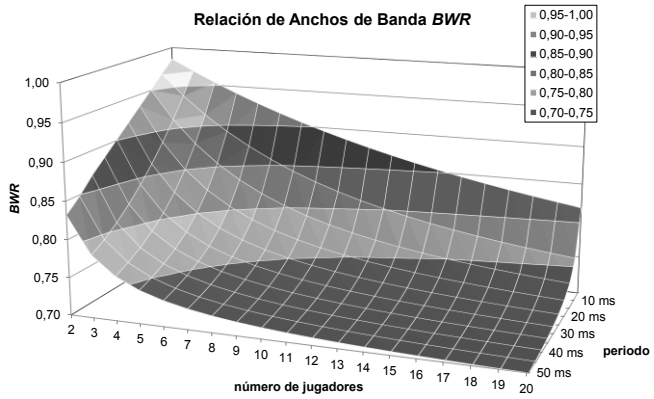
Podemos hacer una última observación: como se ha dicho en las secciones previas, una limitación de los *router* comerciales es el número de paquetes por segundo que pueden gestionar. El método presentado también reduce esta cantidad por un factor de *E[k]*.

C. *Retardos en el sistema*

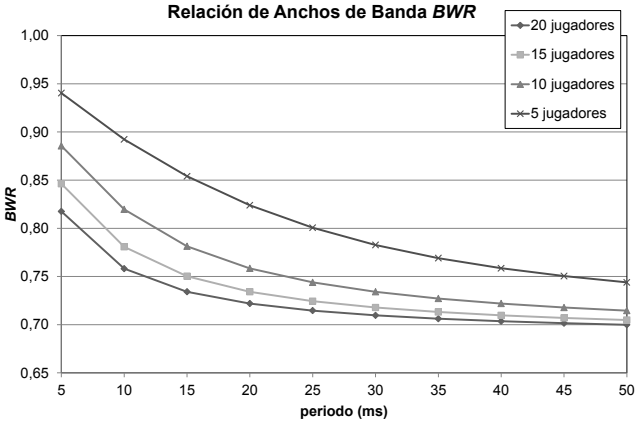
En este apartado se estudia el impacto del método propuesto en el *SRT*. La Fig. 5 muestra un esquema del sistema, con los retardos que se suman para obtener el valor de *SRT*.

- $T_{retención}$ es el tiempo que un paquete pasa en la cola del multiplexor.
- $T_{proceso}$ representa el tiempo que el paquete pasa en el multiplexor y demultiplexor. En [21] se implementó un multiplexor para tráfico RTP, y el tiempo de procesado era menor de 1 ms.
- $T_{encolado}$ es el tiempo de espera en la cola del *router* de acceso. El método presentado no lo modifica directamente, aunque al multiplexar cambiará el tráfico ofrecido a la cola.
- T_{red} es el retardo de red, que tampoco se ve afectado.

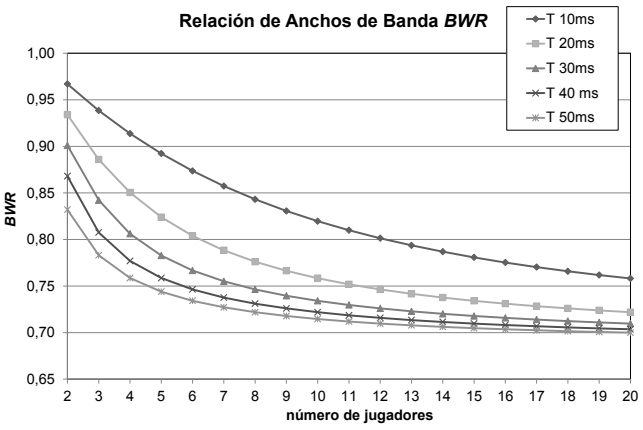
Por tanto, el único retardo significativo que se añade es $T_{retención}$, que es *T/2* en media. En [7] se concluye que el retardo tolerado para Quake 3 está entre 150 y 180 ms, mientras que para Counter Strike está por encima de 200 ms. Por tanto, el tiempo añadido al multiplexar puede ser asumido sin problemas para el usuario.



(a)



(b)



(c)

Fig. 4 *BWR* en función de a) número de jugadores y periodo. b) periodo. c) número de jugadores.

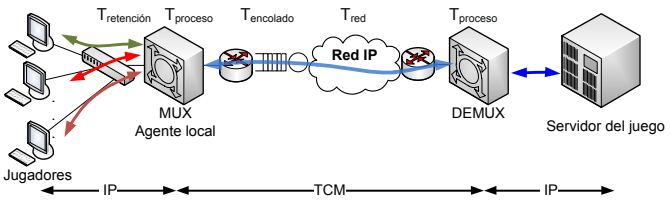


Fig. 5. Retardos del sistema

En este trabajo no se ha considerado la posibilidad de modificar la aplicación pero, si se pudiese hacer, se podría incluir una primera fase de sincronización para hacer que todas las máquinas de la misma partida generasen los paquetes en el mismo instante. De esta manera, este retardo se podría reducir significativamente para los juegos que usan un tiempo entre paquetes fijo.

IV. PRUEBAS Y RESULTADOS

En esta sección presentamos algunos resultados de simulación, obtenidos con trazas reales de un juego FPS. En primer lugar describiremos el método usado para generar los tráficos usados en las pruebas.

Para poder comparar los resultados de simulación con los teóricos, se ha usado el mismo juego: Half Life Counter Strike 1. Las trazas de tráfico se han obtenido del proyecto CAIA (por ejemplo, la traza para 5 jugadores se puede encontrar en [22]). Hay disponibles trazas desde 2 hasta 9 jugadores. Los primeros 10.000 paquetes no se consideran, y sólo se incluyen los siguientes 5.000*número de jugadores. Así se asegura que el tráfico capturado sólo se corresponde con tráfico de juego activo, que es el que queremos estudiar.

Para obtener trazas correspondientes a un mayor número de jugadores, se han sumado las que hay disponibles. Por ejemplo, la traza de 20 jugadores se ha obtenido sumando las de 9, 6 y 5 jugadores. Esto es posible gracias a la propiedad del tráfico del cliente al servidor, cuya distribución es independiente del número de jugadores [8], [20]: el tráfico cliente a servidor de una partida de 20 jugadores es similar al que generarían una partida de 9, otra de 6 y otra de 5 jugadores. Lógicamente, se ha cortado la duración de las trazas a la más corta, obteniendo 110 segundos de tráfico activo.

Se han realizado simulaciones con Matlab para obtener trazas comprimidas y multiplexadas, como se muestra en la Fig. 6. En primer lugar, se separan las trazas de los distintos jugadores, y se elimina el tráfico del servidor a los clientes. Para generar el tráfico en las pruebas, solamente necesitamos la información del instante de generación, el usuario y el tamaño del paquete. Después, se aplica la compresión a las cabeceras IP/UDP de cada flujo. Finalmente, usando el periodo T se obtienen los tamaños e instantes de los paquetes multiplexados.

El juego estudiado tiene tres diferentes comportamientos posibles, dependiendo del método de *render* [11], [20]. En nuestro caso, las trazas se han obtenido con OpenGL, que es el método más usado, y tiene dos posibles valores para el tiempo entre paquetes: 33 y 50 ms, cada uno con una probabilidad del 50%. Esto hace que el valor de λ sea 24 paquetes por segundo. El análisis de las probabilidades para este método se ha incluido en el Apartado B del Apéndice.

La Fig. 7 compara los valores teóricos de *BWR* con los obtenidos en las simulaciones. Se puede comprobar que los valores son muy similares, excepto por pequeñas diferencias para pocos jugadores y valores pequeños del periodo. La causa de estas diferencias es que el tiempo entre paquetes no es exactamente el esperado, sino que hay pequeñas variaciones en torno a 33 y 50 ms, como se puede ver en el histograma (Fig. 8).

La Fig. 9 presenta el número de paquetes por segundo que el *router* tiene que gestionar. Como se ha explicado en la sección II.A, la reducción de este parámetro también resulta interesante. Se observa que a mayor valor del periodo, más se reduce la cantidad de paquetes por segundo, que tiende a ser el inverso del periodo, independientemente del número de jugadores. La Fig. 10 presenta el tamaño medio de los paquetes multiplexados, que crece linealmente con el periodo.

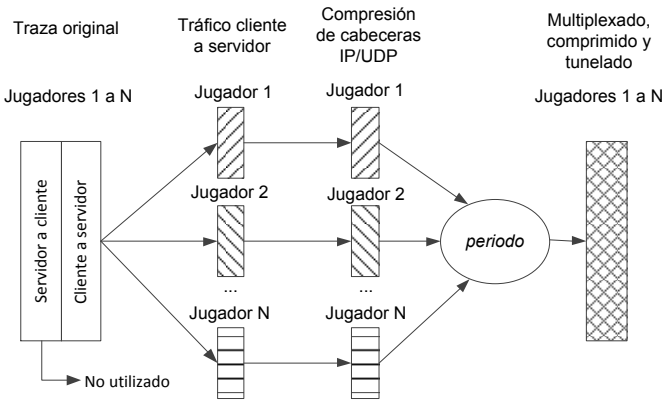


Fig. 6. Método utilizado para construir las trazas

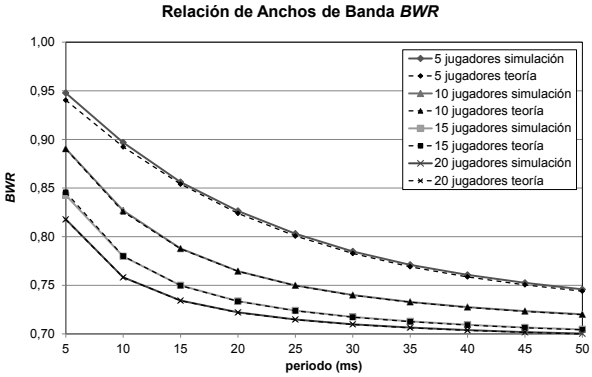


Fig. 7. Comparación de los resultados teóricos y de simulación para *BWR*

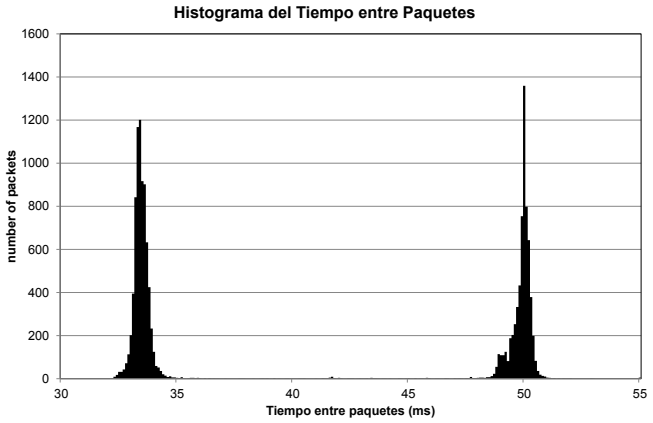


Fig. 8. Histograma del tiempo entre paquetes en ms

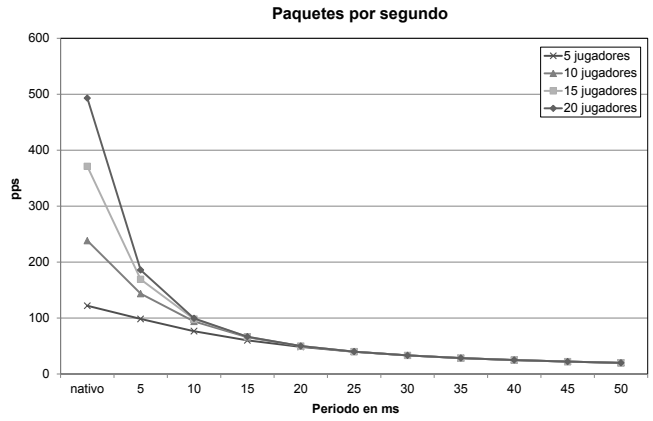


Fig. 9. Paquetes por segundo gestionados por el *router*

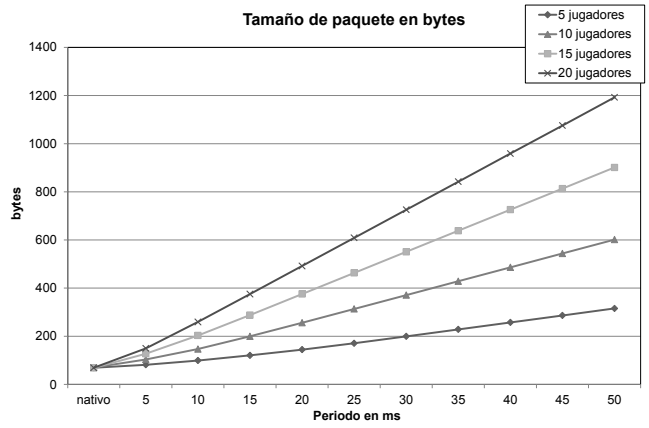


Fig. 10. Tamaño de los paquetes multiplexados

V. CONCLUSIONES

En este trabajo se ha presentado una técnica de tunelado, compresión y multiplexión, que se puede usar para ahorrar ancho de banda mediante la compresión de cabeceras y la unión de paquetes pequeños en otros más grandes. Puede reducir el *overhead* del tráfico de juegos *online,* ya que estas aplicaciones generan altas tasas de paquetes pequeños. El método utiliza un protocolo de compresión de las cabeceras IP/UDP, multiplexión con PPPMux y un túnel L2TP para poder trabajar extremo a extremo.

Las empresas que desarrollan juegos podrían estar interesadas en reducir el ancho de banda utilizado, y también el número de paquetes por segundo que tienen que gestionar. El ahorro de ancho de banda también puede proporcionar un mejor comportamiento en redes de acceso con ancho de banda limitado.

El método se ha probado con tráfico de juegos FPS, ya que estos programas tienen unos requerimientos temporales muy estrictos, pues los jugadores demandan una gran interactividad. Se han realizado simulaciones para estudiar el ahorro de ancho de banda, y los resultados muestran que se puede reducir un 38% para IPv4, y más de un 50% para IPv6. Los retardos añadidos se pueden mantener en niveles bajos si hay un número suficiente de jugadores que compartan la misma ruta. Como línea futura se considera el desarrollo de un algoritmo que ajuste dinámicamente los parámetros de la multiplexión según el número de jugadores y las estadísticas de la red, como el retardo y las pérdidas.

APÉNDICE

Se presentan aquí los cálculos necesarios para obtener una expresión analítica de *E[k|k>1],* que se requiere para construir las gráficas de *BWR* presentadas en la sección III. En primer lugar, podemos calcular *E[k]* como:

$$E[k] = Pr(k=0)\ E[k|k=0] + Pr(k=1)\ E[k|k=1] + \\ + Pr(k>1)\ E[k|k>1] \quad (5)$$

Pero si tenemos en cuenta que $E[k|k=0]=0$ y que $E[k|k=1]=1$, obtenemos:

$$E[k \mid k>1] = \frac{E[k] - \Pr(k=1)}{\Pr(k>1)} \quad (6)$$

Por lo tanto, necesitamos obtener las expresiones de $Pr(k=0)$, $Pr(k=1)$ y $Pr(k>1)$. En los análisis previos hemos definido k como el número total de paquetes llegados al multiplexor en un periodo, es decir, la suma de los paquetes de cada jugador. Ahora definiremos l como el número de paquetes de un solo jugador. Si consideramos que las llegadas de paquetes de los N jugadores son independientes, tenemos $E[k]=N\,E[l]=N\,\lambda T$.

A. Tiempo entre paquetes constante

Las probabilidades que tenemos que calcular dependen de la distribución estadística del tiempo entre paquetes de cada juego. Consideraremos en primer lugar un tiempo entre paquetes constante, como ocurre en muchos juegos [11]. Denominaremos t al tiempo entre paquetes. Consideramos que se cumple $T < 2t$, para evitar grandes retardos. Por tanto, el valor máximo de l será 2, y entonces:

$$E[l] = \lambda T = Pr(l=1) + 2\,Pr(l=2) \quad (7)$$

En el caso de $T \le t$, tenemos que $Pr(l=2)=0$, por lo tanto si usamos (7) obtenemos:

$$Pr(l=1) = E[l] = \lambda T \quad (8)$$

$$Pr(l=0) = 1 - Pr(l=1) = 1 - \lambda T \quad (9)$$

Y en el caso de $T > t$, tenemos que $Pr(l=0)=0$, así que teniendo en cuenta que la suma de las probabilidades es la unidad y usando (7) de nuevo, obtenemos:

$$Pr(l=1) = 2 - E[k] = 2 - \lambda T \quad (10)$$

$$Pr(l=2) = 1 - Pr(l=1) = E[k] - 1 = \lambda T - 1 \quad (11)$$

Por tanto, podemos hallar $Pr(k=0)$:

$$Pr(k=0) = [Pr(l=0)]^N \quad (12)$$

Pero $Pr(k=1)$ es nula en el caso de tener más de un jugador y $T > t$, ya que cada jugador habrá enviado al menos un paquete en el periodo. Por tanto, la expresión para $Pr(k=1)$ con $T \le t$ será:

$$Pr(k=1)|_{T \le t} = \binom{N}{1} Pr(l=1)[Pr(l=0)]^{N-1} \quad (13)$$

B. Dos posibles valores para el tiempo entre paquetes

En este apartado se considera el caso de un juego que genera paquetes usando dos diferentes valores para el tiempo entre paquetes. Denominaremos t_1 al menor y t_2 al mayor, siendo p_1 y p_2 lar probabilidades respectivas de tener t_1 y t_2. Consideraremos, como ocurre en el juego considerado en este trabajo, que $T < 2t_1$, y $t_2 < 2t_1$. En este caso podemos obtener el valor de λ como:

$$\lambda = \frac{1}{p_1 t_1 + p_2 t_2} \tag{14}$$

Ahora distinguiremos dos casos diferentes. En primer lugar, si tenemos $T < t_1$, se trata del mismo caso que el visto en (8) y (9), ya que $Pr(l=2)=0$.

Y en el caso de $t_1 \leq T < t_2$, deberemos encontrar $Pr(l=0)$, es decir, la probabilidad de que no hayan llegado paquetes en el periodo T, que se corresponde con la probabilidad de que el periodo comience en los primeros t_2-T segundos de un tiempo entre paquetes de duración t_2. Por tanto, considerando que los tiempos entre paquetes consecutivos son independientes:

$$\Pr(l = 0) = p_2 \frac{t_2 - T}{p_1 t_1 + p_2 t_2} = p_2 \lambda (t_2 - T) \tag{15}$$

Y ahora, usando (7) y sabiendo que la suma de las probabilidades debe ser la unidad, podemos obtener:

$$Pr(l=1) = \lambda[T - 2p_1(T-t_1)] \tag{16}$$

$$Pr(l=2) = p_1 \lambda (T-t_1) \tag{17}$$

Finalmente, podemos usar los resultados de (12) y (13) para obtener las probabilidades de los distintos valores de k.

AGRADECIMIENTOS

Este trabajo ha sido financiado parcialmente por el Proyecto CPUFLIPI (MICINN TIN2010-17298), por el Proyecto MBACToIP, de la Agencia I+D del Gobierno de Aragón e Ibercaja Obra Social, y por el Proyecto NDCIPI-QQoE de la Cátedra Telefónica, de la Univ. de Zaragoza.

REFERENCIAS

[1] C. Chambers, W. Feng, S. Sahu, D. Saha: Measurement-based Characterization of a Collection of On-line Games. In Proceedings of the 5th ACM SIGCOM conference on Internet Measurement (IMC'05). USENIX Association, Berkeley (2005)

[2] K. Chen, P. Huang, C. Lei: Game traffic analysis: An MMORPG perspective. In Proceedings of the international workshop on Network and operating systems support for digital audio and video (NOSSDAV'05), pp. 19-24. ACM, New York (2005)

[3] M. Mauve, S. Fischer, J. Widmer: A Generic Proxy System for Networked Computer Games. In Proceedings of the 1st workshop on Network and system support for games (NetGames'02), pp. 25--28. ACM, New York (2002)

[4] D. Bauer, S. Rooney, P. Scotton: Network Infrastructure for Massively Distributed Games. In Proceedings of the 1st workshop on Network and system support for games (NetGames'02), pp. 36-43. ACM, New York (2002)

[5] M. Degermark, B. Nordgren, D. Pink: RFC 2507: IP Header Compression (1999)

[6] B. Thompson, T. Koren, D. Wing. RFC 4170: Tunneling Multiplexed Compressed RTP (TCRTP), Nov. 2005.

[7] S. Zander, G. Armitage: Empirically Measuring the QoS Sensitivity of Interactive Online Game Players. Australian Telecommunications Networks & Applications Conference 2004 (ATNAC2004), Sydney, Australia, Dic. 2004.

[8] P. Branch, G. Armitage: Extrapolating Server To Client IP traffic From Empirical Measurements of First Person Shooter games. In Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games (NetGames '06). ACM, NY, USA.

[9] W. Feng, F. Chang, W. Feng, J. Walpole: Provisioning On-line Games: A Traffic Analysis of a Busy Counter-Strike Server. SIGCOMM Comput. Commun. Rev. 32, p. 18 (2002)

[10] W. Feng, F. Chang, W. Feng, J. Walpole: A Traffic Characterization of Popular On-Line Games. IEEE/ACM Trans. Netw., pp. 488-500 (2005)

[11] S. Ratti, B. Hariri, S. Shirmohammadi, A Survey of First-Person Shooter Gaming Traffic on the Internet, IEEE Internet Computing, pp. 60-69, Sept./Oct. (2010)

[12] K. Lee, B. Ko, S. Calo: Adaptive Server Selection for Large Scale Interactive Online Games. In Proc. 14th International Workshop on Network and operating systems support for digital audio and video (NOSSDAV'04), pp. 152--157. ACM, New York (2004)

[13] L. Gautier, C. Diot: Design and Evaluation of MiMaze, a Multi-player Game on the Internet. In Proceedings of the IEEE International Conference on Multimedia Computing and Systems (ICMS'98), IEEE Computer Society, pp. 233. Washington (1998)

[14] V. Jacobson: RFC 1144: Compressing TCP/IP Headers for Low-Speed Serial Links (1990)

[15] M. Mauve, V. Hilt, C. Kuhmünch, W. Effelsberg: RTP/I-Toward a Common Application Level Protocol for Distributed Interactive Media. In Proceedings of IEEE Transactions on Multimedia, pp. 152-161 (2001)

[16] G. Pelletier, K. Sandlund, RFC 5225: RObust Header Compression Version 2 (ROHCv2) (2008)

[17] A. Couvreur, L. M. Le-Ny, A. Minaburo, G. Rubino, B. Sericola, L. Toutain, Performance analysis of a header compression protocol: The ROHC unidirectional mode, Telecommunication Systems, vol. 31, no. 6, pp. 85–98 (2006)

[18] E. Ertekin, C. Christou: Internet protocol header compression, robust header compression, and their applicability in the global information grid. IEEE Communications Magazine, vol. 42, pp. 106-116 (2004)

[19] S. Zander, G. Armitage, A traffic model for the Xbox game Halo 2. In Proc. International Workshop on Network and operating systems support for digital audio and video (NOSSDAV'05). ACM, New York, NY, USA, pp 13-18 (2005)

[20] T. Lang, G. Armitage, P. Branch, H. Choo: A Synthetic Traffic Model for Half-Life. In Australian Telecom, Networks and Applications Conference (ATNAC) Melbourne (2003)

[21] H. Sze, C. Liew, J. Lee, D. Yip: A Multiplexing Scheme for H.323 Voice-Over-IP Applications. IEEE J. Select. Areas Commun. Vol. 20, pp. 1360-1368 (2002)

[22] L. Stewart, P. Branch: HLCS, Map: dedust, 5 players, 13Jan2006. Centre for Advanced Internet Architectures SONG Database, http://caia.swin.edu.au/sitcrc/hlcs_130106_1_dedust_5_fragment.tar.gz

Modelado de tráfico para un servicio de videochat

Wilmar Yesid Campo Muñoz, Gabriel Elías Chanchí, José L. Arciniegas H.
Departamento de Telemática
Universidad del Cauca
Calle 5 No. 4 -70. Popayán, Cauca, Colombia.
{wilicampo, gabrielc, jlarci}@unicauca.edu.co

Roberto García, Xabiel García Pañeda, David Melendi Palacio
Departamento de Informática
Universidad de Oviedo
Campus de Viesques, sn, 33204. Gijón, Asturias, España
{garciaroberto, xabiel, melendi}@uniovi.es

***Resumen*- Cada vez existen más servicios corriendo sobre las redes de comunicaciones, por ello, se hace necesario llevar a cabo estudios del comportamiento del tráfico generado y su impacto sobre las redes y otros servicios. En este artículo se presenta la construcción de un modelo de tráfico y su evaluación mediante técnicas de emulación. Se lleva a cabo la caracterización del tráfico para la obtención del modelo matemático que describe el comportamiento del servicio y permite su simulación. Se describe el entorno de experimentación y los escenarios de evaluación, donde se inyecta tráfico real sobre una red virtual. Finalmente, los resultados muestran las previsiones a tener en cuenta en cuanto a consumo de recursos en el uso de este tipo de técnicas de emulación y el análisis del comportamiento del modelo.**

Palabras Clave- Modelado, análisis de tráfico, emulación, videochat

I. INTRODUCCIÓN

Internet ha tenido una marcada evolución en los últimos años, permitiendo el acceso y el intercambio de información de manera ágil, con características de flexibilidad en la tecnología de acceso y con capacidades de integración a nivel de servicios. La tendencia actual de Internet es fomentar la colaboración y el trabajo en comunidad a través de servicios construidos bajo el concepto de la Web 2.0. Estos servicios están siendo agrupados o integrados en plataformas tales como redes sociales, la cuales a pesar de haber nacido con fines de ocio han dado surgimiento a las Comunidades Académicas Virtuales (CAV), usadas como instrumento de apoyo a los procesos de E-Learning [1]. Es importante conocer el comportamiento de los diferentes servicios, como los foros, chats, blogs, wikis o servicios más complejos, como un servicio de Videochat.

Actualmente existe el interés de incorporar este tipo de servicios dentro de plataformas de E_Learning para no tener que recurrir a servicios externos, por esta razón se ha elegido construir el modelo del tráfico para un servicio de Videochat. Para el estudio del modelo del servicio, se hizo uso de las técnicas de emulación, entendida como la ejecución de simulaciones hibridas con tráfico real, usando para ello la herr

ventos discretos (DES). Para el primero de estos, se usa el tráfico background, para caracterizar y simular el comportamiento de la red a un nivel abstracto, donde en cada ejecución se calcula el efect
cargas background sobre el tráfico de interés. Para la las trazas de tráfico real. Luego, se caracteriza el servicio, identificando cada uno de los actores, los protocolos y sus fases de conexión, transmisión de datos y desconexión. Con este proceso se obtiene el comportamiento del servicio y el modelo matemático denominado tráfico explícito, que describe el flujo de tráfico entre los diferentes actores, para su posterior programación a través de la herramienta de simulación. Finalmente se crea una red simulada o virtual, sobre la cual se ejecuta el modelo ant

El tráfico real se toma desde los dispositivos reales qu
en el servicio de Videochat. Es llevado hasta la red virtual, mediante el modulo em In The Loop" (SITL) de OPNET Modeler y fluye por esta red hasta alcanzar su destino, permitiendo que los dispositivos reales interactúen entre sí, a través de la red simulada.

Así, mediante estas técnicas de emulación donde se combinan entidades simuladas con componentes reales, se obtiene un mayor realismo y se proporciona a los usuarios la oportunidad de interactuar en la red y ajustar los parámetros del modelo mediante repeticiones de los experimentos. El tráfico real puede circular por redes simuladas complejas que sirven de interconexión entre los dispositivos reales, creando modelos de tráfico más exactos y evitando la necesidad de contar con toda una infraestructura de red. La simulación DES captura todos los mecanismos y efectos de los protocolos y los servicios, reduciendo drásticamente los cálculos necesarios para modelar el tráfico en la red, y manteniendo los recursos necesarios para capturar en detalle, parámetros específicos de interés. Mediante estos entornos de emulación se obtiene una mayor precisión con un consumo de recursos asequible.

La contribución de este artículo es la construcción de un modelo de tráfico y su evaluación mediante técnicas de emulación, además se presenta el desarrollo de la plataforma de emulación. Se muestran las limitaciones que aún existen en este campo para trabajar con servicios masivos. Para la evaluación del servicio de Videochat. Se inicia con la obtención del tráfico explicito. Se construyen los escenarios de evaluación y se analiza el rendimiento del servidor encargado de ejecutar la herramienta de emulación. Finalmente se analiza los efectos que se generan sobre los modelos de tráfico.

Para su descripción, el artículo tiene la siguiente organización: En la sección 2 se presenta los trabajos

relacionados con esta investigación. En la sección 3 se describe el servicio de Videochat. En la sección 4 se presenta la caracterización del tráfico con cada uno de sus procesos. En la sección 5 se describe el entorno de experimentación, incluyendo la red real y la red simulada. En la sección 6 se exponen los diferentes escenarios de prueba. En la sección 7 se presentan los resultados y finalmente en la sección 8 se presentan las conclusiones y los trabajos futuros.

II. Trabajos Relacionados

Actualmente existe un gran número de investigaciones relacionadas con vídeo, procesos de codificación y sus entornos de simulación. En [2] las investigaciones se realizan con una aplicación de videochat, donde se presenta un algoritmo de segmentación automática de imágenes de vídeo captadas por una cámara web. Así, la construcción de modelos de tráfico y las pruebas de evaluación de los servicios es un gran reto para los diseñadores de redes e investigadores, donde la fidelidad está reñida con la velocidad. Su estudio puede ser abordado mediante, bancos de pruebas, modelos analíticos, simulaciones o entornos de emulación.

Los bancos de prueba son una representación a escala del entorno real del servicio, que permite capturar con precisión las transacciones detalladas de la red, la desventaja de este enfoque son los costos en infraestructura. PlanetLab [3] es un ejemplo de una colección de máquinas distribuidos a través de Internet como un laboratorio para que los investigadores desarrollen nuevos servicios. Los modelos analíticos presentan la formulación matemática de los servicios para ser desplegados sobre las redes [4]. Pero cuando el sistema se vuelve demasiado complejo, se requiere hacer supuestos, para mantener los modelos manejables lo cual va en contraposición a los detalles de implementación.

En los procesos de simulación se puede combinar en sus experimentos tráfico analítico y tráfico explicito, presentando las ventajas de la velocidad de ejecución y la flexibilidad por estar conformada solo por entidades virtuales, aunque existe dentro de la comunidad dudas acerca de la fiabilidad y la precisión de las simulaciones [5]. La dificultad en convencer a los proveedores de servicios de la necesidad de realizar cambios sobre su infraestructura, ofreciendo como argumento solo datos basados en simulaciones. Este hecho impulso a crear modelos diferentes. Así, surge la emulación como alternativa, la cual es más realista al involucrar en sus experimentos entidades reales. Es viable cuando se cuenta con suficientes recursos, como dispositivos y tiempo. Su costo computacional es intermedio entre las simulaciones y los bancos de pruebas [6].

En [7] se presenta una plataforma de emulación la cual permite ejecutar una red virtual de cientos de nodos en una sola máquina de usuario final, sin embargo esta propuesta se enfoca en un solo tipo de redes. Surgen propuestas de investigación orientadas hacia la creación de emuladores a la medida, como la descrita en [8], lo cual puede convertirse en un problema adicional, alejándose del propósito de esta investigación, que es la construcción de un modelo de tráfico y su evaluación mediante técnicas de emulación para un servicio liviano, cuando es llevado a entornos masivos.

En [9] los autores presentan la emulación como una alternativa a usar antes de una implementación real, para servicios de internet a gran escala. Pero no tienen por qué ser exclusiva de estos entornos. Es más puede ser una gran alternativa para servicios no masivos.

Las herramientas de emulación de mayor interés en la comunidad de investigadores son NS-3 y OPNET Modeler, la principal diferencia es que la primera de ellas no es un producto terminado y al ser open source no se garantiza el soporte, por su parte la segunda requiere licenciamiento. Con estas herramientas se han desarrollado diversos trabajos en el campo de la emulación. En NS-3, por ejemplo para redes IPv6 [10], o para redes Wimax [11]. Con OPNET Modeler, en [12] se propone el Software-in-the-Loop en la construcción de modelos, para evitar realizar la validación de los mismos. En [13] se describe el diseño y la implementación de una red de comunicaciones del ejército de los Estados Unidos, realizando pruebas con el módulo SITL en diferentes fases del proyecto. Para esta investigación se amplía dicho escenario, puesto que además de trabajar con dicho módulo, se trabaja también con tráfico analítico y explicito.

En el Laboratorio del Grupo de Investigación de Sistemas de Distribución Multimedia (DMMS) de la Universidad de Oviedo, se desarrolló una metodología para la construcción de entornos de emulación, la cual sirve de soporte al trabajo aquí presentado [14]. Como caso de estudio se presenta un servicio de Videochat.

III. Servicio De Videochat

En la Universidad de Oviedo se desarrolló el servicio de Videochat, implementado mediante un servicio Web y diseñado para ser utilizada solo por usuarios autorizados. El ancho de banda total que consume el servicio es de 444 Kbps, (Audio 44 Kbps). El servidor de streaming utilizado fue el Flash Media Server (FMS), el cual obedece a una arquitectura cliente servidor. Para la captura y codificación del audio y el vídeo en vivo, se utiliza el Flash Media Encoder. La comunicación se hace mediante una conexión persistente, usando el protocolo Real Time Messaging Protocol (RTMP). Este protocolo usa TCP a nivel de capa de trasporte y soporta el flujo de streaming del servidor FMS [15].

RTMP tiene tres variaciones: RTMP simple, que funciona sobre TCP y utiliza el puerto 1935, RTMPT (RTMP Tunneled) que es encapsulado dentro de peticiones HTTP, para atravesar cortafuegos y RTMPS (RTMP Secure) que funciona como RTMP pero sobre una conexión HTTPS segura; este último fue el que se usó para esta investigación.

El servicio de Videochat consta de tres actores o roles: Un Administrador, el cual se encarga de gestionar el servicio, de permitir o restringir el acceso y de controlar el intercambio de información de audio, vídeo o texto entre los diferentes actores. Un Locutor, quien presenta desde un computador a todos los actores una temática usando una cámara web y un micrófono. Un Cliente, el cual recibe el audio y el vídeo. Además de las anteriores funcionalidades, todos los actores pueden intercambiar información entre sí, a través de texto.

Para mantener la confidencialidad de los datos y de las transmisiones por la red, todas las comunicaciones entre el servidor y los diferentes actores, se realizan utilizando comunicaciones seguras mediante el protocolo HTTPS.

El Administrador inicia el servicio ingresando a través de una URL en el navegador, el cual le presenta un formulario de validación con usuario y contraseña. Al acceder al servicio se muestra una página donde se pueden crear un Locutor y se tiene acceso a las URL de los otros dos actores, ver Fig. 1. Una vez el Administrador habilite el Videochat, el Locutor y el Cliente pueden acceder al servicio mediante las URL correspondientes. El Locutor debe ingresar su usuario y su contraseña, mientras que los Clientes ingresan simplemente digitando un nombre en el formulario que le presenta el servicio. La emisión de audio y vídeo la inicia el Locutor, con lo cual el servicio entra en operación.

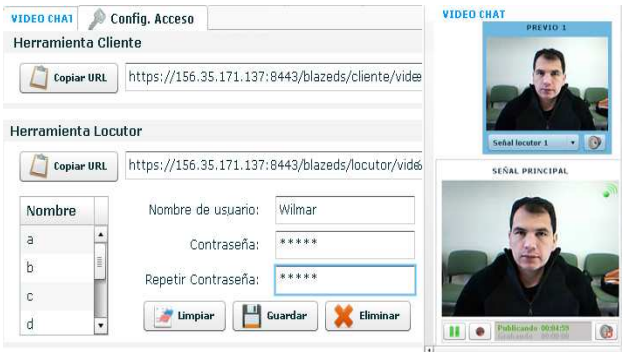


Fig. 1. Formulario y señal de vídeo sobre el Administrador

IV. Caracterización del Tráfico

En este apartado se describe el proceso de construcción del modelo de tráfico del servicio de Videochat. Se inicia con las capturas de las trazas de tráfico, a partir de las cuales se identifican las fases y el comportamiento que presenta el servicio. Se continúa con el modelado donde se describe el proceso de obtención de las FDP y finalmente la validación, donde se muestra que el tráfico generado estadísticamente, describe al tráfico real del servicio.

A. Captura del Tráfico

Se inicia con la puesta en funcionamiento de la red real del Laboratorio DMMS, donde el servicio de Videochat se ejecuta sin la red simulada. En este entorno de red se realizan las capturas del tráfico mediante el analizador de protocolos tcpdump, el cual se ejecuta en cada uno de los equipos de los actores. El proceso general del intercambio de información entre los diferentes actores se puede dividir en la fase de conexión, la fase de transmisión de datos y la fase de desconexión, ver Fig. 2.

La fase de conexión se presenta únicamente al inicio del servicio. El Administrador a través de un navegador ingresa su nombre de usuario y contraseña, posteriormente crea un Locutor y habilita el servicio de Videochat. El Locutor y los Clientes acceden al servicio una vez está habilitado. En esta fase, el tráfico que se genera entre cada actor y el servidor es del orden de los 6 a los 10 paquetes, con un tamaño promedio de 600 bytes por paquete, siendo este tráfico demasiado pequeño para estudiar el comportamiento de la red.

La fase de Transmisión de Datos la inicia el Locutor, quien transmite la información de audio y vídeo hacia el Administrador, siendo este último quien habilita el envío de dicha información a los Clientes.

Haciendo un análisis temporal del comportamiento de los datos, se observa que el envío de paquetes se da en instantes de tiempo, esto se debe a que el tráfico tiene un comportamiento a ráfagas. En la Fig. 3, se verifica este efecto, donde se muestra el número de paquetes en función del tiempo. Las ráfagas de tráfico son analizadas, obteniendo los componentes de cada una de ellas, como son: el tamaño de los paquetes, el tiempo entre los paquetes, el número de paquetes que conforman una ráfaga y el tiempo entre ráfagas.

La fase de Desconexión corresponde al intercambio de paquetes por medio de los cuales un actor abandona el servicio. Cualquiera de los actores puede dar por terminada su participación en la sesión de Videochat, para esto basta con desconectarse del servicio. Este proceso genera un tráfico menor en número de paquetes y en tamaño, respecto a la fase de conexión.

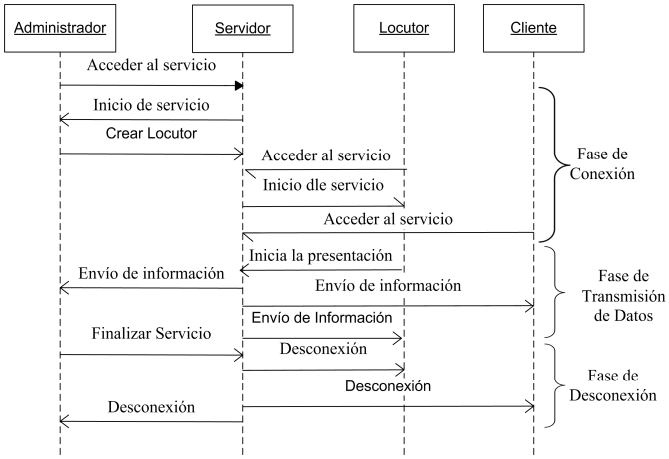


Fig. 2. Fases del servicio de Videochat

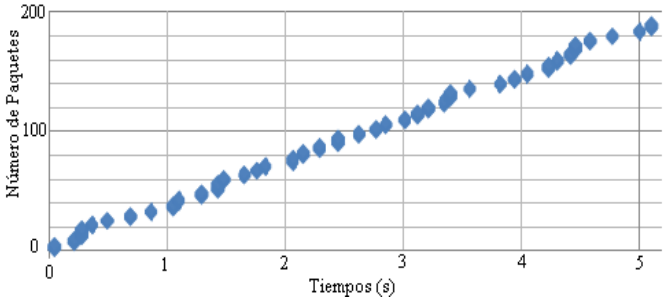


Fig. 3. Comportamiento a ráfagas del tráfico

B. Modelado de tráfico

En este apartado se describe el proceso de modelado para la fase de Transmisión de Datos, debido a su mayor complejidad y mayor volumen de tráfico generado. El proceso de modelado es análogo para las otras dos fases.

El propósito es llevar el tráfico real hasta el entorno de simulación. A partir de las trazas de tráfico capturadas para la fase de Transmisión de Datos, se observa que el Locutor inicia la transmisión de la información. El Servidor de Videochat la recibe y la reenvía hacia el Administrador y hacia el Cliente; además el Servidor reenvía tráfico hacia el propio Locutor. Todo el tráfico es capturado por el analizador de protocolos en forma de series a través del tiempo. La información tabulada en este formato, no es de utilidad, por lo

que se debe obtener una conducta basada en variabilidad descrita a través de un comportamiento probabilístico.

Para esta investigación se han seguido dos enfoques, como son, el uso de las funciones de densidad de probabilidad (FDP) y el uso de scripts directamente. La caracterización de cada componente de tráfico de la fase de Transmisión de Datos, se realiza mediante una FDP encargada de describir el comportamiento de cada componente y su validación se realiza a través de la prueba de bondad de ajuste de Kolmogorov-Smirnov (K-S).

Los scripts corresponden a archivos de texto que entregan los datos de cada componente, de las fases de conexión y desconexión, a la herramienta de simulación directamente.

Así, en la fase de conexión del Administrador donde se envían 6 paquetes, resulta más práctico programar la herramienta mediante dos scripts, uno que entregue 6 valores para el tamaño de los paquetes y otro que entregue 5 valores para el tiempo entre estos, no siendo práctico encontrar una FDP, por ser tan pequeño el número de paquetes que se intercambian en esta fase. En contraste para la fase de Transmisión de Datos, donde el orden del número de paquetes es de las decenas o centenas por segundo, se encuentran las FDP que describen cada componente de esta fase, ver Tabla 1. Los parámetros de la Tabla 1, son obtenidos del análisis estadístico, del tráfico capturado en un escenario real que involucra todos los actores.

Existen diferentes FDP teóricas que caracterizan el comportamiento de un componente, por ejemplo, el tamaño de los paquetes del Servidor al Cliente puede ser descrito mediante una distribución Normal o una de Weibull (ver Tabla 1). Como criterio de selección de una distribución sobre otra se usa el menor valor estadístico Dn global K-S, el cual calcula la distancia máxima entre la distribución acumulada de la muestra Fn(x) y la función de distribución que se ajusta al comportamiento de la muestra F(x) [15].

C. Validación del tráfico

Una vez modelado el tráfico para cada una de las fases del servicio, a través de las FDP y/o los scripts, los componentes de las fases deben ser programados como los parámetros de entrada a la herramienta de simulación, obteniendo un escenario basado en tráfico explícito.

Para validar este modelo, se crean escenarios básicos para cada pareja de actores de la Tabla 1, por ejemplo un escenario básico entre el locutor y el servidor, consta de un computador al cual se le asigna el nombre de Locutor, un router o un switch y un servidor. En el Locutor, es necesario cargar cada uno de los scripts de las fases de conexión y desconexión, además de los parámetros definidos por las FDP en la fase de Transmisión de Datos (columnas 2 y 5 de la Tabla 1). El Servidor de Videochat es necesario configurarlo con las capacidades para enviar y recibir datos hacia y desde el Locutor. De manera análoga se realiza la construcción de un escenario de validación para cada pareja de actores, como fase previa a la construcción de un escenario que involucre todos los actores sobre una misma infraestructura de red, equivalente al escenario real.

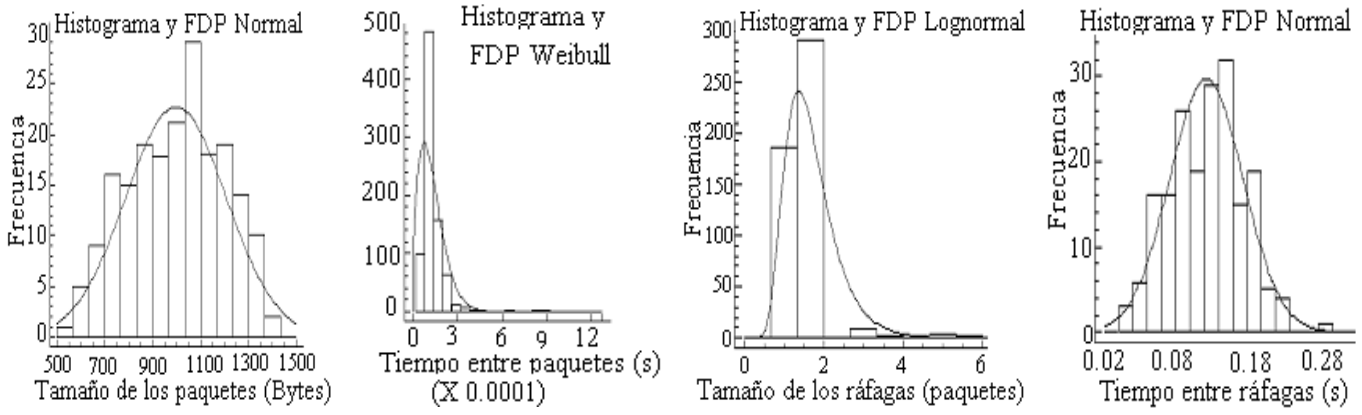
Para asegurar la validez estadística de las simulaciones de eventos discretos, es necesario ejecutar varias instancias de la simulación. Como regla general, de dos a tres docenas de ejecuciones de una simulación [16], una forma de reducir los tiempos empleados en este proceso es ejecutar simultáneamente las simulaciones mediante la distribución en varias máquinas.

Una vez configurado un escenario básico, se capturan las trazas de tráfico generadas en la herramienta de simulación y se analizan mediante las pruebas de bondad y ajuste K-S, que permiten obtener las FDP de cada parámetro, Ver Fig.4. Finalmente se compara estadísticamente el tráfico real generado por el servicio de Videochat, con el generado por la herramienta de simulación. Un escenario construido de esta forma corresponde a un modelo de simulación por eventos discretos basado en tráfico explicito.

Se debe tener en cuenta que los parámetros de modelado presentados son independientes del tiempo de duración de cada una de las fases del servicio y de los escenarios de evaluación, los cuales son construidos en la etapa de emulación mediante redes virtuales incrementando el número de usuarios e inyectando diferentes tipos de tráfico. En la Fig. 4. Se presentan los histogramas del tráfico real con su distribución teórica que mejor se ajusta a su forma. De los 16 casos que contiene la Tabla 1. Se han escogido 4 (uno por cada parámetro) diferentes, a manera de ilustración.

Componentes	**Locutor al Servidor**	**Servidor al Administrador**	**Servidor al Cliente**	**Servidor al Locutor**
Tamaño de los paquetes (Bytes)	Normal μ = 997.23, σ = 209.83 Dn = 0.04	Weibull k = 7.35, λ = 1383.5 Dn = 0.049	Weibull k = 8.53, λ = 1398.8 Dn = 0.048	Normal μ = 1297.15, σ = 104.77 Dn = 0.04
Tiempo entre los Paquetes (s)	Normal μ = 0.015, σ = 0.0094 Dn = 0.049	Weibull k = 1.59, λ = 0.00014 Dn = 0.047	Normal μ = 0.00014 σ = 0.000077 Dn = 0.046	Lognormal μ = 0.00015, σ = 0.00005 Dn = 0.046
Tamaño de las ráfagas (en paquetes)	Lognormal μ = 6.3, σ = 1.8 Dn = 0.048	Lognormal μ = 1.98, σ = 0.81 Dn = 0.048	Lognormal μ = 1.68, σ = 0.65 Dn = 0.47	Lognormal μ = 4.54, σ = 1.82 Dn = 0.47
Tiempo entre ráfagas (s)	Lognormal μ = 0.05, σ = 0.04 Dn = 0.04	Lognormal μ = 0.031, σ =1.1 Dn = 0.049	Lognormal μ = 0.23, σ = 39.14 Dn = 0.046	Normal μ = 0.12, σ = 0.051 0.044

Tabla 1. FDP para el tráfico de la fase de Transmisión de Datos



Locutor al servidor | Servidor al Administrador | Servidor al Cliente | Servidor al Locutor

Fig. 4. Histogramas del tráfico real con su distribución teórica

En la Fig. 5 se presenta la función de distribución acumulada (CDF) para la validación del tamaño de los paquetes del Locutor al Servidor de los valores simulados con los valores reales.

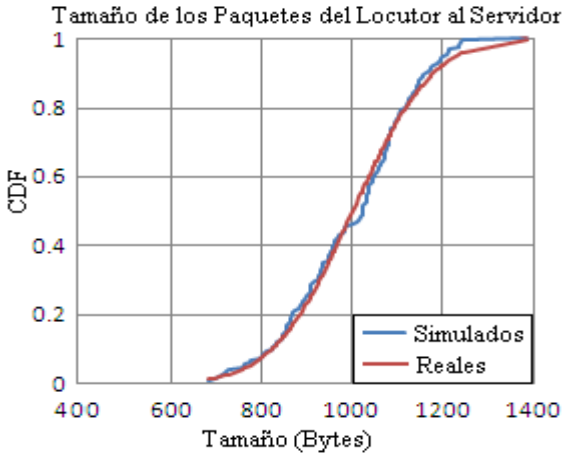


Fig. 5. CDF para validación valores reales y simulados

En las Fig. 4. y Fig. 5. se observa como las distribuciones teóricas siguen el comportamiento de las distribuciones obtenidas de los datos reales.

V. Entorno De Experimentación

A continuación se describe la maqueta del Laboratorio de DMMS de la Universidad de Oviedo, utilizada para el desarrollo de esta investigación. Se muestra el plano de la red real y el plano de la red simulada y el proceso de interacción entre los dos planos de la maqueta. En la Fig. 6, Se observa el diagrama del entorno de experimentación usado, el cual consta de dos planos denominados Red Real del Laboratorio DMMS o plano uno y Red Simulada o plano dos.

A. Plano de la Red Real

El plano uno de la Fig. 6, contiene los computadores reales del servicio asociados a cada actor del Videochat, a los cuales se les ha dado los siguientes nombres: Servidor de Videochat_R, Administrador_R, Cliente_R1 y Cliente_R2 y Locutor_R. Este último debe estar dotado con una cámara web y un micrófono, que le permita a un ponente emitir su conferencia.

El servidor de Videochat_R que aloja al FMS, corresponde a un equipo con 750 megabytes de memoria, un procesador a 3 GHz y Sistema operativo Ubuntu Release 8.04. Los equipos para cada uno de los actores corresponden a computadores dotados de un navegador y un cliente flash.

El equipo Dell PoweEdge r410 que se observa en el plano uno de la Fig. 6, corresponde a un servidor de rack, llamado Eleanor, con 16 gigabytes de memoria y un procesador a 2.1 GHz. Sistema operativo Red Hat Enterprise Linux Server Release 5.4, dotado de 6 tarjetas de red Fast Ethernet. Su función es hacer de router entre las diferentes interfaces de red y alojar la herramienta de simulación, con la cual se construyó la red simulada del plano dos de la maqueta de la Fig. 6.

B. Plano de la red Simulada

La red simulada está conformada por la red troncal la cual consta de 4 routers con enlaces que soportan 100 Mbps; computadores que representan los dispositivos del servicio; el servidor de Videochat y módulos SITL.

Los módulos SITL son usados para la conexión con el hardware externo, conecta una red o dispositivo de la red simulada con un dispositivo de red real a través de una conexión IP. Por lo tanto, este modulo permite entregar paquetes simulados a dispositivos reales y paquetes reales a los dispositivos simulados. Para esta investigación dicho módulo toma el tráfico generado por los dispositivos de computo reales del plano uno y los hace circular por la red simulada o plano dos. Por ejemplo, si desde el Locutor_R se envía información hacia el Administrador_R (ver la línea punteada en la Fig. 6.), el proceso es el siguiente: los paquetes con la información de audio, vídeo o texto, circularán desde el Locutor_R hasta la interfaz de red Eth2 del servidor Eleanor en el plano uno, dichos paquetes son tomados por el modulo SITL_5 del plano dos y enviados al router 4 (R4), en donde, de acuerdo a la tabla de direccionamiento de red, los paquetes son enviados hasta el router 3 (R3), para luego ser entregados a la interfaz SITL_4, la cual entrega el tráfico a la interfaz de red Eth3 del servidor Eleanor. Finalmente dicha información llega hasta su destino el Administrador_R; de esta forma el tráfico real fluye a través de la red simulada antes de alcanzar su destino.

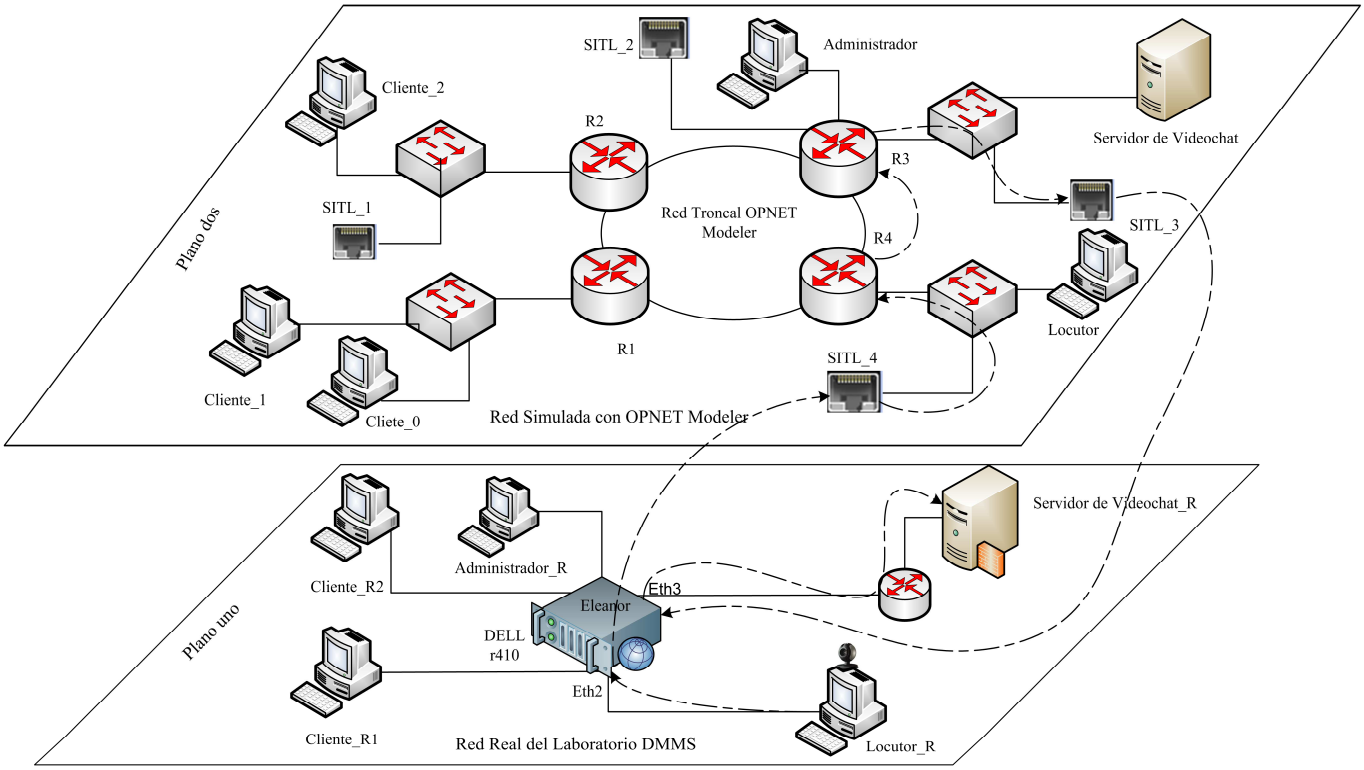


Fig. 6. Maqueta del Laboratorio DMMS

VI. Escenarios De Evaluación

La red virtual se construye partiendo de un escenario básico, el cual está conformado por un router con dos interfaces SITL, las cuales permiten el acceso al servidor de Videochat y al Administrador. Una vez operativo este escenario, se van adicionando elementos, hasta alcanzar la topología deseada del plano dos de la Fig. 6, obteniendo así un escenario virtual con tráfico real. Obsérvese que para esta sección no se ha tenido en cuenta el tráfico explícito caracterizado. Así, lo que se obtiene es una topología de red virtual sobre la cual se inyecta tráfico real.

Teniendo en cuenta el direccionamiento de la red, se crea el escenario de emulación completo, haciendo que los dos planos de la maqueta del Laboratorio interactúen entre sí. Además, para este escenario se adicionan cargas de tráfico sobre los enlaces, mediante tráfico background. Así, este escenario de emulación contiene una red virtual, tráfico explícito cargado sobre elementos simulados, tráfico real que parte y llega a dispositivos reales a través de la red virtual y tráfico analítico.

Los escenarios de evaluación están conformados como se muestra en la Tabla 2. El tráfico del escenario 1 está conformado únicamente por el tráfico explicito, es decir solo contiene el plano dos de la Fig. 6. El escenario 2 está conformado únicamente por el tráfico real del plano uno que fluye sobre el plano dos de la Fig. 6. Los demás escenarios incluyen tráfico real más tráfico explicito. Los escenarios del 7 al 10 corresponden al escenario 5 más tráfico mixto de 75 Mbps, el cual se carga a nivel de la capa IP entre el servidor de Videochat y un Cliente.

El tráfico mixto no tiene en cuenta cada detalle como por ejemplo la función de distribución que sigue, sino que está conformado por tráfico sintético el cual calcula una cantidad de tráfico basado en una descripción de extremo a extremo y tráfico paquete a paquete donde se especifican, un flujo que conecta un nodo fuente y destino, un período de tiempo de simulación dividido en intervalos de tiempo y una tasa en paquetes por segundo para cada intervalo.

Escenarios	Tráfico explicito (plano dos de la maqueta)		
1	SI		
2	NO		
3	SI, Unión de los escenarios 1 y 2		
4	Escenario 3 más 20 usuarios simulados		
5	Escenario 3 más 40 usuarios simulados		
6	Escenario 5, más tráfico background en el backbone		
Con tráfico mixto		Paquete a Paquete	Sintético
7	75 Mbps	5%	95%
8		10%	90%
9		20%	80%
10		30%	70%

Tabla 2. Escenarios de evaluación

VII. Resultados

A. *Análisis y Monitorización de Prestaciones*

La emulación exige un gran procesamiento de datos en tiempo real. Así, una de las limitaciones con las que se encuentra la comunidad científica para realizar emulación de servicios, son las altas capacidades de cómputo exigidas. A continuación se presentan los datos para el servidor que soporta la herramienta de simulación.

En la Fig. 7, se muestra una grafica del número de eventos generados versus el porcentaje de uso de la CPU. Los números sobre la línea corresponden al escenario de pruebas.

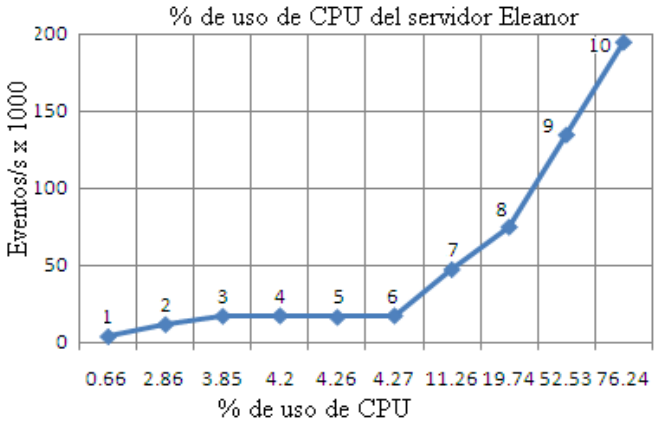


Fig. 7. Eventos generados versus uso de CPU

El tráfico real (escenario dos) genera un mayor número de eventos que su modelo equivalente con tráfico explícito (escenario uno). El número de eventos por segundo y el consumo de la CPU, generados en el escenario tres corresponden aproximadamente a la suma de los valores correspondientes (eventos y CPU) de los escenarios uno y dos. Los escenarios cuatro, cinco y seis, prácticamente generan el mismo número de eventos y el consumo de la CPU se incrementa en unas pocas centésimas, a pesar de que la complejidad de los escenarios ha aumentado en número de Clientes simulados. Incluso el escenario seis contiene trafico backgraound con el 75% de ocupación sobre el backbone de la red simulada.

En los escenarios 7 a 10 se observa que al incrementar el tráfico mixto paquete a paquete, se incrementa el número de eventos por segundo, lo que a su vez incrementa el uso de la CPU. Para valores superiores a 30% de tráfico paquete a paquete, cuyo tamaño sea el de la MTU de Ethernet, el uso de la CPU es superior al 85 %. Con estos valores el proceso de emulación no se ejecuta en tiempo real.

Respecto al uso de memoria el proceso de emulación consume valores entre 0.37% para el escenario uno, hasta 0.65% para el escenario diez, los cuales no son valores críticos. De acuerdo al anterior análisis el consumo de CPU del equipo es un factor a tener en cuenta en los procesos de emulación.

B. Análisis del comportamiento del modelo

Una vez validado el modelo, su propósito es el de estudiar el tráfico que inyecta en la red el servicio en funcionamiento, para poder medir el impacto que tendría su implantación sobre una red de comunicaciones.

El estudio del tráfico se ha realizado bajo diferentes escenarios. No se observa ningún parámetro que afecte el servicio de manera directa. Una razón es que la red virtual está sobredimensionada para este tipo de servicios, con bajo requerimiento de ancho de banda.

No se observa jitter ni perdida de paquetes sobre el servicio real. El retardo de los paquetes reales, extremo a extremo sobre la red virtual es del orden de 1 ms. Los tiempos de conversión de tráfico real a simulado son del orden de los 11 µs. Los tiempos de conversión de tráfico simulado a real, son del orden de los 7 µs. Así este proceso de conversión que se da en los módulos SITL no tiene un impacto importante sobre los retardos que puedan sufrir los paquetes.

Respecto al tráfico mixto se tienen las siguientes medidas: el valor medio de la latencia entre la creación de los paquetes y la recepción en el nodo destino es de 2 ms. El jitter es de 0,43 ms para el escenario 7, hasta alcanzar los 0.7 ms para el escenario 10. Estos valores son lo suficientemente pequeños, por lo que no tienen un impacto sobre el tráfico mixto. Y tampoco se observa que dicho tráfico afecte el tráfico real del servicio de Videochat.

Al agregar tráfico mixto sobre el modelo se genera un incremento del throughput desde el servidor de Videochat hacia los clientes. En la Fig. 8 (a) se observa el tráfico generado para los escenarios sin tráfico mixto. En la Fig. 8 (b) se observa el incremento de magnitud del throughput al incluir el tráfico mixto sobre los escenarios respectivos.

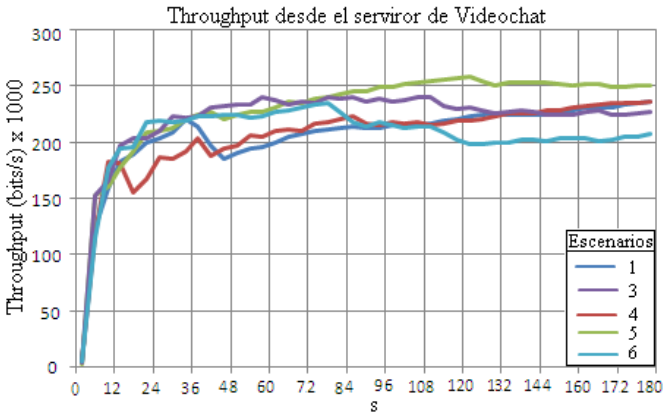


Fig. 8 (a). Escenarios sin tráfico mixto

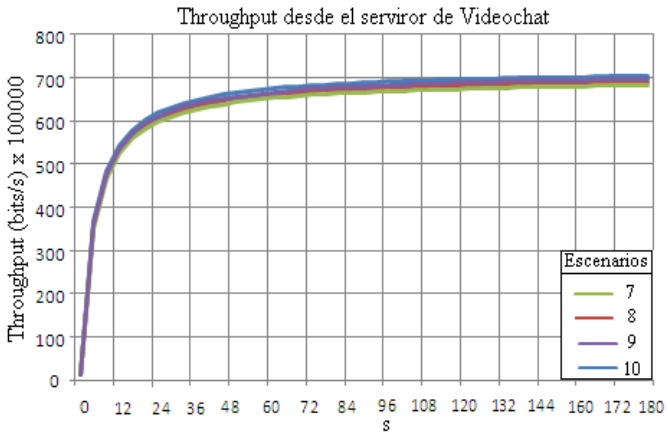


Fig. 8 (b). Escenarios con tráfico mixto

Se realizo un escenario 11, en el cual se congestionó al 100% el anillo de la red virtual mediante tráfico background. Se observó que el servicio de Videochat continuaba en funcionamiento a través de la red virtual, aunque su calidad percibida se iba degradando. Al analizar las trazas de tráfico para este experimento se observa que el throughput sobre los módulos SITL y para el tráfico explicito intercambiado entre los actores simulados se reduce en 10 veces respecto a los otros experimentos. En la Fig. 9 se observa este comportamiento.

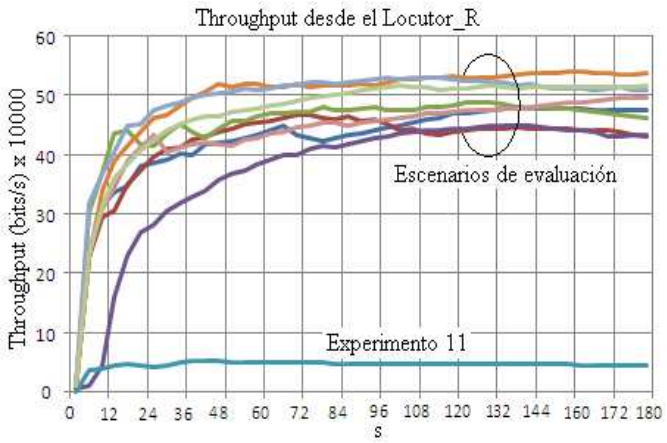


Fig.9. Escenario 11 versus escenarios de evaluación

VIII. CONCLUSIONES Y TRABAJOS FUTUROS

Este artículo provee una descripción del proceso de modelado de tráfico para un servicio de Videochat, el cual presenta como característica un comportamiento a ráfagas. El modelo se construyó con base en tráfico real generado en el Laboratorio DMMS de la Universidad de Oviedo.

Unificando los modos de actuación, de los tráficos explícito, real y background de forma concurrente, se obtiene un escenario completo capaz de generar un modelo de tráfico para un servicio de Videochat. El proceso puede ser extendido a otro tipo de redes y servicios.

Los entornos de emulación donde se trabaja con tráfico real soportado sobre una red virtual, permiten crear escenarios más completos y complejos, sin requerir de toda una infraestructura de red.

De acuerdo al análisis de los resultados el tráfico mixto construido a partir del tráfico real y del tráfico explícito, como generador de carga permite simular el comportamiento de un número elevado de Clientes orientado hacia servicios masivos. Este es un parámetro que se debe monitorear, puesto que el exceso de tráfico mixto paquete a paquete genera un elevado consumo de CPU de la máquina donde se lleva a cabo la emulación, generando como consecuencia la ejecución del modelo en tiempos mayores al tiempo real, invalidando los resultados del modelo.

El tráfico background no genera un aumento en el número de eventos, por lo que las capacidades de procesamiento se mantienen. Pero es necesario crear escenarios de referencia que permitan analizar el comportamiento del modelo y así obtener resultados correctos.

Para que los resultados no pierdan validez, se debe tener en cuenta tanto la correcta generación de tráfico explícito, como las capacidades de procesamiento de la maquina que soporta la emulación.

Como trabajos futuros se plantea investigar la emulación combinando servicios de alta calidad con servicios de menor calidad o livianos y analizando cual es el efecto de los unos sobre los otros. Así, es posible realizar estudios en entornos con múltiples servicios que permitan el análisis de diferentes parámetros de red. Además se puede extender el proceso de modelado a otro tipo de redes como la redes de cable o las redes inalámbricas. Utilizar la emulación para el estudio de las métricas de calidad del servicio y de la experiencia.

AGRADECIMIENTOS

Este trabajo ha sido realizado gracias a la utilización del simulador OPNET Modeler, disponible a través del programa de universidades “Teaching and Research with OPNET”, al operador TeleCable de Asturias SAU, quien financió el desarrollo del servicio de Videochat, el proyecto SOLITE financiado por CYTED, al proyecto ST-CAV de Colombia y al Programa de Promoción de la Investigación (UNOV-11-MA-03) de la Universidad de Oviedo.

REFERENCIAS

[1] Qingling Yue y Yi Jiang, “Research on Construction of Virtual Community in Academic Library,” *Services Science, Management and Engineering, 2009. SSME '09. IITA International Conference on*, 2009, págs. 200-203.

[2] P. Yin, A, Crisminisi, J. Winn, I. Essa “Bilayer Segmentation of Webcam Videos Using Tree_Based Classifiers,” IEEE Transactions on pattern analysis and machine intelligence, 2001, págs 30-42, Vol. 33. No. 1

[3] L. Peterson, T. Anderson, D. Culler, and T. Roscoe. A blueprint for introducing disruptive technology into the Internet. In Proceedings of the 1st Workshop on Hot Topics in Networking (HotNets-I), October 2002.

[4] M. Grossglauser. 10 papers on network models. SIGCOMM Comput. Commun. Rev., 36(5):63–65, 2006.

[5] K. Pawlikowski, Do not trust all simulation studies of telecommunication networks, in: Information Networking, 2003: págs. 899–908.

[6] J. Liu, “A Primer for Real-Time Simulation of Large-Scale Networks,” *Simulation Symposium, 2008. ANSS 2008. 41st Annual*, 2008, págs. 85-94.

[7] M. Puzar y T. Plagemann, “NEMAN: a network emulator for mobile ad-hoc networks,” *Telecommunications, 2005. ConTEL 2005. Proceedings of the 8th International Conference on*, 2005, págs. 155-161.

[8] “DUO: MICA: A Minimalistic, Component-Based Approach to Realization of Network Simulators and Emulators.” 2007.

[9] A. Alvarez, R. Orea, S. Cabrero, X.G. Pañeda, R. García, y D. Melendi, “Limitations of network emulation with single-machine and distributed ns-3,” *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, ICST, Brussels, Belgium, Belgium: ICST, 2010, págs. 67:1–67:9.

[10] Hyon-Young Choi, Sung-Gi Min y Youn-Hee Han y Jungsoo Park, , y , Hyoungjun Kim, «Implementation and Evaluation of Proxy Mobile IPv6 in NS-3 Network Simulator», in *Ubiquitous Information Technologies and Applications (CUTE), 2010 Proceedings of the 5th International Conference on*, 2010, págs. 1-6.

[11] W. P. Furlong y R. Guha, «OFDMA Extension of NS-3 WiMAX Module», in *Computer Modeling and Simulation (EMS), 2010 Fourth UKSim European Symposium on*, 2010, págs. 426-431.

[12] S. Demers, P. Gopalakrishnan, y L. Kant, «A Generic Solution to Software-in-the-Loop», in *Military Communications Conference, 2007. MILCOM 2007. IEEE*, 2007, págs. 1-6.

[13] K. McNeill, J. Deal, T. Haynes, y D. Bradford, with R. Martinez, Wenji Wu, «Hardware and software-in-the-loop techniques using the OPNET modeling tool for JTRS developmental testing», in *Military Communications Conference, 2003. MILCOM 2003. IEEE*, 2003, vol. 1, págs. 469-474 Vol.1

[14] A. Álvarez, R. García, X. Pañeda, D. Melendi yR. Orea, « In pursuit of massive service emulation: a methodology for testbed building», IEEE Communication Magazine, Network Testing series (aceptado)

[15] W.Y. Campo, J.L. Arciniegas, R. García, y D. Melendi, “Análisis de Tráfico para un Servicio de Vídeo bajo Demanda sobre Recles HFC usando el Protocolo RTMP,” *Información Tecnológica*, vol. 21, 2010, págs. 37–48.

[16] OPNET Modeler, “Verifying Statistical Validity of Discrete Event Simulations,” 2008.

Similitud difusa basada en nombres y relaciones taxonómicas entre conceptos para el *matching* de ontologías.

Susel Fernández, Juan R. Velasco, Miguel A. López-Carmona.
Departamento de Automática
Universidad de Alcalá
Edificio Politécnico. Ctra. N-II Km. 31,600. 28871 Alcalá de Henares. Madrid. España.
susel@aut.uah.es, juanra@aut.uah.es, miguellop@aut.uah.es

***Resumen*- Este artículo está enfocado a ofrecer mecanismos de ayuda a los expertos en la primera fase del *matching* de ontologías, utilizando técnicas de lógica difusa para determinar el grado de similitud entre conceptos. Para cada pareja de conceptos de ontologías diferentes se calculan dos medidas de similitud: la similitud semántica utilizando el coeficiente de Jaccard sobre muestras de documentos relevantes tomados de la Web, y la similitud lingüística. En este trabajo hemos definido un sistema difuso de tres capas, la primera para calcular la similitud lingüística, la segunda para calcular la similitud básica a partir de los valores de similitud semántica y lingüística, y la tercera para calcular la similitud avanzada teniendo en cuenta la influencia de las similitudes entre los hijos, padres y hermanos de los conceptos. El sistema fue validado utilizando ontologías públicas que modelan aspectos del mundo real.**

***Palabras Clave- matching* de ontologías, lógica difusa, similitud, Web semántica.**

I. INTRODUCCIÓN

La investigación sobre los servicios de Web semántica promete una mayor interoperabilidad entre agentes software y servicios Web, permitiendo descubrimiento de servicios automatizado basado en contenidos e interacción, utilizando ontologías compartidas publicadas en la Web semántica. Sin embargo, los servicios producidos y descritos por diferentes desarrolladores pueden utilizar diferentes o quizás parcialmente solapados conjuntos de ontologías. El *matching*, que no es más que el proceso de encontrar correspondencias entre los conceptos de dos ontologías debe ser expresado por algunas reglas que expliquen esta correspondencia. Obviamente, encontrar tales correspondencias puede constituir un valioso paso en la solución del problema de la traducción de ontologías.

La incertidumbre está relacionada con todos los aspectos de la Web semántica, puesto que una descripción de un concepto desconocido o un objeto puede ser incierta, y a menudo se da el caso de que un concepto definido en una ontología empareja solo parcialmente con uno o más conceptos en otra.

Existen algunos trabajos encaminados al *matching* de ontologías, que han realizado aportaciones interesantes. Noy y Musen han desarrollado las herramientas SMART [1], PROMPT [2], y PROMPTDIFF [3], que usan emparejamiento de similitud lingüística entre los conceptos para iniciar un proceso de fusión o alineación y, entonces, utilizan las estructuras ontológicas para proporcionar un conjunto de heurísticos para identificar más coincidencias entre las ontologías.

McGuinness y colegas desarrollaron Chimaera [4], donde el ingeniero es el encargado de tomar las decisiones que afectan el proceso de fusión. Chimaera analiza las ontologías que se van a fusionar, y si encuentra coincidencias lingüísticas, la fusión se realiza automáticamente, de lo contrario el usuario es avisado de la adopción de nuevas medidas. Es similar a PROMPT en que ambos están incrustados en los entornos de edición de la ontología, pero difieren en las sugerencias que hacen a sus usuarios con respecto a los pasos de fusión.

Otras investigaciones han seguido enfoques probabilísticos. Por ejemplo, Doan y colegas desarrollaron el sistema GLUE [5], que emplea técnicas de aprendizaje automático para encontrar correspondencias entre ontologías. Para cada concepto en una ontología, GLUE encuentra la mayoría de los conceptos similares en la otra utilizando la distribución de probabilidad conjunta de los conceptos para calcular la similitud. Este enfoque no tiene en cuenta por completo la incertidumbre en el *matching*, ya que se ignoran los conceptos similares en un menor grado.

Ron Pang y colegas desarrollaron un marco probabilístico de *matching* automático de ontologías basado en Redes Bayesianas (BNs) [6]. Las ontologías son primero traducidas a BNs, y a continuación el *matching* de conceptos se realiza por razonamiento evidencial entre las dos BNs. Las probabilidades necesarias tanto en la traducción como en el *matching* se obtienen mediante el uso de programas de clasificación de textos asociando distintos conceptos con documentos de texto relevantes recuperados de la Web. Este enfoque solo tiene en cuenta la probabilidad de aparición de los conceptos en la Web, por lo que falla en el caso en que dos conceptos muy similares no tengan el mismo índice de popularidad.

Nuestra propuesta se centra en la primera etapa del *matching* de ontologías, hallando similitudes entre conceptos por nombre. Primeramente se calcula una similitud semántica entre los conceptos utilizando el coeficiente de Jaccard, a partir de documentos relevantes recuperados de la Web. Luego se calculan la similitud lingüística y la similitud total utilizando un sistema basado en reglas difusas de tres capas. La organización del trabajo es la siguiente: en la sección 2 se

ISBN: 978-84-694-5948-5

describe el uso del coeficiente de Jaccard para calcular la similitud semántica y el método utilizado para obtener la similitud lingüística. En la sección 3 se muestra el sistema basado en reglas difusas y en la sección 4 se reflejan los resultados de aplicar nuestro método sobre dos ontologías del mundo real, y una ontología para un sistema de gestión de ideas. Finalmente en la sección 5 se presentan las conclusiones y las líneas de trabajo futuras.

II. MEDIDAS DE SIMILITUD ENTRE CONCEPTOS

En esta sección se definen las medidas de similitud empleadas. Estas son: la similitud semántica y la similitud lingüística. Finalmente detallamos como influyen las similitudes de los hijos, padres y hermanos de los conceptos en las taxonomías sobre la similitud total.

A. Similitud Semántica. Coeficiente de Jaccard

El coeficiente de Jaccard [7] es uno de los índices binarios de similitud más conocidos y utilizados. Se define como el tamaño de la intersección dividido entre el tamaño de la unión entre dos conjuntos de datos y su valor está entre 0 y 1. Para dos observaciones *i* y *j*, el coeficiente de similitud de Jaccard se calcula a través de la siguiente fórmula:

$$S_{ij} = \frac{a}{a+b+c} \tag{1}$$

Donde *a* es el numero de veces que ambas observaciones tienen valor 1, *b* es el numero de veces que la observación *i* tiene valor 1 y la observación *j* tiene valor 0, *c* es el numero de veces que la observación *i* tiene valor 0 y la observación *j* tiene valor 1.

En nuestro trabajo los valores que intervienen en esta fórmula fueron obtenidos a través de búsquedas sucesivas de documentos relevantes a cada concepto en la Web. De manera similar a la desarrollada en [6], para asegurar que la búsqueda solo devolviera documentos relevantes al concepto, la consulta de búsqueda se forma mediante la concatenación de todos los conceptos que se encuentran en el camino desde la raíz hasta el nodo actual en la taxonomía. Los ejemplares que contengan al concepto A se obtendrán mediante la búsqueda de A y todos sus ancestros en la taxonomía, mientras que los ejemplares que no contengan a A serán aquellos en los que estén presentes todos sus ancestros en la taxonomía, y no esté presente el concepto A. Para cada par de conceptos A y B, pertenecientes a las ontologías origen y destino respectivamente se contarán (a) los documentos en los que aparezcan los dos conceptos, (b) los documentos en los que aparezca A y no aparezca B, y (c) los documentos en los que aparezca B y no aparezca A. Una vez obtenidos estos valores, para cada par de conceptos de las ontologías origen y destino respectivamente, se calcula su similitud semántica por la fórmula (1).

B. Similitud Lingüística

La similitud lingüística constituye el indicador más fuerte del parecido entre dos conceptos, debido a que por lo general los desarrolladores de ontologías dentro de un mismo dominio emplean términos relacionados lingüísticamente para expresar conceptos equivalentes.

En este trabajo se calculan dos tipos de similitud según la relación lingüística de los conceptos: una basada en la sinonimia, y otra basada en las equivalencias lemáticas de los conceptos.

Dados los conceptos A y B de dos ontologías diferentes, el primer paso de nuestra propuesta para calcular la similitud lingüística consiste en obtener las listas de sinónimos y de palabras derivadas de cada uno de ellos, auxiliándonos de la herramienta WordNet [8]. Utilizando las listas de sinónimos y las listas de palabras derivadas de ambos conceptos calculamos la similitud lingüística por sinonimia y la similitud lingüística por equivalencias lemáticas. El algoritmo para calcular la similitud es el siguiente:

Primeramente a cada una de las palabras de las listas obtenidas se le aplica el algoritmo de stemming de Porter [9], que no es más que el proceso de eliminar los finales morfológicos de las palabras en inglés, para obtener las raíces morfológicas de las mismas. Si llamamos L_A y L_B a las listas de raíces morfológicas de cada concepto obtenidas en este paso, podemos calcular la similitud lingüística entre los conceptos como sigue:

$$S = \min\left[\frac{c_A}{T_A}, \frac{c_B}{T_B}\right] \tag{2}$$

Donde c_A es la cantidad de palabras de la lista L_A que están presentes en la lista L_B, T_A es la cantidad total de palabras de la lista L_A, c_B es la cantidad de palabras de la lista L_B que están presentes en la lista L_A, y T_B es la cantidad total de palabras de la lista L_B.

C. Similitud EXTRA

Para mejorar el cálculo de la similitud final entre los conceptos hemos tenido en cuenta la influencia de las similitudes de los hermanos, de los padres y de los hijos de los conceptos en las taxonomías. Si dos conceptos A y B de taxonomías diferentes tienen cierto parecido, y sus hermanos, hijos, y padres también se parecen, es muy probable que A y B sean el mismo concepto, o conceptos muy similares.

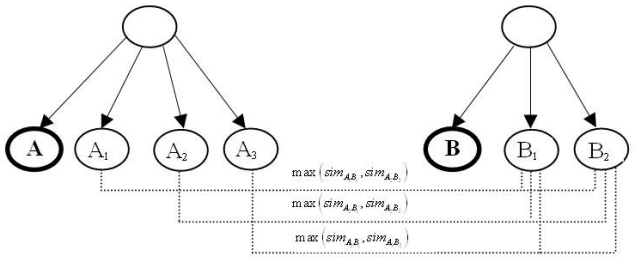


Fig. 1. Similitud *Extra* entre dos conceptos basada en la similitud de sus hermanos.

Dados dos conceptos A y B de dos taxonomías diferentes, llamamos similitud *Extra* al grado de similitud que aportan los hermanos, hijos y padres, y se calcula a través de la fórmula 3. La figura 1 muestra un ejemplo de cómo se calcularía la similitud *Extra* de los hermanos. Si denominamos A_i al *i-esimo* hermano del concepto A, y B_j al *j-ésimo* hermano del concepto B, la similitud *Extra* sería el promedio de los máximos de las similitudes entre todos los hermanos de A y todos los hermanos de B. Fórmula 3.

$$Sim_{extra} = \frac{1}{n}\sum_{i=1}^{n} \max\left\{sim(A_i, B_j)\right\}_{j=1}^{m} \tag{3}$$

III. SISTEMA DIFUSO DE TRES CAPAS

En este trabajo hemos utilizado el entorno de desarrollo XFuzzy 3.0 [10], que integra varias herramientas que cubren las diferentes etapas del diseño de sistemas difusos. Hemos definido un sistema basado en reglas difusas de tres capas: la primera para el cálculo de la similitud lingüística, la segunda para el cálculo de la similitud básica, y la última para el cálculo de la similitud avanzada.

A. Similitud Lingüística

La primera capa del sistema basado en reglas difusas ha sido diseñada para calcular la similitud lingüística entre los conceptos. Se han definido dos variables de entrada, para representar las similitudes por sinonimia y por derivación respectivamente y una variable de salida, que representa la similitud lingüística total, calculada por el sistema difuso. Las tres variables tienen asociado el siguiente conjunto de términos lingüísticos: D= {*Baja*, *Regular*, *Media*, *Alta*, *MuyAlta*}. Se definieron funciones triangulares de pertenencia con conjuntos difusos equiespaciados debido a la distribución de los valores de entrada. Estas funciones se muestran gráficamente en la figura 2.

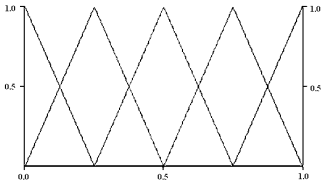


Fig. 2. Funciones triangulares de pertenencia de las variables de la primera capa del sistema difuso. Cálculo de la similitud lingüística

B. Similitud Básica

La segunda capa del sistema basado en reglas difusas ha sido diseñada para el cálculo de la similitud básica. Se han definido dos variables de entrada y una de salida y cada una de ellas tiene asociado un conjunto de términos lingüísticos cuya semántica ha sido representada por medio de funciones triangulares de pertenencia. Las variables del sistema son las siguientes:

Similitud_Jaccard: Es una variable de entrada y representa el valor de la similitud semántica, calculado a través de coeficiente de Jaccard. Tiene asociado el siguiente conjunto de términos lingüísticos: D_{jacc}= {*Baja*, *Regular*, *Media*, *Alta*, *MuyAlta*}.

Para definir las funciones de pertenencia primeramente fue necesario dividir los datos de la similitud semántica en conjuntos difusos, por lo que utilizamos los cuartiles del conjunto de valores de similitud para acotar los triángulos de pertenencia de la siguiente manera: *Baja*: (-0.00224168, 0, 0.00224168), *Regular*: (0, 0.00224168, 0.03031929), *Media*: (0.00224168, 0.03031929, 0.10712543), *Alta*: (0.03031929, 0.10712543, 1), *MuyAlta*: (0.10712543, 1, 1.10712543). Las funciones triangulares de pertenencia para esta variable se muestran gráficamente en la figura 3a).

Similitud_Ling: Es una variable de entrada que representa el valor de la similitud lingüística. Tiene asociado el siguiente conjunto de términos lingüísticos: D_{ling}= {*Baja*, *Regular*, *Media*, *Alta*, *MuyAlta*}. Debido a la distribución de los valores de la similitud lingüística, se definieron conjuntos difusos equiespaciados. Las funciones triangulares de pertenencia para esta variable se muestran gráficamente en la figura 3b).

Similitud_Básica: Es la variable de salida y representa el valor de la similitud inicial, obtenida por el sistema difuso. Tiene asociado el siguiente conjunto de términos lingüísticos: $D_{Similitud}$= {*MuyBaja*, *Baja*, *MedioBaja*, *Regular*, *MedioAlta*, *Alta*, *MuyAlta*}. Las funciones triangulares de pertenencia para esta variable se muestran gráficamente en la figura 3c).

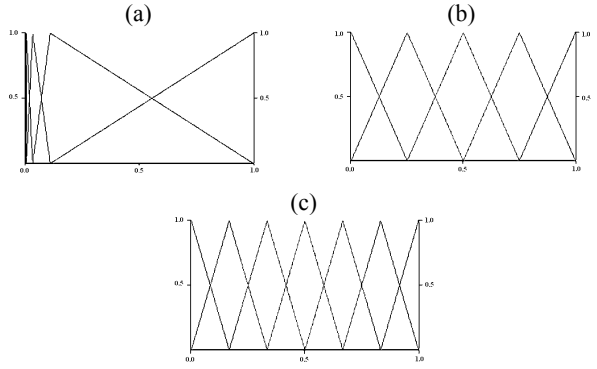


Fig. 3. Funciones triangulares de pertenencia para: a) Similitud_Jaccard b) Similitud_Ling c) Similitud_Básica

C. Similitud Avanzada

Una vez calculada la similitud básica entre los conceptos, hemos tenido en cuenta otros factores para mejorar los valores de similitud, como son las relaciones taxonómicas con otros conceptos.

En la última capa del sistema difuso se han definido cuatro variables de entrada y una de salida. A cada una de ellas hemos asociado un conjunto de términos lingüísticos cuya semántica ha sido representada por medio de funciones triangulares de pertenencia. Estas variables son las siguientes:

Sim_Básica: Variable de entrada que representa el valor de la similitud calculado a partir de las similitudes semántica y lingüística.

Extra_hermanos: Variable de entrada que representa el valor *Extra* que aportan las similitudes entre los hermanos de ambos conceptos.

Extra_padres: Variable de entrada que representa el valor *Extra* que aportan las similitudes entre los padres de ambos conceptos.

Exta_hijos: Variable de entrada que representa el valor *Extra* que aportan las similitudes entre los hijos de ambos conceptos.

Sim_avanzada: Es la variable de salida y representa el valor de la similitud final.

Las 4 variables tienen asociado el siguiente conjunto de términos lingüísticos: *D*= {*MuyBaja*, *Baja*, *MedioBaja*, *Regular*, *MedioAlta*, *Alta*, *MuyAlta*}. Las funciones de pertenencia se muestran gráficamente en la figura 3c).

D. Generación evolutiva de las bases de reglas

Los algoritmos genéticos (AG) pueden ofrecer métodos de búsqueda potentes e independientes del dominio para una gran variedad de tareas de aprendizaje [11]. Uno de los enfoques en los cuáles se han utilizado los AG para procesos de aprendizaje ha sido el de Pittsburgh, donde cada cromosoma codifica la base completa de reglas. El proceso de cruce sirve para obtener una nueva combinación de reglas,

y la mutación proporciona nuevas reglas. En este trabajo utilizamos el algoritmo THRIFT [12], uno de los pioneros en el aprendizaje de reglas difusas siguiendo el enfoque de Pittsburgh. Los cromosomas se obtienen recorriendo una tabla de decisión completa por filas y codificando cada conjunto difuso de salida como un entero. El conjunto de datos utilizado contiene información de 40 ontologías mapeadas por expertos, y ha sido particionado utilizando el método de validación cruzada en 10 partes. Los parámetros de entrada del algoritmo fueron los siguientes:

Tamaño de la población. Número de cromosomas para el algoritmo genético: 61

Número de evaluaciones. Número máximo de llamadas a la función de evaluación para detener la búsqueda: 1000

Probabilidad de cruce. Probabilidad para realizar el cruce de un par de cromosomas: 0.6

Probabilidad de mutación. Probabilidad de realizar la mutación de un cromosoma: 0.1

Las bases de reglas para calcular la similitud lingüística y la similitud básica se muestran en las tablas 1 y 2 respectivamente. Por razones de espacio no mostramos la base de reglas para obtener la similitud avanzada.

Sim_Sin	**Sim_Deriv**				
	Baja	**Regular**	**Media**	**Alta**	**Muy alta**
Baja	Baja	Regular	Regular	Media	Media
Regular	Regular	Regular	Regular	Media	Media
Media	Media	Media	Media	Alta	Alta
Alta	Alta	Alta	Alta	Muy alta	Muy alta
Muy alta	Muy alta	Muy alta	Muy alta	Muy alta	Muy alta

Tabla 1. Base de reglas para la similitud lingüística

Jaccard	**Ling**				
	Baja	**Regular**	**Media**	**Alta**	**Muy alta**
Baja	Muy baja	Baja	Medio Baja	Regular	Medio alta
Regular	Baja	Medio baja	Regular	Medio alta	Alta
Media	Baja	Medio baja	Regular	Medio alta	Alta
Alta	Medio baja	Regular	Medio alta	Alta	Muy Alta
Muy Alta	Medio baja	Medio alta	Alta	Alta	Muy alta

Tabla 2. Base de reglas para la similitud básica

IV. EXPERIMENTOS

Hemos realizado varios experimentos con ontologías del mundo real extraídas de la Web, y con taxonomías generadas a partir de una ontología de un sistema de gestión de ideas.

A. Experimento 1.

El primer experimento se ha realizado utilizando dos ontologías del mundo real extraídas de la Web. Estas son, la ontología ACM [13], y la ontología DMOZ [14]. Hemos seleccionado el tópico de *Inteligencia Artificial*, al igual que en [6], haciendo una poda de conceptos en ambas ontologías de manera que hubiera cierto grado de solapamiento entre ambas. La figura 4 muestra los conceptos seleccionados en ambas ontologías, y la relación que a priori debería existir entre ellos.

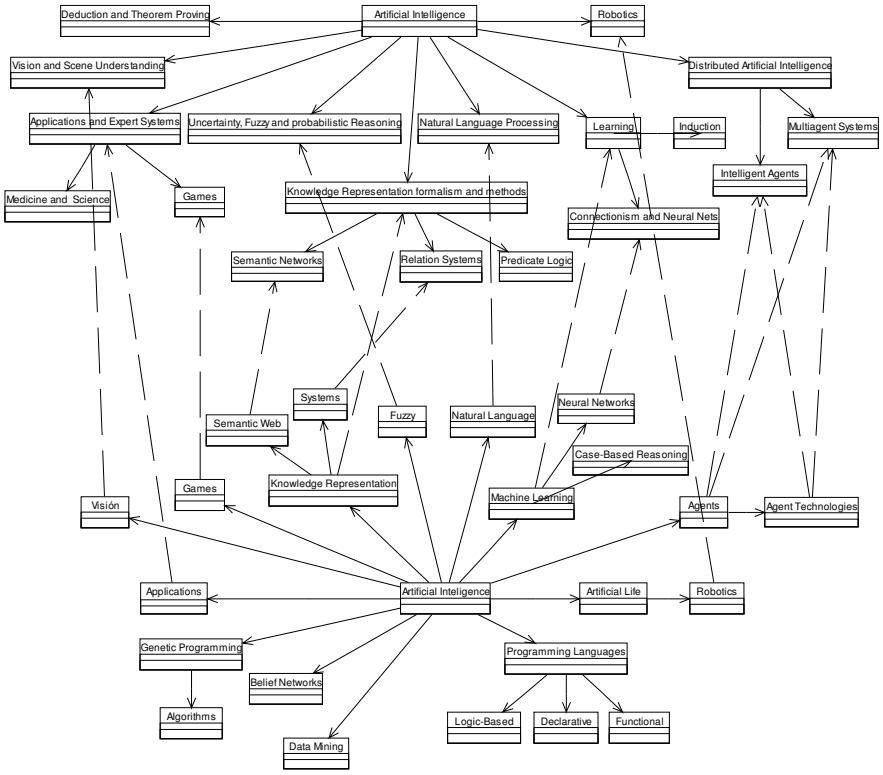


Fig. 4. Conceptos relacionados en las ontologías ACM y DMOZ

En el trabajo de [6], utilizando fragmentos similares de las ontologías ACM y DMOZ, plantean que según su método para los conceptos "*connectionism and neural nets"* y "*Neural Networks"* se obtuvo un índice de similitud de 0.61, debido a que solo se tuvo en cuenta la probabilidad de aparición de ambos conceptos en la Web, y el término *connectionism* es poco popular. En la tabla 3, que muestra los conceptos para los cuáles hemos obtenido los mayores índices de similitud, podemos observar que con nuestro método para estos dos conceptos se ha mejorado el grado de similitud, debido fundamentalmente a su relación lingüística. También se puede apreciar que sólo se obtuvo una pareja de conceptos con similitud más alta de lo esperado.

Conceptos ACM	Conceptos DMOZ	Similitud
Robotics	**Robotics**	**0.99**
Games	**Games**	**0.99**
Multiagent Systems	**Agent Technologies**	**0.91**
Multiagent Systems	Systems	0.88
Natural Language Processing	**Natural Language**	**0.88**
Learning	**Machine learning**	**0.87**
Knowledge Representation Formalisms and Methods	**Knowledge Representation**	**0.85**
Intelligent Agents	**Agents**	**0.75**
Vision and Scene Understanding	**Vision**	**0.74**
Uncertainty, fuzzy, and probabilistic reasoning	**Fuzzy**	**0.74**
Semantic Networks	**Semantic Web**	**0.73**
Connectionism and neural nets	**Neural Networks**	**0.72**
Multiagent Systems	**Agents**	**0.71**

Tabla 3. Conceptos con similitud más alta. Ontologías ACM y DMOZ

B. Experimento 2.

El segundo experimento está enfocado a localizar similitudes entre ideas en un sistema de gestión de ideas dentro de una red social de consultoría, teniendo en cuenta la necesidad de gestionar ideas similares en este entorno colaborativo en el que participan diferentes empresas, clientes, proveedores, colaboradores donde cada uno puede poseer su propia plataforma interna de gestión de ideas. Para este experimento hemos seleccionado 3 pequeños conjuntos de ideas sin clasificar y hemos generado taxonomías automáticamente utilizando el algoritmo Lingo[16] de clustering.

La figura 5 muestra los fragmentos de las taxonomías utilizadas. Las parejas de conceptos cuya similitud debería ser mayor están enlazados con líneas discontinuas. En la parte izquierda se muestran las taxonomías A y B, seleccionadas por tener muchas ideas similares distribuidas por todas las ramas. En la tabla 4, que muestra las ideas para las cuáles se han obtenido los valores mayores de similitud entre las taxonomías A y B, se puede observar que para las ideas equivalentes, los valores de similitud obtenidos son muy altos, encontrándose solo tres parejas de ideas con valores de similitud más altos de lo esperado.

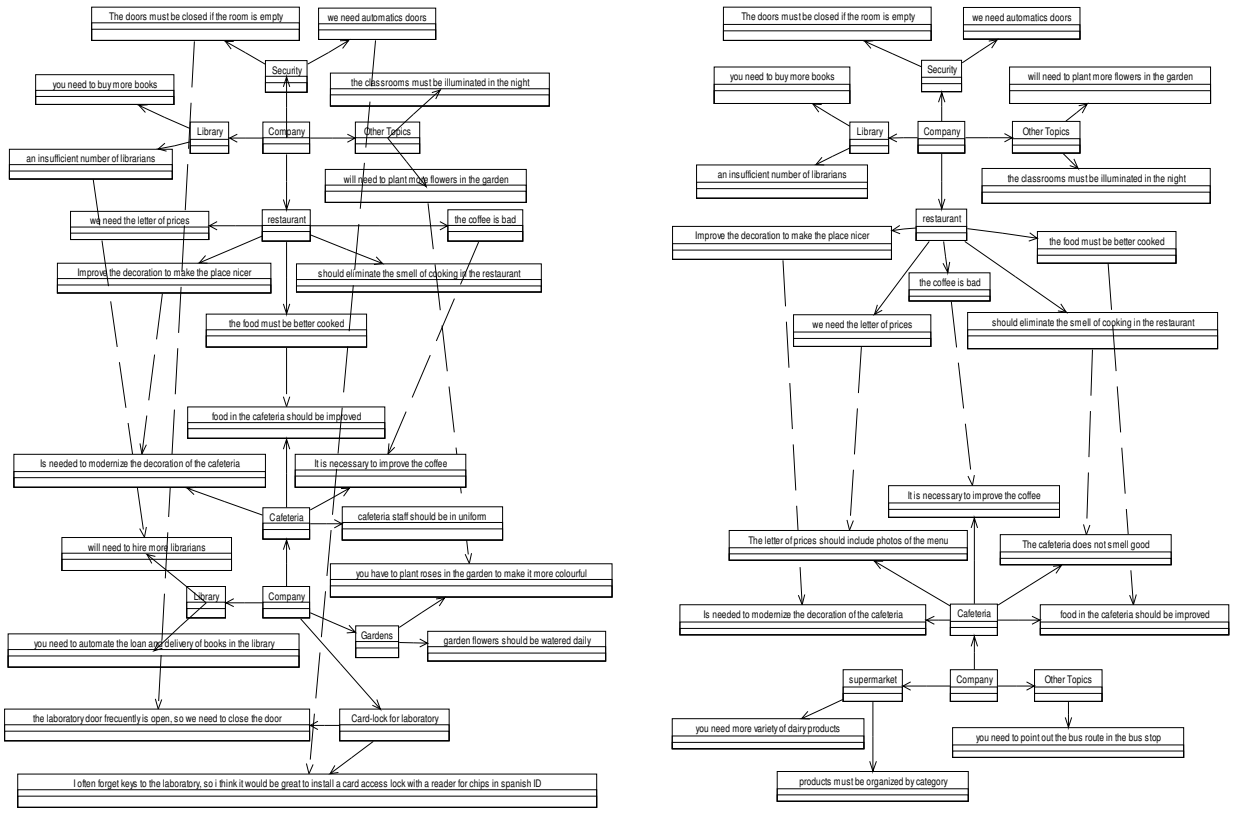


Fig. 5. Taxonomías y relaciones a priori entre las ideas.(Izquierda) Taxonomías A y B. (Derecha) Taxonomías A y C

Taxonomía A de Ideas	Taxonomía B de Ideas	Similitud
the doors must be closed if the room is empty	**the laboratory door is frequently opened, so we need to close the door**	**0.95**
the food must be better cooked	**food in the cafeteria should be improved**	**0.91**
the coffee is bad	**it's necessary to improve the coffee**	**0.89**
improve the decoration to make the place nicer	**is needed to modernize the decoration of the cafeteria**	**0.85**
you need to buy more books	you need to automate the loan and delivery of books in the library	0.85
will need to plant more flowers in the garden	**you have to plant roses in the garden to make it more colorful**	**0.82**
improve the decoration to make the place nicer	you have to plant roses in the garden to make it more colorful	0.79
will need to plant more flowers in the garden	garden flowers should be watered daily	0.74
we need automatics doors	**I often forget keys to the laboratory, so i think it would be great to install a card access lock with a reader for chips in spanish ID**	**0.73**
an insufficient number of librarians	**will need to hire more librarians**	**0.71**

Tabla 4. Ideas con similitud más alta en las taxonomías A y B

Luego comparamos las taxonomías A y C, que se muestran en la parte derecha de la figura 5, seleccionadas de manera que solo fueran equivalentes las ideas relacionadas con *restaurante* y *cafetería*. En la tabla 5 se puede apreciar que los valores de similitud para las ideas equivalentes son muy elevados respecto al resto, lo que constituye un indicador de la efectividad del método.

Taxonomy A of Ideas	Taxonomy C of Ideas	Similarity
we need the letter of prices	**the letter of prices should include photos of the menu**	**0.93**
the food must be better cooked	**food in the cafeteria should be improved**	**0.91**
the coffee is bad	**it's necessary to improve the coffee**	**0.89**
should eliminate the smell of cooking in the restaurant	**the cafeteria does not smell good**	**0.89**
improve the decoration to make the place nicer	**is needed to modernize the decoration of the cafeteria**	**0.85**
the food must be better cooked	the cafeteria does not smell good	0.28
the letter of prices should include photos of the menu	should eliminate the smell of cooking in the restaurant	0.25

Tabla 5. Ideas con similitud más alta en las taxonomías A y C

V. CONCLUSIONES Y LINEAS DE TRABAJO FUTURO

En este artículo se describe nuestro trabajo encaminado a ofrecer un método de ayuda a los expertos en la fase inicial del *matching* de ontologías. La propuesta utiliza técnicas de lógica difusa para obtener similitudes entre conceptos de diferentes ontologías. El sistema ha sido probado en varias taxonomías de conceptos del mundo real obteniendo valores de similitud satisfactorios, que mejoran las propuestas existentes basadas métodos semánticos o lingüísticos exclusivamente, debido a la combinación de ambos enfoques. Como limitación podemos decir que el tiempo de ejecución del sistema se incrementa considerablemente al procesar ontologías completas debido a la gran cantidad de información, por lo que como trabajo futuro nos hemos propuesto mejorar la escalabilidad de la aplicación. También pretendemos extender la técnica empleada, para proponer un modelo de integración que permita el *matching* de conceptos por atributos, por valores y por tipos además de por el nombre, y tener en cuenta en el modelo de integración no solo las relaciones taxonómicas entre conceptos, sino también relaciones inter conceptuales en dominios reales.

AGRADECIMIENTOS

Este trabajo es parte del proyecto RESULTA, financiado por el ministerio de industria, turismo y comercio de España, TSI-020301-2009-31.

REFERENCIAS

[1] Noy, N. F, Musen, M. A.: SMART: Automated Support for Ontology Merging and Alignment. In: 12th Workshop on Knowledge Acquisition, Modelling and Management (KAW'99), Banff, Canada (October 1999).
[2] Noy, N. F, Musen, M. A.: The PROMPT suite: Interactive tools for ontology merging and mapping. International Journal of Human-Computer Studies, 59(6), pp. 983–1024 (2003.)
[3] Noy, N. F, Musen, M. A.: PROMPTDIFF: A Fixed-Point Algorithm for Comparing Ontology Versions. In: 18th National Conference on Artificial Intelligence (AAAI'02), Edmonton, Alberta, Canada (August 2002).
[4] McGuinness, D., Fikes, R., Rice, J., Wilder, S.: An Environment for Merging and Testing Large Ontologies. In: 17th International Conference on Principles of Knowledge Representation and Reasoning (KR-2000), Colorado, USA (April 2000).
[5] Doan A., Madhavan, J., Domingos, P., Halevy, A.: Ontology Matching: A Machine Learning Approach. Handbook on Ontologies in Information Systems. In: S. Staab and R. Studer (eds.), Invited paper. Pp. 397-- 416. Springer-Velag, (2004).
[6] Pan, R., Ding, Z., Yu, Y., Peng, Y.: A Bayesian Network Approach to Ontology Mapping. The Semantic Web – ISWC 2005, Vol. 3729/2005, pp. 563—577. Springer Berlin / Heidelberg (October 2005)
[7] Rijsbergen, V., C. J.: Information Retrieval. Butterworths. Second Edition, London (1979).
[8] Fellbaum, C. WordNet: *An Electronic Lexical Database*. MIT Press. 3.0, Cambridge, MA. 1998.
[9] Porter Stemming algorithm, http://tartarus.org/~martin/PorterStemmer/
[10] XFuzzy 3.0 framework, http://www.imse.cnm.es/Xfuzzy/Xfuzzy_3.0/tools/xfuzzy_sp.html.
[11] Cordón, O., Herrera, F., Hoffman, F., Magdalena, L.: Genetic Fuzzy Systems. Evolutionary Tuning and Learning of Fuzzy Knowledge Bases. World Scientific, Singapore (2001).
[12] Thrift, P. Fuzzy Logic Synthesis with genetic algorithms. In: Proceedings 4th International Confer-ence on Genetic Algorithms, Morgan Kaufmann, 1991, 509-513.
[13] ACM Topic, http://www.acm.org/about/class/1998/
[14] DMOZ hierarchie, http://www.dmoz.org/

[15] Osinski, Stanislaw, Jerzy Stefanowski, and Dawid Weiss. "Lingo: Search Results Clustering Algorithm Based on Singular Value Decomposition." In: *Proceedings of the International IIS: IIPWMÂ´04 Conference, 2004*, Zakopane, Poland. 359-68.
[16] Barwise, J., Seligman, J.: Information Flow: The Logic of Distributed Systems. Cambridge University Press, 1997.
[17] Calvanese, D., Giacomo, G., Lenzerini, M.: Ontology of integration and integration of ontologies. In: Description Logic Workshop (DL 2001), pp 10–19, 2001.
[18] Fernández-Breis, J., Martínez-Béjar, R.: A cooperative framework for integrating ontologies. In: International Journal of Human-Computer Studies, 56: pp. 665–720, 2002.
[19] Gruber, T., Olsen, G.: An ontology for engineering mathematics. In: J. Doyle,P. Torasso, E. Sandewall, (eds). Fourth International Conference on Principles of Knowledge Representation and Reasoning, pp. 258–269, San Mateo, CA, USA (1994).
[20] Grüninger, M.: Ontologies for translation: Notes for refugees from Babel. EIL Technical Report, Enterprise Integration Laboratory (EIL), University of Toronto, Canada (November 1997).
[21] Jannink, J., Pichai, S., Verheijen, D., Wiederhold, G.: Encapsulation and Composition of Ontologies. In: AAAI'98 Workshop on Information Integration, Madison, WI, USA (July 1998).
[22] Kalfoglou, Y., Schorlemmer, M.: Ontology mapping: the state of the art. The Knowledge Engineering Review, 18(1) pp. 1–31, (2003).
[23] Mitra, P., Noy, N. F., Jaiswal, A. R. 2004. OMEN: A Probabilistic Ontology Mapping Tool. In: Workshop on Meaning Coordination and Negotiation at the Third International Conference on the Semantic Web (ISWC-2004). Hiroshima, Japan (2004).

Sesión 2.A
Criptografía y seguridad en redes

Sistema de Tarificación Automático con Anonimato Revocable: Evaluación de Rendimiento

Andreu Pere Isern-Deyà[1], Arnau Vives-Guasch[2], M. Magdalena Payeras-Capellà[1],
Macià Mut-Puigserver[1], Jordi Castellà-Roca[2]
[1] Dpt. Ciències Matemàtiques i Informàtica, UIB, Palma de Mallorca, Spain
[2] Dpt. Enginyeria Informàtica i Matemàtiques, UNESCO Chair in Data Privacy, URV, Tarragona, Spain
{andreupere.isern, mpayeras, macia.mut}@uib.es {arnau.vives, jordi.castella}@urv.cat

***Resumen*—Los sistemas de tarificación masiva (AFC, Automatic Fare Collection) calculan la tarifa que deben pagar los usuarios de un servicio dependiendo del tiempo (time-based AFC) o de la distancia recorrida (distance-based AFC). Estos sistemas deben mantener la seguridad de las transacciones y asegurar la privacidad de sus usuarios. En este trabajo se evaluan los requerimientos de seguridad para este tipo de sistemas y se propone un protocolo AFC basado en distancia con anonimato revocable y en donde diferentes sesiones de un mismo usuario no pueden enlazarse. El protocolo se ha implementado sobre Android y se ha evaluado sobre dos smartphones, comparando el sistema basado en distancias con un trabajo previo basado en tiempo. Los resultados indican que los dos protocolos son usables en dispositivos actuales de gama media y en un entorno real.**

***Palabras Clave*—sistema tarificación seguro, anonimato, privacidad, implementación, rendimiento**

I. Introducción

La incorporación de las Tecnologías de la Información (TIC) en los sistemas de tarificación masiva (AFC, Automatic Fare Collection) permite la reducción de los costes operativos, mejora el control de las infraestructuras e incrementa la facilidad de uso por parte de los usuarios. Los sistemas AFC se diseñan para medios de transporte pero también pueden ser aplicados en escenarios donde se requiera de algún tipo de tarificación por uso. Como ejemplos se pueden nombrar los sistemas de peajes, los aparcamientos tarificados, los transportes públicos o el uso de instalaciones públicas o privadas como campos de deportes. Estos sistemas pueden calcular las tarifas a abonar basándose en el tiempo de uso del servicio (*time-based AFC*) o en función de la distancia recorrida (*distance-based AFC*).

Los sistemas AFC requieren protocolos eficientes y seguros, que sean capaces de controlar las entradas y salidas de los usuarios en el sistema. Por tanto, se requiere que los usuarios puedan mantener el anonimato para evitar la creación de perfiles de uso que puedan comprometer su privacidad. No obstante, también se desea que el anonimato pueda revocarse en caso de usuarios con mala fe, desmotivando así los intentos de fraude. Hay que tener también muy en cuenta la eficiencia del sistema, ya que se trata de escenarios con un gran número de usuarios que interactuan con la infraestructura.

A. Contribución

El trabajo presenta un esquema de gestión para sistemas AFC basado en distancias, producto de la mejora de un sistema AFC basado en tiempo ya presentado por los mismos autores [1]. El esquema asegura la privacidad y mantiene el anonimato para los usuarios honestos aunque este puede ser revocado ante comportamientos fraudulentos. Para conseguir este objetivo, se usa un esquema de firma de grupo [2]. Además, los usuarios no necesitan obtener nuevas credenciales para cada sesión, hecho que mejora la usabilidad del sistema. Finalmente, se han implementado los dos sistemas AFC sobre la plataforma Android, se han realizado pruebas de rendimiento y se han analizado los resultados obtenidos. La evaluación de estos resultados indica que ambos protocolos son usables para un entorno real.

El artículo se organiza de la siguiente forma: en §III se describe el uso del esquema de firma de grupo; en §IV se especifican los requerimientos necesarios que tiene que cumplir un sistema para esta finalidad; en §V se describe el protocolo en detalle; seguidamente en §VI se realiza un análisis de seguridad del protocolo; por su parte, en §VII se muestra la implementación y el análisis de rendimiento del protocolo; y finalmente en §VIII se exponen las conclusiones y el trabajo futuro.

II. Estado del Arte

En la literatura existen diversas propuestas de sistemas AFC que consideran el anonimato revocable de sus usuarios [3], [4], [5], [6], [7], [8]. Las propiedades más importantes de estos sistemas se listan en la Tabla I: el nivel de anonimato de cada propuesta, la trazabilidad de diferentes ejecuciones realizadas por el mismo usuario y los dispositivos para los que están diseñados estos sistemas.

Ref.	Anonimato	No Trazabilidad	Dispositivo
[3]	Revocable	No	Móvil
[4]	Revocable	No	Smart-card
[5]	Revocable	Sí	Móvil y Smart-card
[6]	Revocable	No	Smart-card
[7]	Revocable	No	Móvil
[8]	Revocable	No	Móvil

Tabla I
Comparación de las propuestas analizadas.

La tarifa a pagar es calculada independientemente para cada servicio prestado y puede basarse en el tiempo (*time-based AFC*) o en la distancia recorrida (*distance-based AFC*). El uso de instalaciones públicas como piscinas o campos de deportes son ejemplos de servicios basados en tiempo, mientras que los servicios de bus o tren son ejemplos claros de servicios basados en distancia.

ISBN: 978-84-694-5948-5

III. Background

La propuesta usa el esquema de firma de grupo BBS [2] que proporciona la capacidad a los usuarios que pertenecen a un grupo de firmar datos de forma anónima. Un usuario del grupo o externo a él puede verificar que la firma está generada por un usuario perteneciente al mismo grupo pero no puede averiguar de quién se trata. En la definición del protocolo AFC se usan los procedimientos $KeyGen_G$, $Sign_G$, $Verify_G$ y $Open_G$ que pueden ser consultados en [2] y que no se reflejan en el artículo debido a la limitación de espacio. Además de los anteriores, se han definido dos nuevos $SignLinkable_G$, $VerifyLinkable_G$, basado en los anteriores y que se comentan tangencialmente a continuación.

A. Enlazabilidad entre Firmas de Grupo

En el sistema todos los usuarios son anónimos y sus firmas no son enlazables. No obstante, por seguridad y solo en determinados casos, algunas firmas del mismo usuario deben ser enlazables.

Procedimiento $SignLinkable_G$*:* Se define un procedimiento de firma enlazable $SignLinkable_G(gpk, gsk[i], M)$ en el que dada la clave pública de grupo gpk, la clave privada de usuario $gsk[i]$ y el mensaje M, el usuario genera una firma de conocimiento σ de la siguiente forma:

Primer uso: $Sign_G(gpk, gsk[i], M)$ estándar:

- generar un cifrado lineal de A: $(T_1, T_2, T_3) \leftarrow (u^{\alpha}, v^{\beta}, Ah^{\alpha+\beta})$ para $\alpha, \beta \overset{R}{\leftarrow} \mathbb{Z}_p$;
- dado el mensaje M, lo firma y genera $\sigma \leftarrow (T_1, T_2, T_3, c, s_{\alpha}, s_{\beta}, s_x, s_{\delta_1}, s_{\delta_2})$ donde $c \leftarrow H(M, T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5) \in \mathbb{Z}_p$ y $s_{\alpha}, s_{\beta}, s_x, s_{\delta_1}, s_{\delta_2} \in \mathbb{Z}_p$;

Siguientes usos: $SignLinkable_G(gpk, gsk[i], M)$:

- usa el mismo par (α, β) produciendo el mismo cifrado lineal de A que en el primer uso: $(T_1, T_2, T_3) = (u^{\alpha}, v^{\beta}, Ah^{\alpha+\beta})$;
- dado el mensaje M, lo firma y genera $\sigma' \leftarrow (T_1, T_2, T_3, c', s'_{\alpha}, s'_{\beta}, s'_x, s'_{\delta_1}, s'_{\delta_2})$ donde $c' \leftarrow H(M', T_1, T_2, T_3, R'_1, R'_2, R'_3, R'_4, R'_5) \in \mathbb{Z}_p$;

Notar que los valores $s_{\alpha}, s_{\beta}, s_x, s_{\delta_1}, s_{\delta_2}$ deben ser diferentes a $s'_{\alpha}, s'_{\beta}, s'_x, s'_{\delta_1}, s'_{\delta_2}$ para no revelar información.

Procedimiento $VerifyLinkable_G$*:* Se define un nuevo procedimiento $VerifyLinkable_G(\sigma, \sigma')$ para verificar que dos firmas han sido generadas por el mismo usuario. Este algoritmo toma como entrada dos firmas:

$$\sigma = (T_1, T_2, T_3, c, s_{\alpha}, s_{\beta}, s_x, s_{\delta_1}, s_{\delta_2}) \text{ y}$$

$$\sigma' = (T_1{}', T_2{}', T_3{}', c', s'_{\alpha}, s'_{\beta}, s'_x, s'_{\delta_1}, s'_{\delta_2})$$

y devuelve $true$ o $false$ dependiendo de si las firmas son producidas por el mismo firmante:

$$(T_1 \overset{?}{=} T_1{}', T_2 \overset{?}{=} T_2{}', T_3 \overset{?}{=} T_3{}')$$

IV. Requisitos de los Sistemas AFC

A. Requisitos Comunes de Seguridad

Los servicios sujetos a tarificación entregan a los usuarios un recibo o un *eticket* para después ser verificado. Estos servicios deben garantizar los siguientes requisitos de seguridad:

- *Autenticidad*: la generación de un *eticket* debe ser realizada por un emisor autorizado.
- *No repudio*: el emisor no puede denegar la emisión de un ticket.
- *Integridad*: una vez generado, el *eticket* no puede ser modificado.

Además de estos, otros requisitos deben garantizarse:

- *Período de validez*: cualquier *eticket* incluye un período de validez hasta el cual puede usarse. Los *etickets* expirados pueden borrarse de las bases de datos optimizando el espacio usado.
- *Imposibilidad de múltiples usos*: un *eticket* solo puede usarse una vez y dejar de ser válido una vez verificado.
- *Anonimato revocable*: el sistema debe proporcionar anonimato al usuario pero tiene que ser revocable si éste actúa incorrectamente.
- *No trazable y no enlazable*: el proveedor solo puede trazar la entrada del usuario con su correspondiente salida, pero no puede enlazar diversas sesiones del mismo usuario.

B. Requisitos Adicionales para Sistemas Basados en Tiempo y en Distancias

Dependiendo del servicio al que se le aplica el sistema automático de tarifas, éste puede basarse en el tiempo de servicio (por ejemplo aparcamientos) o en la distancia recorrida (por ejemplo peajes).

En los sistemas basados en tiempo, es necesario incluir en cada *eticket* el instante temporal de emisión para entonces calcular la tarifa a pagar por el servicio.

En cambio, en los sistemas basados en distancias cada *eticket* de entrada incluye el identificador de la estación origen y el sentido de la marcha para evitar ataques confabulados. Entonces la tarifa a pagar es proporcional a la distancia recorrida entre la entrada y la salida del sistema.

V. Protocolo AFC Basado en Distancias

En esta sección se describe el sistema AFC basado en distancia que proporciona anonimato a los usuarios gracias al uso de firmas de grupo [2]. Primero se describen las diferencias entre esta propuesta y el protocolo anterior [1]. Luego se describen los participantes en el protocolo, los requerimientos de seguridad, la información contenida en los *etickets* de entrada y salida, y finalmente las fases que definen el sistema.

A. Diferencias entre las Dos Propuestas para AFC

En [1] se presentó una primera propuesta de protocolo AFC donde la tarificación se realizaba en función del tiempo consumido en el sistema y donde posteriormente se detectó un posible ataque de confabulación en el que los usuarios podrían intercambiarse los *etickets*. Ahora este problema de seguridad ha sido solucionado con el uso de firmas enlazables (ver III-A) a cambio de introducir un mayor grado de complejidad en la propuesta. Además, esta última propuesta es más general y puede usarse en un mayor número de escenarios.

B. Participantes

Los siguientes actores participan en el sistema propuesto:

- Usuario $\mathcal{U}$: accede al sistema y paga por el servicio recibido en la salida. $\mathcal{U}$ ejecuta sus operaciones con la ayuda de su dispositivo móvil.
- Proveedor de servicio ($\mathcal{P}_\mathcal{S}$ estación origen, $\mathcal{P}_\mathcal{D}$ estación destino): punto de control de los *eticket* usados por $\mathcal{U}$. La tarifa a pagar por $\mathcal{U}$ la calcula $\mathcal{P}_\mathcal{D}$ de acuerdo a la distancia recorrida.
- TTP de grupo $\mathcal{M}_\mathcal{G}$: maneja las claves de grupo, la lista de revocación y puede revocar el anonimato de los usuarios en caso de fraude.
- TTP de pago $\mathcal{M}_\mathcal{C}$: maneja los pagos de $\mathcal{U}$ cuando este sale del sistema.

C. Información del eticket y Notación

La información contenida en el *eticket* de entrada y en el *eticket* de salida se describen en la Tabla II. La notación usada en la descripción del protocolo se describe en la Tabla III.

eticket DE ENTRADA ($\mathsf{t_{in}}^*$)		
NAME	NOTATION	DESCRIPTION
Número de serie	Sn	generado por $\mathcal{P}_\mathcal{S}$
Estación de entrada	Ps	identificador de $\mathcal{P}_\mathcal{S}$
Timestamp de entrada	τ_1	instante de entrada
Tiempo de validez	τ_v	
Compromiso de $\mathcal{U}$	σ^*	firmado por $\mathcal{U}$
Sentido de la marcha	ξ	
Firma digital	$Sign_{\mathcal{P}_\mathcal{S}}(\mathsf{t_{in}})$	firmado por $\mathcal{P}_\mathcal{S}$
eticket DE SALIDA ($\mathsf{t_{out}}^*$)		
Número de serie de $\mathsf{t_{in}}$	$\mathsf{t_{in}.Sn}$	enviado por $\mathcal{U}$
Estación de destino	Pd	
Tarifa a pagar	a	
Timestamp de pago	τ_2	instante de salida
Firma digital	$Sign_{\mathcal{P}_\mathcal{D}}(\mathsf{t_{out}})$	firmado por $\mathcal{P}_\mathcal{D}$

Tabla II
INFORMACIÓN EN LOS *etickets* DE ENTRADA Y SALIDA.

D. Especificación del Protocolo

SETUP: Esta fase solo se ejecuta una vez al principio. $\mathcal{M}_\mathcal{G}$ ejecuta $KeyGen_G(n)$ para generar un grupo de tamaño n, y obtiene $(gpk, gsk[\,], grt[\,], \alpha, p, q)$, donde gpk es la clave pública de grupo, $gsk[i]$ es la clave privada para cada $\mathcal{U}_i$, $grt[\,]$ es la lista de revocación, y (α, p, q) son parámetros públicos, donde α es la base pública de exponenciación y (p, q) son dos números primos tales que $p = 2q + 1$, y cardinales de sus respectivos grupos $\mathbb{Z}_p$ y $\mathbb{Z}_q$. Además, cada proveedor de servicio genera su par de claves y comparte su clave pública.

REGISTRO DE USUARIO: $\mathcal{U}$ se registra en la TTP de grupo $\mathcal{M}_\mathcal{G}$ y recibe un par de claves de grupo $(gpk, gsk[i])$. En este punto, los usuarios están de acuerdo que su identidad pueda ser revelada si actúan deshonestamente o si un juez requiere revocar su identidad.

A continuación, $\mathcal{U}$ se registra anonimamente en la TTP de pago $\mathcal{M}_\mathcal{C}$ con la autorización de $\mathcal{M}_\mathcal{G}$ y demostrando que él es el poseedor del pseudónimo $y_\mathcal{U}$ mediante una prueba de conocimiento nulo Schnorr [9]. Entonces, la privacidad de los usuarios está asegurada aunque el anonimato puede revocarse por acción de $\mathcal{M}_\mathcal{G}$. El protocolo de registro se define así:

generatePseudonym. El usuario $\mathcal{U}$ calcula:

1. genera su psedónimo de pago generando $x_\mathcal{U} \xleftarrow{R} \mathbb{Z}_q$ y calcula $y_\mathcal{U} = \alpha^{x_\mathcal{U}} \pmod p$;
2. envía su identidad $\mathcal{U}_i$, su certificado $Cert\mathcal{U}_i$ y un mensaje firmado con su pseudónimo $Sign_\mathcal{U}(y_\mathcal{U}, \text{'Hello'})$ a $\mathcal{M}_\mathcal{G}$;

keyIssue. $\mathcal{M}_\mathcal{G}$ envía el par de claves de grupo $(gpk, gsk[i])$ juntamente con (α, p, q) y la firma $Sign_{\mathcal{M}_\mathcal{G}}(y_\mathcal{U})$ a $\mathcal{U}$;

startingZKP. $\mathcal{U}$ sigue los pasos:

1. genera $r_0 \xleftarrow{R} \mathbb{Z}_q$ y calcula $s_0 = \alpha^{r_0} \pmod p$;
2. envía $(y_\mathcal{U}, s_0, Sign_{\mathcal{M}_\mathcal{G}}(y_\mathcal{U}))$ a $\mathcal{M}_\mathcal{C}$;

challengeGeneration. $\mathcal{M}_\mathcal{C}$ genera un reto $c_0 \xleftarrow{R} \mathbb{Z}_q$ y lo envía a $\mathcal{U}$;

proofGeneration. $\mathcal{U}$ calcula la prueba ZKP de Schnorr $\omega_0 = r_0 + c_0 \cdot x_\mathcal{U} \pmod q$ y lo envía a $\mathcal{M}_\mathcal{C}$;

verifyPseudonym. $\mathcal{M}_\mathcal{C}$ verifica que $\alpha^{\omega_0} \stackrel{?}{=} s_0 \cdot (y_\mathcal{U})^{c_0}$ y abre una cuenta para $y_\mathcal{U}$.

ENTRADA EN EL SISTEMA: Cuando $\mathcal{U}$ entra correctamente en el sistema, recibe un *eticket* de entrada $\mathsf{t_{in}}^*$. $\mathsf{t_{in}}^*$ será usado posteriormente para autorizar el pago y salir del sistema en la estación destino. El protocolo de entrada se define así:

getService. El usuario $\mathcal{U}$ sigue los pasos:

1. genera $r_1 \xleftarrow{R} \mathbb{Z}_q$ y calcula $s_1 = \alpha^{r_1} \pmod p$;
2. calcula $\delta_\mathcal{U} = PK_{\mathcal{M}_\mathcal{C}}(y_\mathcal{U})$ [1];
3. compone $\sigma = (s_1, \delta_\mathcal{U})$, y lo firma con $gsk[i]$: $\sigma^* = (\sigma, \bar{\sigma} = Sign_G(gpk, gsk[i], \sigma))$;
4. envía σ^* a $\mathcal{P}_\mathcal{S}$;

generateTicket. Proveedor de origen $\mathcal{P}_\mathcal{S}$ sigue los pasos:

1. verifica la firma de σ^* comprobando que el firmante pertenece al grupo: $Verify_G(gpk, \sigma, \bar{\sigma})$;
2. genera un timestamp τ_1;
3. construye el *eticket* de entrada $\mathsf{t_{in}} = (\mathsf{Sn}, \mathsf{Ps}, \tau_1, \tau_v, \sigma^*, \xi)$ y lo firma $\mathsf{t_{in}}^* = (\mathsf{t_{in}}, Sign_{\mathcal{P}_\mathcal{S}}(\mathsf{t_{in}}))$;
4. envía $\mathsf{t_{in}}^*$ a $\mathcal{U}$;

verifyEntrance. $\mathcal{U}$ verifica la firma de $\mathsf{t_{in}}^*$ y su contenido;

SALIDA DEL SISTEMA: Cuando el usuario sale del sistema, envía el *eticket* de entrada $\mathsf{t_{in}}^*$ al proveedor destino $\mathcal{P}_\mathcal{D}$ quién calcula la tarifa a pagar. Si $\mathcal{U}$ se comporta correctamente, recibe el *eticket* de salida $\mathsf{t_{out}}^*$ probando que ha actuado correctamente y que puede salir del sistema. El protocolo de salida se describe a continuación:

previousStep. $\mathcal{P}_\mathcal{D}$ genera $\Phi \xleftarrow{R} \mathbb{Z}_p$ y lo envía a $\mathcal{U}$;

showTicket. $\mathcal{U}$ sigue los pasos:

1. firma Φ como el mismo usuario en la entrada: $\Phi^* = (\Phi, \bar{\Phi} = SignLinkable_G(gpk, gsk[i], \Phi))$;
2. envía $(\mathsf{t_{in}}^*, \Phi^*))$ a $\mathcal{P}_\mathcal{D}$;

verifyTicket. El proveedor destino $\mathcal{P}_\mathcal{D}$ sigue los pasos:

1. verifica la firma de $\mathsf{t_{in}}^*$;
2. verifica la firma de grupo de Φ^*: $(Verify_G(gpk, \Phi, \bar{\Phi}))$ y comprueba que sea el mismo miembro que en la entrada: $VerifyLinkable_G(\mathsf{t_{in}}.\sigma^*, \Phi^*)$;
3. verifica que $\mathsf{t_{in}.Sn}$ no haya sido usado;
4. verifica que τ_v no esté expirado y que el sentido ξ es correcto;
5. genera un timestamp τ_2 ($\tau_1 \leq \tau_2$);
6. calcula la tarifa a pagar dependiendo de la estación de entrada ($\mathsf{t_{in}.Ps}$), la estación de salida (Pd) y los correspondientes timestamps (τ_1, τ_2): $a = f_d(\mathsf{t_{in}.Ps}, \mathsf{Pd}, \mathsf{t_{in}}.\tau_1, \tau_2)$;
7. genera un reto $c_1 \xleftarrow{R} \mathbb{Z}_q$;
8. compone $\beta = (\mathsf{t_{in}}^*, a, c_1, \tau_2, \mathsf{Pd})$ y lo firma $\beta^* = (\beta, Sign_{\mathcal{P}_\mathcal{D}}(\beta))$;
9. envía β^* a $\mathcal{U}$ (en caso de disputa β puede usarse por $\mathcal{U}$ como evidencia que ha salido en τ_2;

[1] El criptosistema es probabilístico

NOTACIÓN			
NOMBRE	NOTACIÓN	NOMBRE	NOTACIÓN
Clave pública de grupo	gpk	Listado de claves privadas de grupo	$gsk[\,]$
Lista de revocación	$grt[\,]$	Base de exponenciación	α
Número primo	p	Número primo	q
Pseudónimo de $\mathcal{U}$ (para pago)	$y_{\mathcal{U}}$	Exponenciación inversa de $y_{\mathcal{U}}$ (secreto)	$x_{\mathcal{U}}$
Número aleatorio j-ésimo	r_j	Exponenciación de r_j	s_j
Reto j-ésimo para $\mathcal{U}$ para demostrar posesión de $y_{\mathcal{U}}$	c_j	Respuesta del reto c_j por $\mathcal{U}$	ω_j
Cifrado probabilístico de $y_{\mathcal{U}}$	$\delta_{\mathcal{U}}$	j-ésimo timestamp	τ_j
Parámetro de verificación	k	Hash del parámetro k	h_k
Firma digital del contenido c generada por E	$Sign_E(c)$	Compromiso de $\mathcal{U}$	σ^*
eticket de entrada firmado por $\mathcal{P}_{\mathcal{S}}$	$\mathsf{t_{in}}^*$	Número de serie de $\mathsf{t_{in}}$	Sn
Identificador del proveedor de servicio de origen	Ps	*eticket* de salida firmado por $\mathcal{P}_{\mathcal{D}}$	$\mathsf{t_{out}}^*$
Reto y tarifa firmado por $\mathcal{P}_{\mathcal{D}}$ para $\mathcal{U}$	β^*	Función de cálculo de la tarifa	$f()$
Tarifa a pagar	a	Identificador del proveedor de servicio de destino	Pd
Cifrado probabilístico de los datos de verificación de $\mathcal{U}$	$\gamma_{\mathcal{U}}$	Cifrado probabilístico de los datos de verificación de $\mathcal{P}_{\mathcal{D}}$	$\gamma_{\mathcal{P}_{\mathcal{D}}}$
Aprobación de pago firmada por $\mathcal{M}_{\mathcal{C}}$	ok^*	Denegación de pago firmado por $\mathcal{M}_{\mathcal{C}}$	ko^*

Tabla III
NOTACIÓN, EN ORDEN DE APARICIÓN.

10. construye $\gamma_{\mathcal{P}_{\mathcal{D}}} = (\beta.a, \mathsf{t_{in}}.\mathsf{Sn}, \mathsf{t_{in}}.\sigma, c_1)$;

setPayment. $\mathcal{U}$ sigue los pasos:

1. verifica la firma de β^*;
2. calcula $\omega_1 = r_1 + c_1 \cdot x_{\mathcal{U}} \pmod q$;
3. compone y cifra $\gamma_{\mathcal{U}} = PK_{\mathcal{M}_{\mathcal{C}}}(\omega_1, \mathsf{t_{in}}.\mathsf{Sn}, \beta.a)$;
4. envía $\gamma_{\mathcal{U}}$ a $\mathcal{P}_{\mathcal{D}}$;

sendingPaymentInfo. $\mathcal{P}_{\mathcal{D}}$ reenvía $\gamma_{\mathcal{U}}$ y $\gamma_{\mathcal{P}_{\mathcal{D}}}$ a $\mathcal{M}_{\mathcal{C}}$;

verifyPayment. $\mathcal{M}_{\mathcal{C}}$ ejecuta los pasos:

1. descifra $\gamma_{\mathcal{U}}$ para obtener la prueba Schnorr ω_1;
2. descifra $\mathsf{t_{in}}.\sigma.\delta_{\mathcal{U}}$ para obtener $y_{\mathcal{U}}$ y aplicar el cargo en la correspondiente cuenta de usuario;
3. verifica la identidad de $\mathcal{U}$: $\alpha^{\omega_1} \stackrel{?}{=} s_1 \cdot (y_{\mathcal{U}})^{c_1}$;
4. si es correcto, la tarifa a se carga en la cuenta correspondiente a $y_{\mathcal{U}}$. Si no genera un mensaje de error $ko =$ ('error autenticación', $\gamma_{\mathcal{U}}$), lo firma $ko^* = (ko, Sign_{\mathcal{M}_{\mathcal{C}}}(ko))$, lo envía a $\mathcal{P}_{\mathcal{D}}$ y para el protocolo;
5. construye $ok = (\mathsf{t_{in}}.\mathsf{Sn}, \beta.a,$ 'ok') y lo firma $ok^* = (ok, Sign_{\mathcal{M}_{\mathcal{C}}}(ok))$;
6. envía ok^* a $\mathcal{P}_{\mathcal{D}}$;

setExit. $\mathcal{P}_{\mathcal{D}}$ sigue los pasos:

1. construye $\mathsf{t_{out}} = (\mathsf{t_{in}}.\mathsf{Sn}, \mathsf{Pd}, \beta.a,$ 'salida a $\beta.\tau_2$') y lo firma $\mathsf{t_{out}}^* = (\mathsf{t_{out}}, Sign_{\mathcal{P}_{\mathcal{D}}}(\mathsf{t_{out}}))$;
2. envía $\mathsf{t_{out}}^*$ a $\mathcal{U}$ y permite la salida del usuario;

checkTicket. $\mathcal{U}$ verifica la firma $\mathsf{t_{out}}^*$ y su contenido.

E. Reclamaciones de usuario

Durante el protocolo de salida, $\mathcal{P}_{\mathcal{D}}$ puede que no actúe correctamente ($\mathcal{P}_{\mathcal{D}}$ puede caerse, cometer errores o actuar incorrectamente) de forma que un usuario honesto reciba un servicio incorrecto. Para solventar el problema, el protocolo puede admitir dos quejas por parte de los usuarios.

QUEJA 1. SE RECIBE UN β^* INCORRECTO: En la salida $\mathcal{U}$ puede enviar la información de validacion $(\mathsf{t_{in}}^*, \Phi^*)$, pero $\mathcal{P}_{\mathcal{D}}$ puede actuar mal y enviar un β^* erróneo a $\mathcal{U}$ o simplemente no enviarlo. Entonces, $\mathcal{U}$ puede reclamar recibir un β^* válido siguiendo estos pasos:

claim1Request. $\mathcal{U}$ reenvía $(\mathsf{t_{in}}^*, \Phi^*)$ y la β^* incorrecta a $\mathcal{M}_{\mathcal{C}}$;

claim1Response. $\mathcal{M}_{\mathcal{C}}$ sigue los pasos:

1. verifica la firma de $\mathsf{t_{in}}^*$;
2. verifica la firma de grupo de Φ^*: $(Verify_G(gpk, \Phi, \bar{\Phi}))$ y comprueba si es el mismo miembro que en la entrada: $VerifyLinkable_G(\mathsf{t_{in}}.\sigma^*, \Phi^*)$;
3. verifica que τ_v no esté expirado y que el sentido ξ es correcto;
4. en caso de β^* incorrecto, $\mathcal{M}_{\mathcal{C}}$ verifica si los parámetros $\beta.\tau_2$ o $\beta.a$ son correctos (p.e. $\beta.\tau_2$ es mayor que el momento actual);
5. genera un nuevo timestamp τ_2;
6. calcula la tarifa a pagar dependiendo de la estación de entrada ($\mathsf{t_{in}}.\mathsf{Ps}$), la estación de salida (Pd) y los correspondientes timestamps (τ_1, τ_2): $a = f_d(\mathsf{t_{in}}.\mathsf{Ps}, \mathsf{Pd}, \mathsf{t_{in}}.\tau_1, \tau_2)$;
7. genera un reto $c_1 \stackrel{R}{\leftarrow} \mathbb{Z}_q$;
8. compone $\beta = (\mathsf{t_{in}}^*, a, c_1, \tau_2, \mathsf{Pd})$ y lo firma $\beta^* = (\beta, Sign_{\mathcal{M}_{\mathcal{C}}}(\beta))$;
9. envía β^* a $\mathcal{U}$;

resume. El protocolo de *salida del sistema* continúa normalmente.

QUEJA 2. SE RECIBE UN $\mathsf{t_{out}}^*$ INCORRECTO: En el protocolo de salida, $\mathcal{U}$ puede enviar información de validación $(\mathsf{t_{in}}^*, \Phi^*, \gamma_{\mathcal{U}})$, pero $\mathcal{P}_{\mathcal{D}}$ puede generar y enviar un $\mathsf{t_{out}}^*$ incorrecto a $\mathcal{U}$ o no enviarlo. Entonces, el usuario puede contactar con $\mathcal{M}_{\mathcal{C}}$ y reclamar recibir un $\mathsf{t_{out}}^*$ válido siguiendo estos pasos:

claim2Request. $\mathcal{U}$ reenvía $(\mathsf{t_{in}}^*, \Phi^*, \beta^*, \gamma_{\mathcal{U}})$ a $\mathcal{M}_{\mathcal{C}}$;

claim2Response. $\mathcal{M}_{\mathcal{C}}$ sigue los pasos:

1. verifica la firma de $\mathsf{t_{in}}^*$;
2. verifica que τ_v no esté expirado y que el sentido ξ es correcto;
3. verifica la identidad de $\mathcal{U}$: $\alpha^{\omega_1} \stackrel{?}{=} s_1 \cdot (y_{\mathcal{U}})^{c_1}$;
4. verifica la firma de grupo Φ^* con $Verify_G(gpk, \Phi, \bar{\Phi})$ y que es el mismo usuario que en la entrada con $VerifyLinkable_G(\mathsf{t_{in}}.\sigma^*, \Phi^*)$;
5. calcula la tarifa a pagar: $a = f_d(\mathsf{t_{in}}.\mathsf{Ps}, \mathsf{Pd}, \mathsf{t_{in}}.\tau_1, \tau_2)$. Entonces, $\mathcal{M}_{\mathcal{C}}$ verifica que a es igual a $\beta.a$;
6. construye $\mathsf{t_{out}} = (\mathsf{t_{in}}.\mathsf{Sn}, \beta.a,$ 'salida a $\beta.\tau_2$') y lo firma $\mathsf{t_{out}}^* = (\mathsf{t_{out}}, Sign_{\mathcal{M}_{\mathcal{C}}}(\mathsf{t_{out}}))$;
7. envía $\mathsf{t_{out}}^*$ a $\mathcal{U}$;

resume. El protocolo de *salida del sistema* continúa de forma normal.

En ambas quejas, $\mathcal{M}_{\mathcal{C}}$ tiene que avisar a $\mathcal{P}_{\mathcal{D}}$ sobre su comportamiento incorrecto o sus problemas de comunicaciones con sus usuarios. $\mathcal{M}_{\mathcal{C}}$ también tiene que avisar a $\mathcal{P}_{\mathcal{D}}$ de futuras acciones si el problema persiste.

F. Reclamaciones de Proveedor

En la salida del sistema, $\mathcal{U}$ puede que no siga el protocolo por distintas razones (por ejemplo, $\mathcal{U}$ puede cometer errores o ejecutar acciones deshonestas). Entonces, el proveedor puede solucionar el problema ejecutando una de las siguientes reclamaciones.

QUEJA 3. SE RECIBE UN $(\mathsf{t_{in}}^*, \Phi^*)$ INCORRECTO: En la salida, $\mathcal{P}_{\mathcal{D}}$ recibe la información de verificación $(\mathsf{t_{in}}^*, \Phi^*)$, pero puede que no sea correcta. Entonces, el proveedor de servicio puede reclamar que se revele la identidad del usuario siguiendo este procedimiento:

claim3Request. El proveedor origen $\mathcal{P}_\mathcal{D}$ envía $(\mathsf{t_{in}}^*, \Phi^*)$ a $\mathcal{M}_\mathcal{G}$;

appealingUser. $\mathcal{U}$ tiene que enviar $(\mathsf{t_{in}}^*, \Phi^*)$ a $\mathcal{M}_\mathcal{G}$ para evitar falsas acusaciones;

claim3Response. Si $\mathcal{U}$ no envía los parámetros requeridos, $\mathcal{M}_\mathcal{G}$ ejecuta los pasos:

1. verifica la firma de $(\mathsf{t_{in}}^*, \Phi^*)$;
2. verifica la firma de grupo de Φ^*: $(Verify_G(gpk, \Phi, \bar{\Phi}))$ y que es el mismo usuario que en la entrada $VerifyLinkable_G(\mathsf{t_{in}}.\sigma^*, \Phi^*)$;
3. verifica la firma de grupo de $\mathsf{t_{in}}.\sigma^*$ generada por $\mathcal{U}$ y revela la identidad del usuario mediante $Open_G(gpk, gmsk, \mathsf{t_{in}}.\sigma^*)$;
4. envía la identidad $\mathcal{U}_i$ a $\mathcal{P}_\mathcal{D}$ y $y_\mathcal{U}$ a $\mathcal{M}_\mathcal{C}$;
5. $\mathcal{U}_i$ se añade a la lista de revocación del grupo;

QUEJA 4. SE RECIBE UN $\gamma_\mathcal{U}$ INCORRECTO*:* En la salida, $\mathcal{P}_\mathcal{D}$ y $\mathcal{M}_\mathcal{C}$ reciben la información complementaria de verificación $\gamma_\mathcal{U}$, pero puede que no sea correcta. Entonces, el proveedor puede reclamar que la identidad del usuario sea revelada mediante el siguiente proceso:

claim4Request. $\mathcal{M}_\mathcal{C}$ genera un rechazo de pago $ko =$ ('error información verificación', $\gamma_\mathcal{U}$), lo firma $ko^* = (ko, Sign_{\mathcal{M}_\mathcal{C}}(ko))$ y lo envía a $\mathcal{P}_\mathcal{D}$. $\mathcal{M}_\mathcal{C}$ también envía $(sk_{\mathcal{P}_\mathcal{D}}(\gamma_\mathcal{U}), \gamma_{\mathcal{P}_\mathcal{D}})$ a $\mathcal{M}_\mathcal{G}$ y detiene el protocolo.

providerInfo. El proveedor destino $\mathcal{P}_\mathcal{D}$ envía $(\mathsf{t_{in}}^*, \Phi^*)$ a $\mathcal{M}_\mathcal{G}$;

appealingUser. $\mathcal{U}$ tiene que enviar $(\mathsf{t_{in}}^*, \Phi^*, \gamma_\mathcal{U})$ a $\mathcal{M}_\mathcal{G}$ para evitar falsas acusaciones;

claim4Response. Si $\mathcal{U}$ no envía los parámetros requeridos, $\mathcal{M}_\mathcal{G}$ ejecuta los pasos:

1. verifica si la información $\gamma_\mathcal{U}$, $\gamma_{\mathcal{P}_\mathcal{D}}$ y $(\mathsf{t_{in}}^*, \Phi^*)$ coincide;
2. verifica la firma de grupo de $\mathsf{t_{in}}.\sigma^*$ generada por $\mathcal{U}$, y revela la identidad del usuario mediante $Open_G(gpk, gmsk, \mathsf{t_{in}}.\sigma^*)$;
3. envía la identidad $\mathcal{U}_i$ a $\mathcal{P}_\mathcal{D}$ y $y_\mathcal{U}$ a $\mathcal{M}_\mathcal{C}$;
4. $\mathcal{U}_i$ se añade a la lista de revocación del grupo;

VI. ANÁLISIS INFORMAL DE SEGURIDAD

En esta sección se analizará la seguridad de la propuesta.

PROPOSICIÓN 1. El sistema propuesto preserva la autenticidad, el no repudio y la integridad para los tickets de entrada y salida.

ARGUMENTO 1. *La creación de etickets fraudulentos es hoy en día computacionalmente inviable.*

Prueba. Por una parte, los *etickets* van firmados, $\mathsf{t_{in}}^* = (\mathsf{t_{in}}, Sign_{\mathcal{P}_\mathcal{S}}(\mathsf{t_{in}}))$ y $\mathsf{t_{out}}^* = (\mathsf{t_{out}}, Sign_{\mathcal{P}_\mathcal{D}}(\mathsf{t_{out}}))$, así como la información enviada antes del pago $\beta^* = (\beta, Sign_{\mathcal{P}_\mathcal{D}}(\beta))$. Si una entidad no autorizada es capaz de crear un *eticket* válido (de entrada o salida) sin el conocimiento de las claves privadas de $\mathcal{P}_\mathcal{S}$ o $\mathcal{P}_\mathcal{D}$, también podría generar firmas digitales impersonando esos proveedores. Suponiendo que se usa un esquema seguro de firma digital, esta operación se considera inviable. Por otra parte, el usuario envía la información de verificación firmada con su clave pública de grupo $\sigma^* = (\sigma, Sign_\mathcal{G}(\sigma))$. Por el mismo motivo, esta firma garantiza que el mensaje es auténtico y que ha sido generado por un usuario válido y no revocado del grupo.

ARGUMENTO 2. *El emisor de un eticket no puede denegar la expedición del eticket.*

Prueba. Los *etickets* son firmados por un emisor autorizado (proveedores de servicio) y considerando que el esquema de firma usado es seguro, esta operación solo puede ser ejecutada por estos emisores. Además, la identidad del emisor está enlazada al ticket y por la propiedades de la firma electrónica, el emisor no puede denegar la autoría de esa firma. Lo mismo ocurre con la firma de grupo, donde si la identidad es revelada, la autoría del mensaje puede ser comprobada.

ARGUMENTO 3. *El contenido de los etickets no puede ser modificado.*

Prueba. Si se supone que el esquema de firma es seguro y que la función *hash* es resistente a colisiones y que su inversa es computacionalmente inviable hoy en día, entonces si se modifica el contenido de un *eticket* la verificación de la firma debería ser incorrecta. Para superar la verificación, la firma se debería regenerar a partir del nuevo contenido del *eticket*. Esta operación es computacionalmente inviable hoy en día. Lo mismo ocurre con la firma de grupo.

RESULTADO DE LA PROPOSICIÓN 1. De acuerdo con las definiciones dadas en §IV y las *Quejas 1, 2 y 3*, se puede asegurar que el protocolo alcanza los requerimientos de seguridad de autenticidad, no repudio e integridad.

PROPOSICIÓN 2. El sistema descrito posee anonimato revocable para los usuarios y los movimientos llevados a cabo por estos no son trazables.

ARGUMENTO 4. *Un eticket es anónimo.*

Prueba. La información relativa a la identidad del usuario está cifrada con la clave pública de la TTP de pago. El proveedor de servicio ($\mathcal{P}_\mathcal{S}$ y $\mathcal{P}_\mathcal{D}$) no puede acceder a esta información porque necesita la clave privada de la TTP de pago. Los usuarios del sistema calculan una firma de grupo ($\mathsf{t_{in}}.\sigma^* = (\sigma, Sign_\mathcal{G}(\sigma))$) que certifica que el que firma es un miembro válido del grupo. Entonces, los proveedores no pueden inferir la identidad de quien ha generado la firma. En caso de una situación anómala, la identidad del usuario que ha firmado puede ser revelada a través de la cooperación de la TTP de pago $\mathcal{M}_\mathcal{C}$ y la TTP de grupo $\mathcal{M}_\mathcal{G}$. Si el usuario aparece en la lista de revocación, su identidad es revelada, permitiendo así posteriores acciones legales o de otro tipo.

ARGUMENTO 5. *El usuario es anónimo frente a los proveedores de servicio durante la fase de pago.*

Prueba. Toda la información relacionada con el pago se cifra y solo $\mathcal{M}_\mathcal{C}$ puede acceder a ella. Los proveedores de servicio son externos al pago y solo reciben la confirmación del mismo de $\mathcal{M}_\mathcal{C}$. Por tanto, $\mathcal{M}_\mathcal{C}$ tiene conocimiento de $y_\mathcal{U}$ a partir del par $(x_\mathcal{U}, y_\mathcal{U})$ donde $y_\mathcal{U} = \alpha^{x_\mathcal{U}} \pmod p$, que identifica al usuario como válido. Entonces, el usuario se autentica mediante $x_\mathcal{U}$ a través de una prueba de conocimiento nulo Schnorr [9].

ARGUMENTO 6. *Múltiples firmas de grupo generadas por el mismo usuario no deben ser trazables entre si por los proveedores de servicio u otras entidades externas al sistema.*

Prueba. El esquema de firma de grupo usado [2] hace uso de un esquema de firma probabilístico, es decir, no es posible predecir un texto cifrado dado un determinado texto en claro. Esta propiedad permite la no trazabilidad entre diferentes firmas de grupo realizadas por el mismo usuario.

RESULTADO DE LA PROPOSICIÓN 2. De acuerdo a las definiciones dadas en §IV y las *Quejas 4, 5 y 6*, se puede

asegurar que el protocolo garantiza los requerimientos de anonimato revocable y la no trazabilidad.

PROPOSICIÓN 3. El protocolo no permite reusar *eticket* y garantiza también el control del tiempo de validez del mismo.

ARGUMENTO 7. *El protocolo no permite el doble gasto.*

Prueba. Si un usuario intenta reusar un *eticket* de entrada, el número de serie ya estará marcado como usado. Si este comportamiento irregular puede ser probado, la TTP de grupo $\mathcal{M}_\mathcal{G}$ puede incluir a este usuario en la lista de revocación.

ARGUMENTO 8. *El eticket ya no puede ser válido si su tiempo de validez τ_v ha expirado.*

Prueba. La estación destino $\mathcal{P}_\mathcal{D}$ recibe el *eticket* del usuario para someterlo a verificación. Entonces el instante actual se compara con el tiempo de validez τ_v contenido en el *eticket* de entrada $\mathsf{t_{in}}^*$ y firmado por $\mathcal{P}_\mathcal{S}$.

RESULTADO DE LA PROPOSICIÓN 3. De acuerdo con la definición dada en §IV y las *Quejas 7 y 8*, se puede asegurar que el protocolo garantiza los requerimientos de seguridad de impedir el reuso y el control del tiempo de validez del *eticket*.

PROPOSICIÓN 4. El protocolo propuesto evita ataques de usuarios confabulados.

ARGUMENTO 9. *El sistema AFC basado en distancia descrito en §V no puede ser atacado por usuarios confabulados.*

Prueba. En los sistemas basados en distancias, un ataque basado en el intercambio del *eticket* sería provechoso solo en algunos casos. En estos casos, el sentido de los usuarios debe ser diferente. Si los dos sentidos del sistema están separados físicamente, entonces los usuarios viajando en sentido opuesto no son capaces de intercambiarse los *etickets* para salir, ya que el sentido incluído en el *eticket* sería diferente al sentido de la marcha del usuario.

Los sistemas donde los sentidos de la marcha no están separados físicamente, requieren el uso de firmas de grupo enlazables. Con este procedimiento, donde el usuario conserva el anonimato, el proveedor puede asegurarse que el usuario que sale del sistema es el mismo que obtuvo el *eticket* de entrada. Por este motivo, los usuarios no pueden atacar al sistema intercambiándose los *etickets*.

RESULTADO DE LA PROPOSICIÓN 4. De acuerdo al protocolo descrito en §V, se puede asegurar que el protocolo no puede ser atacado por usuarios confabulados.

VII. IMPLEMENTACIÓN Y PRUEBAS

El protocolo presentado ha sido implementado juntamente con el anterior protocolo basado en tiempo [1] para medir su rendimiento y demostrar su uso práctico. La implementación se ha desarrollado sobre la plataforma móvil Android. El motivo de esta elección por delante de otras plataformas similares (Symbian, Windows Mobile o iOS) es por qué Android actualmente es la plataforma con mayor crecimiento y aceptación para todo tipo de terminales [10], [11] y cuenta con una amplia comunidad de desarrollo. Además el lenguaje de programación utilizado para desarrollar aplicaciones para Android es Java.

El primer reto a superar es la implementación del esquema de firma de grupo utilizado [2]. Después de una profunda búsqueda, no se encontró ninguna implementación del esquema escrita en Java. El esquema está basado en pairings por lo que los cálculos se delegan a una librería especializada llamada jPBC (Java Based Pairing Cryptography) [12] que es capaz de calcular operaciones de pairings complejas sobre curvas elípticas. Además de la firma de grupo, jPBC también se usa para generar valores aleatorios, exponenciaciones y operaciones aritméticas en el protocolo AFC. Como complemento a jPBC, también se usa la librería Bouncycastle [13] que implementa multitud de algoritmos criptográficos como por ejemplo RSA.

La implementación consta de un cliente y de un lado servidor. La comunicación entre ambos se realiza mediante mensajes XML. Seguidamente se describe brevemente cada lado del protocolo.

- **Lado Cliente**. La aplicación de usuario está desarrollada sobre el sistema operativo Android, usando el *API level 7* por compatibilidad entre los dos dispositivos móviles.
- **Lado Servidor**. Comprende la *TTP de Grupo* ($\mathcal{M}_\mathcal{G}$), la *TTP de Pago* ($\mathcal{M}_\mathcal{C}$), la *Estación Origen* ($\mathcal{P}_\mathcal{S}$) y la *Estación Destino* ($\mathcal{P}_\mathcal{D}$). Cada una de ellas está desarrollada sobre Java JDK 6.0 usando su propia base de datos MySQL y exponiendo sus servicios mediante un puerto TCP.
- **Comunicaciones**. Los mensajes intercambiados entre el cliente y la infraestructura son serializados mediante XML, asegurando así la portabilidad del sistema.

A. Escenario de Pruebas

El escenario de pruebas está compuesto por un portátil donde se aloja la parte servidor mientras el cliente se prueba sobre dos smartphones Android: un HTC Desire (gama media-alta) y un HTC Wildfire (gama baja-media). Las características de los dispositivos están listadas en la Tabla IV. La conectividad entre el cliente Android y el sistema AFC se realiza sobre una red 802.11g usando Java Sockets, mientras que los mensajes entre servidores se transmiten a través de la interfaz de red local del portátil. Finalmente, con el fin de obtener valores medios para cada ejecución de cada protocolo, se realizan 20 iteraciones para obtener sus valores medios descartando las medidas extremas.

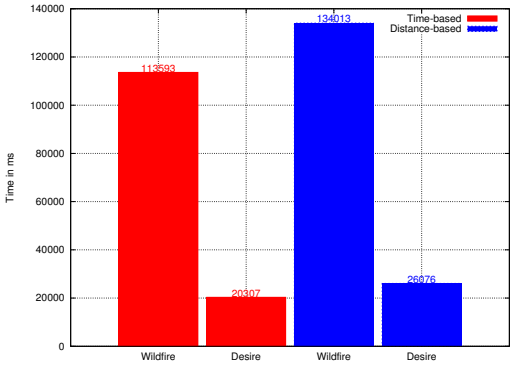


Fig. 1. Tiempo total consumido para cada protocolo sobre cada smartphone.

Dispositivo	CPU	RAM	ROM	OS
Portátil	Intel Core Duo 2 1.6GHz	4GB		Debian Linux 5.0
HTC Desire	Qualcomm Snapdragon 1GHz	576MB	512MB	Android 2.2
HTC Wildfire	Qualcomm MSM7225 528MHz	384MB	512MB	Android 2.1

Tabla IV
CARACTERÍSTICAS TÉCNICAS DE LOS DISPOSITIVOS DE PRUEBA.

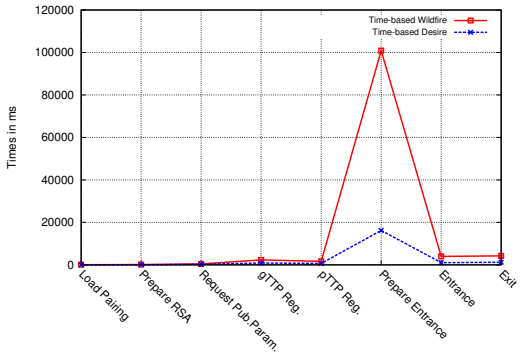


Fig. 2. Flujo temporal del protocolo basado en tiempo.

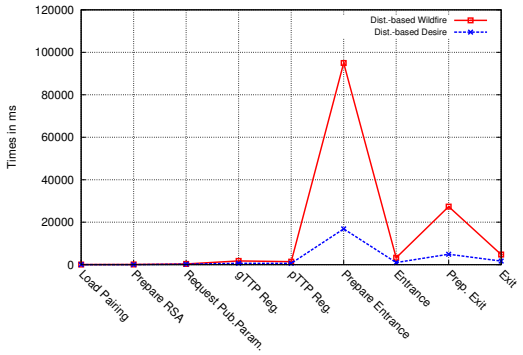


Fig. 3. Flujo temporal del protocolo basado en distancia.

B. Discusión

Una vez realizada la implementación del protocolo, se van a analizar y discutir los resultados de rendimiento obtenidos. El primer gráfico en la Fig. 1 muestra que la ejecución es más rápida sobre el HTC Desire, como era de esperar. Si se analizan los tiempos para cada versión del protocolo, se observa que para el protocolo basado en tiempo el Wildfire es 5.6 veces más lento que el Desire, mientras que en la versión para distancias es 5.2 veces más lento. Pero lo interesante es observar como la penalización en tiempo para el protocolo de distancias no es tan grande como cabía esperar.

El siguiente paso es analizar en qué etapas de cada protocolo se consume más tiempo de ejecución para cada smartphone. Para analizar la Fig. 2 y la Fig. 3, se tiene que explicar el significado de cada etapa del eje x:

- **Load Pairing**. Aquí se cargan todos los datos necesarios para ejecutar los pairings (cargar los parámetros de la curva elíptica y preparar objetos Java).
- **Prepare RSA**. Es el tiempo de carga del par de claves RSA desde un fichero PEM en la SD del smartphone.
- **Request Pub.Param**. Es el tiempo requerido para solicitar a $\mathcal{M}_\mathcal{G}$ el parámetro público α.
- **gTTP Reg**. Es el tiempo necesario para completar el registro del usuario en $\mathcal{M}_\mathcal{G}$ (*generatePseudonym* y *keyIssue* de §V-D).
- **pTTP Reg**. Es el tiempo consumido para el registro del usuario en $\mathcal{M}_\mathcal{C}$ (*startingZKP*, *challengeGeneration*, *proofGeneration* y *verifyPseudonym* de §V-D).
- **Prepare Entrance**. Es el tiempo de precomputación realizado antes de entrar en el sistema donde se calculan los parámetros precomputables de la firma de grupo.
- **Entrance**. Es el tiempo consumido para entrar en el sistema, es decir, el paso especificado en §V-D.
- **Prepare Exit**. Es el tiempo de precomputación ejecutado antes de salir del sistema. Solo aplica a la versión basada en distancia.
- **Exit**. Es el tiempo necesario para salir del sistema, es decir, el consumido por el protocolo de §V-D.

Entonces la Fig. 2 muestra el plano de tiempo para el protocolo basado en tiempo. Como se puede observar, la mayor parte del tiempo se consume en la precomputación antes de la entrada en el sistema. Por otra parte, la Fig. 3 muestra como el protocolo basado en distancia arroja unos resultados similares hasta la salida, donde otro conjunto de precomputaciones deben ser ejecutadas antes de llegar al punto de control. Dos resultados importantes se deducen de estas gráficas:

- El HTC Desire solo es claramente superior al HTC Wildfire donde se requiere más potencia de cálculo a la hora de realizar las precomputaciones.
- En el protocolo basado en distancia, el HTC Wildfire solo es 3 segundos más lento que el HTC Desire en la salida.

Por lo tanto, se puede afirmar que la versión basada en distancia incrementa el coste respecto de la versión basada en tiempo pero permanece usable para ambos dispositivos si se realiza la precomputación previa.

La Fig. 4 y la Fig. 5 muestran los diferentes tiempos de ejecución para cada etapa del protocolo (Inicialización, Registro de usuario en las dos TTP, Entrada, Salida y Precomputaciones agregadas) sobre ambos dispositivos tanto si se usa precomputación como si no.

Por una parte, la Fig. 4 muestra como en ambos protocolos la entrada es el cuello de botella del sistema. Además, en el protocolo basado en distancias la salida consume más tiempo que en el protocolo basado en tiempo ya que los cálculos son más complejos como contrapartida para obtener un mayor nivel de seguridad.

Por otra parte, en la Fig. 5 todas las precomputaciones son extraídas de las operaciones de entrada y salida del sistema y acumuladas en la última barra de las gráficas. Como se puede comprobar, el hecho de extraer la precomputación mejora ostensiblemente el rendimiento del sistema en ambas versiones en el momento de la llegada a los puntos de control de entrada y salida. Por lo tanto, es claro que es imprescindible separar el tiempo de precomputación del tiempo requerido en

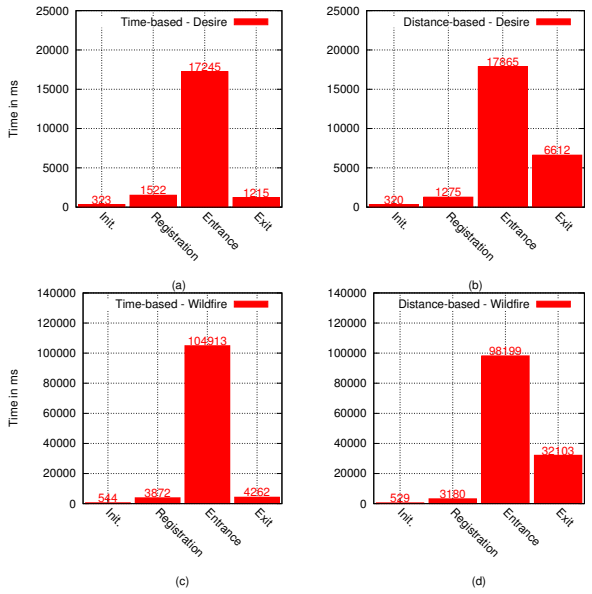


Fig. 4. Fases para cada versión y smartphone sin precomputación.

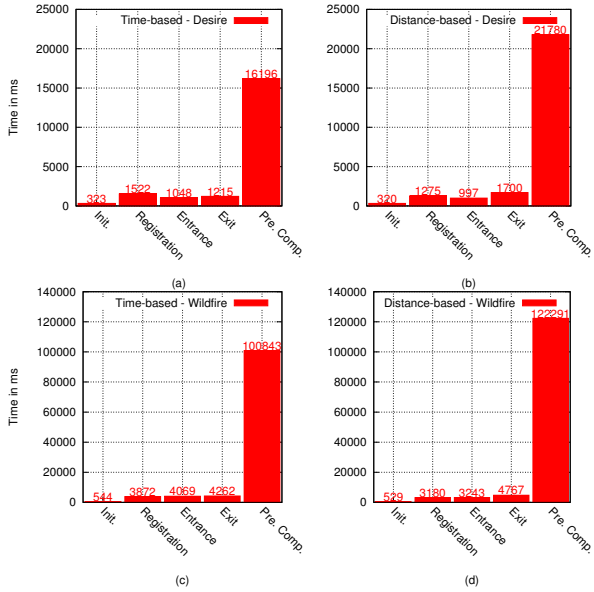


Fig. 5. Fases para cada versión y smartphone con precomputación.

los puntos de control para mejorar el rendimiento general del sistema.

Por tanto, se puede concluir el análisis remarcando que los dos esquemas AFC y la implementación desarrollada es apta para su uso en un entorno realista. Además, las diferencias entre la versión basada en tiempo y la basada en distancia no es tan grande. Por tanto, si se requiere un nivel mayor de seguridad pero asumiendo un pequeño coste añadido, se puede aplicar la versión basada en distancia. Entonces, si lo que se prima es la eficiencia se puede elegir el protocolo basado en tiempo. En ambos casos, el protocolo es adecuado para dispositivos móviles actuales, lo que hace pensar que en un futuro próximo cuando aparezcan dispositivos más potentes, el rendimiento del protocolo será aún mejor.

VIII. Conclusiones

En este artículo se ha presentado un sistema electrónico AFC basado en distancia fruto de la mejora de un trabajo previo. El uso de firmas de grupo ha permitido preservar el anonimato y la privacidad de los usuarios, al mismo tiempo que se sigue permitiendo su autenticación. Además, el sistema proporciona la posibilidad de revocar el anonimato en caso de comportamientos fraudulentos por parte de los usuarios y, a diferencia de otras propuestas, las sesiones de usuario están protegidas contra la trazabilidad, lo que imposibilita la creación de perfiles de usuarios.

Una vez presentada y evaluada la propuesta, se ha implementado el protocolo sobre la plataforma móvil Android y se han desarrollado pruebas de rendimiento que demuestran que el esquema es eficiente y que puede ser aplicado en un escenario realista.

El trabajo futuro irá en la dirección de extender la implementación a nuevas tecnologías de comunicaciones más adecuadas para los procedimientos de comercio electrónico, como NFC (Near Field Communication) y estudiar la aplicabilidad de este esquema en otros escenarios que requieran tarificación.

Disclaimer and Acknowledgements

Soportado parcialmente por el MICINN (proyectos TSI2007-65406-C03-01, ARES-CONSOLIDER INGENIO 2010 CSD2007-00004, “RIPUP” TIN2009-11689 y Audit Transparency Voting Process PT-430000-2010-31), el MITYC (proyecto “eVerification’ TSI-020100-2009-720 y SeCloud TSI-020302-2010-153), y el Gobierno de Catalunya (subvención 2009 SGR1135). Los autores solo son responsables de las opiniones expresadas en este artículo que no reflejan necesariamente la posición de la UNESCO ni comprometen esa organización.

Referencias

[1] A. Vives-Guasch, J. Castellà-Roca, M. Payeras-Capella, and M. Mut-Puigserver, “An electronic and secure automatic fare collection system with revocable anonymity for users,” in *MoMM*, 2010.

[2] D. Boneh, X. Boyen, and H. Shacham, “Short group signatures,” in *CRYPTO*, ser. LNCS, vol. 3152, 2004, pp. 41–55.

[3] H. Wang, J. Cao, and Y. Zhang, “Ticket-based service access scheme for mobile users,” *Aust. Comput. Sci. Commun.*, vol. 24, no. 1, pp. 285–292, 2002.

[4] L. Buttyn, T. Holczer, and I. Vajda, “Providing location privacy in automated fare collection systems,” in *15th IST Mobile and Wireless Communication Summit*, 2006.

[5] T. S. Heydt-Benjamin, H.-J. Chae, B. Defend, and K. Fu, “Privacy for public transportation,” in *6th Workshop on Privacy Enhancing Technologies (PET 2006)*, 2006, pp. 1–19, lNCS 4258.

[6] S.-P. Hong and S. Kang, “Ensuring privacy in smartcard-based payment systems: A case study of public metro transit systems,” in *Communications and Multimedia Security*, 2006, pp. 206–215.

[7] O. Jorns, O. Jung, and G. Quirchmayr, “A privacy enhancing service architecture for ticket-based mobile applications,” in *ARES 2007*, 2007, pp. 374–383, vol. 24.

[8] G. Madlmayr, P. Kleebauer, J. Langer, and J. Scharinger, “Secure communication between web browsers and nfc targets by the example of an e-ticketing system,” in *EC-Web ’08*, 2008, pp. 1–10.

[9] C.-P. Schnorr, “Efficient signature generation by smart cards,” *J. Cryptology*, vol. 4, no. 3, pp. 161–174, 1991.

[10] Gartner, “Gartner Says Worldwide Mobile Phone Sales Grew 35 Percent in Third Quarter 2010; Smartphone Sales Increased 96 Percent,” http://www.gartner.com/it/page.jsp?id=1466313, November 2010.

[11] IDC, “IDC Corporate USA: Android drives big smartphone growth in 2010,” http://www.computerworld.com/s/article/9208478/ Android_drives_big_smartphone_growth_in_2010_IDC_says, February 2011.

[12] A. De Caro, “jPBC: Java Pairing-Based Cryptography Library,” http://gas.dia.unisa.it/projects/jpbc/.

[13] B. Castle, “Bouncy Castle Crypto APIs,” http://www.bouncycastle.org/.

Acceso seguro a nodos RFID en una arquitectura de red personal

Pablo Najera, Rodrigo Roman, Javier Lopez
Departamento de Lenguajes y Ciencias de la Computación
Universidad de Málaga
Edificio Institutos Universitarios de Investigación (4.0.8), Parque Tecnologico de Andalucía. CP 29590
{najera | roman | jlm}@lcc.uma.es

***Resumen*-El paradigma de red personal (PN) permitirá la interacción y colaboración del creciente abanico de dispositivos personales. Con tal fin, la PN ha de integrar en su seno de forma segura múltiples tecnologías heterogéneas con diversas capacidades computacionales y de comunicación. En particular, la incorporación de la tecnología RFID en objetos personales conlleva múltiples riesgos de seguridad y privacidad que han suscitado un elevado interés de la comunidad investigadora en los últimos años. Más allá de su seguridad de forma aislada, su integración en la PN y la interacción de ésta con redes de área extensa como Internet of Things requieren una arquitectura de red personal adecuada para tal contexto. Este artículo proporciona los fundamentos de tal arquitectura segura incluyendo el análisis de aspectos como la incorporación e inicialización de las restringidas etiquetas RFID en la PN, la autenticación tanto de miembros de la PN como de usuarios y servicios remotos en su acceso a las tecnologías de contexto, el control de las políticas de privacidad y el establecimiento de canales seguros de comunicación supervisados.**

***Palabras Clave*-** Seguridad RFID, red personal, arquitectura software

I. INTRODUCCIÓN

El emergente paradigma de red personal habilita la comunicación de todos los dispositivos y servicios del usuario de forma flexible, segura y auto-organizada. Dicha topología de red ha de proporcionar las bases para la provisión de servicios relativos al contexto del usuario así como permitir la comunicación con redes de área extensa (ej. Internet of Things) con objeto de interactuar con dispositivos y redes remotas, así como facilitar la complementación y agregación de los servicios personales.

Dentro de las tecnologías clave en la realización de este modelo de red se encuentran las redes de sensores corporales (body sensor networks, BSNs) que permiten desde el control de los parámetros fisiológicos del usuario al reconocimiento de sus actividades personales o profesionales lo que está llevando a su adopción en múltiples áreas, desde el cuidado de la tercera edad y monitorización de pacientes a aplicaciones nóveles en áreas militares y de consumo.

Si bien no analizada usualmente como miembro de la PN, otra tecnología clave en la integración de capacidades computacionales y de comunicación en objetos cotidianos es la tecnología RFID (*Radio Frequency IDentification*, identificación por radiofrecuencia), la cual permite la identificación única de un objeto y la obtención de datos relacionados (ej. características o historial), gracias a la incorporación de un circuito miniaturizado (etiqueta RFID) en el objeto a controlar. De hecho, la tecnología RFID puede considerarse como un sensor adicional, donde en lugar de parámetros como temperatura o humedad, la red siente los objetos que se encuentran presentes y sus metadatos. Desde esta perspectiva, el lector RFID actúa como otro nodo sensor, que obtiene este tipo particular de datos sobre el contexto basado en el soporte de las etiquetas RFID.

La ITU describe la tecnología RFID como uno de los pivotes que habilitarán la venidera Internet of Things, convirtiendo los objetos cotidianos en objetos inteligentes [1], mientras la Comisión Europea espera que el uso de esta tecnología se multiplique por cinco durante la próxima década. Sin embargo, hay que tener en cuenta las amenazas potenciales a la privacidad y seguridad que puede generar su integración en objetos personales y documentación de los usuarios. Debido a esto, la comunidad investigadora ha dedicado notables esfuerzos a minimizar los riesgos de seguridad proporcionando un amplio rango de protocolos de autenticación mutua [2,3], esquemas de protección de la privacidad [4,5] y primitivas criptográficas ligeras [6,7] para esta tecnología con objeto de evitar accesos no autorizados a las etiquetas RFID personales, así como el seguimiento y perfilado del usuario.

Tal y como se presenta más tarde en este artículo, la integración segura de la tecnología RFID en la PN como tecnología de percepción del contexto complementa a las BSNs y proporciona notable beneficios al conocimiento y servicios potenciales de la PN. La seguridad de RFID como tecnología independiente está alcanzado un adecuado nivel de madurez gracias a los avances de investigación en los últimos años; sin embargo, su integración en el modelo de PN, interacción con otros recursos de la red, usuarios remotos y proveedores de servicio requiere un análisis de seguridad específico y una arquitectura de PN preparada para tales tecnologías heterogéneas. Aunque un creciente volumen de investigación se está enfocando a los paradigmas de PN con la propuesta de diversas arquitecturas de red [8,9,10], y los beneficios de la integración de redes de sensores y RFID ha motivado ya la propuesta de diversas arquitecturas en diferentes escenarios [11,12,13], ninguna de ellas ha introducido la integración segura de RFID y redes de sensores inalámbricas en PNs.

Este artículo expone los beneficios de la colaboración de RFID y tecnologías de sensores en redes PN, analiza como esta integración podría lograrse y define los fundamentos de una arquitectura de PN segura. Tal arquitectura permitiría

ISBN: 978-84-694-5948-5

proporcionar diversas funcionalidades como registrar y mantener de forma segura las etiquetas personales como miembros de la PN, autenticar y autorizar tanto nodos PN como dispositivos remotos en sus solicitudes para acceder a estas tecnologías sensibles al contexto, proporcionar un túnel de comunicación segura con las entidades sin capacidad IP y asegurar el cumplimiento de las políticas de seguridad y privacidad en estas comunicaciones.

Este artículo está organizado de la siguiente forma. La Sección 2 muestra las ventajas y limitaciones de la integración de RFID y BSNs en las PNs. La Sección 3 presenta nuestro concepto de red personal y los tipos de nodos contemplados. La Sección 4 introduce los módulos de nuestra propuesta de arquitectura de PN segura. La Sección 5 analiza la gestión segura de nodos PN y comunicación con las tecnologías de contexto en la arquitectura. Finalmente, la Sección 6 concluye el artículo.

II. ADECUACIÓN DE LA INTEGRACIÓN DE RFID EN PNS

A pesar de que las BSNs proporcionan a la PN cierta consciencia sobre el contexto del usuario a través de los parámetros fisiológicos del propietario, sus actividades y entorno, la representación lograda de la realidad que le rodea no es completa y el conocimiento manejado por el sistema de información para monitorizar y dar soporte al usuario está abierto a otras contribuciones. La tecnología RFID complementa adecuadamente a la BSN. En particular, RFID mejora las características de la red en los siguientes aspectos:

- *Mayor alcance*: la extrema miniaturización de las etiquetas RFID, su habilidad para obtener energía durante el propio proceso de lectura y su bajo coste permite llevar las capacidades de computación y comunicación a un rango mayor de productos de consumo, mobiliario, elementos de infraestructura y objetos personales, incrementando sustancialmente la calidad y cantidad de datos manejados por la PN. Sin embargo, al mismo tiempo, tales objetos personales RFID sólo disponen de recursos altamente restringidos y criptografía ligera, incrementando los riesgos de seguridad y privacidad en la PN.
- *Detectar presencia*: la tecnología RFID permite reconocer la presencia de objetos individuales portados por el usuario o en su contexto, denotando información sobre las herramientas que el usuario tiene disponibles, su actividad actual y rango de acciones potenciales. Basado en esta información, la PN puede proporcionar información específica para apoyar al usuario, habilitar servicios de red, o lograr privilegios especiales en el entorno gracias a la posesión de llaves, equipamiento profesional, tarjetas de identificación u otros objetos distinguidos.
- *Características de los objetos personales*: las etiquetas RFID pueden proporcionar además del reconocimiento del objeto, metadatos sobre sus características. De esta forma, la PN puede incrementar su conocimiento sobre la situación donde el usuario está inmerso, así como capacidades y propiedades de los objetos accesibles, utilizando estos datos para mejorar sus servicios.
- *Registro de actividad en objeto*: la información mantenida en las etiquetas puede incluir también un registro sobre interacciones previas de los objetos personales con otros dispositivos y PNs, lugares donde el objeto ha estado, propietarios previos o hechos relevantes relacionados con el objeto. Este tipo de información histórica del objeto, definida y adaptada a las características específicas y propósito de cada tipo de objeto personal, incrementaría la calidad de los datos gestionados por la PN, así como la información forense recopilada para detectar nodos comprometidos, intrusiones o ataques.
- *Gestión transparente y segura de información personal*: una parte significativa de la información personal (incluyendo certificados de eventos personales, calificaciones académicas, documentos médicos o económicos, informes y estudios) se mantiene actualmente como documentación en papel. La integración de la tecnología RFID en la documentación personal proporcionará un enlace transparente con el mundo digital para un procesado ágil y automatizado de sus contenidos. Además, habilitará el uso de mecanismos de seguridad avanzados que han sido tratados extensamente tanto en documentación electrónica como en los primeros pioneros en documentación híbrida (ej. la extensa suite de mecanismos de seguridad en ePassport), sin sacrificar la fiabilidad y conveniencia del soporte físico.
- *Autenticación de usuario*: como beneficio adicional de la integración de RFID en documentación personal, la integración de esta tecnología en las tarjetas de identificación y documentación habilita la identificación segura y autenticación del usuario en su PN, en el contexto que rodea al usuario o incluso el acceso a redes y servicios remotos con mínima interacción del usuario.

Por lo tanto, la integración segura de RFID en la PN puede mejorar de forma sustancial los servicios relativos al contexto. Aunque la integración de las tecnologías de RFID y sensores aporta múltiples beneficios a la PN, la mayoría de etiquetas RFID sólo implementan criptografía ligera y muestran capacidades computacionales y de comunicación muy limitadas elevando potenciales riesgos de seguridad en la PN. Además, la heterogeneidad de recursos entre RFID, sensores y otros dispositivos personales muestran la necesidad de un modelo de comunicaciones seguro adecuado para la integración de las etiquetas personales en la arquitectura PN.

III. ARQUITECTURA HARDWARE DE RED PERSONAL

Nuestra visión del paradigma de red personal se centra en la definición de una arquitectura segura de red para la integración de la tecnología RFID en la esfera de nodos que rodea al usuario, el corazón de la PN, y la comunicación de este núcleo avanzado con dispositivos remotos (ej. clusters de dispositivos personales en localizaciones remotas, otras PNs o servicios centrales de monitorización). Al igual que en la literatura relacionada [9,10], consideramos una arquitectura de red centralizada donde el dispositivo maestro da soporte a las comunicaciones y gestión de la red, mientras proporcionamos un especial énfasis a la integración de dos tecnologías base para el reconocimiento del contexto: redes de sensores inalámbricas y RFID. En particular, consideramos los siguientes tipos de nodos (véase Fig. 1):

- *Dispositivo maestro*: sin restricciones computacionales y de memoria. El usuario interactúa con él

frecuentemente garantizando la recarga de su batería o incorpora técnicas de recolección de energía de forma que es posible asumir su estado operativo. Integra interfaces de comunicación para interactuar con redes de área extensa (ej. 3G/UMTS, LTE o WiMax) y es portado normalmente por el usuario. Aunque podrían surgir dispositivos concretos para desempeñar tal rol, los omnipresentes teléfonos inteligentes ya satisfacen este perfil.

- *Sensores inalámbricos*: recaban información sobre los parámetros fisiológicos del usuario y estado del contexto. Son posibles diversas características y localizaciones en el usuario y deben ser adaptadas al propósito y aplicaciones de la PN. La red podría incluir una estación base que gestione los nodos sensores y agregue la información, aunque dicha función podría integrarse en otro nodo de la red como el dispositivo maestro.
- *Etiquetas RFID*: múltiples tipos de tecnología RFID pueden coexistir en la PN. Por ejemplo, las etiquetas pasivas UHF EPC Gen2 son más adecuadas para objetos personales (ej. ropa, gafas o herramientas profesionales) cubriendo requisitos básicos de identificación y gestión de información, con un bajo coste por etiqueta y largas distancias de lectura. Por otra parte, la documentación personal requiere mecanismos criptográficos avanzados tales como los disponibles en las etiquetas pasivas HF basadas en ISO/IEC 14443. La tecnología RFID activa proporciona capacidades más avanzadas de computación y medición de características físicas, si bien, en la mayoría de escenarios consideramos que ésta puede ser sustituida por nodos sensores (ej. la familia Mica [14]) a medida que su coste se reduzca en el futuro.
- *Lector(es) RFID*: encargados de identificar y recuperar la información almacenada en los objetos personales. Pueden ser necesarios lectores multi-estándar para comunicarse con los diferentes tipos de tecnologías RFID. Los lectores UHF portables son capaces de acceder a los objetos personales en la esfera del usuario (radio aproximado de 2m) mientras los lectores pasivos HF (tales como los integrados en algunos modelos de teléfonos inteligentes [15,16]) requieren alta proximidad a las etiquetas durante el proceso de comunicación. En caso de que se requiera lectura a corta distancia, podría ser necesaria la notificación al usuario (a través de dispositivos de entrada/salida) para su interacción explícita en el proceso de comunicación.
- *Dispositivos de entrada/salida*: además de los omnipresentes *smartphone*, se prevé que emerjan otras tecnologías para PNs con objeto de proporcionar métodos no intrusivos para la entrada de datos (ej. paneles táctiles en ropa, brazaletes equipados con sensores) y presentación de la información (ej. gafas de realidad aumentada).
- *'Gadgets' o dispositivos avanzados*: diseñados para tareas específicas (ej. GPS, reproductores de música, cámaras digitales y dispositivos de juego). Pueden participar de forma no continua en la red habilitando servicios adicionales. Presentan recursos computacionales y de comunicación menos restringidos que las tecnologías ubicuas.

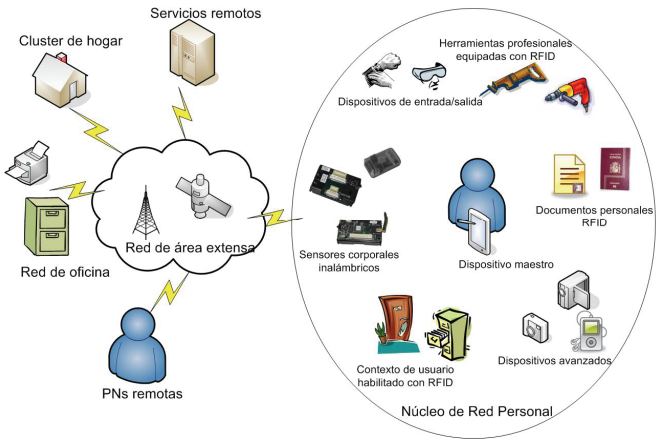


Fig. 1. Esquema de comunicaciones en la red personal

IV. COMPONENTES SOFTWARE EN LA ARQUITECTURA DE RED PERSONAL

Nuestra propuesta no es la primera contribución de una arquitectura software para PNs. En la literatura existente se ha trabajado ya en este área [8,9,10] proporcionando una arquitectura general para este novel paradigma de red que ya considera un amplio rango de aspectos de la gestión de red para dispositivos personales genéricos. Mientras estos trabajos previos proporcionan un adecuado soporte para el desarrollo de PNs, tales enfoques genéricos no analizan cómo lograr la integración segura de la tecnología RFID en la PN.

Las entidades remotas que requieran comunicarse con las etiquetas no pueden acceder a ellas directamente (las etiquetas RFID no poseen dirección IP y las entidades remotas requerirían localizar su ubicación actual en la PN y lectores RFID al alcance). Además, dada la potencial pérdida de información personal, debe poder garantizarse el cumplimiento de las políticas de privacidad del usuario en cualquier comunicación con dichos dispositivos. Debido a esto, la PN debería gestionar el direccionamiento y acceso seguro a etiquetas personales, asegurando el cumplimiento de los requisitos de seguridad en tales comunicaciones.

En la materialización de nuestra visión, la PN debe proporcionar soporte a la colaboración segura de los nodos heterogéneos que coexisten en la red, así como su interacción con entidades externas. Para lograr dicho objetivo, los dispositivos personales deben ser reconocidos como miembros de la PN, proporcionando mecanismos seguros para inicializar nuevos nodos o transferir la propiedad desde otras entidades. Los miembros de la PN y las entidades externas autorizadas deben disponer de las claves y credenciales actualizadas de la red, así como ser capaces de establecer comunicaciones seguras con otros nodos (incluyendo nodos basados en tecnologías de red incompatibles). Durante las comunicaciones, las entidades deben ser autenticadas y se debe asegurar el cumplimiento de las políticas de privacidad. Con objeto de lograr estos objetivos, proponemos una arquitectura de PN basada en los siguientes módulos y comportamiento (véase Fig. 2):

• *PN Members Database* (Base de datos de miembros): encargada de mantener una base de datos de los nodos que se reconocen como entidades de la PN. La base de datos debería mantener metadatos relativos a cada nodo único durante el tiempo que pertenezcan a la red, incluyendo direccionamiento (ej. IP, MAC, dirección PN), materiales criptográficos (ej. certificados digitales y claves), roles, niveles de reputación y privilegios.

• *Member Discovery and Maintenance Module* (Modulo de descubrimiento y mantenimiento de miembros): PN es un paradigma de red dinámico donde se necesita incorporar bajo demanda nuevos dispositivos personales, mientras los antiguos miembros de PN pueden cambiar de propietario, ser comprometidos o desechados. Este módulo gestiona del ciclo de vida seguro de los dispositivos asociados a PN, ya sea con una relación permanente o temporal en la red, incluyendo la incorporación a PN (es decir, proceso de inicialización, intercambio de claves y material criptográfico), actualización de claves y recursos, así como protocolos de desasociación.

• *Naming Resolution and Communication Management* (Resolución de nombres y gestión de comunicaciones): recibe las peticiones de miembros de PN o dispositivos remotos que desean comunicarse con un nodo de PN identificado por una convención de nombres reconocible. El módulo gestiona la solicitud resolviendo la identidad del nodo final, comprobando los privilegios del nodo solicitante (a través del módulo Authentication and Authorization Module), y transfiriendo la conexión al módulo de red apropiado (es decir, PN Routing o Secure Context Management).

• *Authentication and Authorization Module* (Módulo de autenticación y autorización): para (re)conectar a la PN y establecer conexiones a dispositivos de la red, tanto los miembros de PN como nodos remotos necesitan autenticarse en la red. Este módulo gestiona tal proceso y, en base a los privilegios de los nodos, proporciona autorización para futuras interacciones con los miembros de PN.

• *PN Routing* (Enrutamiento en PN): determina la ruta más adecuada para interconectar al solicitante (local o remoto) con la entidad solicitada de PN. La ruta tiene en cuenta la movilidad de los nodos en la red, así como la heterogeneidad en tecnologías de comunicación y capacidades computacionales con objeto de determinar la posición actual del nodo final e incluir los nodos proxy necesarios en la ruta.

• *Secure tunnel Manager* (Gestor de túnel seguro): se encarga de habilitar tales comunicaciones seguras entre los nodos finales, incluyendo el uso de proxies y pasarelas en PN que actúen como puente entre diferentes tecnologías de red, adaptando los mecanismos de seguridad empleados en cada conexión salto a salto con objeto de maximizar el nivel de seguridad de acuerdo a las capacidades de cada par de nodos.

• *Privacy policies and profile DB* (Base de datos de políticas de privacidad y perfiles): gestiona la información relativa al perfil del usuario, así como las políticas de privacidad que definen cómo se debe tratar su información personal y datos almacenados o generados por la PN.

• *Secure Context Management* (Gestión segura del contexto): encargado de gestionar la información generada por las tecnologías sensibles al contexto (es decir, redes de sensores y RFID). Esta información debe ser procesada de acuerdo a las restricciones de privacidad y seguridad deseadas por el usuario. Basado en esta entrada, se filtra, anonimiza y agrega la información de contexto en función de la entidad solicitante y sus privilegios de acceso.

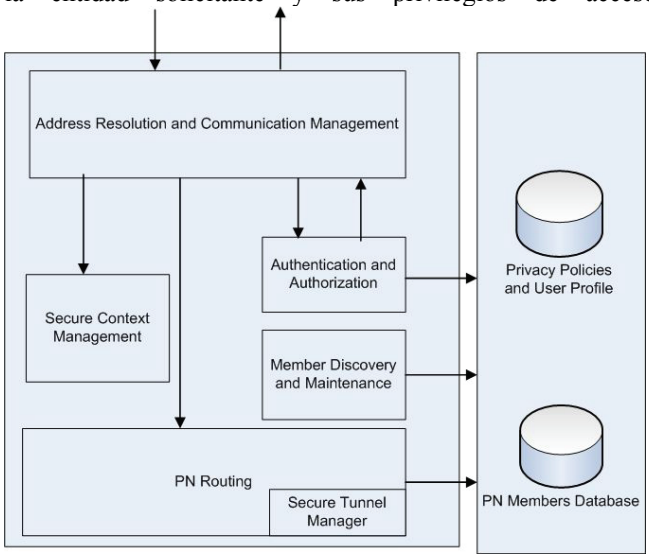


Fig. 2. Componentes software de la arquitectura de PN

En nuestro modelo de PN centralizado, el dispositivo maestro mantiene una posición distinguida disponiendo de una visión global de la red de dispositivos personales, y proporcionando interfaces externos a las redes de área extensa, así como una presencia previsiblemente continua en la red. Como resultado, la arquitectura completa de PN podría desplegarse en el dispositivo maestro que estaría encargado de las funciones de gestión de las comunicaciones en la red. Sin embargo, parte de los módulos de la arquitectura y sus funciones relacionadas podrían delegarse a otros dispositivos de la PN con capacidades computacionales y de comunicación adecuadas, así como una apropiada disponibilidad en la red. Por ejemplo, una estación base inalámbrica podría encargarse del módulo Secure Context Management o un dispositivo avanzado podría mantener PN Members Database o el repositorio de Privacy Policies and User Profile. Tal arquitectura de red distribuida podría definirse de forma estática, aunque nuevas propuestas podrían proporcionar mecanismos seguros para delegación dinámica de las funciones de PN en la red.

V. GESTIÓN SEGURA DE RFID Y NODOS SENSORES

La integración de la tecnología RFID en la PN requiere que se tengan en cuenta consideraciones específicas en cuanto a las funciones llevadas a cabo por los diferentes módulos de la arquitectura. A continuación, discutiremos cómo se puede lograr dicha integración, y los aspectos que han de ser contemplados en la arquitectura. En particular, analizaremos el descubrimiento y gestión de los objetos personales etiquetados, la comunicación segura con las tecnologías pervasivas y el cumplimiento de las políticas de seguridad y privacidad.

A. Descubrimiento y gestión de objetos RFID en la arquitectura

Como miembros de la PN, las etiquetas RFID personales deberían ser incluidas en la PN Members Database para saber qué etiquetas del contexto del usuario pertenecen a la red, y cómo autenticar y acceder a dichas etiquetas. La base de datos debe almacenar información de identificación como el código único de identificación (UID), esquemas de nombrado propios de PN, dirección IPv6 móvil como se propone en [18] o pseudónimos para esquemas de protección de la privacidad. De forma adicional, la base de datos debe mantener el material criptográfico de forma que los nodos autorizados puedan realizar con éxito los protocolos de autenticación mutua, acceder y actualizar sectores específicos de memoria o incluso desactivar las etiquetas.

En una situación ideal, el despliegue de una PN permitiría la selección de mecanismos de seguridad y protocolos de autenticación comunes para todas las etiquetas RFID empleadas en objetos personales. Sin embargo, las características hardware de las etiquetas RFID varían ampliamente de etiquetas básicas que se comportan como máquinas de estado a etiquetas avanzadas capaces de realizar criptografía de clave pública, coexistiendo así etiquetas basadas en diferentes ramas de RFID y protocolos de autenticación. Por lo tanto, la arquitectura PN (incluyendo la PN Members Database, o los módulos de Secure túnel manager y Authentication and Authorization) deberá estar preparada para gestionar los materiales criptográficos y protocolos requeridos por la etiquetas RFID adoptadas en la PN.

A medida que el usuario obtiene o despliega nuevos objetos equipados con RFID, las etiquetas han de ser reconocidas e incluidas en la esfera personal de forma segura. El proceso de incorporar una etiqueta RFID en la PN es gestionado por el módulo Member Discovery and Maintenance. En el caso de etiquetas RFID vírgenes, desplegadas específicamente para aplicaciones de la PN, deberá emplearse un protocolo de inicialización para intercambiar los materiales criptográficos adecuados con la etiqueta (ej. claves, pseudónimos, y/o certificados) y registrar la etiqueta en la PN Members Database. El mecanismo específico para identificar de forma segura la etiqueta y grabar los materiales criptográficos adecuados dependerá de los protocolos de autenticación seleccionados del amplio rango disponible en la literatura. El proceso de incorporación podría requerir interacción explícita del propietario de la PN con el nodo maestro (o algún otro miembro de la PN con capacidades de entrada/salida) con objeto de confirmar qué objetos etiquetados deberían ser aceptados como miembros de la red (ej. mediante selección por pantalla, o acercando físicamente el lector a la etiqueta) y participar en el establecimiento o generación de claves con un alto nivel de entropía (ej. moviendo un dispositivo equipado con acelerómetro o proporcionando una entrada por teclado).

Si la etiqueta es adoptada en la PN, podría requerirse el empleo de un protocolo de transferencia de propietario para obtener los permisos para gestionar de forma segura la etiqueta y regenerar su material criptográfico, pudiéndose adoptar y adaptar esquemas tales como [19][20] en el contexto de la PN. Sin embargo, nuevas propuestas de protocolos podrían tomar en consideración los servicios y recursos disponibles en la PN, la integración de ésta en redes de área extensa y la potencial interacción explícita del usuario con el fin de obtener la transferencia segura de etiquetas entre entidades distantes. En aquellos escenarios donde la etiqueta es necesaria aún en la aplicación original (ej. productos en garantía, o documentos de identificación privados o públicos), el objetivo del proceso de incorporación podría derivar en la compartición de la propiedad de forma segura [21].

B. Acceso seguro a sensores y nodos RFID

El módulo de Naming and Connection Management tiene una particular importancia en el acceso a las etiquetas RFID dado que permite emplear un pseudónimo o esquema de nombrado de la PN en lugar del identificador físico reconocido directamente por la etiqueta. Más aún, el módulo PN Routing libera al nodo solicitante de la necesidad de conocer el camino hasta el lector RFID en cuyo rango de lectura se encuentre la etiqueta. En nuestra visión, un miembro de la PN o un dispositivo remoto podría estar interesado en la información provista por una etiqueta RFID de dos formas posibles:

- *Acceso directo*: el dispositivo desea establecer una comunicación directa con la etiqueta con objeto de identificar el objeto, autenticarlo, actualizar su memoria o extraer información específica.
- *Conocimiento agregado*: el dispositivo requiere obtener consciencia sobre el contexto en el que el usuario se encuentra inmerso. Para su conveniencia, dicho conocimiento puede representarse mejor mediante la agregación de la información provista por los diferentes objetos personales etiquetados y sensores, en lugar de acceder directamente a cada nodo y componer el contexto por si mismo.

Nuestra arquitectura se encuentra preparada para gestionar ambos tipos de requisitos de interacción. En el caso de peticiones de acceso directo, el solicitante es requerido en primer lugar a autenticarse en la PN y obtener autorización para dicho acceso, tras esto, los módulos de nombrado y direccionamiento son responsables de resolver la identidad de la etiqueta, así como su ubicación actual en el seno de la PN y proporcionar una ruta adecuada para alcanzarlo. Si se requiere llevar a cabo una comunicación segura, el submódulo Secure Tunnel Manager participa en el establecimiento de un túnel desde el punto de acceso de la PN hasta el lector RFID próximo a la etiqueta solicitada o, en caso de que los nodos intermedios no sean capaces de participar en dicho túnel, crear enlaces seguros salto a salto dentro de la PN con objeto de maximizar la seguridad del canal extremo-a-extremo de acuerdo a los recursos computacionales y de comunicación de cada nodo en la ruta.

Por otra parte, si se desea hacer uso del conocimiento agregado, tras la autenticación y autorización inicial se emplea el módulo Secure Context Management para proporcionar la información de contexto requerida sobre los parámetros físicos del entorno y objetos personales cercanos. La información relativa al contexto es recopilada y procesada por el módulo como procedimientos en segundo plano que a su vez hacen uso de los servicios de nombrado y direccionamiento seguros proporcionados por la PN. Estos

procedimientos pueden ser iniciados directamente por una petición al módulo o tener lugar periódicamente, desacoplando las consultas de las comunicaciones seguras que realmente tienen lugar con los nodos sensores y RFIDs.

El mecanismo de acceso directo permite al nodo solicitante controlar la comunicación con la etiqueta final a bajo nivel, con objeto de leer o actualizar información específica. Este enfoque es muy adecuado, por ejemplo, en la interacción de entidades remotas con documentación personal RFID para autenticar al propietario de la PN e incluso obtener pruebas no repudiables de interacción con la PN.

Sin embargo, en este caso el control del cumplimiento de los requisitos de seguridad y las políticas de privacidad proporciona un bajo grado de granularidad. Las peticiones y comandos enviados a la etiqueta pueden ser bloqueados o enviados, pero, sin mayor filtrado y procesado de la información directa intercambiada, no es posible ajustar adecuadamente la granularidad de la información personal transmitida. En este caso, los mecanismos de autorización podrían ser reforzados incrementando los requisitos a cumplir por el nodo solicitante antes de que se le otorguen privilegios de acceso directo, dado que la información transmitida a bajo nivel podría potencialmente contener información sensible. La Sección 5.C proporciona una discusión más detallada sobre las alternativas posibles en el acceso directo.

Por otro lado, el acceso mediante conocimiento agregado proporciona un mayor control del cumplimiento de los requisitos de seguridad y privacidad deseados realizando un filtrado de los datos generados por las tecnologías sensibles al contexto, anonimizando los nodos origen de la información antes de que la información sea presentada a la entidad solicitante. Por lo tanto, este mecanismo permitiría reducir los requisitos sobre el nodo solicitante (ej. niveles de reputación o privilegios explícitos otorgados al usuario) para autorizar la interacción del solicitante con el módulo de Secure Context Management, responsabilizando a este último de asegurar la privacidad de los datos personales finalmente mostrados, a costa de reducir la flexibilidad del nodo solicitante en su interacción con las entidades finales de la red y requerir a la PN tareas adicionales de procesado. La Sección 5.D proporciona discusión adicional sobre el uso de las políticas de privacidad en la arquitectura de PN.

C. Alternativas en el acceso directo seguro a nodos RFID

En el enfoque de acceso directo, una entidad remota o local solicita establecer una comunicación con un nodo específico de la PN (estación base, nodo sensor, nodo RFID o dispositivo avanzado). Mientras el módulo de enrutamiento podría proporcionar una ruta directa a nodos PN que incorporen conectividad IP (incluyendo nodos sensores [22]), uno o más nodos proxy serán necesarios en caso de dispositivos basados en tecnologías de comunicación incompatibles o recursos computacionales y criptográficos extremadamente restringidos.

En particular, en el caso de etiquetas RFID personales que carecen de pila TCP/IP e incorporan recursos de comunicación, computación y memoria muy reducidos, el modo de acceso directo (para lectores RFID no locales) requiere del uso de nodos proxy que establezcan un puente entre las diferentes tecnologías de comunicación y reenvíen las consultas y comandos del nodo solicitante a la etiqueta final. Estos nodos pasarela deberían tener también un rol predominante para asegurar el cumplimiento de las políticas de seguridad y privacidad durante la comunicación con las etiquetas RFID.

En el enrutamiento seguro de las comunicaciones de acceso directo a las etiquetas RFID personales, se podrían adoptar las siguientes alternativas (véase Fig. 3):

• *Nodo proxy como repetidor de comandos*: tras la autenticación del nodo solicitante, resolución y localización de la etiqueta a acceder, se establece un túnel seguro desde el nodo remoto al lector RFID en rango de lectura. Uno o más nodos proxy participan en la ruta, sin embargo, los enlaces de comunicación seguros entre las entidades son empleados únicamente para retransmitir la comunicación entre las entidades extremo.

En este caso, la entidad remota debe conocer la tecnología RFID concreta de la etiqueta y enviar comandos que sean compatibles con dicha entidad final. El lector RFID o nodo inteligente más próximo a la etiqueta extrae los comandos recibidos a través del canal seguro y los envía a la etiqueta personal. Tras su respuesta, se encapsula el mensaje proveniente de la etiqueta RFID y se envía de vuelta al dispositivo remoto a través del túnel.

En este esquema, el nodo solicitante es además responsable de completar el protocolo de autenticación (mutua) con la etiqueta, debiendo conocer o ser capaz de obtener el material criptográfico necesario (ej. claves o certificados digitales). En caso de que la etiqueta adoptada en la PN pertenezca a una aplicación exterior (ej. etiquetas

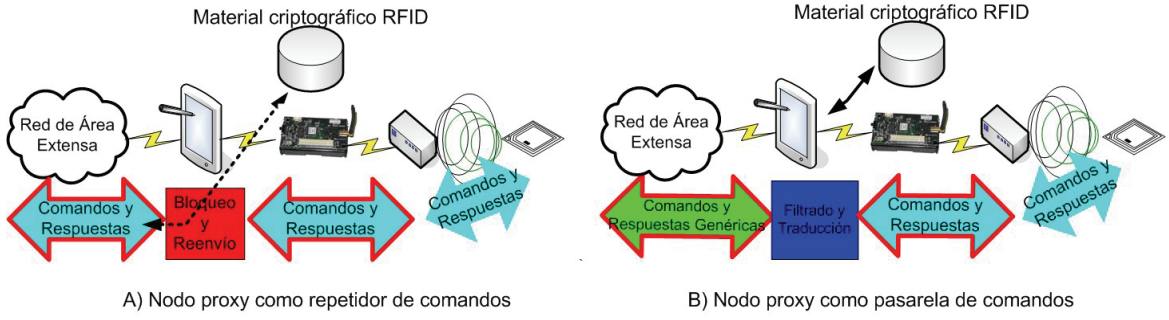


Fig. 3. Alternativas en el acceso directo seguro a nodos RFID

RFID en documentación personal privada o gubernamental), el solicitante podría obtener los materiales criptográficos de terceras partes (ej. un servidor de gestión de claves RFID [23]). En caso contrario, la PN podría directamente proporcionar tales materiales a la entidad solicitante una vez autenticado en la PN. En el último caso, la PN sería responsable de refrescar las claves involucradas por medio del módulo Member Discovery and Maintenance (ej. una vez la comunicación ha finalizado o de forma periódica) con objeto de prevenir futuras comunicaciones no autorizadas.

Dado que los comandos directos se envían a la etiqueta, la PN tiene un bajo control sobre la información personal o privada recuperada o modificada por el solicitante; sin embargo, podría otorgarse un rol adicional a un nodo proxy en la ruta (ej. el dispositivo maestro o el lector RFID) con objeto de analizar el flujo de tráfico y bloquear aquellos mensajes que no se ajusten a las políticas de seguridad.

- *Nodo proxy como pasarela de comandos*: un nodo pasarela en la ruta segura entre solicitante y etiqueta permite intermediar y traducir la comunicación entre ambas entidades. En este caso, el solicitante no requiere conocer el estándar RFID en que se basa la etiqueta, sus características de memoria, comandos compatibles o materiales criptográficos para llevar a cabo la autenticación (mutua) con la etiqueta personal. El solicitante enviaría sus comandos en base a un conjunto normalizado de operaciones para etiquetas RFID genéricas, mientras el nodo pasarela sería responsable de traducir los comandos específicos que serán enviados a la etiqueta RFID, así como interpretar y traducir las respuestas proporcionadas por la etiqueta.

En esta solución, el solicitante sólo necesita mantener las credenciales para autenticarse en la PN. La pasarela se encargaría de recopilar los materiales criptográficos necesarios a través de los mecanismos provistos por la PN y de llevar a cabo la autenticación (mutua) con la etiqueta personal, liberando por tanto a la entidad solicitante del proceso de autenticación doble y de la gestión de credenciales de los nodos individuales de la PN. La gestión segura de las etiquetas personales también se beneficiaría de esta solución dado que los materiales criptográficos necesarios en las comunicaciones internas a la red no se transmiten a entidades externas. Además, es posible obtener un mayor control durante la comunicación 'directa' con la etiqueta habilitando una supervisión más adecuada de las operaciones y datos transferidos (ej. comandos enviados o zonas de memoria) con objeto de comprobar la sensibilidad de los datos, privilegios del solicitante y asegurar el cumplimiento de las políticas de seguridad.

Aunque se mejora la seguridad y privacidad en la PN con esta solución, este enfoque podría no satisfacer aquellos escenarios en los que la entidad solicitante requiera un control más detallado del proceso de comunicación con la etiqueta personal (ej. durante la autenticación y validación de documentos personales RFID).

El rol de pasarela podría ser asumido tanto por el interfaz exterior de la PN (ej. el dispositivo maestro) como por el lector RFID o nodo inteligente que envía los comandos finales. El primero permitiría analizar y filtrar solicitudes inadecuadas en el punto mismo de entrada a la red, por lo tanto controlando la propagación de mensajes no deseados y previniendo posibles ataques potenciales (ej. mensajes mal formados) así como mejorando el uso de recursos de red (ej. batería y ancho de banda); mientras que el segundo enfoque permitiría concentrar las funcionales RFID (tales como formación de mensajes correctos y conocimiento de protocolos empleados) en las entidades de red directamente relacionadas.

D. Políticas de privacidad

Las políticas de privacidad tendrán un rol clave en la integración de la tecnología RFID en la PN. Estas políticas deberían de ser suficientemente flexibles para gestionar el ecosistema de elementos personales etiquetados, dado que estos podrán pertenecer a un amplio rango de categorías y tipos de objetos, así como la potencial diversidad de dispositivos remotos personales o profesionales y proveedores de servicios que pueden requerir acceso a las etiquetas personales y su información asociada. En este contexto, las políticas de privacidad deberían de proporcionar un mecanismo para representar qué categorías o etiquetas individuales mantienen información privada, cuáles no representan un riesgo a la privacidad, en qué condiciones es posible proporcionar acceso público o restringido a los actores seleccionados, e incluso qué datos personales deberían ser filtrados y desasociados de las fuentes donde se generaron antes de ser compartidos con entidades remotas.

En el caso de acceso directo a etiquetas individuales por parte de actores externos, se podrían emplear mecanismos de control de acceso (ej. ACL o RBAC) para definir a qué actores se les permite ejecutar qué comandos sobre qué etiquetas. Se podrían emplear parámetros adicionales relativos al contexto del usuario en las políticas de acceso (ej. localización, actividad actual u otras PN en el entorno).

En el caso de solicitudes de conocimiento agregado, la solución podría estar también basada en estas técnicas, pero en este caso, los elementos a acceder serían los tipos de conocimiento agregado que la PN es capaz de generar tras procesar y filtrar la información, en lugar de las instancias particulares de sensores y etiquetas RFID.

En la literatura, una solución relevante en esta dirección es el dispositivo RFID Guardian que mantiene una política de seguridad centralizada definiendo qué lectores RFID están autorizados a acceder a qué etiquetas en qué condiciones. El dispositivo logra su propósito intercediendo en el proceso de comunicación y llevando a cabo tácticas de simulación de etiquetas para bloquear lectores no autorizados. A pesar de ser un buen punto de comienzo, dicho dispositivo es un claro ejemplo de las soluciones de seguridad existentes en la literatura sobre RFID: no se integra en una PN y únicamente considera a las etiquetas RFID como una tecnología aislada, sin tener en consideración la información generada por otras tecnologías tales como BSNs ni evaluar el contexto del usuario. Además, se centra en el acceso local a las etiquetas RFID por lectores que se encuentran físicamente próximos al usuario, y no considera la integración de los dispositivos personales en redes de área extensa y las comunicaciones con PNs o proveedores de servicio remotos.

Nuestra visión integrada de la tecnología RFID en la PN tiene en consideración ambos aspectos y proporciona la base arquitectural para acceder de forma segura a estas

tecnologías también desde Internet, dejando la puerta abierta al desarrollo de políticas de privacidad específicas para este contexto.

VI. Conclusiones

La PN podría beneficiarse de la integración de los objetos personales habilitados con RFID, sin embargo, sus especiales características (ej. pasividad, no direccionamiento IP, reducidos recursos de comunicación y computación) y los riesgos potenciales a la seguridad y privacidad hacen que sea necesaria una arquitectura de PN preparada para soportar tales tecnologías pervasivas.

En este artículo, hemos definido las bases de una arquitectura de PN adecuada para dicho propósito. En nuestro modelo, las etiquetas personales deberían ser reconocidas como nodos de la PN, manteniendo tanto los materiales criptográficos relacionados como información sobre direccionamiento y metadatos, junto con posible información sensible que habilite el acceso e interacción segura con otros miembros y entidades externas. El despliegue e integración desde sus inicios de los objetos etiquetados en la PN permitiría la selección y definición de un conjunto común de protocolos de autenticación que estandaricen la gestión de etiquetas personales. Sin embargo, desde una perspectiva más práctica, la PN debería soportar la adopción de etiquetas heterogéneas e incorporar mecanismos para la transferencia y compartición segura de propiedad.

La arquitectura controla asimismo la autenticación y autorización de las entidades antes de otorgar privilegios en la red y habilitar las comunicaciones. En nuestro enfoque, las peticiones relativas a las tecnologías pervasivas restringidas en recursos (ej. RFID) pueden ser provistas de dos formas: acceso directo a los nodos finales e información agregada relativa al contexto. Como se ha discutido previamente, cada enfoque presenta sus propios beneficios y dificultades y deberían ser tratados de forma independiente, mediante la gestión segura del contexto y esquemas de acceso directo.

En el acceso directo, la PN ha de resolver y establecer una ruta segura para alcanzar el nodo final, incluyendo etiquetas RFID no basadas en IP. Como se ha mostrado, el rol de los nodos proxy como retransmisores de mensajes o nodos pasarela tiene un impacto sobre los requisitos aplicables al nodo solicitante y el cumplimiento de los requisitos de seguridad. Por último, las políticas de seguridad tienen un rol crucial en la PN y deben ser capaces de representar qué miembros de la PN y entidades externas son capaces de acceder a nodos de contexto o tipos de conocimiento en determinadas situaciones.

La investigación previa en aspectos tales como la integración de RFID y sensores, seguridad en RFID, transferencia segura de propietario o esquemas de control de acceso en RFID pueden ser adoptadas a este propósito proporcionando las bases para la realización de dicha arquitectura. Sin embargo, en lugar de analizarlas como tecnologías aisladas, una visión global de tales como componentes de la heterogénea PN, centrada en el usuario, e integrada en Internet, abre una puerta a la propuesta de soluciones específicamente diseñadas para los requisitos y recursos de este emergente paradigma.

Agradecimientos

Este trabajo ha sido financiado parcialmente por la Comunidad Europea a través de NESSoS (FP7-256890) y por el Ministerio de Ciencia a través de ARES (CSD2007-00004) y SPRINT (TIN2009-09237), el último cofinanciado por fondos FEDER. El primer autor ha sido financiado por el Ministerio de Educación a través del Programa F.P.U.

Referencias

1. International Telecommunication Union, ITU Internet Reports: The Internet of Things, November 2005.
2. Yum, D. H. et al., Distance Bounding Protocol for Mutual Authentication, IEEE Transactions on Wireless Communications, pp. 592-601, 2011
3. Piramuthu, S. RFID Mutual Authentication Protocols, Decision Support Systems, Elsevier (In press), 2010.
4. Armknecht, F. et al. Impossibility Results for RFID Privacy Notions, Transaction on Computational Science XI (6480), 2010, pp. 39-63.
5. Alomair, B. and Poovendran, R. Privacy versus Scalability in Radio Frequency Identification Systems, Computer Communication, 2010.
6. Peris-Lopez, P. et al., Cryptographic Puzzles and Distance-bounding Protocols: Practical Tools for RFID Security, 'IEEE International Conference on RFID 2010', Orlando, USA, 2010, pp. 45-52.
7. Kavun, E. B. and Yalcin, T., A Lightweight Implementation of Keccak Hash Function for Radio-Frequency Identification Applications, in RFIDSec'10, Springer, Istanbul, Turkey, 2010, pp. 258-269.
8. Anggraeni, P. N., Prasad, N. R. and Prasad, R. Secure personal network, Personal, Indoor and Mobile Radio Communications, IEEE 19th International Symposium on, 2008, pp. 1-5.
9. Ibrohimovna, M., et al., Secure and Dynamic Cooperation of Personal Networks in a Fednet, 6th IEEE CCNC 2009, pp. 8 -14.
10. Project IST-FP6-IP-027396, Magnet Beyond, http://magnet.aau.dk, last accessed: March 2011
11. Anggorjati, B., et al., RFID Added Value Sensing Capabilities: European Advances in Integrated RFID-WSN Middleware, 7th IEEE Conference on SECON 2010, pp. 1 -3.
12. Xiaoguang, Z. and Wei, L. The research of network architecture in warehouse management system based on RFID and WSN integration, IEEE International Conference on ICAL 2008, pp. 2556 -2560.
13. Tolentino, R. S., et al., Next Generation RFID-Based Medical Service Management System Architecture in Wireless Sensor Network, in Communication and Networking, Springer Berlin Heidelberg, 10.1007/978-3-642-17587-9_17, 2010, pp. 147-154.
14. Memsic WSN product family, http://www.memsic.com/products/wireless-sensor-networks.html, last accessed: March 2011
15. NFC-enabled Google Nexus S, http://www.google.es/nexus/#/tech-specs, last accessed: March 2011
16. NFC-enabled Nokia smartphones, http://www.nearfieldcommunicationsworld.com/2010/06/17/33966/all-new-nokia-smartphones-to-come-with-nfc-from-2011/, last accessed: March 2011
17. Yang, H., Yang, L. and Yang, S.-H., Hybrid Zigbee RFID sensor network for humanitarian logistics centre management, Journal of Network and Computer Applications (In Press), 2010.
18. Dominikus, S. and Schmidt, J.-M. Connecting Passive RFID Tags to the Internet of Things, 'Interconnecting Smart Objects with the Internet Workshop', Prague, Czech Republic, 2011.
19. Yu Ng, C., Susilo, W., Mu, Y. and Safavi-Naini, R. Practical RFID Ownership Transfer Scheme, Journal of Computer Security - Special Issue on RFID System Security, 2010.
20. Song, B. and Mitchell, C. J. Scalable RFID Security Protocols supporting Tag Ownership Transfer, Computer Communication, Elsevier, 2010.
21. Kapoor, G. et al. Single RFID Tag Ownership Transfer Protocols, IEEE Transactions on Systems, Man, and Cybernetics, 2011, pp. 1-10.
22. Mulligan, G., The 6LoWPAN architecture, 4th workshop on Embedded networked sensors, ACM, New York, NY, USA, 2007, pp. 78-82.
23. Najera, P., Moyano, F., Lopez, J., Security Mechanisms and Access Control Infrastructure for e-Passports and General Purpose e-Documents, Journal of Universal Computer Science, 15, 2009,970-991.

Extensión de los pseudoprimos de Mersenne para criptografía basada en curva elíptica en el MSP430

Leandro Marin
Departamento de Matemáticas Aplicadas,
Universidad de Murcia
Facultad de Informática - Murcia
leandro@um.es

Antonio J. Jara and Antonio F. Gomez Skarmeta
Departamento de Ingeniería de la Información y las Comunicaciones,
Universidad de Murcia
Facultad de Informática - Murcia
jara@um.es, skarmeta@um.es

***Resumen*—El soporte de la seguridad para dispositivos inteligentes es uno de los mayores retos para la nueva generación de Internet, donde a través de tecnologías como 6LoWPAN, dispositivos inteligentes son ya capaces de conectarse a Internet. La mayoría de estos dispositivos se basan en el procesador MSP430, de Texas Instrument, por tener un bajo consumo, tamaño y coste. El problema es que también presentan una baja capacidad de cálculo, memoria y autonomía, es por ello que en el caso de querer dar soporte a la seguridad es necesaria la implementación de una pila de seguridad optimizada. Nuestra investigación se ha centrado en la optimización de las primas criptográficas para soportar criptografía basada en clave pública con curva elíptica (ECC). La contribución presentada es la implementación de ECC con una serie de optimizaciones basadas en la operación de multiplicación de Montgomery con desplazamientos de bits, y con la definición de un conjunto especial de primos que aprovecha las propiedades del microprocesador para la implementación de la multiplicación basada en desplazamiento de bits. Los resultados obtenidos presentan un tiempo de 1.2665 segundos para la multiplicación escalar (operación usada para las operaciones criptográficas), cual presenta una notable mejoría frente a los $2,217$ segundos de la solución planteada por TinyECC.**

***Palabras Clave*—Seguridad; 6LoWPAN; ECC; pseudoprimos de Mersenne; Internet de las cosas; MSP430; cryptosuite ligero.**

I. Introducción

El soporte de la seguridad es uno de los temas más importantes de la actual sociedad de la información y las comunicaciones. La evolución, por un lado, de las tecnologías con el desarrollo de nuevos dispositivos personales como móviles, sistemas embebidos y ahora objetos inteligentes, y por otro lado, de los servicios que permiten la definición y gestión de toda nuestra información en la nube, servicios online y el acceso desde cualquier lugar a nuestra información, son las causas que están llevando a la redefinición de Internet a como fue concebido originalmente, a nueva versión que se conoce como el Internet del futuro, aunque cada vez es más del presente. El Internet del futuro define una extensión de la capacidad de direccionamiento del Internet con el protocolo IPv6, el cual ofrece una capacidad que permite concebir la conexión de todos los objetos y dispositivos que nos rodean, concibiendo lo que se ha denominado Internet de las cosas, o de los objetos inteligentes (IoT). El objetivo del IoT es permitir que, a partir de una comunicación y computación global, todos los sistemas puedan colaborar y hablar entre sí para alcanzar una mayor conciencia del estado y las necesidades reales en cada momento, permitiendo así alcanzar una mejor personalización y adaptación de los servicios.

Esos dispositivos inteligentes con capacidad de comunicación y computación es lo que se encuentra desde hace unos años en las redes personales de bajo consumo denominadas "Low-power Wireless Personal Area Networks" (LoWPANs). El grupo de trabajo 6LoWPAN del IETF ha definido en el RFC4944 [1], los estándares para soportar IPv6 sobre esas LoWPANs (6LoWPAN), con el objetivo de ofrecer la conectividad a Internet a esos dispositivos inteligentes.

6LoWPAN ofrece a las LoWPANs todas las ventajas de Internet, como son escalabilidad, flexibilidad, ubicuidad, que es abierto y que ya está muy extendido, por lo que también podría ser considerado que con 6LoWPAN también podríamos tener la ventaja de poder utilizar los protocolos ya desarrollados para Internet, como MIPv6 para movilidad, SNMP para gestión de red, IPSec para seguridad etc. Sin embargo, esos protocolos no se pueden aplicar sobre los dispositivos que se utilizan en las redes 6LoWPAN, debido a las restricciones que presentan en cuanto a capacidad, recursos, memoria, autonomía energética etc.

Por lo tanto, dadas las restricciones mencionadas, el soporte de la seguridad, algoritmos criptográficos y en definitiva cryptosuites supone un reto a resolver para los servicios que vayan a ser usados en redes 6LoWPAN. A día de hoy, la seguridad en las LoWPANs está basada en criptografía simétrica (SKC), la cual es directamente soportada por el hardware de los microprocesadores utilizados en la actualidad. SKC es adecuado para ofrecer soluciones de ámbito local, tal como estaban inicialmente pensadas estas redes de área personal. Pero con la introducción a Internet, dicha localidad desaparece y en contraposición es requerida una mayor escalabilidad. Por lo tanto, criptografía de clave pública (PKC) es necesaria ser considerada también para este tipo de redes.

Por esa razón, el objetivo de nuestra investigación es la optimización matemática de las primitivas criptográficas para criptografía de clave pública basada en curva elíptica (ECC), dichas primitivas criptográficas es lo que utilizan los protocolos y mecanismos de seguridad para llevar a cabo sus operaciones. Específicamente, nuestras optimizaciones van a estar enfocadas para los dispositivos 6LoWPAN que utilizan el microprocesador MSP430 de Texas Instrument, el cual ha sido elegido por ser uno de los más extendidos, y podemos encontrar en motas tales como Tmote Sky. Además, este microprocesador es también utilizado en tecnologías como identificación por radiofrecuencia (RFID) activa [2], y en nuevas soluciones hibridas como DASH7 [3].

Las optimizaciones matemáticas para primitivas crip-

tográficas han sido previamente llevadas a cabo para dispositivos en redes de sensores, donde en algunas de ellas también ha sido considerado el MSP430, es por ello que un estado del arte es presentado en la sección II. De todos los trabajos relacionados, TinyECC [11] es una de las implementaciones más relevantes de ECC para redes de sensores y dispositivos basados en el MSP430. TinyECC está basado en el algoritmo de Barret para reducción del modulo p, pero ha sido demostrado que la multiplicación de Montgomery es más adecuada para los dispositivos basados en el MSP430 que Barret [18]. El trabajo mencionado presentó una solución con resultados (tiempos de computación) cercanos a los obtenidos en TinyECC, además de demostrar que la técnica basada en desplazamiento de bits con la multiplicación de Montgomery era más eficiente que la multiplicación de 16-bits ofrecida por el compilador del MSP430.

En este artículo se presenta la evolución de la solución mencionada con la definición de un conjunto especial de primos que reducen a prácticamente la mitad los ciclos necesarios para la multiplicación de Montgomery basada en desplazamiento de bits. La selección de primos con propiedades especiales, es una técnica bien conocida y aplicada en estándares como la FIPS 186-, que recomienda usar curvas especificas sobre unos primos especiales, conocidos como los primos generalizados de Mersenne, para los que la aritmética modular se simplifica y hace estos más óptimos de computar, ver [21, Section D.2]. Los primos definidos por el FIPS186-3 son muy buenos para procesadores con operaciones de 32 o 64 bits, nuestro avance ha sido determinar los primos ideales para la multiplicación de Montgomery basada en desplazamiento de bits para procesadores de 16 bits. Además, de las dos optimizaciones mencionadas, con el objetivo de reducir aún más el tiempo final, han sido consideradas las optimizaciones generales definidas en [7].

En resumen, este trabajo propone una implementación de la multiplicación para ECC basada en desplazamiento de bits. Este enfoque presenta resultados interesantes para microprocesadores que no soportan la multiplicación de forma nativa como el MSP430, donde ahora mismo es simulada por el compilador con un bajo rendimiento, por lo que la alternativa de llevar a cabo la multiplicación con desplazamiento de bits obtiene una carga computacional mucho más reducida. Esta multiplicación es presentada en la sección IV. Además, han sido definidos unos primos especiales que extienden a los pseudoprimos de Mersenne, los cuales simplifican aún más la operación de multiplicación, los cuales son presentados en la sección V.

II. Trabajos relacionados

Antes del Internet de las cosas y 6LoWPAN, algunas soluciones de seguridad para las redes de sensores ya habían sido planteadas, así como para RFID activo [4].

Las soluciones normalmente definidas para redes de sensores estaban basadas en criptografía simétrica (SKC), pero tal como ya sido mencionado, SKC no es escalable para el Internet de las cosas, ya que requiere que el nodo origen y el nodo destino de la comunicación compartan una clave, la cual es usada para encriptar y desencriptar, de forma que un tercer nodo que no conozca dicha clave, no es capaz de entender el contenido del mensaje.

Nuestra investigación, se ha centrado en criptografía de clave pública o asimétrica, donde hay dos claves, una clave privada y otra pública. Las cuales tienen la propiedad, de que cualquier operación realizada con la clave privada puede ser invertida por la clave pública, y vice versa. Estas primitivas permiten la generación de mecanismos y protocolos, pues a partir de ellas podemos proveer confidencialidad, integridad, autenticidad, y la propiedad de no repudio.

La criptografía de clave pública fue considerada impracticable para dispositivos con las restricciones que presentan microprocesadores como el MSP430, pero eso fue hace tiempo. Algunos estudios fueron llevados a cabo para RSA [14], donde quedó demostrada su viabilidad pero aún presentaban resultados que para algunos casos de uso no eran viables, no obstante estos resultados fueron mejorados con la solución basada en curva elíptica (ECC). Por lo tanto, criptografía de clave pública para dispositivos inteligentes está principalmente enfocada en ECC, por ser más ligera que RSA, presentar unos menores requisitos en computación, memoria y por usar un menor tamaños de claves [15].

Los trabajos relacionados con la implementación eficiente de ECC se suelen centrar en la operación de multiplicación, ya que esta es la operación más cara tanto en RSA como en ECC. Hay varias implementaciones de RSA y ECC, por ejemplo para ECC puede definirse una implementación sobre un cuerpo $GF(2^m)$ o sobre un cuerpo $GF(p)$, este trabajo se centra en la aritmética modular, o sea $GF(p)$, ya que es válida tanto para RSA como ECC, y sobre todo porque tiene una misma naturaleza que las operaciones ofrecidas por los microprocesadores convencionales. Sin embargo, algunas implementaciones interesantes han sido definidas sobre $GF(2^m)$, por ejemplo [16] ha presentado la multiplicación sobre $GF(2^m)$ que es más rápida que sobre $GF(p)$, siempre y cuando se dispongan de multiplicaciones optimizadas para $GF(2^m)$, es por ello que también se han planteado algunos diseños de nuevos microprocesadores que implementase y trabajase de forma nativa sobre $GF(2^m)$ en [17]. Pero mientras esos microprocesadores no se llevan al mercado, sigue resultando más interesante centrarse en las soluciones sobre $GF(p)$.

Algunas de las implementaciones sobre aritmética modular ($GF(p)$) son, por un lado, para RSA con código ensamblador sobre un procesador de 8-bits ATmega128 en [14], donde presentan un algoritmo de multiplicación que explota las ventajas de hacer un escaneo de los productos en el algoritmo de multiplicación, con el objetivo de reducir el número de acceso a memoria. Por otro lado, para ECC encontramos TinyECC [11], [12] y NanoECC [13], los cuales implementan la generación de firma y verificación basada en ECC (ECDSA), encriptado y desencriptado (ECIES), y acuerdo de claves con Diffie-Hellman (ECDH). TinyECC implementa varias técnicas de optimización, como por ejemplo la reducción modular optimizada basada en los pseudoprimos de Mersenne, método de ventana deslizante, sistema de coordenadas Jacobiano, código ensamblado en línea y multiplicación hibrida, con el objetivo de alcanzar una alta eficiencia. Específicamente, los requisitos en memoria y capacidad computacional de esos algoritmos son, por ejemplo para ECDSA se requiren 19308 bytes de ROM y 1510 bytes RAM para el procesador MICAz,

generando la firma en 2 segundos, y verificándola en 2.43 segundos.

Este articulo presenta una mejora a esos tiempos encontrados con TinyECC, a partir de una implementación donde son tenidas muy presentes las características del hardware, y las instrucciones soportadas, con el objetivo de simplificar y optimizar la operación de la multiplicación. En las siguientes secciones del articulo se presenta nuestra aproximación y sus ventajas con respecto a lo encontrado en el estado del arte.

III. Optimizaciones matemáticas para mecanismos de seguridad basados en PKC

Varias optimizaciones para hacer más eficiente la implementación de ECC son encontradas en la literatura, ver sección II. Específicamente, las optimizaciones para reducir la complejidad de la multiplicación de Montgomery son encontrados en [6], [7].

Con el objetivo de optimizar nuestra implementación, en este trabajo se han seguido las siguientes optimizaciones:

- **Curvas elípticas, puntos y coordenadas** Una curva elíptica E sobre un cuerpo es una curva cubica no singular definida sobre el plano proyectivo. En concreto, el campo considerado es $\mathbb{Z}_p$ con p un primo de 160 bits, y E en la forma normal de Weierstrass, $E : y^2 = x^3 + ax + b$. Ha sido considerado el caso especial con $a = -3$, el cual reduce la cantidad de operaciones. Hay diferentes sistemas de coordenadas que pueden ser utilizados para representar los puntos. En este trabajo ha sido considerado un sistema de coordenadas mixto definido en [7], para el cual la multiplicación escalar se lleva a cabo en $1610.2M$, donde M es el tiempo de la multiplicación modular básica de 160 bits con modulo p, el objetivo por lo tanto de nuestra investigación es optimizar la multiplicación modular, ya que la multiplicación escalar se obtiene en función de ella.
- **Representación de las coordenadas** Tal como ha sido mencionado, el sistema de representación de los puntos de coordenadas utilizado es el definido por Montgomery. Esta representación es bastante utilizada en diferentes implementaciones y los detalles de la misma pueden ser encontrados en [8] y en [9].
- **Acumulador** Todas las operaciones deberían ser implementadas en ensamblador para optimizarlas al máximo. En la implementación propuesta, una de las decisiones tomadas fue usar 10 registros del MSP430 para almacenar el acumulador, el cual hace más simples las operaciones de suma y desplazamiento de bits que son utilizadas para la implementación de la multiplicación modular. Esta decisión hace que sólo queden libres 2 registros del MSP430, pero las ventajas obtenidas en el manejo del acumulador, hace que la decisión sea acertada.

Además, a estas optimizaciones inicialmente consideradas, nuestra principal contribución está basada en la implementación de la multiplicación modular de Montgomery, la cual se presenta en la sección IV, y la mejora de dicha implementación con los primos definidos que hacen aún más simple y óptima dicha implementación, estos primos son presentados en la sección V. Finalmente, la solución con todas las optimizaciones aplicadas es presentada en la sección VI.

IV. Optimización matemática basada en desplazamiento de bits para la multiplicación modular en lugar de la del procesador

Tal como ha sido mencionado son varias las ventajas de la representación de Montgomery para el cálculo de la multiplicación modular [18]. La implementación presentada para ECC, utiliza un entero (k) con un tamaño de 160-bits, i.e. $k = 160, R = 2^{160}$).

La representación de Montgomery ha resultado más interesante que otras implementaciones, tales como la reducción de Barret usada en TinyECC [11], por la razón de que con la representación de Montgomery, para representar los números a y b, los cuales van a ser multiplicados, se tiene aR y bR mod n (n es un primo para ECC). Las operaciones de suma y resta con esos números, no causan ningún problema, ya que R es un factor común. El problema aparece con la operación de multiplicación, cuando aR y bR son multiplicados, lo que se obtiene es abR^2, pero sin embargo lo que se busca es abR. Por lo tanto, es necesario reducir el resultado por el factor R. La gran ventaja que se obtiene con la representación de Montgomery es justamente evitar dicha reducción, ya que Montgomery ofrece llevar a cabo la reducción del factor R durante la multiplicación, alcanzado de esa manera una multiplicación mucho más efectiva, y como se ha mencionado, la operación de multiplicación modular es la que consume la mayor parte del tiempo, ya que esta se tiene que repetir miles de veces. Por esa razón, la multiplicación modular es lo que se va a optimizar y discutir más en detalle en las siguientes subsecciones, donde se presenta como se lleva a cabo con desplazamiento de bits y también como se llevaría a cabo con la multiplicación ofrecida por el compilador.

A. Implementación basada en desplazamiento de bits

Sean a y b dos enteros en representación de Montgomery. Entonces, tal como se ha mencionado, tenemos aR y bR mod n, con valor entre 0 y $n-1$. Estos valores son almacenados en representación binaria, por lo que $aR = \sum_i a_i 2^i$ y $bR = \sum_i b_i 2^i$.

Si se quiere calcular $(aR)(bR)R^{-1} = (ab)R$, es necesario llevar a cabo k desplazamientos a la derecha (con $k = 160$).

Nótese que el modulo n es impar, ya que es primo, por lo tanto cuando este es divido por 2 mod n, aparecen dos opciones: o el resultado es par y este puede ser directamente desplazado, o este es impar y necesita previamente ser añadido n, con el objetivo de que quede un 0 en el bit menos significativo, y entonces poder desplazarlo sin perder información del valor.

Para llevar a cabo el proceso de multiplicación es necesario acumular el resultado actual, para ello es definida la variable P, cuyos dígitos son $P = \sum_i P_i 2^i$. Cada uno de esos digitos B_i va a ser multiplicado por $\sum_i A_i 2^i$, y divido por 2. Como al inicio P es 0, si $B_i = 0$ para algunos valores iniciales pueden ser ignorados. Por lo tanto, la multiplicación puede empezar directamente con el digito en la posición i_0, tal que $B_{i_0} = 1$, y copiar el valor de A_i en P_i.

A partir de la posición i_0 se puede encontrar: $B_i = 0$ o 1. Por un lado, cuando $B_i = 0$, se tiene que dividir P por 2, y añadir n cuando P es par. Por otro lado, cuando $B_i = 1$, entonces es necesario añadir el valor de aR a P, antes de dividir por 2.

Para hacer la estimación del tiempo necesario, podemos considerar que las posibilidades de encontrar $B_i = 1$ o 0 son igual de probables, i.e. 0.5. Por lo tanto, para cada k bits de B_i, cuando este es 1 necesita llevar a cabo una suma de k bits (i.e. suma de a_i) y una división por 2 de P. Por otro lado, cuando este es 0 solamente es necesario un desplazamiento a la derecha. Por lo tanto, k divisiones por 2 y $k/2$ sumas. Teniendo en cuenta que para cada división por 2, va a ser necesario, con probabilidad 0.5, también llevar a cabo una suma del modulo n. En resumen, el tiempo total es:

$k(d + s/2) + (k/2)s = k(d + s)$, donde d es el tiempo para k desplazamientos a la derecha, y s es el tiempo para la suma de k bits.

Dado que el microprocesador MSP430 ofrece operaciones de 16-bits, las sumas y desplazamientos de bits se pueden hacer en bloques de 16-bits. Por lo tanto, α define el tiempo para la suma y desplazamientos de 16-bits (que serán normalmente entre 1 y 4 ciclos de CPU para el desplazamiento, y entre 1 y 6 ciclos de CPU para las sumas, esta diferencia de ciclos viene en función de si las variables están en registros o en memoria). El tiempo final es:

$$2\alpha k^2/16 = \alpha k^2/8.$$

El código de programa para la multiplicación modular con desplazamiento de bits es presentada en el algoritmo 1, el cual ha sido implementado en código ensamblador con las otras optimizaciones comentadas, este código ensamblador es basado en MSPGCC, la información sobre los ciclos para cada operación e instrucción es definida en [20].

Algorithm 1 Código basado en desplazamiento de bits

```
acumulador = 0
for i = 0 a k do
  if B_i igual 1 then
    acumulador = acumulador + A
  end if
  if acumulador es par then
    acumulador = (acumulador + p)/2
  else
    acumulador = acumulador/2
  end if
end for
```

B. Multiplicación modular ofrecida por el microprocesador-compilador

La instrucción del conjunto de instrucciones ofrecidas por el microprocesador-compilador que permite llevar a cabo la multiplicación de dos registros de 16 bits, y guardar el resultado de 32 bits, en dos registros de 16 bits es simulada en el MSP430, es décir no es directamente soportada por el hardware.

En [19, página. 478-480] se muestra el código que se lleva a cabo para soportar la multiplicación de 16-bits x 16-bits. Al tiempo de esta operación lo vamos a denominar μ, y α para el tiempo de las sumas y desplazamientos de 16-bits.

Sean los siguientes números, sobre los que hay que llevar a cabo la multplicación, $aR = \sum_j \overline{A}_j 2^{16j}$ y $bR = \sum_j \overline{B}_j 2^{16j}$. En este caso el valor de j está entre 0 y $k/16$, en lugar de entre 0 y k del Programa 1. Por lo tanto, para cada multiplicación de k bits, es necesario llevar a cabo $k/16$ multiplicaciones de 16-bits y $2k/16$ sumas, quedando un resultado de $k+16$ bits. El tiempo final es igual a: $(\mu + 2\alpha)k/16$.

Para cada paso de la multiplicación de aR por los dígitos de $\overline{B}_i$, es necesario añadir el resultado obtenido con el resultado acumulado, es decir una suma es necesaria (que se traduce en una suma de dos números de k bits, y otra suma de 16 bits, por lo que $\alpha(k + 1)/16$ sumas), entonces la operación necesita dividirla por 2^{16} mod n, esto es denominado δ para el tiempo empleado para la división por 2^{16}.

El tiempo total para cada uno de los bloques de 16 bits ($k/16$ bloques, $\overline{B}_i$) es $(\mu + 2\alpha)k/16 + \alpha(k + 1)/16 = \frac{\mu k+\alpha(3k+1)}{16}$, y además δ, por lo que el tiempo total es $\frac{\mu k^2+\alpha(3k+1)k}{256} + \frac{k\delta}{16}$.

La división de un número de $k+16$ bits por 2^{16} mod n es llevada a cabo añadiendo (o restando) n hasta que el resultado es múltiplo de 2^{16}. Sí el último digito de n en base 2^{16} es 1, entonces el proceso es simple, ya que el número de veces a restar n es indicado por el último digito del número que queremos dividir por 2^{16} mod n. Por lo tanto el tiempo total es:

$$\delta_t = \tfrac{(\mu+3\alpha)k+\alpha}{16}.$$

δ_t es el tiempo ideal, cuando n ha sido elegido de tal forma que su último digito vale 1, y por lo tanto la división se puede llevar a cabo de una forma simple. Por norma general, no se puede asumir que el último valor de n vaya a ser 1, por lo tanto esta estimación no es realista. Pero, se puede aplicar el algoritmo extendido de Euclides, para solucionar esto, a partir de pre-calcular el inverso modular multiplicativo del último digito de n mod 2^{16}, y este puede ser usado junto con el último digito de p. De esta manera, si que podemos obtener un tiempo más realista:

$$\delta = \tfrac{k}{16}(\mu + 3\alpha) + 2\mu + 2\alpha.$$

Esta estimación puede ser simplificada asumiendo que los términos que no tienen k no son relevantes para el tiempo total, de esa manera δ_t y δ son muy similares, en el orden de $\frac{k}{16}(\mu+3\alpha)$. Por lo tanto, a partir de esa expresión y reduciendo adicionalmente todos los término que no contienen k^2, el tiempo total para la operación de multiplicación modular con la instrucción ofrecida por el microprocesador es:

$$M \simeq \tfrac{\mu k^2}{256} + \tfrac{(\mu+3\alpha)k^2}{256} = \tfrac{(2\mu+3\alpha)k^2}{256}.$$

C. Comparación entre ambas implementaciones de la multiplicación modular

La comparativa entre ambas implementaciones, presenta que la versión basada en desplazamiento de bits es mejor que la basada en la instrucción ofrecida por el microprocesador-compilador, cuando se cumple que:

$$\tfrac{\alpha k^2}{8} < \tfrac{(2\mu+3\alpha)k^2}{256} \Rightarrow 32\alpha < 2\mu + 3\alpha \Rightarrow \tfrac{29}{2} < \tfrac{\mu}{\alpha}.$$

En resumen, cuando el número de ciclos para llevar a cabo la multiplicación con la instrucción ofrecida es más de 15 veces mayor que el número de ciclos para llevar a cabo la suma o el desplazamiento de bits es preferible la solución basada en desplazamiento de bits. Dado que la instrucción para la multiplicación ofrecida por el MSP430

requiere una gran cantidad de ciclos de reloj (150 ciclos en el MSP430), mientras que el desplazamiento de bit y la suma son soportadas en tan sólo entre 1 y 4 ciclos para el desplazamiento de bit, y entre 1 y 6 ciclos para la suma (como ya se ha mencionado esa diferencia de ciclos depende de si la variable está en memoria o en un registro, por ejemplo rrc R4, i.e. desplazamiento de bits para registros es sólo 1 ciclo, pero rrc 0(R1), el cual es el desplazamiento de bit en memoria para la dirección con valor $R1$, es de 4 ciclos, y de la misma manera para la suma).

Por lo tanto, para el caso del MSP430, la evaluación presenta que la solución basada en desplazamiento de bits es mejor que la solución basada con la instrucción ofrecida, ya que la relación de coste es menor de 15 veces, i.e. $\mu < 15\alpha$, ya que MSP430 tiene una relación μ/α entre 38 y 150.

V. Primos especiales para la multiplicación de Montgomery basada en desplazamiento de bits

Estos primos especiales son una extensión de los pseudoprimos de Mersenne, pero con las consideraciones de la implementación de la multiplicación de Montgomery con desplazamiento de bits. Estos primos son definidos como:

Definition Es dicho que p es un primo especial para la multiplicación basada en desplazamiento (de tipo α y λ) si p es un primo y existe u tal que $p = u \cdot 2^{\lambda-\alpha+1} - 1$ y $2^{\alpha-2} < u < 2^{\alpha-1}$.

Este trabajo, está enfocado en el caso de que $\alpha = 16$ y $\lambda = 160$, ya que es el caso que necesitamos, pues la implementación de ECC es para claves de 160-bits de longitud y para ser ejecutadas sobre el microprocesador MSP430, el cual es de 16-bits. El parámetro α determina la longitud de la palabra para las sumas y λ la longitud del número primo. Nótese, que si $2^{\alpha-2} < u < 2^{\alpha-1}$ entonces $2^{\alpha-2+\lambda-\alpha+1}-1 < p < 2^{\alpha-1+\lambda-\alpha+1} - 1$. Por lo tanto, $2^{\lambda-1} - 1 < p < 2^{\lambda} - 1$ y entonces p tiene λ-bits de longitud.

El número de primos especiales depende de λ y α. Por ejemplo, para $\alpha = 8$ y $\lambda = 160$ hay sólo uno (con $u = 100$). Para $\alpha = 16$ y $\lambda = 160$ hay 288 primos especiales.

Las operaciones básicas para estos primos especiales y para llevar a cabo la multiplicación de Montgomery son detalladas en las siguientes subsecciones.

A. Operaciones básicas

Siguiendo la misma metodología definida en la sección IV, y las optimizaciones definidas en la sección III. Esta implementación utiliza 10 registros para una variable de 160-bits denominada acumulador. Las operaciones con el acumulador son muy rápidas, y cuando este acumulador es combinado con estos primos especiales, el resultado es también muy rápido.

1) División por 2: La operación más básica de la multiplicación de Montgomery es $x \mapsto x \cdot 2^{-1}$ en $\mathbb{Z}_p$. Supongamos $x = x_0 + x_1 \cdot 2^{16} + \cdot + x_9 \cdot 2^{16 \cdot 9} < p$.

El algoritmo de esta operación es como sigue:

if x es par **then**
  $resultado$ es desplazar x una posición a la izquierda
else
  $resultado$ es $x + p$ desplazado una posición a la izquierda.
end if

Incluso cuando es usado el acumulador, se necesitan 3 ciclos para comprobar si x es impar y saltar dependiendo de ello. Una vez ha tomado esa decisión, entonces necesita 10 ciclos para desplazar el acumulador en el mejor de los casos y 30 ciclos para sumar un primo normal p y desplazarlo. Esto hace que el algoritmo requiera para un primo general entre 13 y 33 ciclos. El algoritmo para desplazamiento de bits es:

desplazar x
if no hay carry (i.e. si x era par) **then**
  saltar a (END), porque el resultado ya está en x.
end if
Ignora el carry y añade u a la palabra más significativa de x.
(END) El resultado está en x.

El resultado es directo cuando x es par. En caso de que x es impar, si x es desplazado (sin tener que haberse desplazado), entonces queda como $(x-1)/2$, y lo que se necesita es $(x+p)/2$. Pero nótese que si u es añadida a la palabra más significativa de x, entonces el resultado obtenido es justamente el resultado que se necesitaba:

$$(x-1)/2 + u \cdot 2^{\lambda-\alpha} = \frac{x - 1 + u \cdot 2^{\lambda-\alpha+1}}{2} = \frac{x+p}{2}$$

El número de ciclos para esta operación, con esta optimización es de 10 ciclos para desplazar x, 2 ciclos para el salto y en caso de que es necesario añadir u para rectificar, entonces son necesarios otros 2 ciclos. Pero nótese que cuando x está entre 0 y p, entonces $(x+p)/2$ está también entre 0 y p, por lo que no es necesario ninguna corrección adicional. El número total de ciclos es de 12 en el caso de que es par y de 14 en el caso de ser impar. Esta reducción es significativa porque esta operación se tiene que llevar a cabo λ veces para multiplicación de Montgomery. Desde que la probabilidad para que x sea impar es 0.5, el algoritmo sin optimizar daría un resultado de $13 \cdot 0.5 + 33 \cdot 0.5 = 23$ ciclos, y el basado en los primos especiales obtiene un tiempo medio de 13 ciclos, lo que supone una reducción del $43,47\%$ del número de ciclos requerido.

2) Suma y corrección modulo p: El algoritmo de multiplicación de Montgomery requiere añadir al segundo operando el acumulador en función de los bits del primer operando, con el objetivo de que el resultado esté entre 0 y $p-1$, cuando la suma obtiene un resultado sobre el valor de p, entonces es necesario aplicar una corrección, es decir restar p, para ofrecer el resultado dentro del rango.

Añadir una variable de λ-bits al acumulador requiere muchos ciclos, ya que la variable debería estar en memoria, en concreto esto requeriría de 10 sumas add(c).w mem,reg, donde para cada una de ellas son necesarios 3 ciclos, por lo que la suma completa son 30 ciclos.

Esta corrección para un primo general requiere comprar el resultado con el primo, para determinar si se ha pasado o no. Para llevar a cabo esta comparación, comprueba la palabra más significativa del acumulador con la más significativa del primo (2 ciclos), y entonces un salto (2 ciclos) es llevado a cabo a la parte del código oportuna en función del resultado de la comparación. Esta correción es más simple con los primos especiales propuestos, ya que:

Proposition 5.1: Sea p un primo especial $p = u \cdot 2^{\lambda-\alpha+1}-1$ y $a = \sum_{i=0}^{\lambda/\alpha-1} a_i 2^{\alpha i}$ el acumulador después de una suma parcial en la multiplicación de Montgomery, de x por y. Entonces:

1) a no puede ser exactamente p.
2) a no necesita corrección, si y sólo si, la palabra más significativa de a es menor que $2u$.

Proof:

1) En la multiplicación de Montgomery, el acumulador tiene productos parciales $h \cdot y$. Si $y \neq 0$, los productos parciales no pueden ser 0 (o p, que es el mimo elemento en $\mathbb{Z}_p$) y en caso de que y es 0, el resultado parcial debería ser siempre 0, en lugar de p.
2) El acumulador necesita corrección, sí y sólo si, $a \geq p$, está usando (1) es equivalente a $a > p$. Sea $k = \lambda/\alpha - 1$, entonces

$$a = \sum_{i=0}^{k} a_i 2^{\alpha i} = a_k 2^{\lambda-\alpha} + \sum_{i=0}^{k-1} a_i 2^{\alpha i}$$

$$p = 2u2^{\lambda-\alpha} - 1$$

El número $\sum_{i=0}^{k-1} a_k 2^{\alpha i}$ está entre 0 y $2^{\lambda-\alpha}-1$, ya que este está grabado con $k-1$ palabras. Por lo tanto,

$$a > p \Leftrightarrow a_k 2^{\lambda-\alpha} + \sum_{i=0}^{k-1} a_i 2^{\alpha i} > 2u2^{\lambda-\alpha} - 1$$

$$\Leftrightarrow (a_k - 2u)2^{\lambda-\alpha} > -\left(\sum_{i=0}^{k-1} a_i 2^{\alpha i} + 1\right)$$

El número $-\left(\sum_{i=0}^{k-1} a_i 2^{\alpha i} + 1\right)$ es negativo, por lo tanto si $a_k - 2u \geq 0$, tenemos que $a > p$. Conservativamente, Si $a_k - 2u < 0$ entonces $a_k - 2u \leq -1$ y por lo tanto

$$(a_k - 2u)2^{\lambda-\alpha} \leq -2^{\lambda-\alpha} \leq -\left(\sum_{i=0}^{k-1} a_i 2^{\alpha i} + 1\right).$$

Ha sido probado que el resultado necesita corrección sí $a > p$ y esto es equivalente a $a_k \geq 2u$. Entonces la corrección es requerida si y sólo sí $a_k < 2u$.

■

3) Suma con desplazamiento y correcciín modulo p: Siguiendo el algoritmo de la multiplicación de Montgomery, es necesario después de la suma, un desplazamiento para el siguiente bucle, por lo tanto es mejor considerar ambas operaciones juntas, pues se reducen los ciclos de una corrección parcial.

Supongamos que a es el acumulador y necesitamos calcular $(a + w)2^{-160}(p)$. Entonces el algoritmo es el siguiente:

añade w a a de derecha a izquierda
desplaza a de izquierda a derecha con carry, sin corrección previa
if no carry **then**
  saltar a (END)
end if
comparar u con la palabra más significativa de a.
if u es menor que este **then**
  añadir u a la palabra más significativa de a y salta a (END)
end if
resta u a la palabra más significativa de a y añade 1 al resultado final.

B. Implementación en ensamblador y tiempos de ejecución

El algoritmo previo y las optimizaciones son implementadas en el MSP430 con las siguientes convenciones: es usado el registro R5 para u, R4 para la dirección del operando, y 10 registros para el acumulador R6, R7, ..., R15.

DIV2

```
        RRC.w R6
        RRC.w R7
        RRC.w R8
        RRC.w R9
        RRC.w R10
        RRC.w R11
        RRC.w R12
        RRC.w R13
        RRC.w R14
        RRC.w R15
        JNC end
        ADD.w R5,R6
end:
```

En DIV2 es necesario que el flag del carry sea 0 antes de ejecutar el código, por lo que se usa la instrucción CLRC, para limpiar el bit del carry. El tiempo de ejecución son 12 ciclos si no se activa el carry, y 13 si no, con probabilidad 0.5. Por lo que la media es de 12.5 ciclos.

modADD R4

```
        ADD.w 18(R4),R15
        ADDC.w 16(R4),R14
        ADDC.w 14(R4),R13
        ADDC.w 12(R4),R12
        ADDC.w 10(R4),R11
        ADDC.w 8(R4),R10
        ADDC.w 6(R4),R9
        ADDC.w 4(R4),R8
        ADDC.w 2(R4),R7
        ADDC.w 0(R4),R6
        JC reqC
        CMP.w R6,2u
        JL end
reqC:   SUB.w 2u,R6
        ADD.w #1,R15
        JNC end
        ADD.w #1,R14
        ADDC.w #0,R13
        ADDC.w #0,R12
        ADDC.w #0,R11
        ADDC.w #0,R10
        ADDC.w #0,R9
        ADDC.w #0,R8
        ADDC.w #0,R7
        ADDC.w #0,R6
end:
```

En modADD R4 son necesarios 30 ciclos para la suma del acumulador, 2 ciclos para comprobar si no se ha desbordado. La probabilidad de corrección es en torno a 0.5. Si no hay corrección, este compara R6 con $2u$ (2 ciclos) y salta al final (2 ciclos). Esto supone $30+2+2+2 = 36$ ciclos. En otro caso, si es requerida la corrección, entonces son $30+2+1+2+\epsilon = 35+\epsilon$ ciclos, donde ϵ es la parte del código con baja probabilidad. El tiempo medio es de 36 ciclos.

modADD+DIV2 R4

```
        ADD.w 18(R4),R15
        ADDC.w 16(R4),R14
        ADDC.w 14(R4),R13
        ADDC.w 12(R4),R12
        ADDC.w 10(R4),R11
        ADDC.w 8(R4),R10
        ADDC.w 6(R4),R9
        ADDC.w 4(R4),R8
        ADDC.w 2(R4),R7
        ADDC.w 0(R4),R6
        RRC.w R6
        RRC.w R7
        RRC.w R8
        RRC.w R9
        RRC.w R10
        RRC.w R11
        RRC.w R12
        RRC.w R13
        RRC.w R14
        RRC.w R15
        JNC end
        CMP.w R5,R6
        JNC pre
        SUB.w R5,R6
        ADD.w #1,R15
        JNC end
        ADD.w #1,R14
        ADDC.w #0,R13
        ADDC.w #0,R12
        ADDC.w #0,R11
        ADDC.w #0,R10
        ADDC.w #0,R9
        ADDC.w #0,R8
        ADDC.w #0,R7
        ADDC.w #0,R6
        JMP end
pre:    ADD.w R5,R6
end:
```

En resumen, es necesario para la suma y el desplazamiento, por un lado cuando el acumulador es par requiere 42 ciclos, y por otro lado necesita 4 ciclos en caso de que sea necesario añadir p y $7+\epsilon$ en caso de que haya que restar p. Por lo tanto el coste final es $42 + 0.5(4 + 0.5(3 + \epsilon)) \equiv 45$ ciclos.

VI. Todas las optimizaciones unidas

La sección V ha descrito las ventajas para multiplicación de Montgomery con los primos definidos. Esta sección describe la unión de la implementación de la multiplicación basada en desplazamiento de bits presentada en la sección IV y los mencionados primos especiales. Finalmente, algunas optimizaciones adicionales han sido añadidas al proceso completo.

Las operaciones sobre curvas elípticas han sido estudiadas profundamente y optimizadas para diferentes arquitecturas. La operación básica para ECC es la multiplicación escalar, $n \times P$, donde P es un punto de la curva, y n es un número de gran tamaño. Esta operación es analizada en [7], donde concluye que el coste de la multiplicación escalar es $1610,12$ veces el coste de la multiplicación modular de números de 160-bits. Es por ello que hemos enfocado nuestra investigación en optimizar la multiplicación modular, tal como otras propuestas presentadas en los trabajos relacionados, donde TinyECC planteaba una multiplicación modular basada en la reducción de Barrett [11]. Para nuestro caso se eligió Montgomery por sus ventajas para el desplazamiento de bits. Esta multiplicación es explicada en detalle en [5].

Sea x e y, los operandos de ls multiplicación y p, un primo especial para un determinado u.

Una implementación básica de la multiplicación de Montgomery con los primos especiales require los siguientes pasos:

1) Cruzar bit a bit el operando x.
2) Si se encuentra un bit con valor 0, entonces el acumulador es rotado i.e. (operación DIV2).
3) En otro caso, si el valor es 1, entonces y es añadido al acumulador, y es también rotado (operación modADD+DIV2).

El coste de la operación DIV2 y modADD+DIV2 han sido ya presentados en la sección V.

Para cruzar x bit a bit han sido usados los siguientes registros:

- 10 registros para almacenar el acumulador, R6,R7,...,R15.
- 1 registro para u.
- 1 registro para leer la palabra que está siendo cruzada de x (x está compuesta de 10 palabras de 16 bits).
- Para acceder a una palabra de x, es almacenado en la pila la dirección de la última palabra que fue accedida de x, aumentada en dos unidades, para que apunte a la siguiente palabra de x.
- Además, también es almacenado en la pila la dirección de la primera palabra de x.
- Al principio del bucle para cruzar el operando x, es almacenado en el registro R5la dirección de la primera palabra x, y entonces se usa el siguiente código.

```
MOV.w 0(R5),R5
SETC
RRC R5
```

Este código introduce un bit de control, el cual permite rotar el registro hasta que el resultado es 0. Cuando el resultado es 0 , ese bit es ignorado, pues ese fue un marcador introducido por nosotros. Entonces pasa a la siguiente palabra de x. Esto hace que se ahorre un contador para saber cuándo se ha recorrido la palabra. Por ejemplo, el siguiente método permite encontrar el primer bit que es 1 de x.

```
      ADD.w #18,R5
      PUSH.w R5
      SUB.w #20,R5
      PUSH.w R5
next0: MOV.w 2(R1),R5
      CMP.w R5,0(R1)
      JZ end0
      SUB.w #2,2(R1)
      MOV.w 0(R5),R5
      SETC
Loop0: RRC R5
      JNC Loop0
      JZ next0
```

El salto aend0 es definido para devolver 0, ya que ha leído todo el operando y no ha encontrado ningún bit a 1. El número de ciclos es hasta 17 para saltar a otra palabra, dado que este es requerido para cada palabra, introduce $1,7$ ciclos por bit, quedando el tiempo como:

1) Sí el bit es 1, entonces necesita 3 ciclos para comprobarlo. Además, necesita comprobar el bit de control (2 ciclos) y llevar a cabo la suma con rotación (45 ciclos). En total 50 ciclos con probabilidad 0.5.
2) En otro caso, si el bit es 0, sólo require 3 ciclos para comprobarlo y 15.5 ciclos para rotarlo, en total 18.5 con probabilidad 0.5.
3) Por lo tanto, el número medio de ciclos por bit es de $25 + 9.25 = 34.25$.
4) Le añadimos los ciclos necesarios para saltar y comprobar el fin de palabra, quedando en 38 ciclos.
5) Si tenemos 160 bits. El resultado son 6080 ciclos, teniendo en cuenta que se necesitan algunos precalculos se estima un total de 6293 ciclos. Dado el reloj del MSP430 a 8 Mhz, el resultado es

$$1610 \cdot 6293/8 \cdot 10^6 = 1.2665 \ segundos.$$

Esta implementación ofrece un resultado mejor que otras implementaciones basadas en otros tipos de primos, como TinyECC que presenta un tiempo para encriptar o desencriptar, donde es usada multiplicación escalar de $3,271$ segundos y $2,217$ segundos respectivamente. Nuestra solución es entorno a $1,2665$ segundos.

VII. Resultados y evaluación

La verificación de la solución se ha llevado a cabo con nuestro propio simulador, el cual genera además del código para la Tmote Sky, un código en C basado en librería LiDIA [10], este ha sido evaluado sobre la mota Tmote Sky con el sistema operativo Contiki 2.4 OS, donde son implementadas las funciones mencionadas para llevar a cabo la multiplicación en código ensamblador en línea.

El algoritmo más rápido de ECC es basado en *Montgomery + método de ventana deslizante*, ver [7], [18]. Este algoritmo además ha sido optimizado para el procesador MSP430 con la implementación de la multiplicación modular de Motgomery, la cual la lleva a cabo para los primos espciales definidos en 6293 ciclos. Dado que la multiplicación escalar son 1610 veces la modular, el tiempo alcanzado ha sido para un MSP430 a 8Mhz de:

$$1610 \cdot 6293/8 \cdot 10^6 = 1.2665 \ segundos.$$

Para alcanzar esta solución, se han utilizado 10 registros del microprocesador para guardar una variable de 160 bits (el acumulador), donde son guardados los resultados parciales de

la multiplicación, con esta optimización se ha reducido casi un 40% el total del número de ciclos, dado que la operación rrc para desplazamiento de bits, y la operación add para la suma, emplean 1 ciclo y 3 ciclos respectivamente, en lugar de 4 y 6. Además, los bucles han sido desenrollados para optimizar aún más el código ensamblador. Finalmente, gracias a la introducción de los primos especiales se ha pasado de los resultados con la solución definida en la sección IV de entorno a 2,5 segundos [18], similar a TinyECC, a $1,2665$ segundos, lo que es un 50% menos que TinyECC y nuestro trabajo previo.

VIII. Conclusiones y trabajo futuro

La evolución de Internet, con la introducción de objetos inteligentes a la red, hace que se requiera el soporte de criptografía asimétrica para dispositivos con capacidad reducida y altas restricciones. Este trabajo ha presentado una implementación de criptografía basada en curva elíptica optimizada para el procesador MSP430 de Texas Instrument, el cual es uno de los más utilizados e integrados con los objetos inteligentes en tecnologías como 6LoWPAN, RFID activo y DASH7.

Las optimizaciones para ECC están basadas, por un lado en la implementación de la multiplicación modular de Montgomery con desplazamientos de bits, y por otro lado con la definición de unos primos especiales que explotan dicha implementación basada en desplazamientos de bits.

El resultado alcanzado con las optimizaciones mencionadas es de $1,2665$ segundos, para la multiplicación de Montgomery escalar. Este tiempo reduce casi al 50%, el tiempo alcanzado por TinyECC, la que ofrece un tiempo de $2,217$ segundos. Por lo tanto, puede ser concluido que los tiempos alcanzados demuestran la viabilidad de criptografía de clave publicaba basada en ECC para el Internet del futuro, donde son conectados a la red dispositivos inteligentes.

Señalar que la selección de primos concretos, no presenta ninguna debilidad o vulnerabilidad, esta es una técnica que se usa comúnmente por estándares como la FIPS 186-3, donde recomienda curvas optimizadas para procesadores de ordenadores convencionales con procesadores de 32 y 64 bits. Por lo que nuestro aporte ha sido determinar los primos óptimos para la multiplicación modular de Montgomery basada en desplazamiento de bits y el microprocesador de 16 bits MSP430.

El trabajo futuro, está enfocado en la reducción del número de sumas, accediendo en bloques de 4 en 4 bits, y el establecimiento de valores precalculados en tiempo de compilación.

IX. Agradecimientos

Este trabajo ha sido llevado a cabo por el grupo Matemáticas aplicadas y por el grupo de Excelencia de la Fundación Séneca Sistemas Inteligentes y Telemática bajo el "Programa de Ayuda a los Grupos de Excelencia 04552/GERM/06". Ambos grupos de la Universidad de Murcia.

Los proyectos involucrados en este trabajo son principalmente el proyecto Modelización Hidrológica en Zonas Semiáridas (MHZS) de la Comunidad Autónoma de la Región de Murcia bajo el programa de Ciencia y Tecnología PCTRM 07/10, el proyecto de la Fundación Séneca 12006/PI/09, y el proyecto del Ministerio de Ciencia e Innovación MTM2009-11696.

Finalmente, también agradecer al Ministerio de Educación y Ciencia por la beca predoctoral de Formación de Personal Universitario (FPU) con referencia AP2009-3981.

Referencias

[1] Montenegro, G.; Kushalnagar, N.; Hui, J.; Culler, D.; *Transmission of IPv6 Packets over IEEE 802.15.4 Networks*, RFC4944, 2007.

[2] Zampolli, S.; Elmi, I.; Cozzani, E.; Cardinali, G.; Scorzoni, A.; Cicioni, M.; Marco, S.; Palacio, F.; Gomez-Cama, J.; Sayhan, I.; Becker, T.; *Ultra-low-power components for an RFID Tag with physical and chemical sensors*, Journal of Microsystem Technologies, Springer, pp. 581-588, Vol. 14, No. 4, 2008.

[3] Norair, J.P.; *Opentag webinar, DASH7: ultra-low power wireless data technology*, 2009.

[4] Engels, D; Fan, X.; Gong, G.; Honggang, H.; Smith, E.; *Ultra-Lightweight Cryptography for Resource-Constrained Devices*, Financial Cryptography and Data Security, Lecture Notes in Computer Science, Springer, pp. 3-18, Vol. 6054, 2010.

[5] Cohen, H.; *A Course in Computational Algebraic Number Theory*, 3rd ed. GTM 138, Springer, 1996.

[6] Cohen, H.; Miyaji, A.; Ono, T.; *Efficient Elliptic Curve Exponentiation*. Advances in Cryptology -Proceedings of ICICS'97, (LNCS 1334), Springer-Verlag 1997.

[7] Cohen, H.; Miyaji, A.; Ono, T.; *Efficient Elliptic Curve Exponentiation Using Mixed Coordinates*. Advances in Cryptology, ASIACRYPT'98 (LNCS 1514) 51-65, Springer-Verlag, 1998.

[8] Hankerson, D.; Menezes, A.; Vanstone, S.; *Guide to Elliptic Curve Cryptography*, Springer-Verlag, New York, 2004.

[9] Montgomery, P.; *Modular Multiplication Without Trial Division*, Math. Computation, vol. 44, pp. 519–521, 1985.

[10] Hamdy, S.; *LiDIA. A library for computational number theory. Reference Manual*. Edition 2.1.1, 2004.

[11] Liu, A.; Ning, P.; *TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks*, 7th International Conference on Information Processing in Sensor Networks, SPOTS Track, USA, pp. 245-256, 2008.

[12] Seo, S.C.; Han, D.G.; Kim, H.C.; Hong, S.; *TinyECCK: Efficient Elliptic Curve Cryptography Implementation over GF(2m) on 8-bit MICAz Mote.* IEICE Transactions on Info and Systems E91-D(5), 1338-1347, 2008.

[13] Szczechowiak, P; Oliveira, L.B.; Scott, M.; Collier, M.; Dahab, R.; *NanoECC: Testing the Limits of Elliptic Curve Crytography in Sensor Networks*, Dublin City University, Ireland; UNICAMP, Brasil, 2008.

[14] Gura, N.; Patel, A.; Wander, A.; Eberle, H.; Shantz, S.C.; *Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs*. Workshop on Cryptographic Hardware and Embedded Systems, 2004.

[15] Hitchcock, Y.; Dawson, E.; Clark, A.; Montague, P.; *Implementing an efficient elliptic curve cryptosystem over GF(p) on a smart card*, ANZIAM Journal, 2003.

[16] Uhsadel, L.; Poschmann, A.; Paar, C.; *Enabling Full-Size Public-Key Algorithms on 8-bit Sensor Nodes*. European Workshop on Security and Privacy in Ad hoc and Sensor Networks, 2007.

[17] Hodjat, A.; Batina, L.; Hwang, D.; Verbauwhede, I.; *HW/SW Co-Design of a Hyperelliptic Curve Cryptosystem using a Microcode Instruction Set Coprocessor* Integration, VLSI Journal 40(1), pp.45-51, 2007.

[18] Ayuso, J.; Marin, L.; Jara, A.; Skarmeta, A.F.G.; "Optimization of Public Key Cryptography (RSA and ECC) for 8-bits Devices based on 6LoWPAN", 1st International Workshop on the Security of the Internet of Things, Tokyo, Japan, 2010.

[19] Bierl, L.; *MSP430 Family Mixed-Signal Microcontroller Application Reports*, http://focus.ti.com.cn/cn/lit/an/slaa024/slaa024.pdf, pp. 478-480, 2000.

[20] MSPGCC project, Instructions set: http://mspgcc.sourceforge.net/manual/x223.html, Cycles for each instruction: http://mspgcc.sourceforge.net/assemble.html, 2011.

[21] Locke, G.; Gallagher, P; "FIPS PUB 186-3: Digital Signature Standard (DSS)", National Institute of Standards and Technology, 2009.

Federando Autenticación y Autorización en Servicios Kerberos mediante GSS-API y EAP

Alejandro Pérez, Fernando Pereñíguez, Rafael Marín-López, Gabriel López
Departamento de Ingeniería de la Información y las Comunicaciones,
Universidad de Murcia
Facultad de Informática. Campus Universitario de Espinardo. 30100. Murcia.
{alex, pereniguez, rafa, gabilm}@um.es

***Resumen*—Kerberos es un protocolo para autenticación y distribución de claves que se está convirtiendo en uno de los estándares más utilizados para el acceso a servicios. Sin embargo, aunque los proveedores de servicios usan este protocolo para controlar a sus propios suscriptores, no existe un gran despliegue de infraestructuras Kerberos para manejar usuarios provenientes de dominios externos. Este tipo de operación se soporta por las denominadas federaciones de identidad, que prefieren el uso de infraestructuras AAA. La falta de una correcta integración entre estas infraestructuras AAA con Kerberos hace que el acceso a los servicios esté limitado únicamente para los suscriptores del mismo dominio. Para evitar esta limitación, se propone diseñar una arquitectura que permita relacionar la autenticación y autorización de los usuarios realizada a través de la infraestructura AAA con la distribución de *tickets* Kerberos en el dominio del proveedor de servicio, además de una gestión avanzada de atributos de usuario mediante tecnologías como SAML y XACML.**

***Palabras Clave*—AAA, autenticación, autorización, EAP, Kerberos, SAML, XACML**

I. Introducción

El gran desarrollo de las telecomunicaciones ha fomentado el establecimiento de acuerdos empresariales entre proveedores de servicios mediante las denominadas *federaciones*, con el objetivo de incrementar los beneficios sobre los servicios de red desplegados. De hecho, dichas federaciones permiten a un suscriptor de un proveedor de servicios acceder a los servicios de otros proveedores afiliados a la federación.

En general, un suscriptor puede acceder a los servicios ofrecidos dentro de una federación tras ejecutar un único proceso de autenticación. Esto es comúnmente conocido como *single sign-on* (SSO) [1]. Kerberos [2] se está convirtiendo hoy en día en uno de los estándares de autenticación y distribución de claves que proporciona SSO con un mayor despliegue [3]. De hecho, diversos sistemas operativos y aplicaciones de red (FTP, SSH, HTTP...) ya integran Kerberos para realizar el control de acceso. Sin embargo, aunque los proveedores de servicio usan este protocolo para controlar el acceso de sus propios suscriptores en su dominio (*single-realm*), es inusual ver despliegues de infraestructuras Kerberos para gestionar usuarios que provengan de diferentes dominios en la federación (*cross-realm*).

En su lugar, los proveedores de servicio suelen desplegar las denominadas infraestructuras AAA (*Authentication, Authorization, and Accounting* [4]) para controlar estas operaciones en redes federadas. Es más, EAP (*Extensible Authentication Protocol* [5]) se ha convertido en uno de los candidatos más prometedores para proporcionar una autenticación flexible y una fácil integración con las infraestructuras AAA subyacentes. Por ejemplo, *eduroam* [6] despliega una infraestructura basada en EAP y AAA que es usada por una comunidad investigadora y educativa internacional formada por más de quinientas instituciones.

La ausencia de una correcta integración entre las infraestructuras AAA y Kerberos limita el acceso a los servicios a sólo suscriptores (en adelante *usuarios*) que pertenecen al dominio del proveedor de servicios, imposibilitando a cualquier usuario de la federación acceder a un servicio sin importar su dominio de origen. Esto ha motivado un incipiente esfuerzo de los organismos de estandarización [7] con el objetivo de integrar infraestructuras AAA con el control de acceso a los servicios en federaciones.

El objetivo principal de este trabajo es solventar estos problemas mediante la definición de una arquitectura que permita integrar las infraestructuras AAA existentes en la federación (como puede ser *eduroam*) con el control de acceso a servicios llevado a cabo mediante el uso de Kerberos en el dominio del proveedor de servicios. Concretamente, se ha diseñado un nuevo mecanismo de *pre-authenticación* Kerberos basado en el uso de a) GSS-API [8] como interfaz genérica de autenticación y b) el uso del *Extensible Authentication Protocol* (EAP) [5] como protocolo específico de autenticación, para permitir la autenticación de usuarios pertenecientes a cualquier dominio de la federación, haciendo uso de las credenciales que usan en su dominio de origen, y utilizando para ello la infraestructura AAA que interconecta el dominio visitado con el dominio origen del usuario. De este modo se evita el despliegue de una infraestructura Kerberos *cross-realm*. Además, considerando que Kerberos carece de gestión de la autorización, esta propuesta realiza una gestión avanzada de la misma mediante la integración con los estándares SAML y XACML.

El resto de este artículo se estructura de la siguiente manera. La sección II proporciona un breve resumen de las principales tecnologías relacionadas. En la sección III se presenta la arquitectura propuesta y la sección IV detalla la fase de autenticación de usuario. En la sección V se describe cómo se podría integrar una autorización avanzada en esta arquitectura y la sección VI discute algunos aspectos que deben ser tenidos en cuenta para el despliegue de nuestra solución. La sección VII describe el trabajo relacionado y, finalmente, la sección VIII presenta algunas conclusiones y vías futuras.

ISBN: 978-84-694-5948-5

II. Estado del arte

A. Kerberos

Kerberos [2] es un protocolo de autenticación y gestión de claves basado en claves simétricas donde participan tres entidades: un *cliente*, un *servidor de aplicación* proporcionando un servicio, y un *servidor de distribución de claves* (KDC - *Key Distribution Center*). El KDC se compone a su vez de dos servidores especiales: un *servidor de autenticación* (AS - *Authentication Server*) y un *servidor de tickets* (TGS - *Ticket Granting Server*). Se requiere que las siguientes relaciones de seguridad estén pre-establecidas para su correcto funcionamiento:

AS ⇔ TGS, cliente ⇔ AS y serv. de aplicación ⇔ TGS

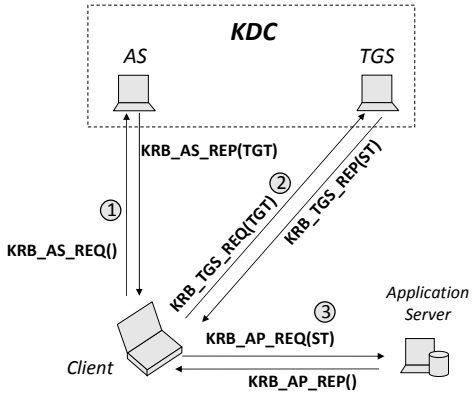


Fig. 1. Intercambio típico de mensajes Kerberos

Un intercambio típico de mensajes Kerberos comienza cuando el cliente solicita un *Ticket Granting Ticket* (TGT) del AS, a través de un mensaje *KRB_AS_REQ (1)*. El TGT es un tipo de ticket especial usado para generar otros tickets. El AS genera una clave de sesión con el TGS que se incluye en el TGT y se envía al cliente en el mensaje *KRB_AS_REP*. Kerberos implementa un mecanismo llamado *pre-autenticación* que permite al KDC autenticar al cliente antes de proporcionarle el TGT. Cuando se utilizan mecanismos de autenticación que requieren más de un intercambio (*multi-roundtrip*), el cliente y el AS ejecutan varios intercambios *KRB_AS_REQ/KRB_ERROR*. Cuando el proceso de autenticación acaba con éxito, el AS responde con un *KRB_AS_REP* final que contiene el TGT solicitado.

Una vez que el cliente posee el TGT, puede solicitar tickets de servicio (ST - *Service Ticket*) al TGS *(2)*, para acceder a un servicio específico. Con este propósito, el cliente envía un *KRB_TGS_REQ* protegido con la clave de sesión obtenida en el mensaje *KRB_AS_REP*. Cuando el TGS valida el TGT, genera una nueva clave de sesión que se incluye tanto en el ST como en el mensaje *KRB_TGS_REP*. Finalmente, el cliente se autentica con el servicio *(3)* mediante el envío del ST al servidor de aplicación, incluido en un mensaje *KRB_AP_REQ*. De forma opcional, se puede hacer uso de un mensaje *KRB_AP_REP* si el cliente necesita autenticar al servidor de aplicación.

Kerberos soporta un modo de operación denominado *cross-realm* que permite que un cliente se autentique contra servicios localizados en dominios externos. Gracias a relaciones de confianza pre-establecidas entre los TGS/KDCs de los diferentes dominios, el cliente sigue el camino desde el origen hasta el TGS visitado adquiriendo los denominados *cross-realm TGTs*. Finalmente, el cliente contacta con el TGS/KDC visitado y obtiene el ST que será presentado al servicio.

B. Generic Security Service Application Program Interface (GSS-API)

GSS-API [8] es un marco genérico que proporciona servicios de seguridad tales como autenticación, integridad y confidencialidad. Las aplicaciones de red que necesitan proteger sus comunicaciones pueden emplear los diferentes servicios ofrecidos por la GSS-API y, aún así, seguir siendo independientes del mecanismo concreto de seguridad. La implementación de la GSS-API más extendida es la proporcionada por el protocolo Kerberos [9], aunque otros mecanismos como EAP también han sido propuestos [10].

GSS-API permite que una entidad (denominada *initiator*) establezca un contexto de seguridad con otra entidad (denominada *acceptor)*. La negociación de dicho contexto comienza cuando el *initiator* llama a la función *GSS_Init_sec_context()*, que indica si son necesarias más interacciones para completar el establecimiento del contexto (GSS_S_CONTINUE_NEEDED) y devuelve un token que debe ser enviado al *acceptor*. El *acceptor* pasa el token recibido a la función *GSS_Accept_sec_context()*. Si se asume un protocolo de autenticación que requiere una única iteración, la función indica un estado finalizado *GSS_S_COMPLETE*, y devuelve un token para que sea enviado al *initiator*. Finalmente, el *initiator* invoca otra vez a la función *GSS_Init_sec_context()* (pasándole el token recibido), que devuelve el estado del establecimiento del contexto.

Una vez que el contexto de seguridad se ha establecido, tanto *initiator* como *acceptor* pueden realizar protección de mensajes de aplicación mediante las funciones *GSS_GetMIC/GSS_VerifyMIC* (para autenticación y protección de la integridad) y *GSS_Wrap/GSS_Unwrap* (para confidencialidad).

C. EAP - Extensible Authentication Protocol

EAP [5] (*Extensible Authentication Protocol*) es un protocolo que permite el uso de diferentes mecanismos de autenticación, denominados *métodos EAP* (por ejemplo, basados en claves simétricas o en certificados). Los métodos EAP se ejecutan entre un *EAP peer* y un *EAP server*, a través de un *EAP authenticator*. Desde el punto de vista de la seguridad, el *authenticator* no forma parte de la autenticación mutua y sólo reenvía paquetes EAP entre *peer* y *server*.

Para llevar a cabo una autenticación EAP, *authenticator* generalmente comienza el proceso solicitando la identidad de *peer* mediante un mensaje *EAP Request/Identity*. El *peer* responde al mismo mediante un *EAP Response/Identity* que contiene su identidad. Con esta información, el *EAP server* será capaz de seleccionar el método de autenticación a utilizar. La ejecución del método puede requerir varios intercambios *EAP Request / EAP Response* entre *peer* y *server*.

La configuración más típica es la denominada *pass-through*, donde el *EAP server* se sitúa en un servidor AAA, y el *EAP authenticator* se implementa en un nodo separado.

Mientras que para transportar los mensajes entre el *peer* y el *authenticator* se utiliza un *EAP lower-layer* (ej. 802.1X [11]), la comunicación entre *authenticator* y *server* se realiza utilizando un protocolo AAA como RADIUS [12] o Diameter [13].

Es importante mencionar que ciertos métodos EAP [14] son capaces de generar material criptográfico. Según [15], la *Master Session Key* (MSK) y la *Extended Master Session Key* (EMSK) se exportan tras una autenticación EAP exitosa. Mientras que la primera se envía al *authenticator* para establecer una asociación de seguridad con el *peer*, la segunda generalmente se usa para la derivación de claves adicionales.

D. SAML/XACML

SAML [16] es una especificación de seguridad basada en XML para representar e intercambiar información de autenticación y autorización. SAML define *Assertions* como el formato de las sentencias de seguridad. Generalmente, cada *Assertion* incluye la identidad del emisor (*Issuer*), la identidad del sujeto a que se refiere (*Subject*), y el conjunto de condiciones bajo las que la sentencia es válida. Además, se incluye distinta información en función del tipo de sentencia de que se trate. Hay dos tipos principales de sentencias: *AuthnStatement*, que indica que un sujeto fue autenticado previamente por una autoridad; y *AttributeStatement*, que indica que un sujeto tiene asociados una serie de atributos.

SAML establece que las entidades intercambian sentencias mediante el uso de mensajes *SAMLRequest* y *SAMLResponse*, pero no define cómo se representa y gestiona la información de autorización o políticas por los proveedores de servicio. Para este propósito se utiliza XACML.

XACML [17] incluye dos especificaciones diferentes: la primera es un lenguaje de políticas de control de acceso, que definen el conjunto de sujetos que pueden realizar determinadas acciones sobre un conjunto de recursos; la segunda es un formato para representar solicitudes y respuestas de control de acceso. XACML también define una arquitectura extensible que puede adaptarse a diferentes escenarios y tipos de políticas. En ella destacan dos elementos importantes: el *Policy Enforcement Point* (PEP), que controla el acceso al recurso y lleva a cabo la decisión tomada; y el *Policy Decision Point* (PDP), que evalúa las políticas y toma una decisión basada en la información disponible.

III. Arquitectura propuesta

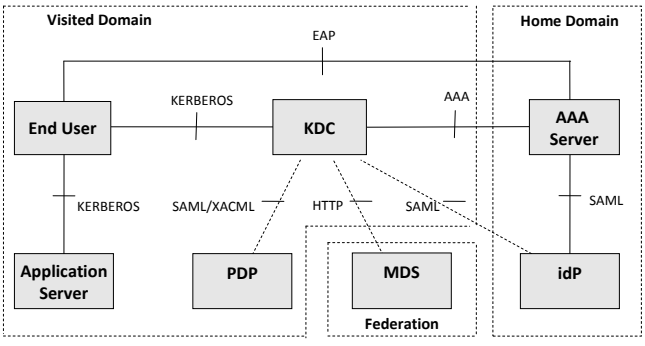


Fig. 2. Arquitectura propuesta

A. Resumen

La arquitectura propuesta, mostrada en la Fig. 2, asume que un usuario desea obtener acceso a un servicio de aplicación de un proveedor de servicio que requiere el uso de Kerberos. Por tanto, el proveedor de servicio necesitará desplegar un KDC en su dominio. El dominio del proveedor de servicio puede ser tanto el dominio origen del usuario (*home domain*) como uno diferente dentro de la federación (*visited domain*).

Tal y como se explicó en la sección II-A, Kerberos permite pre-autenticar al usuario antes de proporcionarle el TGT inicial requerido para acceder a los servicios de aplicación. Esta propuesta hace uso de EAP como protocolo de autenticación para la pre-autenticación Kerberos. Dado que EAP es un protocolo estándar que está bien integrado con infraestructuras AAA, se permite que los usuarios (incluso aquellos provenientes de otros dominios dentro de la federación) se autentiquen usando sus credenciales EAP (compartidas con su dominio origen), a través de la infraestructura AAA. De este modo, sólo el dominio del proveedor de servicio necesita desplegar un KDC, que actuará como *EAP authenticator* en la pre-autenticación Kerberos. Por tanto, no es necesario desplegar una infraestructura Kerberos *cross-realm* en la federación, lo que supone una importante ventaja, como se analizará en la sección VI.

Si el proceso de pre-autenticación basado en EAP termina con éxito, el usuario obtiene un TGT válido para el dominio del proveedor de servicios. De forma más específica, por medio de una *única* autenticación EAP, es posible derivar una *clave compartida* entre el usuario y el KDC requerida para obtener el TGT inicial. Con ese TGT, el usuario puede solicitar tickets de servicio sin necesidad de realizar procesos adicionales de pre-autenticación, incorporando los beneficios de la operación de *Single Sign-On* (SSO) ofrecida por Kerberos.

Dado que ni Kerberos ni EAP gestionan la autorización de usuarios, esta arquitectura incorpora soporte para que los proveedores de servicio gestionen atributos de usuario y tomen decisiones de control de acceso. De esta forma, el acceso a los servicios no se basa únicamente en si el usuario ha sido autenticado o no, sino también en información adicional proporcionada por estos atributos o variables de entorno. El KDC recupera toda esta información mediante protocolos basados en SAML. Para interactuar con la infraestructura de autorización, los proveedores de servicio desplegarán, además, una arquitectura XACML como la descrita en la sección II-D. El rol de PEP será representado por el KDC, mientras que el PDP será una entidad independiente que se encargará de recibir peticiones de control de acceso, comprobar políticas y devolver decisiones de autorización al KDC. De esta forma, la gestión de la autorización es completamente transparente para los servicios.

Para transportar los paquetes EAP entre el cliente y el KDC se propone el uso de una capa de abstracción intermedia basada en el uso de GSS-API. De esta forma, desde el punto de vista del KDC, los datos de pre-autenticación que se transmiten son *tokens* GSS-API, siendo independiente del mecanismo concreto de autenticación que se está ejecutando por debajo (en este caso será EAP). Esto permite que en el futuro la arquitectura sea extensible a tipos de federación que

no estén basados en EAP.

B. Componentes principales de la arquitectura

La arquitectura propuesta está compuesta de los siguientes componentes:

- *End User*. Representa al usuario final, integrando la funcionalidad de un *EAP peer* y un cliente Kerberos para acceder a un servicio particular usando Kerberos sobre GSS-API.
- *KDC*. Es el componente principal de la arquitectura. Para la pre-autenticación Kerberos, el AS en el KDC jugará el rol de un *authenticator EAP*, intercambiando paquetes, EAP encapsulados en *tokens* GSS-API, sobre el protocolo Kerberos con el *end user*, y reenviandolos a través de la infraestructura AAA. Con respecto al proceso de autorización, el servicio TGS del KDC actuará como PEP, ocupándose de la obtención de atributos y la emisión de peticiones de control de acceso, interactuando con el IdP y el PDP respectivamente, a fin de determinar si el ST debe ser emitido al cliente.
- *AAA server*. Es la entidad responsable de autenticar al *end user* mediante EAP, por lo que se comporta como *EAP server*. El *AAA server* contactará con el IdP para solicitar una prueba de autenticación del usuario.
- *Identity Provider (IdP)*. La arquitectura propuesta asume que el dominio origen del usuario dispone de un IdP basado en la tecnología SAML. El IdP recibirá consultas de autenticación del *AAA server* del dominio origen, así como consultas de atributos de proveedores de servicios solicitando información de los usuarios.
- *Policy Decision Point (PDP)*. Es la entidad que se encargará de gestionar el conjunto de políticas de control de acceso para el dominio en el que se encuentra el KDC.
- *MetaData Service (MDS)*. El MDS es un servicio especial [18] donde tanto los proveedores de servicio como de identidad publican la localización (URL) de servicios que están disponibles para usarse dentro de una federación. En particular, la localización del IdP se publica en el MDS. En general, la localización del MDS es un valor fijo y pre-configurado dentro de la federación.
- *Application Server*. Proporciona el servicio específico (ej. WEB, SSH, etc.). Se asume que el control de acceso al servicio se basa en el uso de tickets de servicio Kerberos enviados mediante GSS-API.

C. Operación general

Supongamos que un usuario requiere un ticket de servicio (ST) de Kerberos para acceder a un *application server* en un dominio dentro de la federación. Para obtener dicho ST, el usuario requiere en primer lugar obtener un TGT para el TGS, lo que implica una comunicación con el AS localizado en el KDC del proveedor que controla dicho servicio. Antes de ser capaz de obtener el TGT, el usuario realiza una pre-autenticación Kerberos basada en EAP usando las credenciales compartidas con el *AAA server* de su dominio origen (fase de autenticación).

En cuanto el *AAA server* de su dominio origen autentica al usuario, comienza el primer paso de la fase de autorización. En particular, el *AAA server* origen contacta con el IdP, que emite una sentencia de autenticación indicando que el usuario ha sido autenticado satisfactoriamente. Con esta información, el *AAA server* puede obtener ciertos parámetros como el tiempo de vida de la autenticación o un puntero (*handle*) que será enviado al KDC a través de la infraestructura AAA. Este *handle* referencia a la autenticación realizada y se incluye en el TGT emitido por el KDC.

Cuando el usuario quiere obtener el ST para un servicio, presenta el TGT al KDC (TGS). Usando el *handle* contenido en el TGT, el TGS comienza el segundo paso de la fase de autorización, solicitando los atributos de usuario al IdP mediante el uso de SAML. El KDC puede descubrir la localización del IdP realizando una consulta al MDS. Entonces, el KDC (actuando como PEP) solicita una decisión de autorización al PDP, que gestiona el conjunto de políticas de control de acceso para el dominio de proveedor de servicios. Si el PDP autoriza el acceso del usuario al servicio, el KDC emitirá el ST. En caso contrario, el KDC no distribuirá el ST y emitirá un error. Finalmente, el usuario puede presentar este ST al servicio de modo tradicional, mediante el uso de GSS-API [9], donde el usuario actuará como *GSS-API initiator* y el servicio como *GSS-API acceptor*.

IV. Fase de autenticación

En la arquitectura propuesta el KDC hace uso de la GSS-API para pre-autenticar al usuario. Es decir, los datos de pre-autenticación contendrán tokens GSS-API, de forma que esta pre-autenticación pueda hacer uso de cualquier mecanismo disponible a través de la GSS-API. En particular, uno de los mecanismos de autenticación bajo estudio está basado en EAP [10], donde un token GSS-API contendrá un paquete EAP.

Como se muestra en la Fig. 3, este modelo implica una secuencia de llamadas a las funciones *GSS_Init_sec_context/GSS_Accept_sec_context (1)* de la GSS-API por el usuario (*GSS initiator*) y el KDC (*GSS acceptor*), respectivamente. Los tokens GSS-API generados por el *initiator* serán incluidos en el campo *padata* del mensaje *KRB_AS_REQ*; mientras que los generados por el *acceptor* serán transportados en el campo *e-data* del mensaje *KRB_ERROR*. De forma más específica, definimos un nuevo *padata-type* llamado *PA-GSS-TOKEN* que transportará un token generado por la GSS-API. En el caso particular de usar EAP como mecanismo de autenticación en GSS-API [10], la infraestructura AAA se ve involucrada tanto en el proceso de autenticación como en el autorización *(2)*.

Además, será necesario que el KDC sea capaz de recuperar el contexto GSS-API asociado al proceso de autenticación de un usuario cada vez que se recibe un nuevo mensaje *KRB_AS_REQ*. Para ello, en cada mensaje *KRB_ERROR*, además del elemento de pre-autenticación *PA-GSS-TOKEN*, se incluirá otro elemento *PA-FX-COOKIE* [19] que contendrá el identificador del contexto (*gss_ctx_id*) para el *acceptor*. El cliente incluirá una copia exacta de este *PA-FX-COOKIE* en el siguiente mensaje *KRB_AS_REQ* que envíe. Un mensaje *KRB_AS_REQ* sin un elemento *PA-FX-COOKIE* será considerado por el KDC como un inicio de autenticación y se creará un nuevo contexto para la misma.

Según el estándar Kerberos, el KDC generará un error si el identificador indicado en el campo *cname* del mensaje *KRB_AS_REQ* no se encuentra en su base de datos local. Dado

que el objetivo de este trabajo es permitir el acceso de usuarios federados que no se encuentran registrados en el dominio del proveedor, proponemos que el usuario incluya en su lugar el un identificador genérico *WELLKNOWN:FEDERATED*, siguiendo el modelo propuesto en [20]. De esta forma se indica al KDC que la autenticación se realizará mediante un mecanismo federado. Una vez terminado el proceso de autenticación, el KDC incluirá tanto en el TGT como en el mensaje *KRB_AS_REP* la identidad real del usuario obtenida mediante GSS-API.

Tras una autenticación exitosa con el dominio origen *(1)*, el *GSS initiator* (end user) y *acceptor* (KDC) dispondrán de material criptográfico para proteger los mensajes de aplicación. Concretamente, cuando se utiliza EAP, tanto *initiator* como *acceptor* obtendrán una MSK y derivarán dos claves, denominadas *CK (Checksum Key)* y *SK (Secret Key)*.

En particular, el token GSS-API contenido en el último *KRB_AS_REQ (3)* transportará la estructura de un mensaje *KRB_SAFE*, donde el campo *user-data* contendrá un checksum del mensaje *KRB_ERROR* enviado al usuario, calculado mediante el uso de la CK. Una vez recibido el token, el *acceptor* puede autenticar al *initiator* verificando la estructura *KRB_SAFE* con su propia CK.

Para finalizar, el KDC invoca a la función *GSS-Wrap* para proporcionar integridad y confidencialidad al campo *enc-part* del mensaje *KRB_AS_REP*, mediante el uso de la clave SK derivada. De igual forma, el usuario llamará a *GSS-Unwrap* para procesar el campo *enc-part*. La correcta verificación de este campo permitirá al usuario autenticar al KDC.

Como se puede observar, tras una autenticación correcta, el usuario obtiene un TGT que puede usarse para obtener un ticket de servicio *(4) y (5)*. A continuación nos centraremos en las tareas de autorización requeridas durante la emisión del TGT *(2)* y la solicitud del ticket de servicio *(4)*.

V. Fase de autorización

Esta sección describe cómo se pueden realizar decisiones de control de acceso sobre el modelo de autenticación descrito anteriormente. En particular, mientras que SAMLv2 [16] se emplea para gestionar las sentencias de autenticación y atributos, XACML [17] ofrece un lenguaje estándar de políticas de control de acceso. Asumimos también que la infraestructura AAA subyacente está basada en RADIUS o Diameter. El flujo de mensajes necesario para la autorización está descrito en la Fig. 4.

A. Manejo de la sentencia de autenticación

Una vez que el usuario ha sido autenticado por su servidor AAA origen, y antes de notificar al usuario, el servidor AAA solicita una sentencia de autenticación (*AuthnStatement)* al IdP (*2*). Esta solicitud se realiza mediante un mensaje *AuthnQuery*, donde el campo *Subject* representa la identidad del usuario. El IdP genera un *Assertion* de tipo *AuthnStatement*, indicando que el usuario ha sido correctamente autenticado bajo un contexto específico. De forma similar a [21], se envía un *handle* que referencia a dicha sentencia de autenticación. Este *Assertion* es válido únicamente por un período de tiempo (definido por el IdP), y generalmente se identifica al usuario mediante un seudónimo temporal. Típicamente, las consultas y respuestas SAML se firman digitalmente y se transportan protegidas mediante HTTPS/SOAP.

Una vez que el servidor AAA recibe la sentencia de autenticación, envía al KDC el mensaje *EAP-Success*, junto a cierta información adicional. Por un lado, en base a la información recibida en la sentencia, el servidor AAA puede enviar atributos AAA estándar para dirigir el comportamiento del KDC. Un ejemplo sería el atributo RADIUS *Session-Timeout*, que estaría basado en el período de validez de la sentencia de autenticación, y que el KDC podría usar para limitar el tiempo de vida del nuevo TGT. Por otro lado, también se enviará el *handle*, que podrá ser utilizado posteriormente por el KDC para recuperar atributos de usuario. Este *handle* toma la forma de *pseudonym@homedomain*, protegiendo la identidad del usuario e indicando el dominio de donde pueden obtenerse los atributos. Es necesario definir un nuevo atributo AAA para transportar el *handle* hasta el KDC.

B. Gestión del handle

Como se describe en la sección III, el KDC se encargará de solicitar atributos de usuario y decisiones de autorización, definiendo un escenario de gestión de la autorización totalmente transparente para los servicios. El AS en el KDC es responsable de llevar a cabo el proceso de autenticación, por lo que recibe el *handle* a través del cliente AAA. Sin embargo, el módulo responsable de emitir el ST es el TGS, por lo que parece razonable delegar en él la gestión del proceso de autorización.

El flujo de mensajes necesario para gestionar el *handle* se describe en la Fig. 4. Cuando el AS recibe el *handle (1)*, lo encapsulará como un nuevo elemento *authorization-data* (ADE) dentro del campo *authorization-data (3)* del TGT. Éste TGT se envía entonces al usuario en el mensaje (*KRB_AS_REP)*.

El TGS recibirá el *handle* por medio del mensaje *KRB_TGS_REQ*. Siguiendo el estándar Kerberos, el TGS comprobará si el TGT recibido es válido y, en caso de serlo, emitirá un nuevo ST para el usuario. Se propone extender la funcionalidad del TGS, para permitir que pueda solicitar tanto atributos del usuario a su dominio origen, como una decisión de autorización al PDP local *(4)*. Es importante recalcar que el *handle* contenido en el TGT no se incluirá en el ST emitido.

C. Gestión de atributos y control de acceso

Para poder recuperar atributos *(4)*, el TGS debe descubrir dónde está localizado el proveedor de atributos del usuario. Esta información puede haber sido configurada de forma previa para pequeñas federaciones, o bien puede consultar un servicio de metadatos o *MDS*. El servicio de metadatos almacena descriptores XML con las localizaciones de los servicios públicos de la federación. Estos localizadores serán devueltos a TGS en base a la información de dominio incluida en el *handle*. Los detalles del perfil de MDS [18] están fuera del ámbito de este trabajo. En aras de la simplicidad, consideraremos (como es habitual en las soluciones actuales) que el IdP también juega el rol de proveedor de atributos.

Una vez localizado el IdP, el TGS le envía un mensaje *SAML AttributeQuery*. El elemento *Subject* de este mensaje contiene el seudónimo de usuario incluido en el *handle*. Tras

Fig. 3. Pre-autenticación basada en EAP sobre GSS-API sobre Kerberos

recibir la petición, el IdP comprueba los atributos y emite un nuevo *Assertion* de tipo *AttributeStatement*, que se envía de vuelta al TGS. Como se ha comentado anteriormente, las sentencias de solicitud y respuesta de atributos se firman digitalmente y se transmiten sobre HTTPS/SOAP.

El TGS solicita a su PDP local una decisión de autorización, actuando como PEP y emitiendo un *XACMLAuthzDecisionQuery* indicando: (a) los atributos del usuario; (b) el identificador del recurso objetivo de control de acceso; y (c) la acción requerida. Tras comprobar las políticas, el PDP emite un *XACMLAuthzDecision Response* que contendrá un *XACMLAuthzDecisionStatement* donde se indica la respuesta (PERMIT o DENY) y, opcionalmente, un conjunto de obligaciones. Si la decisión es positiva, el TGS crea un ST y lo envía al usuario. La descripción de las políticas de control de acceso siguen el estándar XACML 2.0 y quedan fuera del ámbito de este trabajo.

Finalmente, el usuario podrá acceder al servicio mediante un mensaje *KRB_AP_REQ* que contendrá el ST en un GSS-API token (*5*). Este intercambio no requiere ningún cambio al modo de operación para acceder a un servicio mediante Kerberos sobre GSS-API y, por tanto, los servicios desplegados no necesitan ser modificados.

VI. Discusión

Como se puede observar, nuestra solución no requiere que el usuario realice una operación *cross-realm*, evitando así que los dominios intermedios tengan que desplegar KDCs para asistir el proceso. Además, a diferencia de las infraestructuras Kerberos, las infraestructuras AAA no sólo sirven para

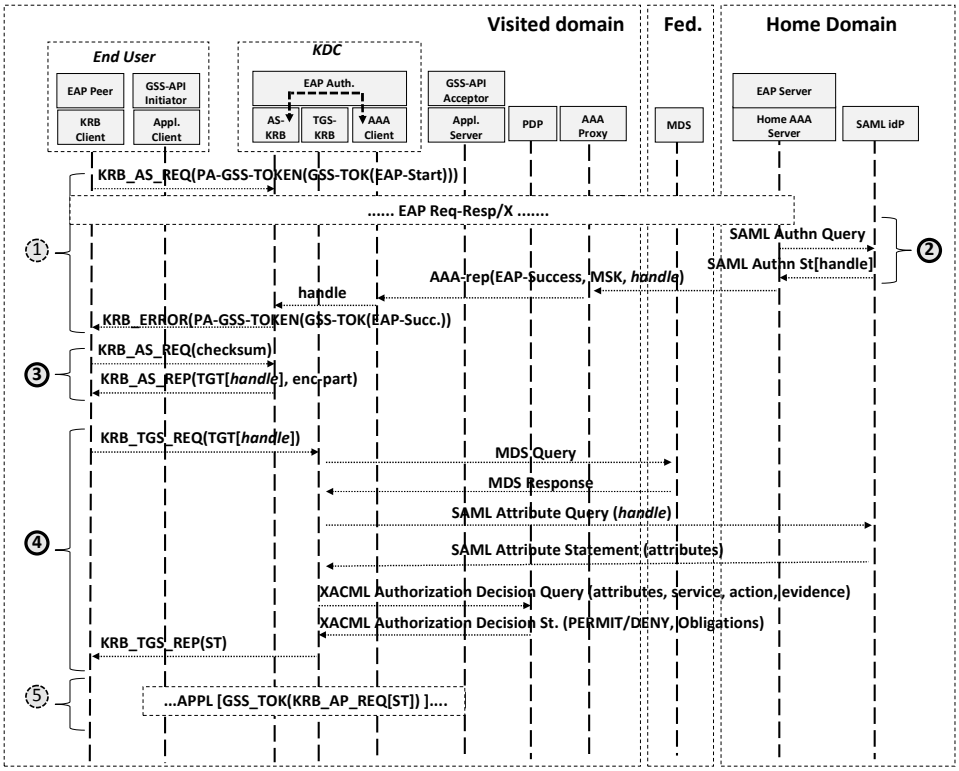


Fig. 4. Gestión de la autorización

gestionar autenticación y distribución de claves, sino también para realizar autorización y procesos de auditoría, que son importantes en las redes desplegadas en la actualidad.

Con respecto a la gestión y distribución de claves, el uso de EAP para la pre-autenticación Kerberos tiene también algunas implicaciones en términos de seguridad. De hecho, tanto el análisis de seguridad descrito en [22] para la gestión de claves en arquitecturas AAA como el descrito en [15] para EAP son aplicables aquí. La clave MSK exportada por el servidor AAA/EAP del dominio origen debe ser transmitida hasta el KDC, que actúa como *authenticator EAP*. Potencialmente, los servidores AAA intermedios (*proxies*) pueden observar dicha clave, que será utilizada para derivar la clave secreta de Kerberos. Este tipo de comportamiento podría afectar a la seguridad de esta clave.

Sin embargo, el modelo de confianza en tales entornos federados asume que los AAA *proxies* puede considerarse como entidades confiables, y por tanto la MSK se puede distribuir de una forma segura hasta el KDC a través de los mismos. Además, como se explica en [22], existen técnicas de envoltura de claves que pueden aplicarse para proporcionar integridad, confidencialidad y protección contra reenvío al material criptográfico distribuido entre entidades AAA.

La sección V describe cómo el KDC gestiona la autorización. También es posible un modelo en el que el KDC simplemente incluye el *handle* en el ST emitido, de forma que sea el propio servicio quien se encargue de realizar la petición de atributos y de contactar con el PDP.

Además, es posible simplificar el flujo de autorización si se permite que el IdP proporcione, junto con el *handle*, un conjunto de *AttributeStatement* durante la primera fase de autorización. Estos atributos serían transportados mediante atributos AAA hasta el AS, que los incluiría estos atributos en el TGT de forma que el TGS pueda utilizarlos directamente sin necesidad de contactar con el el IdP. Esto permite ahorrar algunos mensajes, aunque tiene la desventaja de que el IdP no puede determinar qué atributos son realmente necesarios para el servicio ya que éste es todavía desconocido.

VII. Trabajo relacionado

Existen diversas soluciones que tratan de asegurar que los TGTs sólo se emiten a usuarios autenticados. Uno de los primeros trabajos en este área puede encontrarse en [23], donde los autores proponen que el KDC local distribuya un TGT tras una autenticación EAP. Sin embargo esta solución no está basada en el uso de la pre-autenticación Kerberos, sino que asume que existe un proceso de acceso inicial a la red que sustituye el intercambio *KRB_AS_REQ/KRB_AS_REP*.

La mayoría de las soluciones aparecen tras el desarrollo de la versión 5 del protocolo Kerberos [2]. Por ejemplo, [24] propone un mecanismo (llamado PKINIT) que permite que usuarios y KDC se autentiquen mutuamente durante el intercambio *AS* mediante el uso de certificados. No obstante, esta solución no proporciona flexibilidad dado que exige que los usuarios dispongan de su propio certificado.

En [25], los autores proponen que el cliente se autentique con un servidor de pre-autenticación y obtengan un ticket inicial que se use para autenticar al usuario con el KDC. Sin embargo, esta propuesta implica una modificación de Kerberos para añadir dicho intercambio.

Una solución con el mismo objetivo que evita este inconveniente puede encontrarse en [19]. Esta propuesta del *IETF Kerberos Working Group* define un conjunto de funciones comunes que se ponen a disposición de los

mecanismos de pre-autenticación. Una de estas herramientas es un canal protegido entre el KDC y el usuario para intercambiar datos de pre-autenticación. Sin embargo, este mecanismo no considera la infraestructura AAA ni EAP para realizar el proceso de autenticación.

Pese a que EAP fue diseñado para el acceso a la red, la idea de proporcionar un mecanismo genérico de autenticación para servicios de aplicación basado en EAP se ha descrito en el proyecto *Moonshot* [7], [10]. Éste propone integrar EAP sobre GSS-API para la autenticación en servicios de aplicación, pero sin utilizar Kerberos en el proceso. Este trabajo también propone la integración de AAA y SAML para proporcionar autorización, lo que implica que los servicios deben ser capaces de entender SAML, algo que no es muy común hoy en día. Por contra, en el trabajo expuesto en el presente artículo el proceso de autorización recae sobre el KDC, de forma que resulta trasparente a los servicios.

La integración de SAML y Kerberos sigue siendo un problema a resolver [26], y está bajo estudio en diferentes cuerpos de estandarización como el IETF y OASIS. Existen dos modelos preliminares, Kerberos-en-SAML [27] y SAML-en-Kerberos, aunque, para éste último, todavía no se ha definido una solución. Siguiendo este modelo e involucrando al KDC en las tareas de autorización, nuestra propuesta evita tener que cambiar las aplicaciones existentes.

VIII. Conclusiones y vías futuras

Este trabajo analiza varios protocolos estándares y tecnologías utilizadas para autenticar y autorizar servicios en redes federadas. En particular, Kerberos se utiliza ampliamente para autenticación y distribución de claves en servicios de aplicación dentro de un dominio. Sin embargo, las infraestructuras Kerberos *cross-realm* no han tenido un gran despliegue en redes federadas. En su lugar, los dominios se interconectan por medio de una infraestructura AAA para el control de acceso a servicios. Desafortunadamente, esto dificulta enormemente que los usuarios suscritos en un dominio puedan acceder usando Kerberos a los servicios proporcionados en otro dominio de la federación.

En este artículo, se propone una solución al problema, diseñando una nueva arquitectura que integra, a través de GSS-API, EAP con pre-autenticación Kerberos. Esto permite que cualquier usuario en la federación, incluso aquellos que no tienen una suscripción con el proveedor de servicios, puedan autenticarse con su dominio origen mediante la infraestructura AAA desplegada antes de obtener un ticket Kerberos para acceder al servicio. Por tanto, evitamos desplegar una infraestructura Kerberos *cross-realm* apoyándonos en el despliegue AAA existente. Además, se han integrado las tecnologías SAML y XACML para alcanzar una gestión avanzada de la autorización.

Como vías futuras, está prevista la integración de la pre-autenticación Kerberos basada en EAP con el marco general de pre-autenticación definido en [19]. Además, se etá trabajando en el desarrollo de un prototipo que permita analizar el rendimiento de la solución propuesta. Finalmente se está colaborando con el grupo de trabajo ABFAB [28] del IETF para promover nuevas soluciones para la federación de servicios.

Agradecimientos

Este trabajo está financiado por el proyecto MULTIGIGABIT EUROPEAN ACADEMIC NETWORK (FP7-INFRASTRUCTURES-2009-1). También ha recibido financiación de la Fundación Séneca a través de su Programa de Ayuda a la Formación de Recursos Humanos. Finalmente los autores agradecen a la Fundación Séneca por el Programa de Ayuda a los Grupos de Excelencia (04552/GERM/06).

Referencias

[1] T. Määttänen. *Single Sign-On Systems*, Noviembre 2002.
[2] C. Neuman, T. Yu, S. Hartman, and K. Raeburn. *The Kerberos Network Authentication Service (V5)*. IETF RFC 4120, Julio 2005.
[3] *The MIT Kerberos Consortium*. http://www.kerberos.org.
[4] C. de Laat, G. Gross, L. Gommans, J. Vollbrecht, and D. Spence. *Generic AAA Architecture*. IETF RFC 2903, Aug. 2000.
[5] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz. *Extensible Authentication Protocol (EAP)*. RFC3748, Junio 2004.
[6] K. Wierenga y otros. *DJ5.1.4: Inter-NREN Roaming Architecture. Description and Development Items*, Septiembre 2006. Project Deliverable.
[7] J. Howlett and S. Hartman. *Project Moonshot*. Febrero 2010.
[8] J. Linn. *Generic Security Service Application Program Interface Version 2*. IETF RFC 2743, Enero 2000.
[9] L. Zhu, K. Jaganathan, and S. Hartman. *The Kerberos Version 5 Generic Security Service Application Program Interface (GSS-API) Mechanism: Version 2*. IETF RFC 4121, Julio 2005.
[10] S. Hartman and J. Howlett. *A GSS-API Mechanism for the Extensible Authentication Protocol*. IETF Internet Draft, IETF draft-howlett-eap-gss-01.txt, Febrero 2011.
[11] IEEE 802.1X Std., Standards for Local and Metropolitan Area Networks: Port based Network Access Control, 2004. IEEE Standards for Information Technology.
[12] C. Rigney, S. Willens, A. Rubens, and W. Simpson. *Remote Authentication Dial In User Service (RADIUS)*. IETF RFC 2865, Junio 2000.
[13] P. Calhoun and J. Loughney. *Diameter Base Protocol*. IETF RFC 3588, Sept. 2003.
[14] R. Dantu, G. Clothier, and A. Atri. EAP Methods for Wireless Networks. *Computer Standards Interfaces*, 29(3):289–301, 2007.
[15] B. Aboba, D. Simon, and P. Eronen. *Extensible Authentication Protocol Key Management Framework*. RFC 5247, Agosto 2008.
[16] S. Cantor, J. Kemp, R. Philpott, and E. Maler (Eds.). *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) v2.0*, Marzo 2005.
[17] *eXtensible Access Control Markup Language (XACML) Version 2.0*, Febrero 2005. OASIS Standard.
[18] S. Cantor, J. Moreh, R. Philpott, and E. Maler (Eds.). *Metadata for the OASIS Security Assertion Markup Language (SAML) v2.0*, Marzo 2005.
[19] S. Hartman and L. Zhu. *A Generalized Framework for Kerberos Pre-Authentication*. IETF RFC-to-be 6113, Febrero 2011.
[20] L. Zhu. *Additional Kerberos Naming Constraints*. IETF Internet RFC-to-be 6111, Febrero 2011.
[21] *DAMe Project*. http://dame.inf.um.es.
[22] R. Housley and B. Aboba. *Guidance for Authentication, Authorization, and Accounting (AAA) Key Management*. IETF RFC 4962, Julio 2007.
[23] H. Tschofenig and V. Sankhla. *Bootstrapping Kerberos*. IETF Internet Draft, IETF draft-tschofenig-pana-bootstrap-kerberos-00, Julio 2004.
[24] L. Zhu and B.Tung. *Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)*. IETF RFC 4556, Junio 2006.
[25] P.L. Hellwell, T.W. van der Horst, and K.E. Seamons. *Extensible Pre-Authentication in Kerberos*. In *Proc. of the Twenty-Third Annual Conference on Computer Security Applications, 2007*, Miami Beach, FL, Diciembre 2007.
[26] J. Hodges, J. Howlett, L. Johansson, and RL. Morgan. *Towards Kerberizing Web Identity and Services*, Diciembre 2008. Kerberos consortium.
[27] J. Howlett and T. Hardjono. *SAML V2.0 Kerberos Subject Confirmation Method Version 1.0*, Diciembre 2009. Committee Draft 01.
[28] Application bridging for federated access beyond web (abfab) ietf working group. https://datatracker.ietf.org/wg/abfab/charter/.

Seguridad y movilidad en una VANET real desplegada con diferentes tecnologías inalámbricas

Pedro J. Fernández Ruiz {pedroj@um.es},
Cristian A. Nieto Guerra {cristian.nieto@um.es},
Antonio F. GomezSkarmeta {skarmeta@um.es}

Departamento de Ingeniería de la Información y las Comunicaciones
Campus de Espinardo, Facultad de Informática, C.P. 30100, Murcia, SPAIN

***Resumen*—En este artículo se pretende compartir las experiencias y conclusiones obtenidas a la hora de desplegar una VANET y aplicar sobre ella los servicios de movilidad y seguridad, fundamentales para una red de estas características donde hay elementos móviles que además usan tecnologías inalámbricas, mucho más fáciles de interceptar. En las VANETs suelen estar presentes diferentes tecnologías inalámbricas de acceso. Por tanto nuestra solución debe ser independiente de estas tecnologías para poder cambiar de una a otra fácilmente. Para ello la seguridad se ha aplicado en la capa de red mediante los protocolos IKEv2 y EAP. Gracias a EAP se puede además hacer las funciones de control de acceso utilizando para ello diferentes métodos de autenticación, de los que hemos hecho una comparativa de algunos de ellos. En cuanto a la movilidad se ha empleado la evolución de MIPv6 para VANETs llamada NEMO, en donde se introduce el concepto de "router móvil". Con todo esto dispondremos de un estupendo banco de pruebas real de donde extraer conclusiones reales.**

***Palabras Clave*—Redes Vehiculares, VANETs, WiMAX, Wi-Fi, IKEv2, IPsec, EAP, NEMO, Autenticación, Mobilidad, Seguridad, OpenIKEv2.**

I. Introducción

Las comunicaciones inalámbricas están experimentando un gran auge en los últimos años. En consecuencia, conceptos directamente relacionados con el hecho de funcionar sin cables también están ganando en importancia día a día. Los más destacados son la movilidad y la seguridad. Esto es debido a la propia naturaleza de las comunicaciones inalámbricas, donde los dispositivos pueden moverse por unas determinadas zonas de cobertura y donde además se utiliza un mismo medio compartido para llevar a cabo dicha comunicación. Esto hace que aumente el riesgo de que la información transmitida sea interceptada por terceros no deseados. Además, los dispositivos inalámbricos vienen cada vez mejor provistos en cuanto a comunicaciones inalámbricas, disponiendo de diferentes tecnologías simultáneamente como pueden ser Wi-Fi, WiMAX, 802.11p, Bluetooth; cada una con sus propias características que podemos contemplar en la figura 1, que las hacen más o menos interesantes dependiendo del escenario en el que nos encontremos. Las redes vehiculares también están aprovechando estos avances, y por eso hoy en día los coches empiezan a incluir dispositivos de comunicaciones en su equipamiento de serie.

	802.11p (Wave)	802.11g (Wi-Fi)	3.5G (HSDPA)	802.16e (WiMAX)
Máxima velocidad de transferencia (Mbps)	27	54	14(DL), 0.38(UL)	70
Latencia (ms)	< 50	50 - 250	50 - 250	20
Rango de cobertura (Km)	1	0.1	10	15
Velocidad máxima (Km/h)	100	10	120	140
Ancho de banda (Mhz)	10	20	3	10,2
Banda de frecuencias (Ghz)	5.86 - 5.92	2.4, 5.2	0.8, 1.9	2.5, 3.5, 4.9, 5.8

Fig. 1. Comparación entre tecnologías inalámbricas.

Como tecnología inicial de despliegue de nuestra infraestructura inalámbrica se ha elegido el estandar 802.16e, o también llamado "Mobile WiMAX". El estándar 802.16, o simplemente WiMAX, fue inicialmente concebida para escenarios punto a punto para dotar de conectividad a poblaciones aisladas. Sin embargo, esta evolución a la movilidad ha posibilitado usar dicha tecnología en escenarios donde la movilidad toma suma importancia, como es el caso de las VANETs. Además usa frecuencias más bajas que el estándar original para poder mantener comunicaciones *NLOS*[1], es decir, que no requieran que emisor y receptor tengan linea de visión directa.

Por tanto se han adquirido estaciones base WiMAX para proveer cobertura a lo largo del anillo de circunvalación que rodea el Campus de Espinardo de la Universidad de Murcia, dejando intencionadamente alguna zona donde no exista cobertura, y por otra parte, zonas donde haya solapamientos de cobertura de diferentes estaciones base para encontrarnos con todas las situaciones posibles. Para asegurarnos de estas circunstancias es necesario efectuar un análisis de coberturas de la zona una vez desplegada la infraestructura.

Uno de los temas más importantes a tener en cuenta en las comunicaciones inalámbricas es la seguridad. Al estar compartiendo el mismo medio para realizar las transmisiones, normalmente el aire, estas comunicaciones están expuestas a diversas amenazas y vulnerabilidades que permiten a un tercero interceptar, intervenir o incluso impedir la comunicación

[1]Non-Line-Of-Sight

ISBN: 978-84-694-5948-5

Fig. 2. Estación base real basado en WiMAX.

Fig. 3. Cliente WiMAX real.

(DoS). Para aportar autenticidad y confidencialidad a dichas comunicaciones se va a utilizar el protocolo IPsec que se encarga de transportar los datos de forma segura mediante métodos criptográficos. Para ello requiere que previamente se establezca un material criptográfico compartido en los extremos de la comunicación. Se puede hacer de forma estática, pero tiene el inconveniente de que hay que cambiarla de vez en cuando y siempre manualmente, que puede llegar a ser tedioso para un administrador. Para evitar esto usaremos el protocolo IKEv2, que se encarga de generar automáticamente y de forma segura dicho material criptográfico cada vez que sea necesario. Además IKEv2 permite realizar asignación dinámica de direcciones IP en uno de los extremos y transportar el protocolo EAP que nos permitirá poder usar diferentes métodos de autenticación. También podremos implementar gracias a la conjunción de los protocolos de seguridad anteriores un sistema de control de acceso para restringir el acceso sólo a los terminales móviles que tengan permiso.

En cuanto a la movilidad, otro aspecto fundamental en comunicaciones inalámbricas en general, redes vehiculares[1] en particular, hemos utilizado la implementación del protocolo NEMO[10] aportada por el proyecto UMIP, protocolo similar a MIPv6 pero que introduce el concepto de router móvil al que nos podremos conectar de tal forma que no seremos conscientes de sus movimientos de una red a otra. Un escenario donde comúnmente se usa dicho protocolo es el caso de un tren, autobús o vehículo en general que provee acceso a internet a sus viajeros.

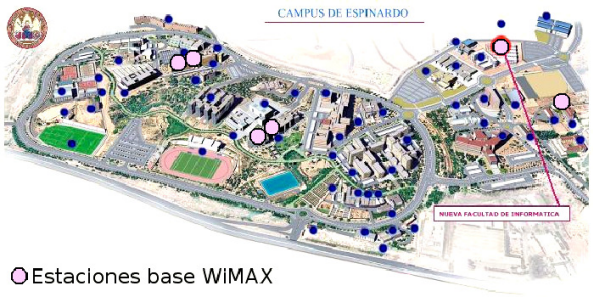


Fig. 4. Localización de las estaciones base en el Campus de Espinardo.

II. Construyendo el escenario

Lo que se plantea aquí es el despliegue de una infraestructura inalámbrica, inicialmente formada sólo por WiMAX pero al que se le añadirá Wi-Fi en un futuro, para ofrecer acceso a Internet a través de un sistema de control de acceso. La autenticación debe ser extensible (EAP) y los datos viajar cifrados (IPsec + IKEv2). Además el terminal debe de poder cambiar de red, y por tanto de IP, sin que las comunicaciones se vean interrumpidas, ni sus sesiones rotas (NEMO), pudiendo a su vez proveer acceso a terceros.

La dificultad reside en ofrecer todos los servicios anteriores a la vez, sin que ninguno de ellos dificulte el desempeño de otro. Por eso se ha creído conveniente ir paso a paso, empezando por el despliegue de la infraestructura, instalación y configuración de equipos, configuración de IKEv2 para la seguridad y finalmente configuración de NEMO para movilidad.

A. *Infraestructura inalámbrica*

El primer paso que tenemos que dar es desplegar nuestra infraestructura WiMAX, que en un futuro se mezclará con otras, en un entorno real (Campus de Espinardo, Murcia) y verificar las zonas de cobertura mediante un análisis de cobertura y comprobar si es como queremos que sea para la realización de las investigaciones.

Se han adquirido hasta un total de 6 estaciones base repartidas por las azoteas de aquellos edificios que por su posición con respecto al Campus facilita la distribución de la cobertura, como se puede apreciar en la figura 4. Acompañando a estas estaciones también se han adquirido clientes suficientes para conectarnos a las estaciones base. Tanto estos últimos como las estaciones base son operables mediante TELNET o SNMP, de tal forma que podemos, por ejemplo, saber la fuerza de la señal que recibe un determinado cliente. Con esto y un dispositivo GPS podemos realizar el mencionado análisis de cobertura, haciendo corresponder las fuerzas de señal registradas con la localización geográfica donde se tomó cada muestra, dándonos como resultado lo mostrado en la figura 5.

La toma de muestras puede realizarse con todas las estaciones base activas, o una por una. La segunda opción es recomendable para saber exactamente los límites de la cobertura

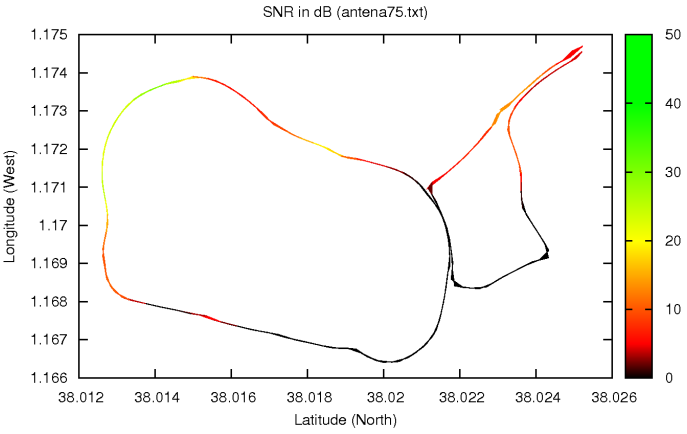


Fig. 5. Mapa de cobertura de algunas de las estaciones base WiMAX instaladas.

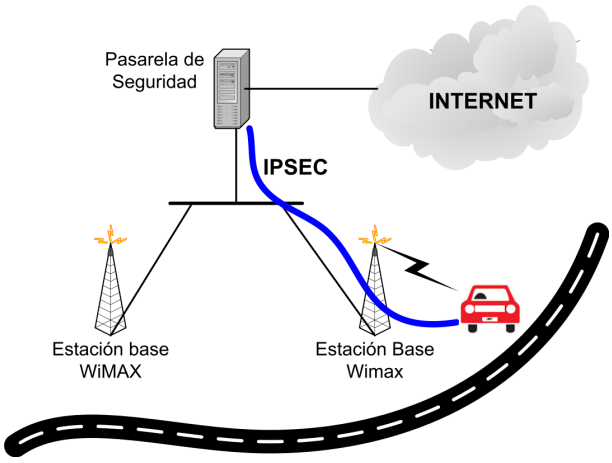


Fig. 6. Escenario "Road warrior" sobre una infraestructura WiMAX.

de cada una, y así saber si existen zonas con solapamiento de dos o más estaciones, o si hay zonas de sombra donde ninguna de ellas llega. Las zonas de solapamiento son muy interesantes para el estudio de escenarios de pre-autenticación en handovers, es decir, anticiparnos al cambio de estación base autenticándonos primero mediante el enlace que todavía tenemos con la estación base que vamos a dejar. Esto hace que en el momento de realizar el handover ya no haya que autenticarse y el cambio sea muy rápido e imperceptible para las aplicaciones que están usando la red.

De estos mapas de cobertura se constata que, a pesar de trabajar con frecuencias relativamente bajas (4.9-5.8Ghz) consideradas NLOS, es decir, que no requieren linea de visión directa entre emisor y receptor, colinas y edificios son difícilmente salvables debido a la alta direccionalidad de las microondas, generando así zonas de sobra tras ellos.

B. Seguridad mediante IPsec, IKEv2 y EAP

Para ser independientes de las diferentes tecnologías presentes en las redes heterogéneas, hemos aplicado los mecanismos de seguridad a nivel de red, y por tanto siempre el mismo en todos los casos. El tráfico se protege mediante el protocolo IPsec, que establece túneles seguros. Como todos los túneles tienen que tener dos extremos. En este caso uno está claro, el vehículo. Sin embargo debemos establecer el otro extremo en algún punto. Para ello vamos a definir un elemento extra dentro de cada red de acceso llamado "Pasarela de Seguridad" (Security Gateway, en adelante SG). Se trata simplemente de un router que nos proveerá acceso a Internet por un lado, y nos servirá de extremo para los túneles IPSec por el otro. Por tanto tendrá dos interfaces, una hacia una red aislada que conecta con las estaciones base y la otra el acceso a Internet.

La pasarela se seguridad se anunciará a través de las estaciones base por medio de "Router Advertisements", en adelante RA) para notificar su presencia a los diferentes vehículos que pudiera haber en el rango de cobertura de dichas estaciones. En el momento en el que un vehículo recibe un RA, entra en juego el protocolo IKEv2 para el establecimiento de material criptográfico entre vehículo y SG. En la misma negociación se realizará simultáneamente un proceso de autenticación, en nuestro caso basado en EAP, que nos servirá como sistema de control de acceso. Por tanto el servicio IKEv2 tiene que estar instalado tanto en vehículo como en SG, es decir, en los extremos de los futuros túneles IPsec. Este escenario IPsec, que podemos ver en la figura 6, es muy similar al típico y conocido "RoadWarrior".

Tanto en el vehículo como en el SG, OpenIKEv2 debe establecer las políticas necesarias para:

- permitir todo el trafico cuyas direcciones origen y destino sean *link-local*. Ya sabemos de antemano que no pueden ir más allá del SG.
- permitir el tráfico IKEv2 e ICMPv6, para permitir también con ello las negociaciones IKEv2 y mensajes ICMP, como el ping.
- proteger el tráfico con origen o destino aquellos vehículos que se hayan autenticado con éxito y hayan establecido un túnel IPsec.
- impedir que circule cualquier otro tráfico distinto al anterior.

Para el proceso de autenticación comentado anteriormente, se ha elegido como punto de partida el método de autenticación EAP-TLS[7]. Para ello necesitamos de una PKI que nos proporcione certificados y un *AAA*, que es simplemente un servidor *Radius*[19] (implementado con *Free Radius*[20]). Después se ha implementado un método de autenticación que nos permite reducir el tiempo consumido en un handover. Se trata de EAP-FRM[8], fruto de la misma Universidad de Murcia. En la figura7 podemos ver la notable diferencia en los tiempos de autenticación de EAP-TLS y EAP-FRM, y lo próximo que está éste último del método de autenticación más rápido de todos, es decir, directamente con IKEv2 con claves pre-compartidas (PSK).

C. Movilidad mediante NEMO

La movilidad es una característica fundamental en las VANETs. Precisamente los vehículos se mueven mucho y

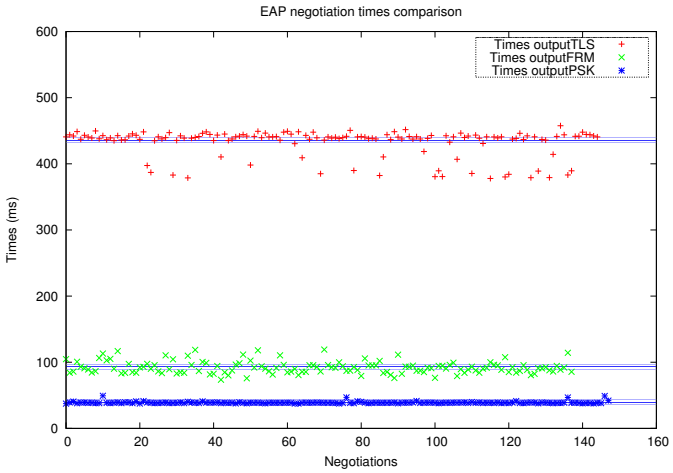


Fig. 7. Comparativa de tiempos de autenticación

además a una gran velocidad. Pero, ¿en qué consiste la movilidad en cuanto a las redes se refiere?. Nos referimos a la capacidad de vehículo de mantener las sesiones de las comunicaciones abiertas a pesar de cambiar de punto de acceso. En otras palabras, que sus aplicaciones trabajen con una IP que no cambie nunca (Home Address, HoA) mientras obtenemos conectividad con otra que va cambiando (Care of address, CoA) cada vez que saltamos de un punto de acceso a otro que requiera un cambio de IP. Inicialmente, para aportar esta característica se podía utilizar el protocolo MIPv6. Sin embargo, para el caso de las VANETs existe una evolución de dicho protocolo llamada NEMO[2]. Lo que realmente distingue y hace interesante este último protocolo es el nuevo concepto de "Mobile Router" (en adelante MR) como vemos en la figura 8. Por decirlo de otra forma, posibilitamos que los propios vehículos que puedan a su vez dar acceso a otros dispositivos que se encuentren en su interior. La evidente ventaja es que sólo el MR tiene que realizar el proceso de handover, haciendo que sea transparente para los dispositivos conectados a través de él, que llamaremos conceptualmente "Mobile Node NENO" (en adelante MNN).

Otra diferencia es que el Home Agent tiene que tener en cuenta un dato más para cada MR: los prefijos de sus redes internas ("Mobile Network Prefix", en adelante MNP), para ser capaz de direccionar de vuelta el tráfico procedente de los diferentes dispositivos MNN. Por tanto, será algo que, además de la CoA, se notificará en el "Binding Update" que se envía al Home Agent.

Hemos realizado también un estudio de cuánto tiempo se invierte en notificar al HA, es decir, el intercambio BU/BA. Como podemos apreciar en la figura 9 el tiempo invertido es muy constante, en torno a 1 segundo, lo cual nos parece mejorable, pues los requisitos de las redes VANETs suelen ser más exigentes.

[2]NEtwork MObility

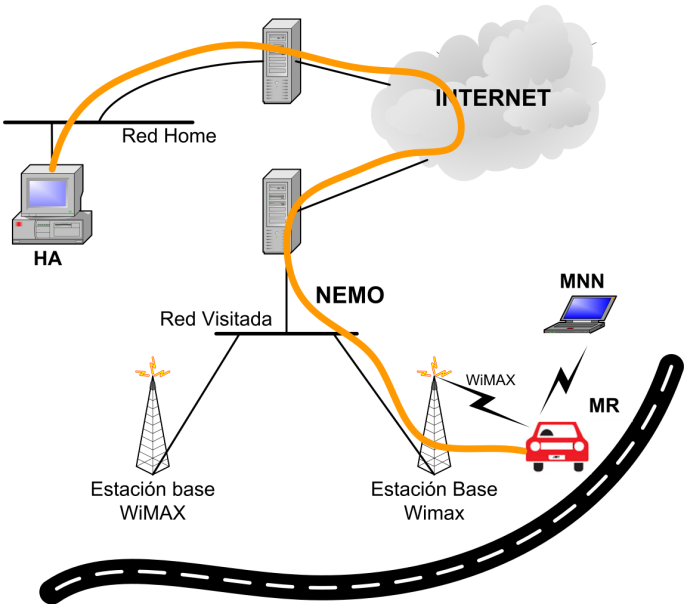


Fig. 8. Soporte de movilidad para una red vehicular.

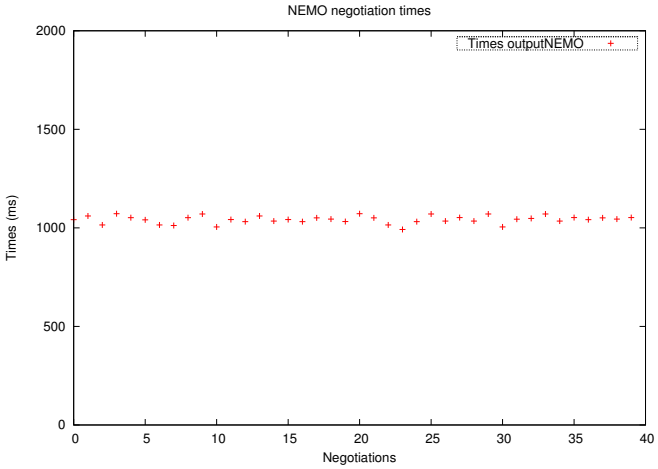


Fig. 9. Tiempos de negociación de la movilidad NEMO

D. *Interoperabilidad entre seguridad y movilidad*

Hasta ahora hemos visto como proporcionar tanto seguridad como movilidad a las comunicaciones, pero de forma independiente. Estos servicios son dependientes entre sí, por lo que hay que establecer un cierto orden a la hora de aportarlos. Sin embargo, este orden no se puede conseguir de forma natural debido a que el servicio IKEv2 no reacciona a los RAs de la misma forma que lo hace el servicio de movilidad. Por tanto, al producirse un cambio de red, y por tanto de IP, el único que se entera de ellos es el servicio de movilidad, actualizando dicha CoA en el HA. Es por esto que hay que dotar al servicio IKEv2 de una forma de colaborar con el servicio NEMO y crear el túnel IPsec antes de reestablecer la mobilidad. De esta forma protegemos también la señalización BU/BA entre RM y HA, además de el tráfico que circule entre RM y la pasarela se seguridad (SG).

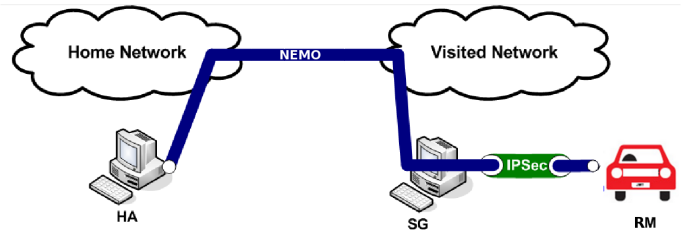


Fig. 10. Anidación de túneles IPsec y NEMO

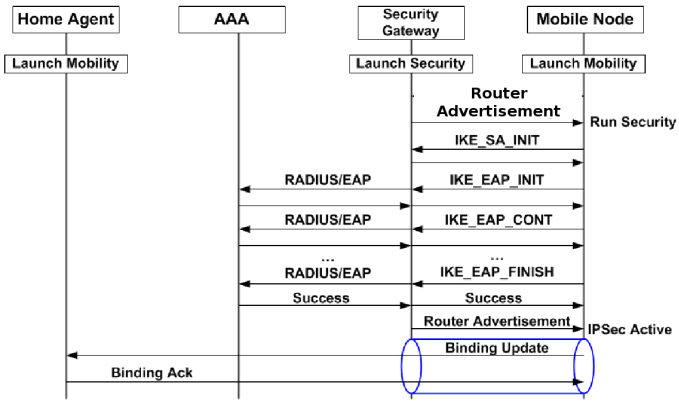


Fig. 11. Colaboración entre seguridad y movilidad.

Si nos fijamos en la figura 10 vemos que en el RM se está produciendo una anidación de túneles, en concreto, el túnel de movilidad va por dentro del túnel de seguridad. Para conseguir esto es necesario que el sistema operativo, en este caso Linux 2.6, pueda aplicar varias políticas a un mismo tráfico. Esto en Linux se resuelve teniendo dos conjuntos de políticas, las llamadas "main" relacionadas con la seguridad, y las llamadas "sub" relacionadas con la movilidad. En este caso se puede aplicar una política de cada conjunto simultáneamente, consiguiendo el anidamiento deseado.

El mecanismo elegido para hacer que movilidad y seguridad cooperen es la de hacer que el servicio NEMO avise a IKEv2 cada vez que reciba un "Router Advertisement" (RA), y espere a que IKEv2 cree el túnel IPsec. Una vez creado, IKEv2 notificará a NEMO para que prosiga con su proceso de movilidad.

III. Handover en redes vehiculares

Hablar de VANETs implica inmediatamente hablar de movilidad y por consiguiente de handover, el proceso de pasar de una red de acceso a otra. Este proceso se realiza a varios niveles y dependiendo de eso será más o menos complejo. Podemos identificar cuatro tipos de handover:

- Handover intra-dominio intra-tecnología: Se cambia de estación base sin cambiar de tecnología ni de dominio. Es el caso más sencillo de handover, pues solo se realiza a nivel de enlace.
- Handover intra-dominio inter-tecnología: Se cambia de estación base cambiando sólo la tecnología. Se complica por hacer interactuar diferentes tecnologías con distintos comportamientos pero sigue siendo sólo a nivel de enlace.
- Handover inter-dominio intra-tecnología: Se cambia de estación base cambiando sólo el dominio. Este handover se produce a nivel de enlace y red, teniendo que poner en marcha los mecanismos de seguridad y movilidad necesarios.
- Handover inter-dominio inter-tecnología: Se cambia de estación base y tecnología, siendo este el más complejo de los handovers, pues cambia absolutamente todo.

Como hemos comentado, cada tipo posee un diferente nivel de complejidad. Nuestro enfoque será ir aumentando dicho nivel en el escenario de prueba paulatinamente, teniendo como objetivo el mejorar el rendimiento de procesos como por ejemplo, métodos de autenticación, tiempos de handover, etc.

1) Disponibilidad del servicio: Cada día la tecnología va avanzando más y por sus características, dichos avances, producen un gran impacto en los usuarios finales; quienes desean aprovechar al máximo estas nuevas funcionalidades. Pensando en que los dispositivos móviles deben estar provistos de la mayor cantidad de tecnologías, para que el usuario pueda hacer uso de ellas en las diferentes situaciones que se pueda encontrar, nos podemos encontrar con dispositivos de comunicación que poseen varias interfaces de red, que manejan diferentes tecnologías. Esta peculiaridad permite al usuario final utilizar diferentes alternativas para conectarse a la red con un mismo dispositivo. Por consiguiente, esto permite a los proveedores de servicios de red ofrecer sus servicios a través de estas diferentes tecnologías.

Como resultado de tener más de una interfaz de red con diferentes tecnologías, la disponibilidad del servicio es mayor. Es también mas eficiente ya que en cada situación la interfaz de red empleada será aquella que obtenga la mejor calidad de la señal o la que menor coste nos suponga económicamente.

Se plantea la situación de que durante el recorrido que realiza un vehículo podemos encontrar diferentes orografías, desde una llanura hasta un entorno urbano, cada una con sus características especificas. En una llanura nos encontramos con una orografía plana, así que es menos complicado el llegar a lograr una comunicaciones entre emisor y receptor con línea de visión directa (LOS). Este hecho, permite implementar infraestructuras de redes utilizando pocos recursos para alcanzar un gran área de cobertura. Sin embargo, en un área urbana se pueden encontrar una gran cantidad de edificaciones y otros tipos de obstáculos que no nos permiten una línea de visión directa entre emisor y receptor, por lo que es necesario utilizar comunicaciones que no la requieran(NLOS), lo que además permite la utilización de dispositivos inalámbricos, con tecnología de corto alcance; pero que por otro lado conlleva el uso de más recursos.

Cada tecnología inalámbrica tiene características que pueden ser consideradas una ventaja o desventaja, basándose estas consideraciones en el área donde será desplegada la tecnología. Por esta razón se justifica el uso de diferentes tecnologías, dependiendo de las características de la zona donde va a ser desplegada. Por ejemplo, WiMAX[11] es la mejor

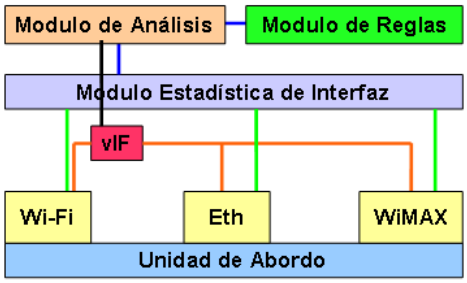


Fig. 12. Diagrama de la estructura de la OBU.

opción para zonas abiertas y donde el vehículo puede recorrer distancias largas en un corto espacio de tiempo. Sin embargo, UMTS es la mejor opción si el vehículo se mueve a través de una zona con muchos obstáculos a su alrededor. Finalmente, WiFi[12] puede ser considerado como la mejor opción dentro de entornos urbanos, donde encontramos muchas edificaciones y una gran densidad de usuarios.

Seleccionando la tecnología propuesta en cada caso se ofrece una mayor eficiencia en el aprovechamiento de los recursos. Debido a esto, es importante encontrar una relación equilibrada entre la tecnología empleada y el entorno en el cual se encuentra localizado el usuario final. Con lo cual la relación de implantación de tecnologías inalámbricas, por los proveedores de servicios y los dispositivos móviles de los usuarios, guardan una estrecha relación.

Por ejemplo, imaginemos el caso de un vehículo que esta viajando a través de una autopista, y por tanto ha tenido que pagar un peaje. Este peaje pudiera incluir el servicio de conexión a Internet, que se ofrece a lo largo del trayecto. La tecnología WiMAX parece la más indicada para cubrir este trayecto. Sin embargo, durante el recorrido se encuentra con un áreas de servicio, la cual también ofrece el acceso a Internet pero mediante la tecnología Wi-Fi; debido a que en un área de servicios existe una concentración mayor de usuario, se hace necesaria una señal con más poder y un mayor ancho de banda.

Dicho vehículo debe estar equipado con una *Unidad de Abordo* (OBU), la cual es la encargada de rastrear los servicios que se encuentran en el entorno y hacer uso de aquel que tenga permiso, con mejor calidad y prestaciones. Esta selección debe ser transparente para el usuario, quien está ocupado conduciendo el coche. Para permitir esta transparencia, un conjunto de reglas deben ser establecidas previamente en la OBU; con el objetivo de configurar el comportamiento de esta selección dinámica de servicios.

2) Unidad de Abordo: A día de hoy, es más común encontrarnos con que los vehículos vienen provistos con ordenador de abordo, con unas capacidades de procesamiento cada vez más que aceptables para implementar una amplia gama de servicios, además de poder disponer de múltiples interfaces de red que manejan distintas tecnologías inalámbricas. Gracias a estas características es posible que la OBU realice mayor cantidad de funcionalidades y de manera más eficiente.

La parte software, como se puede observar en la figura 12, esta dividida en:

- Interfaz Virtual (vIF): sabiendo que la OBU debe poder aceptar conexiones a cualquiera de las tecnologías que posee, se hace necesaria una única interfaz de salida, con el objetivo de hacer independiente los procesos que se realizan por encima del nivel de red. Por lo tanto, la vIF nos proporciona un mayor nivel de abstracción para la solución de posibles problemas que se pudieran presentar como consecuencia del proceso de cambiar la tecnología de comunicación que se esta utilizando en un momento determinado.
- Módulo de Reglas (RM): como se ha comentado anteriormente, cada tecnología tienen sus propias características que en base a ciertas condiciones pueden ser consideradas una ventaja o desventaja. Para hacer un uso, lo más eficiente posible, es necesario establecer reglas que serán consideradas al momento en que la OBU tenga que realizar alguna acción, por ejemplo, seleccionar el mejor punto de acceso.
- Módulo de Estadística de Interfaz (IM): si se tiene una vIF que depende de la interfaz que se encuentre activa, en su momento, se hace necesario que un proceso esté recabando información de las diferentes interfaces físicas. Entre la información reunida está la disponibilidad de punto de acceso, fuerza de la señal, protocolo de seguridad, derecho de acceso. Esta información será procesada para tomar decisiones en base a las reglas preestablecidas.
- Módulo de Análisis (AM): ya se ha comentado que se tienen reglas para poder elegir que hacer cuando se presenten ciertas condiciones en el entorno. Por otra parte tenemos la información recabada del ambiente. En este modulo es donde se toma en consideración la información proporcionada por RM y IM para realizar las decisiones y llevar a cabo las acciones necesarias para hacerlas efectivas en el sistema.

Entre la reglas, más sobresalientes a grandes rasgos a ser consideradas, están:

- Derecho al acceso del proveedor de servicios.
- Tecnología que implementa la interfaz utilizada.
- Fuerza y calidad de la señal.
- Consideraciones para el escenario de handover intra-tecnologías.
- Consideraciones para el escenario de handover inter-tecnologías.

En este escenario la OBU puede ser considerado como un nodo móvil (MN), el cual inicia con una comunicación establecida. De manera transparente al usuario, la OBU puede cambiar de punto de acceso (AP), basándose en la infraestructura disponible. Una de las características que debe satisfacer la AM es la capacidad de anticiparse a los cambios futuros de AP, para reducir el tiempo en que se pueda interrumpir la comunicación (escenario de pre-autenticación). Como se muestra en la figura 13, durante el viaje el vehículo realizará

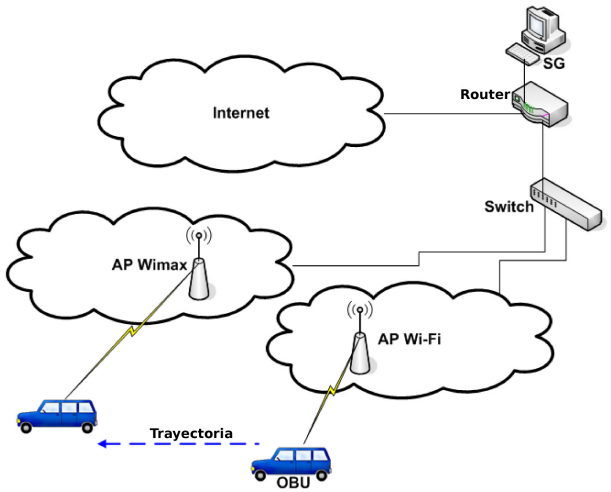


Fig. 13. Diagrama del escenario de movilidad.

cambios de AP en varias ocasiones y se le asignará en cada ocasión una IP diferente.

Por otra parte, la infraestructura también posee un servicio que contempla MR, MI y AM. Por consiguiente, la movilidad puede ser iniciada por el MN o por la infraestructura, que la misma recaba mucha más información. Esto permite a la infraestructura realizar ciertas acciones, como puede ser el balanceo de carga, ya que el MN sólo es consciente de su entorno.

El handover entre AP's tiene dos enfoques:

- Donde la tecnología de ambos AP's son iguales, siendo un handover intra-tecnología.
- Donde la tecnología de los AP's son diferentes, siendo un handover inter-tecnología.

En ambas situaciones descritas, es necesario que la aplicación pueda continuar transmitiendo sin problemas, por lo que es necesario diseñar un procedimiento que realice el handover entre los AP's sin que se interrumpa la comunicación que mantienen los protocolos de la capa de red y superiores. Para resolver esta situación se hace uso de la movilidad para IPv6, la cual permite a un MN mantener la misma dirección IP(HoA), a nivel de aplicación, mientras usa simultáneamente otra dirección IP(CoA), que puede ir cambiando sin problemas, proporcionándole la conectividad a Internet.

IV. Conclusiones y futuras líneas de trabajo

Después de crear nuestra VANET basada inicialmente en WiMAX[14], vemos que éste ha sido un primer gran paso requerido para abrir múltiples líneas de investigación, ya que nos sirve como escenario de pruebas para investigaciones relacionadas con las VANETs, así como las tecnologías de acceso empleadas. En la actualidad está sirviendo como escenario de prueba a proyectos como ITSSv6[21] a nivel europeo.

Hemos abierto el campo de estudio de métodos de autenticación basados en EAP, realizando una comparativa de tiempos de dos de ellos, EAP-TLS y EAP-FRM, concluyendo que EAP-FRM se presenta como un rapidísimo método de autenticación, impresionantemente cercano al tiempo utilizado por la simple autenticación mediante claves pre-compartidas, que por definición es la menos costosa de todas. En un futuro podrán entrar en la comparativa muchos más métodos basados en EAP.

Hemos descubierto que movilidad y seguridad están condenadas a entenderse y no trabajar por separado independientemente, ya que son dos servicios que dependen uno del otro. Hemos propuesto una solución que implica muy pocas modificaciones. Además nos ha servido para probar las implementaciones de NEMO (UMIP) y IKEv2 (OpenIKEv2), así como mejorar la compatibilidad entre ellos en diversos aspectos.

Un despliegue de una infraestructura inalámbrica de este calibre debe realizarse tomando en consideración las circunstancias del entorno que nos rodea, usando en cada caso la tecnología inalámbrica más apropiada. La responsabilidad de elegir la tecnología a usar en cada momento debe recaer en un dispositivo del vehículo (OBU) que basándose en una serie de reglas preestablecidas elegirá en todo momento la interfaz más apropiada para cada entorno o circunstancia. Esto libera al conductor de elegir continuamente a que red moverse.

Por último, decir que las VANETs son un excelente escenario para investigar el comportamiento de protocolos cuando la característica fundamental a potenciar es la movilidad.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el CICYT TIN2008-06441-C02-02 bajo el "Programa de ayuda a los grupos de excelencia de la fundación Séneca 04552/GERM/06".

Referencias

[1] Stephan Olariu, Michele C. Weigle, "Vehicular Networks from Theory to Practice", ISBN: 978-1-4200-8588-4 ,2009
[2] V. Devarapalli, F. Dupont, "Mobile IPv6 Operation with IKEv2 and the revised IPsec Architecture" IETF draft (2006)
[3] C.Kaufman, "Internet Key Exchange (IKEv2) Protocol," IETF RFC 4306, 2006.
[4] D. Harkins, D. Carrel, "The Internet Key Exchange (IKE)," IETF RFC 2409, 1998
http://www.ietf.org/rfc/rfc2401.txt
[5] S. Kent, R. Atkinson, "Security Architecture for the Internet Protocol", IETF RFC 2401, 1998
http://www.ietf.org/rfc/rfc2401.txt
[6] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, H Levkowetz, "Extensible Authentication Protocol (EAP)," IETF RFC 3748, June 2004.
[7] D. Simon, B. Aboba, R. Hurst, "The EAP-TLS authentication protocol", IETF RFC 2716, 2008
http://www.ietf.org/rfc/rfc2716.txt
[8] R. Marín, F. Pereñiquez, F. Bernal, A. Skarmeta, "Architecture for Fast EAP Re-authentication based on a new EAP method (EAP-FRM) working on standalone mode", IETF draft, 2009
http://tools.ietf.org/html/draft-marin-eap-frm-fastreauth-00
[9] OpenIKEv2,
http://openikev2.sourceforge.net/
[10] D. Johnson, C. Perkins, J. Arkko, "Mobility Support in IPv6" IETF RFC 3775, 2004.
[11] MIPL Mobile IPv6,
http://mobile-ipv6.org/
[12] Nautilus 6 Project,
http://www.nautilus6.org/
[13] WIDE Project,
http://www.wide.ad.jp/

[14] IEEE 802.16,
http://wirelessman.org/
[15] IEEE 802.11,
http://standards.ieee.org/getieee802/802.11.html
[16] IEEE 802.15,
http://www.ieee802.org/15/
[17] Descripción del protocolo GPRS
http://www.protocols.com/pbook/gprs.htm
[18] Descripción del protocolo UMTS
http://www.protocols.com/pbook/umts.htm
[19] C. Rigney, S. Willens, A. Rubens, W. Simpson., "Remote Authentication Dial In User Service (RADIUS)", IETF RFC 2865, Jun 2000.
[20] Free Radius,
http://freeradius.org/
[21] IPv6 ITS Station Stack for Cooperative Systems FOTs, ICT Research in FP7

Sesión 2.B
Gestión y optimización en entornos inalámbricos heterogéneos

Entorno de simulación para la evaluación de algoritmos de selección de acceso en redes inalámbricas heterogéneas

Johnny Choque, Ramón Agüero, Luis Muñoz
Departamento de Ingeniería de Comunicaciones
Universidad de Cantabria
Plaza de la Ciencia s/n, Santander
{jchoque, ramon, luis}@tlmat.unican.es

***Resumen*- El presente trabajo expone el diseño de un entorno de simulación flexible, escalable, y de fácil configuración, orientado hacia la evaluación de diversos algoritmos de selección de acceso. A diferencia de otras herramientas similares, el simulador que se presenta permite desplegar escenarios altamente configurables, con diferentes tipos de usuarios, de servicios, de terminales y de tecnologías; así como también con un elevado número de usuarios y estaciones base debido a las técnicas de abstracción incorporadas en su diseño, lo que permite simular complejos escenarios sin incrementar notablemente la carga computacional. A su vez, permite evaluar algoritmos que involucren varios operadores, llegando por tanto ha desarrollarse escenarios de simulación multi-acceso, multi-interfaz, multi-servicio y multi-operador.**

***Palabras Clave*- Simulación, Algoritmo de Selección de Acceso, multi-acceso, multi-interfaz, multi-servicio, multi-operador.**

I. INTRODUCCIÓN

El amplio abanico de tecnologías de acceso radio junto con la proliferación, cada vez más creciente, de terminales que incorporan varios de ellos, permiten la creación de una gran variedad de escenarios en los cuales, principalmente, los procesos de selección de acceso se han convertido en tareas cada vez más complejas. Este comportamiento se ve considerablemente incrementado debido a la existencia, cada vez mayor, de requerimientos por parte de los usuarios y los operadores a la hora de tomar una decisión en cuanto a la selección del acceso más adecuado dentro del abanico tecnológico que puede existir en el entorno. Evidentemente, la información con que se cuenta al momento de tomar dicha decisión tendrá una naturaleza 'local', esto es, limitada al usuario, en tanto en cuanto éste no podrá conocer las posibles consecuencias de su decisión sobre otros nodos de la red.

En este sentido, el presente trabajo describe el diseño de una herramienta de simulación que pretende cubrir todos los requisitos exigidos, tanto por parte de la red, los usuarios, los servicios, etc., con el fin de crear un entorno que permita evaluar de manera flexible y escalable los diferentes algoritmos de selección de acceso que existen en el estado del arte, sin exigir con ello una alta carga computacional. Para mostrar la potencia del simulador se plantea un escenario de red heterogéneo con una riqueza en cuanto a tecnologías de acceso, tipos de usuarios, servicios y operadores. Haciendo uso de un algoritmo de selección de acceso básico, los resultados obtenidos muestran claramente la idoneidad del simulador, sacando a la luz interesantes conclusiones en cuanto al comportamiento de la red analizada.

Para cubrir el objetivo anteriormente establecido, el artículo se ha estructurado en los siguientes puntos: la Sección II ofrece una perspectiva del trabajo relacionado y que se encuentra en la literatura actual, estableciendo las diferencias principales. La Sección III presenta los principios de diseño sobre los que se ha basado la implementación del simulador para cumplir con el requisito de escalabilidad. La Sección IV describe la arquitectura software del simulador, su funcionamiento interno, así como las herramientas que permiten darle flexibilidad. También se describe la estructura del algoritmo de selección de acceso que por defecto incorpora el simulador. En base a ello se especifica en la Sección V los parámetros que serán utilizados en el algoritmo durante la simulación. La Sección VI describe el escenario de red que se utilizará para analizar una serie de estrategias de acceso, cuyas prestaciones se presentan en la Sección VII. Finalmente, la Sección VII concluye el artículo, identificando varias líneas que se abren a partir del trabajo.

II. TRABAJO RELACIONADO

En la actualidad existen una gran variedad de herramientas de simulación que, a grandes rasgos, podrían ser clasificadas de acuerdo a las capas del modelo OSI en las que trabajan [1]. Por una parte se tiene aquellas herramientas diseñadas para caracterizar el rendimiento y las interacciones de la capa física y de enlace, tales como formas de onda, radio-frecuencia, propagación, etc., y, además, se cuenta con herramientas que involucran el resto de capas, principalmente enfocadas a evaluar el rendimiento de protocolos de red, por lo que se suelen catalogar como herramientas de simulación de red. Este trabajo se centra en esta última categoría, por tanto de aquí en adelante se entiende que sólo se hace mención a este tipo de herramientas.

A su vez, dentro de las herramientas de simulación, se puede encontrar aquellas alternativas comerciales y otras que son de código abierto. Ambas categorías presentan ventajas y desventajas que el investigador tendrá que sopesar a la hora de decantarse por una en concreto. Muchas herramientas de

simulación nacen de un proyecto de investigación y, por tanto, son habitualmente de código abierto, como GloMoSim [2] y OMNet++ [3], pero en algunos casos, evolucionan a versiones comerciales, como QualNet [4] y OMNEST [5], respectivamente. Otras herramientas de simulación que deben destacarse son ns2 (y su reciente versión ns3) [6] en la vertiente de código abierto y el simulador OPNET [7], como representante del grupo de herramientas comerciales.

El objetivo de todas ellas es facilitar el desarrollo de la simulación que se pretende llevar a cabo, ofreciendo módulos integrados con los que es posible interactuar rápida y dinámicamente; sin embargo, habitualmente tienen la desventaja de que, en muchos casos, se desconoce su funcionalidad interna y/o es difícil modificar su estructura para adaptarla a un caso particular. Llegado a este punto tiene sentido plantearse si es más conveniente el desarrollo de una herramienta de simulación personalizada. En este sentido, el diseñador tendría la ventaja de conocer las características, capacidades, y limitaciones específicas de la herramienta que ha diseñado. Podría enfocar el diseño de la herramienta hacia sus objetivos concretos y refinar los modelos y resultados de acuerdo a sus necesidades.

Está claro que el diseño de una nueva herramienta de simulación puede requerir un gran esfuerzo e inversión temporal, por lo que cobra mucha importancia las abstracciones que se adopten durante su desarrollo, con el fin de limitar la necesidad de llegar al mínimo detalle en cada aspecto. Incluso en las herramientas de simulación más conocidas se realiza algún tipo de abstracción, debido principalmente a las limitaciones de los recursos, e.g. memoria y tiempo de procesamiento, de los equipos sobre los cuales se ejecuta la simulación. Aunque existen técnicas de procesamiento paralelo y estrategias distribuidas [8] para realizar simulaciones a gran escala, el nivel de exigencia tanto de hardware como de software hace que sea una alternativa poco viable. Como contrapartida, adoptar algún tipo de abstracción conlleva poner en riego la precisión de la simulación. Por esta razón, es necesario llegar a un compromiso entre el grado de abstracción a realizar y la pérdida de precisión en la que se incurre. Anteriores trabajos han evaluado este tipo compromiso, como en [9], en el que a través de casos de uso se estudia el efecto del nivel de detalle de los modelos de propagación radio en las simulaciones de redes inalámbricas. En [9], los autores ponen de manifiesto que un modelo radio más simple puede ser más efectivo en casos donde el objetivo principal de la simulación no dependa fuertemente de las abstracciones de la capa física, aunque a pesar de ello siga siendo una parte importante de la misma.

El adoptar una mayor precisión a la hora de modelar el sistema que se desea simular, conlleva en muchos casos un incremento sustancial en el tiempo de ejecución. Esta situación se acentúa mucho más en el caso de trabajar con sistemas que involucren muchos terminales, estaciones base y, sobre todo, un gran número de restricciones, relacionadas con todos los elementos que forman parte de los algoritmos de selección de acceso. Por este motivo, hay un gran número de trabajos, ver, por ejemplo, [10] y [11], que se han visto en la necesidad de limitar el número de elementos a simular. En el presente trabajo se incluye desde la fase inicial de diseño del simulador los modelos de abstracción adecuados, para permitir evitar dichas limitaciones y, de esta forma, poder trabajar con un número sensiblemente mayor de terminales y estaciones base, así como un mayor y más variado conjunto de restricciones.

III. PRINCIPIOS DE DISEÑO

La herramienta desarrollada se denomina *multi-Constraint Access Selection in Heterogeneous Environment* (mCASE), y es un simulador gestionado por eventos que hace uso de la técnica de programación orientada a objetos (está implementado en C++). Permite la creación de diferentes escenarios de red en base a la especificación, en número y tipo, de los diversos elementos que forman parte de la simulación (tecnologías de acceso, terminales, estaciones base, usuarios, servicios, etc.). Tiene la capacidad de replicar los escenarios analizados con anterioridad, almacenando no solamente las características y el número de elementos de la simulación, sino también los eventos, con su respectiva información asociada, generados por el movimiento de los usuarios y los servicios que han sido inicializados durante la simulación. Esto permite evaluar el impacto de las diferentes estrategias bajo análisis sobre el mismo escenario.

Para lograr lo anterior es necesario, como se mencionó en la Sección II, adoptar una serie de abstracciones que permitan dotar al mCASE de la suficiente flexibilidad y escalabilidad, de tal forma que pueda hacer frente, con recursos computacionales asequibles, al ingente número de restricciones de los algoritmos de selección de acceso, así como también la gran cantidad de eventos generados por los elementos involucrados en la simulación.

A. *Abstracción del modelamiento de tráfico*

En muchos trabajos, como en [12], se ha estudiado el modelado de tráfico a diferentes escalas de detalle, llegándose a distinguir tres niveles: sesión, conexión y paquete. Cada nivel caracteriza un comportamiento diferente del tráfico, y se modela de manera particular. El nivel de sesión describe el comportamiento de cada usuario a la hora de conectarse y desconectarse del sistema. Este está indirectamente caracterizado por mCASE, mediante los patrones de movimiento del usuario. El nivel de paquete caracteriza la distribución de los paquetes dentro de cada conexión. Se decide que éste no es necesario para el caso de mCASE, porque el objetivo no es estudiar el comportamiento interno del tráfico. Por tanto, se modelará el tráfico a nivel de conexión, en el que se caracteriza la distribución de las conexiones que cada usuario (independiente para cada tipo de servicio) realiza. Por otra parte, con el fin de abstraer las diferentes medidas de carga que cada tecnología puede implementar, se define una unidad de capacidad genérica y discreta, denominada unidades de tráfico (*Traffic Unit*, TU) [13], utilizada para caracterizar tanto la capacidad de las estaciones base como la de los servicios solicitados.

B. *Abstracción del modelo de propagación radio*

Teniendo en cuenta que el objetivo de mCASE no es estudiar de manera precisa el canal de propagación, sino que se centra en la evaluación de algoritmos de selección de acceso [14], se propone abstraer los diferentes modelos de propagación a utilizar en el simulador. Esto implica que se utilizarán modelos sencillos pero que en esencia deberían representar, en la medida de lo posible, las características

más relevantes de un modelo más preciso. Esta estrategia ya ha sido utilizada en otras simulaciones, como en [15], en el cual debido a la intrínseca complejidad para modelar un sistema WiMAX se propone un modelo razonablemente sencillo pero equivalente en gran medida a un modelo completo.

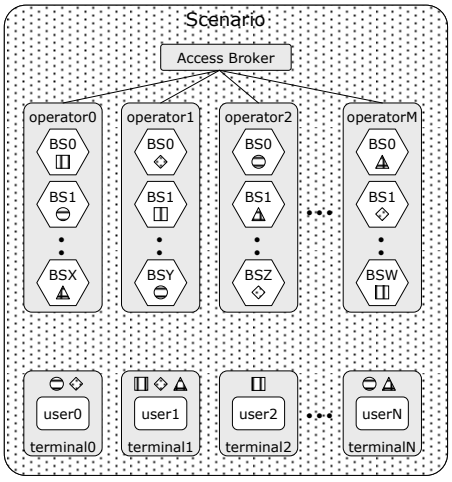


Fig. 1. Diagrama Funcional del simulador mCASE

IV. ARQUITECTURA DEL SIMULADOR

El entorno de mCASE esta constituido por diversas clases c++ relacionadas entre sí y que cumplen un papel específico dentro del simulador. Tal como se muestra en la Fig. 1, la clase *scenario* es la que aglutina todos los objetos que forman parte de mCASE. Así, almacena toda la información de los terminales y estaciones base que son creados en cada proceso de simulación, y también coordina la interacción entre el resto de elementos. Durante la fase de despliegue de la red se crean todos los objetos que representan a los terminales y estaciones base (*Base Station,* BS) que formarán parte de la simulación. Aunque a cada usuario (*User*) se le asocia un único *Terminal*, los dos objetos mantienen sus propias propiedades. Cada BS incorpora una única tecnología de acceso radio (*Radio Access Technology*, RAT), mientras que por el contrario cada terminal puede incorporar una o más RATs. Además, durante la fase de despliegue, se asigna a cada BS el operador al que pertenece, pudiendo cada operador tener asociado un conjunto diferente de BSs, tanto en número como en tipo. Para poder analizar situaciones en las que un terminal tiene que llevar a cabo un traspaso entre BS de diferentes operadores y existen políticas que permiten la interacción entre ellos, se ha incorporado en la arquitectura un *Access Broker*, que se encargará de gestionar adecuadamente las estrategias cooperativas entre operadores Así, mCASE no sólo está diseñado para replicar un entorno multi-RAT sino también uno del tipo multi-operador.

A. Configuración del simulador

mCASE es una herramienta de simulación flexible, escalable y fácil de configurar. Esto es debido a que permite especificar todos los parámetros de la simulación a través de un fichero de configuración general denominado *mCASE.cfg*. En dicho fichero se agrupan, a través de diversas secciones, todas las propiedades de cada uno de los objetos involucrados en la simulación; asimismo se definen los valores de otros parámetros necesarios para el simulador.

Cada tecnología radio empleada se modela con un objeto RAT, caracterizado por el radio de su área de cobertura (*Range*) y por la carga total que puede soportar (*Capacity*), en TUs. La posibilidad de utilizar diferentes tipos de RAT permite crear, por ejemplo, escenarios urbanos con una amplia variedad de tecnologías radio o, por otra parte, escenarios rurales en los que existen, por ejemplo, escasas estaciones base GSM y algunas más del tipo WLAN.

Los diferentes tipos de terminales son implementados a través del objeto *Terminal*. La propiedad *Probability* de este objeto permite indicar qué porcentaje de cada tipo de terminal existirá en la simulación. Como cada terminal se diferencia por los RATs que incorpora (lista de *RATid*) se pueden incluir en la simulación una variada gama de tipos de terminal, desde los más sencillos hasta los más sofisticados. De manera similar al objeto *Terminal*, la propiedad *Probability* del objeto *User* se utiliza para especificar qué porcentaje de cada tipo de usuario existirá en la simulación. Los diferentes tipos de usuarios se caracterizan por los servicios que son capaces de utilizar, permitiendo de esta forma incorporar al simulador usuarios con diferentes requerimientos de tráfico. La asociación entre un usuario particular con un terminal específico es el resultado de la creación de un objeto *userTerminal*, el cual permite implementar en el simulador usuarios con diferentes necesidades de tráfico y conectividad, como uno ejecutivo que podría utilizar servicios avanzados de videoconferencia por una RAT, mientras que sincroniza sus datos con la nube a través de otra, o un usuario más tradicional, que únicamente utiliza el navegador a través de una sola RAT. En cuanto a los servicios asociados a cada tipo de usuario, cada uno de ellos está representado a través del objeto *Service*, que detalla adecuadamente las propiedades de este objeto, tales como tiempo entre llegadas (*Tia*), tiempo de servicio (*Ts*), capacidad requerida (TUs) y características adicionales (por ejemplo, si se trata de un servicio con requerimientos específicos de tiempo real, etc.); así, se pueden crear diferentes tipos de servicios como video, voz y datos. Finalmente, mCASE ofrece la posibilidad de que cada usuario tenga diferentes patrones de movimiento. Esto permite incorporar al simulador usuarios que pueden ir en vehículos u otros que se desplazan como peatones. Para ello se emplea el objeto *movement*, que incorpora una amplia gama de propiedades, permitiendo implementar diversos patrones de movimiento, como el conocido *Random WayPoint* y sus variantes.

Con respecto a las estaciones base, cada una de ellas está representada por el objeto *BS*. Estos incorporan un único RAT y mediante *Mindistance* se puede realizar un mejor despliegue, ya que permite mantener una distancia mínima entre BSs del mismo tipo. Por otra parte, cada operador controla un conjunto de BS de diferente tipo, lo que permite al simulador la posibilidad de implementar escenarios muy diversos, como el de un operador incumbente con un mayor número de BSs de tipo celular y otros menos tradicionales, que basan su negocio en el despliegue de BSs de tipo WLAN o WIMAX. En todo caso, cuando dos operadores se ven involucrados en el proceso de traspaso de un usuario, entra en juego el *Access Broker* (gestor de acceso). Por ello, dicho agente incluye a todos los operadores que forman parte de la

simulación e implementa las políticas y algoritmos que han de regir las operaciones entre operadores.

B. Funcionamiento del simulador

El diseño modular de mCASE permite añadir o modificar cualquier parte de su estructura, con la finalidad de incorporar al simulador nuevas características que permitan mejorar sus prestaciones. En general, el simulador está constituido por las fases descritas a continuación.

- *Despliegue de Terminales*. Durante esta fase se crean cada uno de los objetos *userTerminal* que serán incluidos en la simulación. A cada terminal se le asigna un identificador único, el tipo de terminal que es, el tipo de usuario que lo lleva, el operador al que está suscrito y un patrón de movimiento. El tipo de terminal se asigna mediante una variable aleatoria uniformemente distribuida, en base a las probabilidades asignadas a cada tipo de terminal en el fichero de configuración. De manera similar sucede con la asignación del tipo de usuario y el tipo de movimiento. El tipo de operador también es asignado mediante una función aleatoria uniformemente distribuida pero en base a las probabilidades especificadas en la sección *MarketShare* del mCASE.cfg, que especifica la distribución del mercado entre operadores, de tal forma que se puedan crear escenarios con algún operador tradicional, con mayor cuota de mercado, de manera sencilla. Finalmente, a cada terminal se le asigna una posición inicial dentro de los límites del área de simulación, y se especifican los parámetros de movimiento según el tipo asignado, así como los correspondientes parámetros de servicio, que se relacionan indirectamente con el tipo de usuario.
- *Despliegue de Estaciones Base*. Para el despliegue de las estaciones base se crean los correspondientes objetos *basestation*, cada uno de ellos representado por un identificador único, el tipo de BS, y el operador al que pertenece. Cada operador tiene asignado un número específico de cada tipo de estaciones base, por lo que el despliegue consiste básicamente en asignar su posición dentro del área de la simulación. Para cada operador se tiene cuidado de mantener una distancia mínima entre sus estaciones base del mismo tipo, de acuerdo al parámetro *Mindistance*.
- *Creación de patrones de movimiento y servicios*. Antes de iniciar la simulación, se generan, para cada usuario, todos los eventos que representan los movimientos que realizará durante toda la simulación. Cada movimiento tiene asignado una serie de parámetros que lo caracteriza (identificador, posiciones inicial y final, dirección, velocidad, etc.) y el evento que lo sitúa temporalmente, almacenado en la cola de eventos única de mCASE. De manera similar, para cada usuario se crean todos los eventos correspondientes a los servicios que utilizará durante la fase de despliegue. Según el tipo de usuario asignado a cada terminal se establece un conjunto de tipos de servicios que representan diferentes patrones de tráfico. Para cada usuario se crean simultáneamente los servicios que le han sido asignados y que están representados por los siguientes parámetros: identificador, tipo y estado (*on, off*) del servicio, y un evento que indica el cambio de estado en el tiempo.
- *Inicio de la simulación*. El gestor de eventos almacena todos los eventos que han sido generados durante la fase de despliegue. Los almacena en orden cronológico en función del tiempo en el que deben dispararse. Al inicio de la simulación se selecciona el primer evento de la cola y se ejecuta la tarea correspondiente a la misma. Dependiendo del tipo de evento, algunos podrían generar otros eventos que actuarían posteriormente, por lo que serían almacenados de igual forma en la cola de eventos. Los eventos que indican el inicio de un servicio son un caso especial, ya que implícitamente implica solicitar recursos de una estación base. Por tanto, estos eventos inician el proceso de selección de acceso que será explicado más adelante. Cuando todos los eventos hayan sido atendidos o se haya alcanzado el tiempo límite de la simulación, se finaliza la ejecución y se recogen las estadísticas adecuadas en los ficheros de salida que hayan sido establecidos en el fichero de configuración.

C. Proceso de selección de acceso

Cuando un servicio asignado a un terminal entra en el estado activo implica que comienza a generar tráfico de acuerdo a las características de dicho servicio. Por tanto es necesario solicitar recursos a la red para satisfacer dicho servicio, iniciándose para ello un proceso de selección de acceso. La estrategia adoptada para este proceso ha surgido de las ideas planteadas en el proyecto *Ambient Networks* [16] en el cual los autores han participado, y que implica las siguientes fases:

- *Detección de accesos*. En función de la posición actual del terminal se determinan las estaciones base con las que tiene cobertura, sin tener en cuenta, en esta fase, el tipo de operador al que pertenecen, o si dispone o no de recursos suficientes para satisfacer al servicio. Es decir, se construye un conjunto de estaciones base teniendo en cuenta únicamente la conectividad con dicho terminal, denominado Conjunto Detectado (*Detected Set*, DS).
- *Validación de accesos*. Tomando como entrada el DS, en este punto se aplican las diversas políticas que pueden tener los operadores sobre las estaciones base. En función del tipo de política aplicada, el DS puede reducirse, filtrando aquellas BSs que no la cumplan, o también variar algunos parámetros de las BSs. Así, por ejemplo, puede que un operador aplique una política de seguridad restrictiva que el terminal no pueda cumplir y que, por tanto, tenga que descartarlo, o puede que se apliquen políticas de precio en función de la carga actual de cada BS y que, por consiguiente, se tenga que actualizar el parámetro de precio de cada BS. Para el último caso, ese tipo de políticas involucran las BSs de diferentes operadores, por lo que entraría en juego el *Access Broker*. En resumen, esta fase hace un refinamiento del DS y valida los diversos parámetros de las BSs seleccionadas, por lo que recibe el nombre de Conjunto Validado (*Validated Set*, VS).
- *Accesos Aspirantes*. En esta fase radica la inteligencia del proceso de selección de acceso. El simulador es lo suficientemente flexible para incorporar diversas estrategias o algoritmos de selección de acceso, incluso aquellos más elaborados, como los basados en técnicas de decisión con múltiples atributos (*Multi-Attribute Decision Making*, MADM) [17]. El simulador incluye por defecto un algoritmo basado en la asignación de pesos a diversas restricciones, tanto de la red como del terminal, cuyas particularidades serán explicadas con más detalle en la

siguiente sección. El resultado de esta fase es un conjunto de BSs ordenadas de forma descendente en función de la valoración obtenida por el algoritmo aplicado. Cada una de las BSs de dicho conjunto es un candidato en potencia a ser elegido como el acceso más adecuado, por lo que recibe el nombre de Conjunto Candidato (*Candidate Set*, CS).

Finalmente, para determinar la BS que atenderá la solicitud de recursos del terminal, cada una de las BSs del CS, partiendo de la que obtuvo una mayor valoración, es interrogada, para averiguar si dispone de los recursos exigidos por el servicio. Si es así, se reservan los recursos correspondientes en dicha BS, y en caso contrario se continúa la búsqueda con la siguiente BS del CS. Si se llegara a interrogar a todas las BSs del CS sin que ninguna cumpla con los recursos exigidos, entonces se rechaza la solicitud de conexión, entendiendo que el terminal no tiene a su alcance ninguna BS capaz de satisfacer la demanda particular del servicio.

V. Algoritmo genérico de selección de acceso

El simulador incluye un algoritmo de selección de acceso que se basa en la utilización de una función de coste Φ_{ij}, entre el usuario *i* y la estación base *j*, constituida por la suma modulada, a través de pesos, de las diversas restricciones tanto de la red como de los terminales. A cada restricción se le aplicará un peso diferente para, finalmente, seleccionar aquella estación base que maximice la función de coste empleada. La utilización de pesos permite dar un mayor o menor grado de importancia a una determinada restricción dentro de la función de coste con la finalidad establecer diferentes políticas en el algoritmo de selección de acceso. Las restricciones representan, por su parte, determinados aspectos que tienen que ver principalmente con las preferencias que un usuario podría tener a la hora de decantarse por una alternativa u otra, u otros que podrían ser de interés para el operador. En particular, se han tenido en cuenta las restricciones que se presentan seguidamente.

- *Operador preferido*. Con este parámetro se pretende reflejar la predisposición que los usuarios tienen a conectarse, en caso de que sea posible, con su operador preferente (η_i) debido, por ejemplo, a la existencia de un contrato, mejores tarifas, etc. Este parámetro dependerá del operador asociado (ζ_j) a la BS que esta siendo evaluada. Se utilizará B_{ij} en la función de coste para modelar este parámetro, definida como:

$$B_{ij} = \begin{cases} 1 & si\ \eta_i = \zeta_j \\ 0 & en\ caso\ contrario \end{cases} \quad (1)$$

- *Traspasos*. Una vez que un usuario esté conectado a una estación base, preferirá mantener la conexión con la misma el mayor tiempo posible, de manera que no se incurra en la degradación (sobrecarga) que podría generar un proceso de traspaso. De esta manera, conociendo la estación base con la que el usuario se había conectado previamente, se define el parámetro Γ_{ij} como:

$$\Gamma_{ij} = \begin{cases} 1 & si\ usuario\ i\ tenía\ conexión\ con\ BS\ j \\ 0 & en\ caso\ contrario \end{cases} \quad (2)$$

- *Calidad del enlace*. A la hora de seleccionar entre varias alternativas de acceso, uno de los elementos que tradicionalmente más se emplean es la calidad del enlace radio. Evidentemente se trata de un aspecto que depende fuertemente de la tecnología radio y del modelo de propagación empleados. En general, se puede asegurar que se corresponde con una función decreciente con la distancia a la estación base (d_{ij}); en este caso, se utilizará una función triángulo [18], que tome el valor máximo (1) cuando el terminal se encuentra en la misma posición que la estación base y el mínimo (0), justo en el límite de su área de cobertura (ω_j), de tal manera que se define el parámetro Δ_{ij} como:

$$\Delta_{ij} = \begin{cases} 1 - \dfrac{d_{ij}}{\omega_j} & si\ d_{ij} < \omega_j \\ 0 & en\ caso\ contrario \end{cases} \quad (3)$$

- *Carga*. Posiblemente sea el aspecto que en mayor medida condiciona el comportamiento que tendría la red a la hora de determinar el CS; lo que se pretende es balancear la carga de las diferentes estaciones base, para lo que se utiliza la capacidad relativa (θ_j) de cada una de ellas, de manera que cuando todos sus recursos estén disponibles (θ_{max}) se le asigna un valor de 1, reduciéndose de acuerdo a una función triangular hasta 0 cuando toda la capacidad esté siendo utilizada. Para este parámetro se define E_{ij} como:

$$E_{ij} = \begin{cases} 1 - \dfrac{\theta_j}{\theta_{\max}} & si\ \theta_j \le \theta_{\max} \end{cases} \quad (4)$$

A partir de los parámetros anteriores, se define una función de coste (Φ_{ij}), que los combina, permitiendo establecer una clasificación de las estaciones base disponibles:

$$\Phi_{ij} = \beta \cdot B_{ij} + \gamma \cdot \Gamma_{ij} + \delta \cdot \Delta_{ij} + \varepsilon \cdot E_{ij} \quad (5)$$

Para que dicha función de coste sea lo más flexible posible, se modula cada uno de los aspectos anteriormente mencionados por un peso; así, β promueve el empleo de una estación base del operador preferente; γ se utiliza para minimizar la necesidad de cambiar de estación base; δ potencia el uso de una BS que maximice la calidad del enlace radio; finalmente ε busca balancear la carga de las estaciones base, restringiendo el acceso con las estaciones saturadas. En las definiciones anteriores se puede comprobar que todas las variables que se han definido están acotadas en [0, 1], por lo que si se establece que la suma de los cuatro pesos sea igual a la unidad, $\beta + \gamma + \delta + \varepsilon = 1$, se puede acotar la función de coste en el mismo intervalo.

VI. Uso de mCASE en el análisis de escenarios de red

El amplio abanico de parámetros que pueden ser configurados en el simulador le da la capacidad de admitir una gran diversidad de escenarios de red para su análisis. Como punto de partida, se plantea un escenario que intenta reflejar un escenario heterogéneo, no solo debido a las tecnologías presentes, sino también a la presencia de varios operadores. En concreto, se utilizarán tres tipos de tecnologías, cuyas características de resumen en la Tabla 1. El primero de ellos emula una tecnología de características más similares a las comunicaciones inalámbricas tradicionales (GSM), pues tiene una cobertura sensiblemente superior y, además, también presenta una capacidad mayor. Las otras dos tecnologías son más cercanas a puntos de acceso WLAN, con alcance y capacidad claramente inferiores. Recordar que para modelar la capacidad se utiliza

la abstracción descrita en la Sección III.A, que se basa en unidades de tráfico discretas, TUs.

Operador	RAT ID	Cobertura (m)	Capacidad (TU)	Número de BSs	Tecn. emulada
B	0	80	5	20	WLAN-B
B	1	60	8	30	WLAN-A
A	2	600	20	2	GSM

Tabla 1. Tecnologías empleadas durante el análisis

Se supone, además, que coexisten dos operadores. El primero de ellos (A) es el operador incumbente, que gestiona las estaciones base con tecnología celular, mientras que el segundo (B) sería un nuevo operador, que ofrece un acceso menos convencional, a través de las tecnologías WLAN-B y WLAN-A.

Se considera además un área de 1000×1000 m^2, en el que las estaciones base se despliegan sin una planificación previa, según un despliegue aleatorio, aunque se limita la distancia entre ellos (siempre que sean del mismo operador y de la misma tecnología). Con todo esto, la red que se analizará es la que se muestra en la Fig. 2, en la que se puede ver que las 2 estaciones base GSM cubren gran parte del área bajo análisis, sin existir demasiado solapamiento entre ellas. El área cubierta por el operador (B) es sensiblemente menor pero permite dar cobertura a una zona significativa no cubierta por el operador incumbente (esquina superior izquierda).

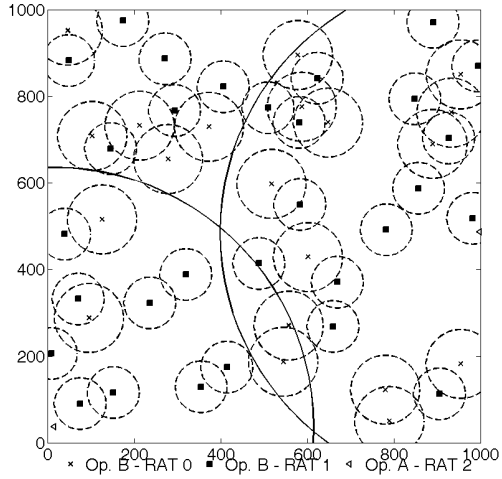


Fig. 2. Despliegue de red empleado durante el análisis

Se despliegan 200 usuarios, asumiendo que el 60% son clientes del operador A, mientras que el resto lo son del menos tradicional, B. Se definen, además, tres tipos de terminales: uno básico que incorpora únicamente como tecnología de comunicación inalámbrica una interfaz GSM; otro de tipo medio, con dos interfaces de red inalámbrica, una GSM y otra WLAN-A; y finalmente una del tipo avanzado que incorpora las tres tecnologías involucradas en la simulación. La asignación de un tipo de terminal específico a cada usuario se hace en función de porcentajes de probabilidad; en este caso se especificaron 0.30, 0.40 y 0.30 para la asignación de terminales del tipo básico, medio y avanzado, respectivamente. Se definen también dos tipos de usuarios, uno domestico y otro profesional, en función de los tipos de servicios que cada usuario utiliza durante la simulación. Para ello se modela todo el tráfico según modelos de Poisson, definiéndose uno o más servicios que cada usuario puede utilizar de forma simultánea, en función de la configuración particular de cada escenario. En el escenario que se analiza en este artículo, se han empleado tres servicios diferentes, cuyas características se resumen en la Tabla 2. En base a dichos servicios el usuario domestico representa el 70% del total de usuarios, utilizando los servicios de voz y datos; mientras que el resto esta constituido por los usuarios profesionales que, además de voz y datos, emplean el servicio de video. Los usuarios inicialmente se sitúan aleatoriamente en el área bajo análisis y, posteriormente, se mueven libremente según el modelo *Random Waypoint* [19].

Servicio ID	Tia (s)	Ts (s)	Capacidad (TU)	Tipo
0	120	60	1	Datos
1	120	180	1	Voz
2	200	180	2	Video

Tabla 2. Características de los servicios utilizados en el análisis

Una vez presentado el escenario sobre el que se llevará a cabo el análisis, la Tabla 3 presenta las diferentes estrategias de selección de acceso. Como se puede ver, se va modificando el valor que se le otorga a cada peso, de manera que cada estrategia dará prioridad a alguno de los parámetros que se han presentado anteriormente. En este sentido, la estrategia A proporciona el mismo peso a todos los parámetros con el objeto de analizar el efecto de una distribución igualitaria de los pesos dentro de la función de coste. Las estrategias B, C, D y E se enfocan (cada uno de ellas) en un único parámetro, con la idea de estudiar el grado de influencia que cada uno de ellos tienen en la función de coste de forma individual. Finalmente, las estrategias F, G, H, I, J y K favorecen a dos de los parámetros anteriormente mencionados, con la finalidad de evaluar el impacto que produce la combinación de dos parámetros de forma simultánea en la selección de acceso.

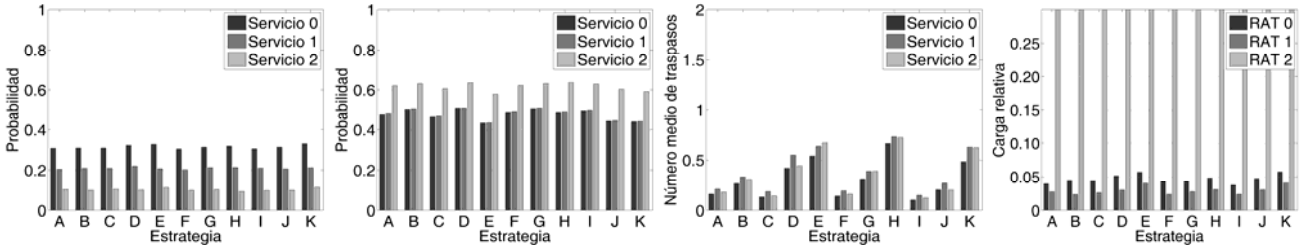
Param.	A	B	C	D	E	F	G	H	I	J	K
β	0.25	1	0	0	0	0.5	0.5	0.5	0	0	0
γ	0.25	0	1	0	0	0.5	0	0	0.5	0.5	0
δ	0.25	0	0	1	0	0	0.5	0	0.5	0	0.5
ε	0.25	0	0	0	1	0	0	0.5	0	0.5	0.5

Tabla 3. Estrategias de selección de acceso analizadas

VII. RESULTADOS

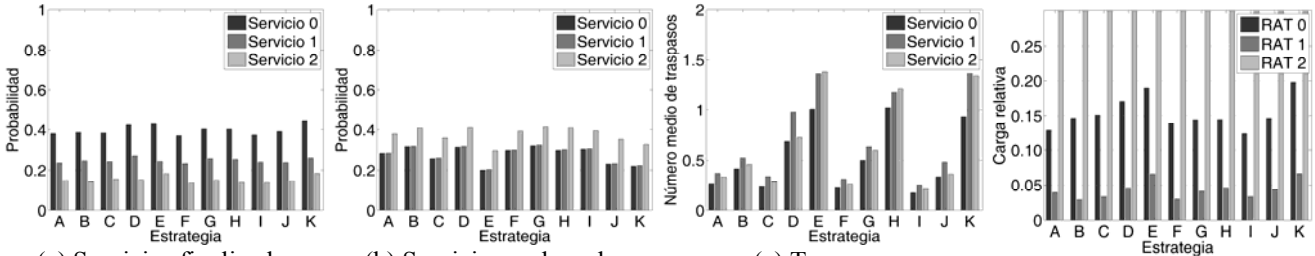
A continuación se describen los resultados obtenidos al emplear las 11 estrategias de selección de acceso que se han presentado previamente. La simulación tiene una duración de 2000 segundos, realizándose 100 simulaciones independientes para cada estrategia, promediando el resultado final, para asegurar la validez estadística de los resultados.

Para valorar el comportamiento de las diferentes alternativas, se pueden extraer de los resultados del simulador un conjunto amplio de métricas, pero siendo el objetivo del presente trabajo el resaltar la flexibilidad y capacidades de mCASE se muestran en la Fig. 3 y Fig. 4 aquellas métricas que dejan más patentes dichas



(a) Servicios finalizados (b) Servicios rechazados (c) Traspasos (d) Carga

Fig. 3. Prestaciones de las estrategias de acceso con tres tipos de terminales



(a) Servicios finalizados (b) Servicios rechazados (c) Traspasos (d) Carga

Fig. 4. Prestaciones de las estrategias de acceso solo con tipo de terminal avanzado

características. Ambos resultados han sido obtenidos utilizando siempre el mismo despliegue de BSs (Fig. 3), así como la misma configuración en lo que se refiere a las RATs, servicios, modelo de movimiento, estrategias de selección de acceso, etc. La única diferencia radica entre ambos conjuntos de medidas es que para el caso de la Fig. 3 se consideraron los tres tipos de terminales, mientras que en la Fig. 4 todos los usuarios utilizaron un único tipo de terminal, siendo aquel que incorpora las tres tecnologías involucradas en al simulación.

La Fig. 3(a) muestra la probabilidad de que un servicio finalice con éxito, y de forma similar la Fig. 3(b) muestra la probabilidad de que un servicio sea rechazado. El simulador también extrae información acerca de la probabilidad de que un servicio no finalice adecuadamente una vez ha sido cursado (*Dropped Services*). En la Fig. 3(a) se puede ver que la probabilidad de que un servicio finalice adecuadamente no se ve afectada en gran medida por las diferentes estrategias, aunque está claro que dentro de cada estrategia los diferentes servicios tienen una mayor probabilidad de finalizar con éxito cuanto menos requerimientos exijan a la red (servicio 0 – datos), decreciendo dicha probabilidad hasta el servicio más exigente (servicio 2 – vídeo). Por el contrario, la Fig. 3(b) muestra una mayor influencia de las estrategias en los servicios rechazados. Se puede ver que para las estrategias E, J y K, los servicios sufren una menor probabilidad de rechazo, y eso es debido a que en dichas estrategias se da mayor peso al parámetro de balanceo de carga, con lo que se consigue mantener una mayor capacidad libre en las diferentes BS. Siguiendo este razonamiento, la estrategia H también tendría que mostrar una menor probabilidad de rechazo, pero en este caso dicha estrategia está influenciada por el peso compartido que tiene con el parámetro de operador preferente, que influye negativamente, ya que los usuarios que pertenecen al operador incumbente son más, y éste, como se podrá ver más adelante, se satura rápidamente. La Fig. 3(c), que representa el número medio de traspasos por servicio, pone de manifiesto una influencia mayor de las diferentes estrategias en los resultados. Es necesario aclarar que para esta métrica sólo se han tenido en cuenta los servicios finalizados correctamente. Se puede observar que las estrategias C, F, e I, muestran un menor número medio de traspasos, debido a que en dichas estrategias se da mayor importancia al parámetro de traspaso. El mismo resultado no se produce en la estrategia J, en la que el efecto combinado con el parámetro de carga hace que el número medio de traspasos se incremente ligeramente. Finalmente, en la Fig. 3(d) se muestra la carga relativa que tienen las BS (divididas por su RAT). Con esta métrica se pretende analizar el uso que se le da a cada tecnología, permitiendo a los operadores establecer la posibilidad de incorporar o no nuevos usuarios. Debido a que la distribución del mercado en el escenario analizado asigna un 60% de los usuarios al operador incumbente, que gestiona las BSs con tecnología de mayor cobertura (GSM - RAT 2), éstas se saturan, llegando a soportar más del 90% de carga relativa para todas las estrategias,. Se puede ver que la carga relativa en las RATs del operador novel es bastante baja, aproximadamente 5%, debido principalmente a la poca cobertura que ofrecen, en conjunto, todas las BSs que pertenecen a este operador, a pesar de la existencia de un mayor número de ellas. Este efecto se ve incrementado por el hecho de no todos los usuarios sean capaces de conectarse con dichas RATs, por no disponer de terminales apropiados. Enfocando el análisis en las tecnologías del operador novel (RAT 0, RAT 1), se puede ver que para las estrategias E y K, el porcentaje de carga relativa supera el 5%. Esto es debido a que en dichas estrategias se otorga un mayor peso al balanceo de carga entre las BSs. Las estrategias H y J deberían seguir esta tendencia pero se ven afectadas de forma *negativa* por el peso combinado de los parámetros de operador preferente y traspaso, respectivamente.

La Fig. 4 muestra el impacto que produce el cambio de un único parámetro en la configuración del escenario analizado. El uso de un tipo de terminal por parte de todos los usuarios, con un abanico de interfaces de comunicación que cubre todas las tecnologías del escenario, hace que los resultados en los parámetros equivalentes de la Fig. 3 se vean

notoriamente afectados. La probabilidad de que un servicio finalice con éxito se incrementa en más de un 10% para todos los tipos de servicio, tal como se muestra en la Fig. 4(a). De forma similar los servicios rechazados (Fig. 4(b)) se reducen aproximadamente un 50% en todas las estrategias, debido a que en este caso se tiene un mayor conjunto de alternativas de acceso con el terminal avanzado. En la Fig. 4(c) se puede observar que el número medio de traspasos se ve incrementado notablemente en las estrategias E, H y K, debido a que en dichas estrategias se otorga una mayor importancia a la carga libre de las BSs, permitiendo al usuario pasar fácilmente de una BS a otra al disponer de un mayor número de alternativas de acceso en su terminal. Esta misma tendencia debería seguirse en la estrategia J, pero el efecto combinado de los pesos con el parámetro de traspaso hace que el número de traspasos se vea reducido, viéndose de esta forma la influencia de los pesos a los parámetros dentro del algoritmo. Finalmente, en la Fig. 4(d) se puede observar que para el caso del RAT 0 el porcentaje de ocupación de las BSs con dicha tecnología se ha incrementado notablemente, siendo el incremento algo menor para RAT 1, ya que la cobertura de ésta es algo menor; destacar que, al igual de lo que ocurría en los resultados presentados en la Fig. 3 la carga relativa de las BS del operador A es siempre superior al 90%, favorecidos por la cobertura (prácticamente global) que tienen.

VIII. CONCLUSIONES

En este trabajo se ha presentado una herramienta de simulación propietaria que se ha diseñado para analizar algoritmos en el ámbito de la selección de acceso en entornos de red altamente heterogéneos. Se han identificado las necesidades y requerimientos que llevan a apostar por el desarrollo de una herramienta propietaria, frente a la utilización de otras disponibles.

Para poner de manifiesto la validez y el funcionamiento de la herramienta, se ha presentado un primer análisis acerca de las prestaciones de varias estrategias de selección de acceso, que otorgan diferentes niveles de prioridad a una serie de figuras de mérito. Los resultados, además de validar la implementación llevada a cabo, permiten establecer grados de compromiso entre los diferentes parámetros a potenciar a la hora de decantarse por uno u otro acceso.

En el futuro se aprovechará el marco que proporciona mCASE para analizar exhaustivamente diferentes estrategias en el ámbito de la gestión de recursos en redes de acceso heterogéneas. Se analizarán estrategias de cooperación entre operadores, asignación de tarifas, algoritmos de selección de acceso, etc. Para un análisis más exhaustivo se contrastarán y corroborarán los resultados con los obtenidos a partir de un estudio más teórico, para el que se utilizarán diferentes herramientas matemáticas, como la programación lineal y la teoría de juegos.

AGRADECIMIENTOS

Los autores querrían expresar su agradecimiento al Gobierno de España por su financiación en el proyecto “Comunicaciones Cognitivas y Cooperativas sobre Entornos Heterogéneos”, C3SEM (TEC2009-14598-C02-01).

REFERENCIAS

[1] Kasch, W.; Ward, J.; Andrusenko, J., “Wireless network modeling and simulation tools for designers and developers”, IEEE Communications Magazine, Volume 47 , Issue 3, Publication Year: 2009, Page(s): 120 – 127.

[2] Xiang Zeng, Rajive Bagrodia, and Mario Gerla. “GloMoSim: A library for parallel simulation of large-scale wireless networks”. Proceedings of the 12th Workshop on Parallel and Distributed Simulations - PADS '98, May 26-29, in Banff, Alberta, Canada, 1998.

[3] A. Varga, “The OMNeT++ discrete event simulation system”. In European Simulation Multiconference (ESM’2001), Prague, Czech Republic, June 2001.

[4] Scalable Network Technologies, Inc., QualNet simulation software, http://www.scalable-networks.com/products/qualnet/, copyright (c) 2006-2010.

[5] Simulcraft Inc. OMNEST simulation software, http://www.omnest.com/, copyright (c) 1992-2010.

[6] The ns-3 network simulator, http://www.nsnam.org/.

[7] OPNET Technologies, Inc., Opnet Network R&D Simulator, http://www.opnet.com/solutions/network_rd/, copyright (c) 2010.

[8] Hyunok Lee; Manshadi, V.; Cox, D.,”High-fidelity and time-driven simulation of large wireless networks with parallel processing”, IEEE Communications Magazine, Volume: 47 , Issue: 3, Publication Year: 2009 , Page(s): 158 – 165.

[9] J. Heidemann, N. Bulusu, J. Elson, C. Intanagonwiwat, K. chan Lan, Y. Xu, W. Ye, D. Estrin, and R. Govindan, “Effects of detail in wireless network simulation”, In Proceedings of the SCS Multiconference on Distributed Simulation, pages 3–11, Phoenix, Arizona, January 2001. Society for Computer Simulation.

[10] Xing, B.; Nalini Venkatasubramanian, “Multi-constraint dynamic access selection in always best connected networks”, The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2005. MobiQuitous 2005. Publication Year: 2005 , Page(s): 56 – 64.

[11] de Sousa, V.A.; de O. Neto, R.A.; de S. Chaves, F.; Cardoso, L.S.; Cavalcanti, F.R.P., “Access selection with connection reallocation for multi-access networks”, International Telecommunications Symposium. Publication Year: 2006 , Page(s): 615 – 619.

[12] Klemm, A.; Lindemann, C.; Lohmann, M., “Traffic modeling and characterization for UMTS networks”, IEEE Global Telecommunications Conference, 2001. GLOBECOM '01. Volume: 3, Page(s): 1741 – 1746.

[13] Poyhonen, P.; Tuononen, J.; Haitao Tang; Strandberg, O., ”Study of Handover Strategies for Multi-Service and Multi-Operator Ambient Networks”, Second International Conference on Communications and Networking in China, 2007. CHINACOM '07. Publication Year: 2007 , Page(s): 755 – 762.

[14] Lucas-Estan, M.C.; Gozalvez, J.; Sanchez-Soriano, J., “Common radio resource management policy for multimedia traffic in beyond 3G heterogeneous wireless systems”, IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. Publication Year: 2008 , Page(s): 1 – 5.

[15] M. Miozzo, F. Bader, Accurate Modelling of OFDMA Transmission Technique using IEEE 802.16m Recommendations for WiMAX Network Simulator Design, in Proceedings of 2nd International ICST Conference on Mobile Networks and Management (MONAMI'2010), 22-24 September 2010, Santander (Spain).

[16] J. Lundsjö, et al., “A Multi-Radio Access Architecture for Ambient Networking”, in Proc. 14th IST Mobile & Wireless Communications Summit, Dresden, Germany, 19-23 June 2005.

[17] B. Xing y N. Venkatasubramanian. “Multi-constraint dynamic access selection in always best connected networks”. The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services. MobiQuitous 2005, July 2005.

[18] Poyhonen, P.; Tuononen, J.; Haitao Tang; Strandberg, O., ”Study of Handover Strategies for Multi-Service and Multi-Operator Ambient Networks”, Second International Conference on Communications and Networking in China, 2007. CHINACOM '07. Publication Year: 2007 , Page(s): 755 – 762.

[19] T. Camp, J. Boleng, V. Davies, “A survey of mobility models for ad hoc network research”. Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, 2(5):483–502, 2002.

Reducción de la latencia de handover en dispositivos multi-interfaz

David Gómez, Ramón Agüero y Luis Muñoz
Universidad de Cantabria, Santander, España
{**dgomez,ramon,luis**}**@tlmat.unican.es**

***Resumen*—En este artículo se presentan, de una forma eminentemente práctica, las posibilidades que aparecen con la proliferación de nuevos dispositivos capaces de acceder a Internet a través de diferentes tecnologías de acceso inalámbricas (WLAN, WiMAX, UMTS...). Esta heterogeneidad está llamada a ser uno de los pilares fundamentales de las redes de comunicaciones venideras, por lo que se hace necesaria la presencia de una serie de entidades capaces de afrontar los nuevos retos inherentes a este nuevo concepto de acceso inalámbrico. A pesar de que el paradigma ABC (Always Best Connected) ha suscitado el interés de la comunidad científica desde hace años, la mayor parte de los trabajos existentes se enfocan desde un punto de vista descriptivo (presentando arquitecturas) o se basan en simulación y/o emulación. Para este trabajo se ha tratado de ir un paso más allá y, partiendo de la arquitectura definida en el proyecto Mobilia del programa Celtic, se ha implementado una plataforma real donde se efectúan dos situaciones de handover diferentes: iniciado por el propio terminal del usuario o por la red. Del mismo modo, se reutilizará la plataforma para analizar cuantitativamente las mejoras que introduce el uso de dispositivos multi-interfaz, en términos de la reducción del retardo en los citados handovers o traspasos.**

***Index Terms*—Redes heterogéneas, Dispositivo multi-interfaz, Handover, IEEE 802.21, SCTP**

I. Introducción y objetivos

La llegada de las nuevas tecnologías inalámbricas permite al usuario disponer de un amplio abanico de tecnologías a la hora de conectarse a la red, dando lugar al concepto de redes de acceso heterogéneas, a partir del que surgen nuevos retos y desafíos. Uno de los más relevantes se basa en el paradigma ABC (*Always Best Connected*), que establece la necesidad de que el usuario final se conecte de manera transparente a la alternativa de acceso más adecuada, a partir de unos criterios y preferencias de usuario prestablecidos, los requerimientos de los servicios utilizados y las condiciones particulares de las redes disponibles. Además, surge la necesidad de establecer acuerdos cooperativos entre las diferentes entidades que componen el sistema. Los mecanismos existentes actualmente no satisfacen plenamente estos requisitos, necesitando normalmente la intervención directa por parte del usuario.

Para la realización de este trabajo se ha propuesto implementar una arquitectura capaz de hacer frente a la heterogeneidad que imperará en las redes de comunicaciones del futuro, ofreciendo al usuario el acceso más adecuado de una manera completamente automática. Esta propuesta es el eje principal del proyecto Mobilia (*Mobility concepts for IMT-Advanced*), perteneciente al programa CELTIC. En concreto, se presenta una plataforma real empleada para comprobar la viabilidad de la solución diseñada en el marco de dicho proyecto y que, además, se ha utilizado en el ámbito del proyecto CENIT mIO!.

Para ello se describen dos casos de uso, que tratan de ilustrar dos formas diferentes de iniciar un proceso de selección de acceso: en la primera es el terminal del usuario, ante la degradación de la calidad del enlace inalámbrico, el que decide comenzar un proceso de handover, mientras que en la segunda será la propia red quien solicite un traspaso debido a una situación de sobrecarga. Por otra parte, en el artículo se analizan cuantitativamente las ventajas que puede aportar el hecho de disponer de más de un interfaz inalámbrico, en términos de reducción de la latencia durante la ejecución de un handover.

Para ello el documento se ha estructurado como se detalla a continuación: la Sección II discute sobre aquellos trabajos que presentan conceptos y objetivos similares, centrándose en aquellos que han buscado una aproximación empírica; la Sección III presenta la arquitectura que ha sido diseñada en el marco del proyecto Mobilia para gestionar la selección de acceso en entornos de red heterogéneos. La Sección IV describe la implementación de la mencionada arquitectura a un escenario real, detallando los dos casos de uso que se han seguido para probar la viabilidad de la arquitectura. La Sección V presenta una serie de resultados obtenidos en dicha plataforma, con el fin de comprobar los beneficios del empleo inteligente de los dispositivos multi-interfaz a la hora de reducir la latencia durante un proceso de handover. Por último, la Sección VI concluye el documento, aludiendo algunas líneas que quedan abiertas para trabajos futuros.

II. Trabajo previo

Durante los últimos años, la presencia de las redes inalámbricas heterogéneas ha atraído la atención de la comunidad científica, dando como resultado un gran número de propuestas que tratan de abordar los desafíos que surgen con la aparición de estos escenarios. Algunas comparten muchas características con la arquitectura de Mobilia, por lo que es conveniente mencionarlas en este apartado.

Dos de las más completas son: *Common Radio Resource Management (CRRM)* y *Joint Radio Resource Management (JRMM)* (véase [1], [2] junto con sus respectivas referencias), así como *Multi-Radio Resource Management (MRRM)*([3] y sus referencias). Todas ellas se corresponden con arquitecturas que desempeñan las tareas de gestión de recursos en escenarios de redes heterogéneas. Funcionalmente, presentan una entidad encargada de obtener información de las capas inferiores de forma transparente, sin importar la tecnología subyacente. Por otro lado, una nueva entidad inteligente emplea toda esta información y, procesándola junto con otros requerimientos, se encarga de tomar las decisiones que estime oportunas para

ISBN: 978-84-694-5948-5

satisfacer las demandas del usuario (esto es, establecer una conexión con el elemento de acceso más adecuado).

De forma paralela, la mencionada heterogeneidad en las redes inalámbricas ha suscitado el interés de los cuerpos de estandarización más relevantes. En este sentido, el estándar IEEE 802.21 define el *Media Independent Handover Framework (MIHF)*, cuyo objetivo consiste en proporcionar la señalización que se utilizará durante el intercambio de información entre las entidades involucradas en un proceso de traspaso (o de selección de acceso) ([4], [5]). Una de las principales características de la arquitectura Mobilia es que fue diseñada para emplear la señalización IEEE 802.21 como el eje central para el intercambio de mensajes entre las entidades que la conforman. El proyecto ODTONE (*Open Dot Twenty ONE*) [6], desarrollado en la Universidad de Aveiro, presenta un estado avanzado en la implementación del estándar IEEE 802.21, contando además con la ventaja de ser un proyecto *open-source*.

El objetivo de este trabajo consiste en presentar una plataforma real que ha sido desarrollada con el fin de comprobar la viabilidad de la arquitectura propuesta; además, utilizando dicha implementación, se pretende proporcionar medidas ilustrativas que muestren el beneficio que se podría obtener al disponer de más de un interfaz inalámbrico. Concretamente, el artículo muestra una serie de medidas empíricas de la latencia observada durante un proceso de handover. Es importante remarcar que todos los resultados obtenidos están realizados con componentes reales, sin emplear emulación alguna.

Existen otros trabajos que han tratado de implementar de una forma tangible este tipo de arquitecturas. Por ejemplo, en [7] los autores presentan una plataforma real que refleja una *Arquitectura Multi Radio* que fue propuesta dentro del marco del proyecto Ambient Networks. Sin embargo, en este caso no se muestran resultados y las diferentes entidades no fueron incluidas en los elementos de acceso (ya sean en forma de punto de acceso o de estación base), limitando la posibilidad de establecer un handover iniciado por la red. Otros trabajos surgidos del mismo proyecto ([8], [9]), analizan la sobrecarga introducida durante un proceso de traspaso vertical, utilizado como solución de movilidad *Host Identity Protocol (HIP)*, y empleando tecnologías reales heterogéneas (concretamente IEEE 802.11 y 3G), pero de nuevo no se incorporan funcionalidades en los elementos de red, con lo que todos los handovers son iniciados por el usuario final. Su solución se basa en una entidad encargada de notificar los eventos que pudieran derivar en la necesidad de ejecutar un traspaso a las entidades interesadas, de una manera inteligente.

Otra aproximación completamente diferente consiste en el uso de complejas plataformas de medidas que emulan el comportamiento de las redes inalámbricas heterogéneas. Una de las más relevantes es la implementada en el ámbito del proyecto AROMA [10], que fue diseñada para analizar el rendimiento del ya mencionado CRRM. Con estas aproximaciones, el objetivo es realizar un exhaustivo análisis del rendimiento, supliendo el papel que normalmente desempeñan los simuladores, pero ofreciendo un mayor grado de flexibilidad y precisión[1]. Por su parte, el presente trabajo se basa en un escenario más concreto, donde pueden caracterizarse procedimientos específicos, estudiándose las mejoras que se producen a través de la utilización de diferentes tecnologías de descubrimiento y selección de acceso sobre escenarios reales. Puede encontrarse más información relacionada con la arquitectura Mobilia en [11] y [12].

[1]Para ello es necesario incorporar modelos para las fuentes de tráfico, los patrones de movilidad, etc.

III. La arquitectura de Mobilia

Mobilia trata de hacer frente a los retos que supone la llegada de los nuevos escenarios de comunicaciones inalámbricos, donde la heterogeneidad aparece como uno de los factores más relevantes. Uno de los objetivos principales consiste en dar soporte a una cooperación eficiente y transparente entre las diferentes redes de acceso, tratando de alcanzar el paradigma *ABC*. Por este motivo, se ha realizado un gran esfuerzo en construir una arquitectura que permita una selección de acceso a redes en escenarios heterogéneos. La estructura global se rige según una serie de principios, como la abstracción de las tecnologías subyacentes, la inclusión de métricas en los procedimientos de selección de acceso, y el empleo del marco de señalización, propuesto por el *Media Independent Handover* (MIH), actuando como medio de transporte para el intercambio de mensajes entre las diferentes entidades en la red [4]. Este marco de señalización se encuentra estandarizado por el grupo de trabajo IEEE 802.21 y supone el factor diferencial que va a explotarse en Mobilia. La señalización definida en IEEE 802.21 se basa en una codificación TLV (Tipo-Longitud-Valor), ofreciendo diferentes ventajas desde el punto de vista de la implementación. La Figura 1 muestra la arquitectura global a implementar en los terminales móviles involucrados en la arquitectura. Adicionalmente, algunas de estas funcionalidades se integrarán también en otro tipo de entidades que aparecen en el prototipo, como los Puntos de Acceso o el Servidor de vídeo (que ejercerá también las funciones de *Access Broker*).

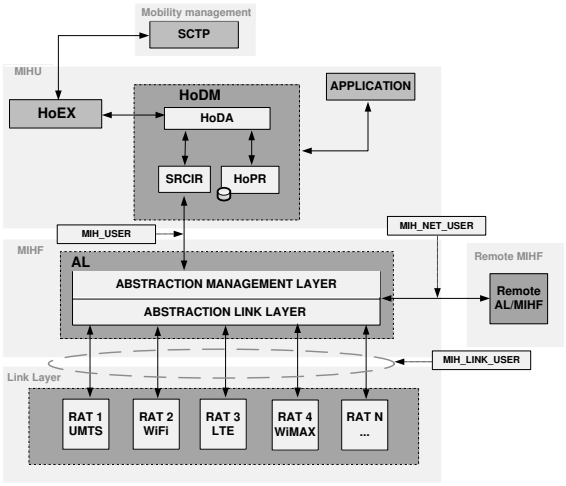


Figura 1. Visión global de la arquitectura propuesta por Mobilia para la realización de handovers verticales

Analizando la Figura 1 puede observarse la presencia de unas entidades responsables de la correcta gestión de los traspasos verticales. A continuación se procederá a realizar una breve descripción de las funcionalidades asociadas a cada una de ellas.

Link Layer - LL. Basada en el MIH_LINK_USER_SAP, definido en la especificación IEEE 802.21, facilita el intercambio de datos entre los diferentes interfaces inalámbricos con el Abstraction Layer o MIHF.

Abstraction Layer - AL (Media Independent Handover Function o MIHF en terminología 802.21). Se trata del elemento central de la arquitectura. Su principal función es permitir, ya sea de manera local o remota, el intercambio de información, comandos y eventos, entre las diferentes entidades responsables de tomar y ejecutar decisiones relacionadas con los traspasos. El MIHF tiene la tarea de realizar la abstracción de las características de las capas inferiores de una forma independiente de la tecnología. Para ello se ha estructurado en dos componentes: el *Abstraction Management Layer* (AML), que está a cargo del interfaz con el MIH User (MIH_SAP) y el interfaz remoto (MIH_NET_SAP), y el *Abstraction Link Layer* (ALL), que gestiona los interfaces con los niveles inferiores (MIH_LINK_SAP).

Handover Decision Manager - HoDM (Media Independent Handover User - MIHU). Está formado por dos elementos: el *Handover Decision Manager* (HoDM) y el *Handover Execution Manager* (HoEM). El primero constituye el bloque más importante de la entidad, siendo el responsable de enviar la señalización para que comience la ejecución de un traspaso. Toda la gestión de la información y criterios de decisión se encuentran localizados en este módulo, que a su vez se compone de los siguientes elementos:

- *Handover Decision Algorithm Module* (HoDA), que incluye la inteligencia necesaria para decidir cuándo debe realizarse un handover.
- *Handover Policies Repository* (HoPR), que es utilizado por el HoDA para establecer las condiciones para proceder a la inicialización de un handover. Esta información se combina con la que se recibe dinámicamente a través del *Service Requirements Collector Information Repository* (SRCIR).
- Service Requirements Collector Information Repository (SRCIR), que implementa el interfaz del MIH_USER con el MIHF/AL, actuando como un repositorio dinámico de información recogida por el terminal de usuario y la red. Esta información es utilizada por el HoDA, donde se compara con la almacenada en el HoPR, tomando entonces las decisiones que se estimen oportunas.

El HoEM se comunica directamente con el MIHF a través del SRCIR para el intercambio de primitivas que derivarán en la ejecución de un traspaso. Debe estar provisto de la inteligencia necesaria para informar de los eventos y manejar los posibles errores surgidos durante la ejecución de un handover.

MIHF Remoto. Proporciona a la arquitectura la capacidad de compartir información con elementos remotos.

Stream Control Transmission Protocol (SCTP) [13]. Los protocolos tradicionales de nivel de transporte, TCP y UDP, fueron diseñados para trabajar en redes con equipos fijos. Con el paso del tiempo y la proliferación de los dispositivos móviles se ha propiciado la necesidad de nuevos mecanismos que manejen la movilidad de una forma eficiente, minimizando el retardo y las pérdidas producidas durante un traspaso. Por ello surgen alternativas en el nivel de transporte, como el protocolo SCTP, que hereda muchas de las propiedades de TCP (como los controles de flujo y congestión), solucionando algunas de sus debilidades mediante la adaptación de funcionalidades procedentes de UDP. Junto a éstas se incorporan una serie de novedades, resumidas a continuación:

- *Multistreaming*. Aunque se mantiene el concepto de asociación (al igual que TCP), el protocolo SCTP soporta la presencia de múltiples flujos dentro de una misma conexión.
- *Multihoming*. SCTP permite a un equipo asociarse a través de múltiples interfaces de manera simultánea, cambiando la conexión dinámicamente, según las condiciones del canal o la movilidad del usuario.
- El protocolo SCTP otorga la posibilidad de cambiar la dirección IP de uno de los extremos de la comunicación de manera completamente transparente, manteniendo en todo instante la misma conexión [14].

Gracias a estas incorporaciones, el protocolo SCTP satisface las necesidades de la solución de movilidad dentro de la arquitectura Mobilia, descartando otras alternativas, como Mobile IP [15], que desperdicia una gran cantidad de recursos en el tunelado IP que transporta la información a través del *Home Agent*. Para los casos definidos en el IMT-Advanced, caracterizados por unas altas demandas de throughput, deberá evitarse cualquier tipo de sobrecarga innecesaria.

IV. Configuración de la demostración

Como se ha comentado anteriormente, uno de los objetivos principales del presente trabajo consiste en la implementación de la arquitectura descrita en la Sección III. Sin embargo, existen ciertas limitaciones que deben tenerse en cuenta: en primer lugar, la disponibilidad de las tecnologías inalámbricas está limitada; aunque existan algunas implementaciones disponibles de tarjetas 3G o WiMAX, la mayor parte de ellas no ofrece la flexibilidad[2] necesaria para dar respuesta a los retos que propone la arquitectura de Mobilia. Teniendo en cuenta estas limitaciones, la presencia de redes inalámbricas heterogéneas será emulada con el despliegue de dos puntos de acceso IEEE 802.11, configurados en dos canales ortogonales, que tomarán el rol de elementos de acceso. Esta aproximación, aunque obviamente presenta ciertas limitaciones, supone del mismo modo claras ventajas: el soporte de los drivers correspondientes es imprescindible para poder obtener elementos de información (que serán tenidos en cuenta para la selección de acceso); además, gracias al proyecto MadWifi [16], cualquier portátil estándar puede desempeñar la labor de punto de acceso, haciendo posible el desarrollo en elementos propios de la red. Teniendo en cuenta lo anterior, la plataforma propuesta se compone por cuatro terminales [3], como puede verse en la Figura 2. Su comportamiento se describe brevemente a continuación:

- **Terminal Móvil (TM)**. Se corresponde con el dispositivo móvil que utiliza el usuario. Su principal característica consiste en que puede estar equipado con más de un interfaz inalámbrico.

[2]Para una correcta implementación de las tecnologías se requiere la disponibilidad de unos drivers apropiados y la posibilidad de introducir cambios en el lado de la red (el elemento de acceso)

[3]Se emplean dos portátiles para actuar como punto de acceso, puesto que deben integrar la arquitectura de Mobilia

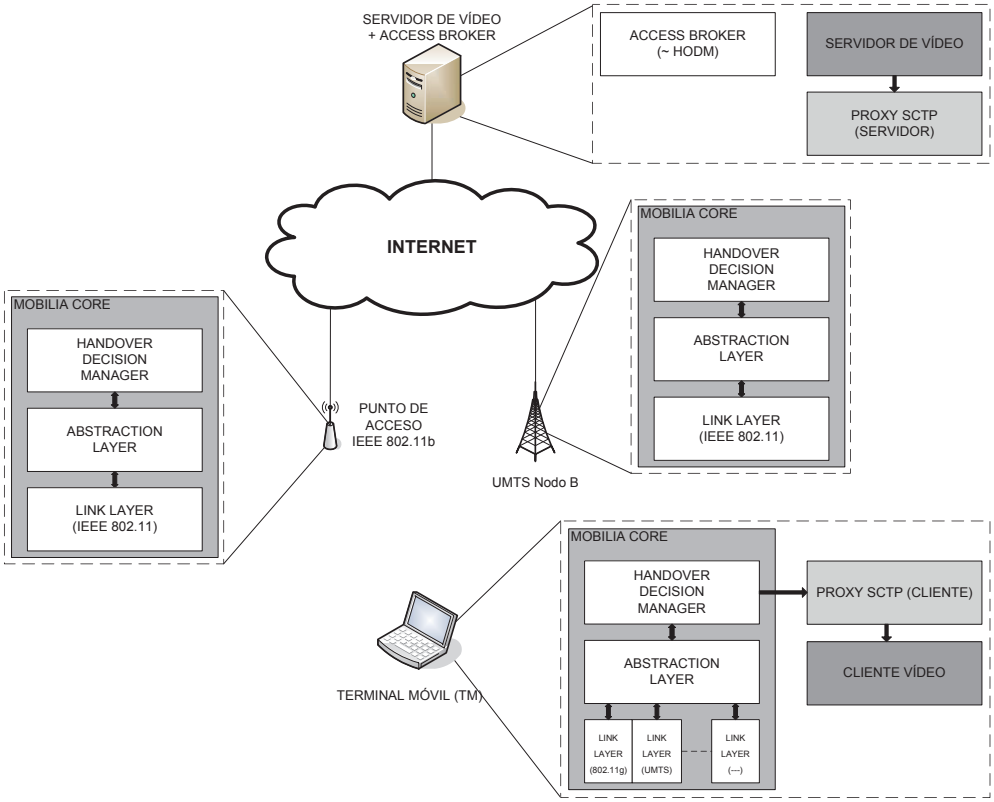


Figura 2. Plataforma empleada en la demostración

- **Elemento de Acceso (EA)**. Emulan el papel de cualquier elemento que proporciona acceso a la red (punto de acceso en WiFi, nodo B en UMTS, etc.). Para representar tecnologías heterogéneas, los dos EAs se configurarán en sendos canales ortogonales, evitando la interferencia entre ellos.
- **Access Broker (AB)**. Para garantizar una gestión eficiente de los recursos disponibles, el papel de un AB (ya sea centralizado o distribuido) puede ser fundamental; en la plataforma desplegada, esta entidad se conecta con los dos Elementos de Acceso a través de una infraestructura cableada. Incluye además aplicaciones que serán empleadas para introducir tráfico al sistema durante los experimentos.

Tanto el TM como el EA incorporan toda la arquitectura de Mobilia, mientras que el AB sólo incluye una instanciación del HoDM, ya que no tiene que gestionar recursos de los niveles inferiores de manera local (AL y LLs).

Como se ha mencionado anteriormente, la plataforma adopta el protocolo SCTP como solución de movilidad, incluyendo un proxy SCTP en cada uno de los extremos de la comunicación (en el terminal móvil, que hará las veces de cliente, y en el servidor de vídeo). El comportamiento de este proxy se basa básicamente en el reenvío del flujo de datos a través de un túnel, pudiendo modificarse el destino de manera dinámica gracias a la capacidad *multihoming* de SCTP, donde un nodo puede recibir paquetes por diferentes direcciones IP.

Se procederá al desarrollo de dos casos de uso: el primero de ellos se corresponde con la situación más convencional, en la que es el equipo del usuario el encargado de iniciar un traspaso al detectar una degradación en la calidad del enlace, buscando una alternativa que se adapte mejor a sus necesidades; en el segundo, es la red quién decide que debe realizarse un handover, debido a una situación de sobrecarga dentro de la misma.

IV-A. Caso de uso I. Handover iniciado por el usuario

Como se ha dicho anteriormente, en este supuesto el dispositivo detecta que la SNR (Signal to Noise Ratio) del enlace actual es inferior a un umbral preestablecido; como respuesta, se inicia una solicitud de handover, con el fin de conseguir obtener una conexión que presente una mejor relación señal a ruido.

Es importante remarcar dos aspectos: en primer lugar, todo el proceso de traspaso es llevado a cabo por los diferentes elementos de la arquitectura definida en Mobilia, sin intervención alguna por parte del usuario, que no va a percibir degradación alguna en la ejecución del servicio; además, gracias a la implementación de una GUI (*Graphical User Interface*), es posible proporcionar a los LLs un valor emulado de la calidad del enlace (por ejemplo la RSSI o Received Signal Strength Indication), para asegurar un comportamiento predecible de todo el proceso, aunque la implementación del LL garantiza la alternativa de implementar medidas reales. [4]

Una vez que el TM se conecte a un EA, se recibirá un vídeo en streaming desde el servidor y se configurarán los umbrales que marcarán los eventos con los que dará comienzo

[4] A efectos prácticos, la emulación de la SNR proporciona el control del evento que desencadenará un traspaso, evitando los errores en la transmisión inherentes a un canal de calidad deficiente

el proceso de handover (que serán almacenados en el HoPR). A través de la funcionalidad de emulación de la SNR de la interfaz gráfica, se decrementa la calidad del enlace a un valor inferior al umbral establecido. En ese instante, el HoDA decide que debe realizarse un traspaso y envía un mensaje LINK_GOING_DOWN a los niveles inferiores, a través de los interfaces MIH_USER y MIH_LINK_USER. En este momento, existen dos alternativas: si el TM cuenta con un único interfaz inalámbrico, no podrá beneficiarse de la solución propuesta, siendo necesario interrumpir el flujo de datos para buscar redes próximas y conectarse a un nuevo EA; por otro lado, en el caso de que tenga más de un interfaz de red, será posible minimizar la latencia producida durante el proceso de handover, llegando incluso a cambiar de una conexión a otra sin que el visionado del vídeo se vea en la percepción subjetiva del usuario. Este último caso se conoce como *make-before-break* handover, en el que se establece en segundo plano la nueva conexión en un interfaz alternativo, cambiando el flujo en el momento en el que se encuentre completamente establecida. En la demostración, el SCTP es el encargado de la realización del último paso en el proceso del traspaso, asignando la dirección obtenida de la nueva dirección como destino en el túnel. El intercambio de mensajes de este caso de uso se detalla en la Sección V.

IV-B. Caso de uso II. Handover iniciado por la red

En este caso es la red la encargada de iniciar el proceso de traspaso; para ello, se emulará una situación de sobrecarga[5]. De nuevo, una vez se cruce el umbral almacenado en el HoPR, el LL notifica la situación al AL que, a su vez, hace llegar el mensaje al HODM. Se presupone que no existe ningún tipo de acuerdo entre los dos elementos de acceso, por lo que se hace necesario consultar al Access Broker sobre la posibilidad de cambiar el flujo de datos, ya que es consciente de la situación de la red en todo momento (por ejemplo si las redes próximas al TM disponen de los recursos suficientes para proporcionar el servicio requerido). En caso afirmativo, se le indicará al HoDM del usuario final que puede realizar el traspaso[6]. A partir de este instante, el proceso de traspaso tiene un comportamiento idéntico al del primer caso de uso. De nuevo, los equipos multi-interfaz pueden beneficiarse de la arquitectura para reducir la latencia durante el proceso de traspaso.

V. Resultados

Esta sección utiliza la plataforma que se ha descrito anteriormente para evaluar el retardo que sufre una estación móvil cuando realiza un handover vertical entre dos elementos de acceso.

Concretamente, se abordarán cuatro casos diferenciados que serán descritos a continuación. Es necesario destacar que el principal objetivo de esta campaña de medidas es demostrar cuantitamente los beneficios que aporta el hecho de contar con más de un interfaz inalámbrico en los equipos de usuario.

A. Para el primero de los casos se cuenta con un terminal que posee un único interfaz IEEE 802.11 (situación bastante típica actualmente), por lo que todo el proceso de handover debe realizarse en el mismo interfaz, incluyendo las tareas de búsqueda de redes próximas y la posterior conexión a la que se considere más adecuada. Éste es el ejemplo que trata de ilustrar la peor de las posibilidades, debido a que la conexión debe romperse en dos momentos (ver Figura 3), pero se incluye simplemente por motivos comparativos. En este caso, así como en los dos siguientes, se asume que el direccionamiento IP en el nuevo acceso se configura de manera estática y, por lo tanto, no introduce ningún retardo adicional al sistema.

B. En este caso se añade un nuevo interfaz IEEE 802.11 al dispositivo. Uno de los interfaces se emplea para conectarse con los elementos de acceso (se le conoce como interfaz activo), mientras que el otro se dedica exclusivamente a las tareas de búsqueda de redes adyacentes (en este caso se le llama interfaz de *scanning*). A diferencia del supuesto anterior, la conexión se rompe únicamente durante el traspaso, ahorrando el tiempo empleado para la búsqueda de redes disponibles. Con esta situación se refleja también la posibilidad de que la red sea la encargada de proporcionar a la estación la información acerca de células adyacentes a través de un canal de señalización común, permitiendo al TM cambiar de EA sin necesidad de buscar redes.

C. Este escenario toma al anterior como punto de partida, donde el terminal dispone de un interfaz activo y otro de *scanning*; sin embargo, en esta configuración se aprovecha del hecho de contar con un dispositivo multi-interfaz, permitiendo la realización de un traspaso conocido como *make-before-break* handover[7]. Como puede observarse en la Figura 4, el interfaz de *scanning* se conecta en segundo plano a una red alternativa tras recibir el mensaje LINK_GOING_DOWN proveniente del HODM, mientras que la transmisión del vídeo se mantiene intacta en el interfaz activo. Cuando se ha completado todo el proceso de handover, el terminal móvil advierte al servidor de vídeo que debe modificarse la dirección IP destino[8], cambiando el flujo hacia una nueva ruta, evitando así el tiempo que se pierde durante el proceso de traspaso.

D. Como se ha mencionado anteriormente, en los tres casos anteriores se hace uso de un direccionamiento IP estático (asumiendo que el terminal fuera consciente del localizador IP a emplear en cada red), eliminando, por tanto, el tiempo adicional que se consumiría debido al intercambio de mensajes necesarios durante una asignación dinámica. Esto podría suponer un comportamiento poco realista, por lo que para este último caso se ha optado por añadir una asignación de dirección IP a través del protocolo DHCP con el fin de analizar la

[5]El driver MadWIFI permite conocer información sobre la carga de la red en una célula WLAN en los propios elementos de acceso (p.e. el número de estaciones asociadas a un punto de acceso o el tamaño del buffer de transmisión), en cualquier caso, siguiendo el ejemplo anterior, se emulará una sobrecarga de usuarios a través de un GUI en el EA.

[6]A través de un mensaje que parte del HODM del Access Broker hasta la misma entidad en el equipo del usuario

[7]Se trata de un handover durante el cual no se produce corte alguno en la conexión

[8]El servidor de vídeo no es capaz por sí solo de cambiar la dirección IP de destino (mientras la aplicación está en marcha), por lo que se ha introducido un proxy SCTP como solución a este problema (como se ha detallado al comienzo de la sección)

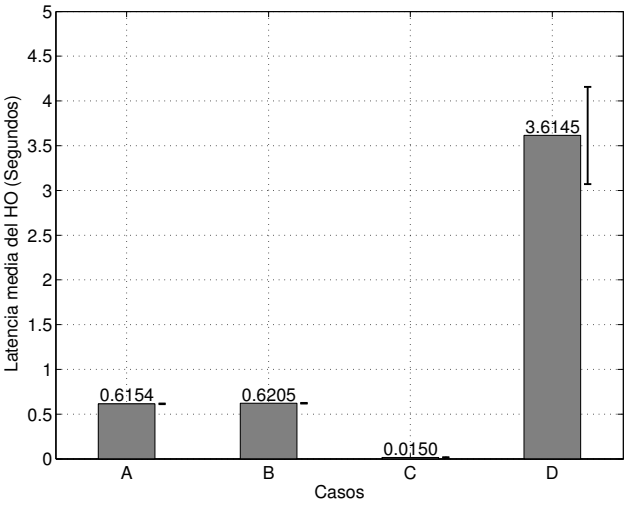


Figura 6. Promediado de la latencia de handover

sobrecarga adicional que se introduce en el proceso. Se toma el primero de los casos como punto de partida (dispositivo con un interfaz). En la Figura 5 puede comprobarse que la latencia se ve afectada en gran medida por la citada asignación dinámica. Es importante recalcar que para el caso C el proceso DHCP resultaría inocuo, puesto que se ejecutaría en background mientras que el tráfico fluye por el otro interfaz.

La Figura 6 muestra la latencia media y el intervalo de confianza del 95 % para una serie de 10 medidas independientes en cada uno de los cuatro casos expuestos.

Estos experimentos se han basado en la medida del retardo en el terminal móvil empleando como herramienta de análisis principal Wireshark [17]. Para ello se define la latencia del handover como el intervalo de tiempo que transcurre desde la recepción del último paquete en el enlace antiguo hasta que se recibe el primer paquete a través de la nueva conexión.

A tenor de los resultados, se observa que los dos primeros casos siguen el mismo proceso de handover, siendo los retardos correspondientes prácticamente idénticos (la diferencia radica en la funcionalidad de los interfaces), deteniendo el flujo de datos por un tiempo ligeramente superior a medio segundo. Sin embargo en el Caso C se emplea un sistema más avanzado de traspaso, en el que se evita la pérdida de conectividad inherente a la ruptura de conexión del interfaz activo previa a la conexión con el nuevo EA. Por último, en el Caso D se muestra el retardo total de combinar el primer supuesto (handover al nivel de enlace) y la negociación necesaria para la asignación de una dirección IP, resultando en un incremento signiticativo en el retardo global del sistema. Cabe destacar asismismo la gran estabilidad de los retardos observados para los casos A, B y C, a la vista de los intervalos de confianza obtenidos. En el caso de emplear la asignación de IP mediante DHCP, se puede observar que se introduce un grado de variabilidad notable en la medida, obteniendo intervalos de confianza notablemente más elevados.

VI. Conclusiones

Este trabajo presenta el diseño y la implementación de una plataforma real, basada en la arquitectura propuesta en el proyecto Mobilia del programa Celtic, con el fin de analizar empíricamente una serie de técnicas de selección de acceso dentro de un escenario caracterizado por la presencia de redes de acceso heterogéneas. Concretamente, se ilustran dos situaciones a través de las cuales tendrá lugar un traspaso: la primera se inicia en el terminal de usuario tras una degradación en la calidad de la señal del enlace, mientras que la segunda trata de replicar una situación donde el evento que inicia el proceso de handover es detectado por la propia red, concretamente cuando el elemento de acceso percibe una sobrecarga en el número de estaciones conectadas.

La plataforma implementada se utiliza para analizar cuantitativamente los beneficios que podrían aportar el uso eficiente de los interfaces inalámbricos en aquellos terminales que los tengan. Para ello se estudian cuatro configuraciones diferentes, demostrando que el hecho de disponer de más de un interfaz inalámbrico puede derivar en notables mejoras en el rendimiento del sistema.

En el futuro se plantea extender la plataforma desde varios puntos de vista: por un lado, la señalización entre las diferentes entidades involucradas se basará completamente en el estándar IEEE 802.21, pudiendo emplear soluciones ya planteadas, como por ejemplo ODTONE, debido a que la arquitectura Mobilia es independiente de la solución de movilidad a emplear, sería interesante un nuevo planteamiento adoptando otras técnicas, tales como Mobile IP o Fast MIPv6. También se pretende extender el abanico de la información empleada durante el proceso de selección de acceso, incorporando preferentemente los requerimientos concretos de los servicios. Muchas de estas líneas de actuación se están llevando a cabo en el ámbito del proyecto CENIT mIO!.

Por otra parte, la arquitectura presentada en este trabajo puede verse potenciada con la inclusión de nuevas funcionalidades, tales como técnicas cognitivas, para garantizar un uso más eficiente del medio radio, característica que está llamada a ser de gran importancia en las redes de comunicación inalámbricas venideras.

Agradecimientos

Los autores querían expresar su agradecimiento al Gobierno de España por su financiación en los proyectos "*Comunicaciones Cognitivas y Cooperativas sobre Entornos Heterogéneos*", C3SEM (TEC2009-14598-C02-01) y "*Mobility Concepts for IMT-Advanced*", Mobilia (Avanza I+D TSI-020400-2008-82). Asimismo, agradecer a Telefónica su financiación en el marco del proyecto CENIT mIO!.

Referencias

[1] L. Giupponi, R. Agusti, J. Perez-Romero, y O. Salient, "Improved revenue and radio resource usage through inter-operator joint radio resource management," in *Communications, 2007. ICC '07. IEEE International Conference on*, 2007, pp. 5793 –5800.

[2] J. Perez-Romero, O. Sallent, R. Agusti, P. Karlsson, A. Barbaresi, L. Wang, F. Casadevall, M. Dohler, H. Gonzalez, y F. Cabral-Pinto, "Common radio resource management: functional models and implementation requirements," in *Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on*, vol. 3, 2005, pp. 2067 –2071 Vol. 3.

[3] J. Sachs, R. Agüero, K. Daoud, J. Gebert, G. Koudouridis, F. Meago, M. Prytz, T. Rinta-aho, y H. Tang, "Generic abstraction of access performance and resources for multi-radio access management," in *Mobile and Wireless Communications Summit, 2007. 16th IST*, 2007, pp. 1 –5.

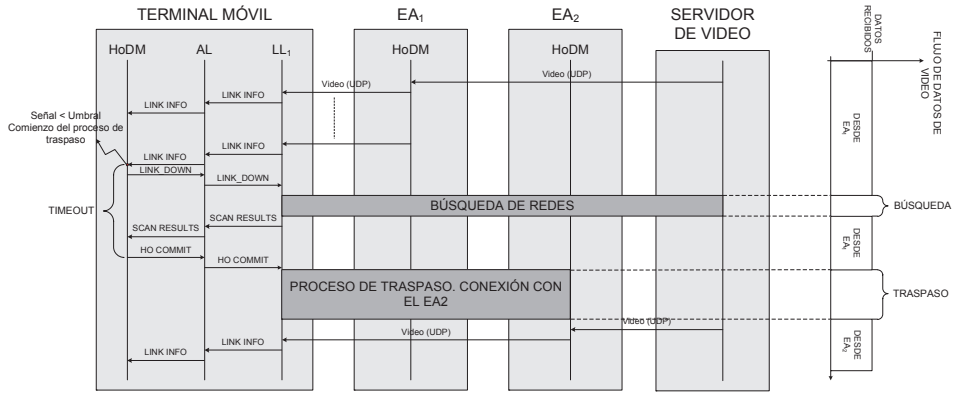


Figura 3. Diagrama de flujo para terminales con un sólo interfaz

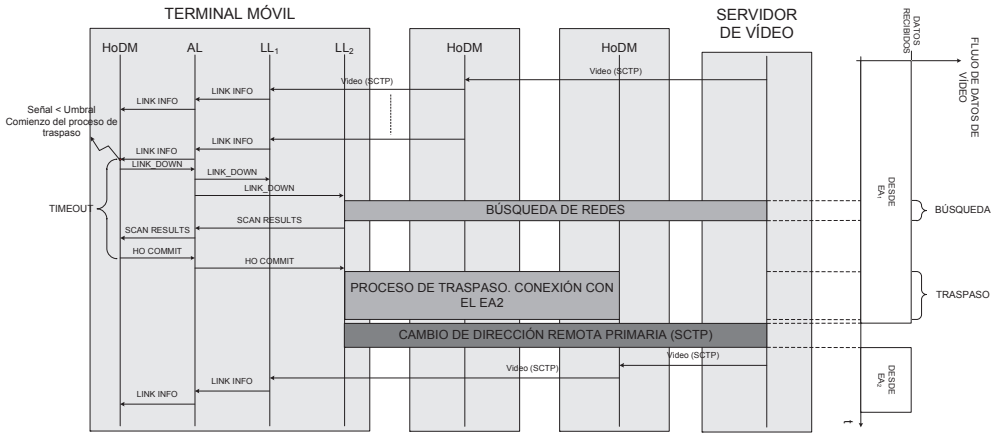


Figura 4. Diagrama de flujo de la ejecución de un soft-handover con SCTP como protocolo de transporte

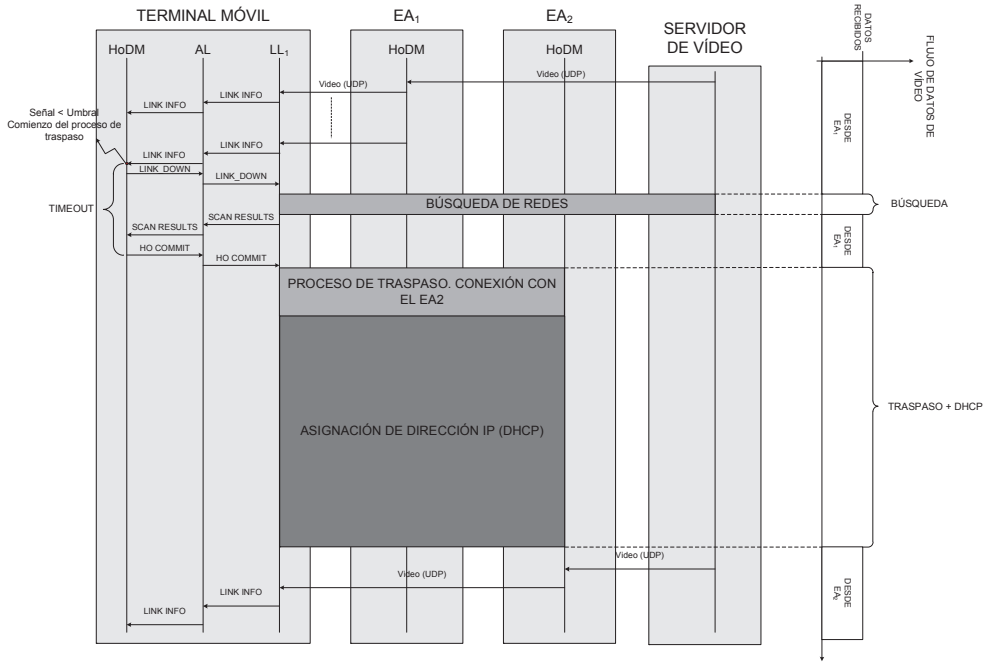


Figura 5. Diagrama de flujo para una asignación IP dinámica con DHCP

[4] *IEEE Standard for Local and Metropolitan Area Networks- Part 21: Media Independent Handover*, Std., 21 2009.

[5] K. Taniuchi, Y. Ohba, V. Fajardo, S. Das, M. Tauil, Y.-H. Cheng, A. Dutta, D. Baker, M. Yajnik, y D. Famolari, "IEEE 802.21: Media independent handover: Features, applicability, and realization," *Communications Magazine, IEEE*, vol. 47, no. 1, pp. 112 –120, 2009.

[6] *ODTONE - Open Dot Twenty ONE, http://hng.av.it.pt/projects/odtone*.

[7] R. Agüero, J. Gebert, J. Choque, y H. Eckhardt, "Towards a multi-access prototype in Ambient networks," in *2007 Proceedings of the 16TH IST Mobile and Wireless Communications, VOLS 1-3*. 345 E 47TH ST, NEW YORK, NY 10017 USA: IEEE, 2007, Proceedings Paper, pp. 1149–1153, 16th IST Mobile and Wireless Communications, Budapest, HUNGARY, JUL 01-05, 2007.

[8] P. Paakkonen, P. Salmela, R. Agüero, y J. Choque, "An integrated ambient networks prototype," in *Software, Telecommunications and Computer Networks, 2007. SoftCOM 2007. 15th International Conference on*, 2007, pp. 1 –5.

[9] ——, "Performance analysis of hip-based mobility and triggering," in *World of Wireless, Mobile and Multimedia Networks, 2008. WoWMoM 2008. 2008 International Symposium on a*, 2008, pp. 1 –9.

[10] M. López-Benítez, N. Vučević, F. Bernardo, y A. Umbert, "Real-time evaluation of radio access technology selection policies in heterogeneous wireless systems: the aroma testbed approach," in *Proceedings of the 11th international symposium on Modeling, analysis and simulation of wireless and mobile systems*, ser. MSWiM '08. New York, NY, USA: ACM, 2008, pp. 294–302. [Online]. Available: http://doi.acm.org/10.1145/1454503.1454553

[11] B. Cendón, J. Herrero, R. Agüero, A. Rodríguez, S. Albillos, A. Sainz, J. Sanz, y D. Gómez, "An access selection prototype based on ieee 802.21," in *Mobilight*, 2010.

[12] D. Gómez, R. Agüero, R. Sanz, M. García, y L. Muñoz, "Dynamic qos configuration of ieee 802.11e wlans: an empirical assesment," 2010.

[13] R. Stewart *et al.*, *SCTP Dynamic Address Reconfiguration*, IETF, Jun. 2004, [Internet Draft]. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-tsvwg-addip-sctp-09.txt

[14] R. Rouil, N. Golmie, y N. Montavont, "IEEE 802.21 transport solution using cross-layer optimized stream control transmission protocol," in *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on*, 2008, pp. 1 –5.

[15] C. Perkins, "IP Mobility Support for IPv4," RFC 3344 (Proposed Standard), Internet Engineering Task Force, Aug. 2002, obsoleted by RFC 5944, updated by RFC 4721. [Online]. Available: http://www.ietf.org/rfc/rfc3344.txt

[16] *The MadWifi Project: Madwifi - Multibrand Atheros Driver for Wireless Fidelity, http://www.madwifi.org/*.

[17] *Wireshark - The Network Protocol Analyzer, http://www.wireshark.org/*.

Integración de MPLS para la gestión de QoS en Fast Handover Proxy Mobile IP

David Cortés-Polo, José Luis González-Sánchez, Javier Carmona-Murillo,
Francisco J. Rodríguez-Perez.
Departamento de Ingeniería de Sistemas Informáticos y Telemáticos
Universidad de Extremadura
dcorpol@unex.es, jlgs@unex.es, jcarmur@unex.es, fjrodri@unex.es.

***Resumen*- La gestión de la movilidad y la Calidad de Servicio (QoS) son dos de los puntos más importantes en el desarrollo de las nuevas redes inalámbricas. Además éstos se vuelven un desafío importante si la velocidad de movimiento del nodo móvil es muy alta. En este artículo se presenta una nueva arquitectura que proporciona ambos, QoS y una alta movilidad usando un protocolo de gestión de movilidad IP. La arquitectura está compuesta por tres mecanismos para proveerlos. El primero de ellos es Fast Handoff para reducir el tiempo de desconexión. El segundo método es la extensión del túnel creado por el mecanismo de Fast Handoff para gestionar de manera eficiente los movimientos del Nodo Móvil (MN) . Por último, el tercer mecanismo es el uso de MPLS para proveer QoS a la red de acceso móvil. Para conseguir esto, se ha desarrollado modelos analíticos para evaluar el coste de actualización de los registros, el ratio de paquetes perdidos y los requerimientos de buffer. Los resultados muestran que los mecanismos propuestos pueden reducir significativamente el coste de la actualización de registro y reducir la latencia producida por el handoff y el ratio de paquetes perdidos.**

***Palabras Clave*- IPv6; Fast Handoff; Proxy Mobile IPv6; MPLS; QoS; Coste de Actualización de Registros**

I. INTRODUCCIÓN

La cuarta generación de redes de comunicación móviles (4G) promueve la integración de redes heterogéneas de manera transparente. De esta manera se podrá satisfacer la creciente demanda de QoS y ancho de banda [1] de los usuarios.

En este sentido, la integración de redes heterogéneas está resuelto usando un protocolo de gestión de la movilidad en redes IP. Para las redes inalámbricas, existen dos aproximaciones Mobile IP (MIP) [2] y Proxy Mobile IP (PMIP) [3]. Ambas están basadas en protocolos IP que permiten la movilidad de un nodo móvil (MN) manteniendo la conectividad a Internet mientras se mueven.

El primer protocolo desarrollado para incorporar la movilidad en el MN fue MIP. Este protocolo tiene ciertos inconvenientes como la necesidad de ser implementado en la pila de protocolo del nodo móvil para mantener la conectividad a Internet mientras se mueve, así como, el consecuente uso de recursos del MN para implementar esta funcionalidad.

PMIP fue desarrollado para evitar estos inconvenientes. Incluye en la red de comunicaciones móviles diversas entidades que resuelven esta problemática. Estas entidades siguen el movimiento del nodo, inician la señalización de la movilidad y gestionan los recursos de la red. Esto reduce la señalización del MN y los recursos hardware reservados para la pila del protocolo.

Sin embargo, estos protocolos presentan ciertos inconvenientes como son el tiempo de latencia durante el handoff o la gran carga de señalización debido a las frecuentes actualizaciones de registro.

Además, las redes 4G intentan resolver la creciente demanda de los usuarios para mejorar los servicios aportando QoS y un ancho de banda estable. Esto puede ser conseguido usando provisión de recursos en la red de acceso para proveer de ciertos parámetros de QoS en la comunicación.

Existen tres arquitecturas que proveen recursos de red para la garantía de QoS en Internet: Integrated Services (Intserv) [4], Differentiated Services (Diffserv) [5] y Multiprotocol Label Switching (MPLS) [6]. Los beneficios inherentes de MPLS en términos de QoS, ingeniería de tráfico y soporte de servicios avanzados IP, inspira diversos trabajos para el uso de esta tecnología en la infraestructura inalámbrica [7].

Para adoptar los requerimientos de la próxima generación de redes de comunicación móvil, en este artículo, se propone un nuevo protocolo Enhanced Fast Handover Proxy Mobile IPv6 MPLS (E-FHPMIP MPLS), que mejora las limitaciones de PMIP/MIP e incluye los beneficios aportados por la capacidad de la reserva de recursos de MPLS.

La propuesta está basada en tres mecanismos que proporcionan la QoS requerida por la comunicación. El mecanismos de Fast Handoff que crea un nuevo túnel a la red visitada antes de que el MN se enlace con el Punto de Acceso (AP). El segundo mecanismo es Enhanced Fast Handoff, que extiende el túnel creado al nuevo salto del MN si la QoS requerida lo permite. El último mecanismo es el túnel-MPLS que introduce soporte de QoS y de ingeniería de tráfico.

Para medir la efectividad de los mecanismos propuestos, se ha desarrollado expresiones analíticas del coste de actualización de los registros, ratio de paquetes perdidos y requerimientos de buffer. Los resultados numéricos muestran que la propuesta puede reducir el coste de la actualización de registros y proveer de un menor ratio de paquetes perdidos con respecto a otros esquemas evaluados (PMIP[3], PMIP-MPLS [8], Fast Handover Proxy Mobile IPv6 [9]).

El resto del artículo se compone de la siguiente manera. La sección II muestra los trabajos relacionados con la propuesta. La sección III introduce la arquitectura E-FH PMIP MPLS. En la sección IV, se desarrolla el modelo analítico. Los resultados numéricos son presentados en la sección V. Por último en la sección VI se presentan las conclusiones del trabajo presentado.

ISBN: 978-84-694-5948-5

II. Trabajos Relacionados

Como se presentó en la sección anterior, Mobile IP, probablemente es el protocolo más extendido para la gestión de la movilidad. Una de las líneas de investigación más importantes del momento son las extensiones de micro-movilidad aplicadas a Mobile IP y el mecanismo de túnel usando MPLS para enviar los paquetes a través de la red visitada garantizando los requerimientos de QoS [10]-[13].

El mecanismo de túnel transporta el tráfico a través de las entidades de movilidad como Home Agent (HA), Foreign Agent (FA), Nodo Móvil (MN) y otros nodos relacionados con la comunicación. Como método usual, se usan túneles IP-en-IP para el transporte y el mecanismo de encapsulado.

Las aproximaciones basadas en Mobile IP necesitan modificar el protocolo incluido en el Nodo Móvil de manera que se incluyan funcionalidades extra o nuevos algoritmos al MN.

Proxy Mobile IPv6 proporciona diversas ventajas comparado con MIPv6. Una de las mejoras principales está relacionada con el mecanismo de handover, ya que es basado en la red. Esto provee una gestión local de la movilidad desde la red hasta el Nodo Móvil y por lo tanto no se debe modificar el protocolo implementado en él mismo.

Por otro lado, en ambos protocolos, la latencia de los paquetes, así como el ratio de paquetes perdidos es muy alto cuando se produce un handover.

[8] analiza el impacto producido en Proxy Mobile IPv6 cuando se crea un túnel usando MPLS. Con esta propuesta, el Local Mobility Anchor (LMA) y el Mobile Access Gateway (MAG) están interconectados por un Label Switched Path (LSP). Esta aproximación reduce la latencia producida por el handover y lo optimiza. Con este modelo, todos los mensajes usados son una extensión de los mensajes definidos en PMIPv6. Por otro lado, diversos parámetros de QoS pueden ser incluidos en la red de acceso, pero la pérdida de paquetes y la latencia aún se mantienen altas con esta aproximación.

En [9] se propone un mecanismo de fast handoff para Proxy Mobile IPv6. Este protocolo denominado Fast Handover Proxy Mobile IPv6 (FH PMIPv6), permite a un MN descubrir rápidamente el movimiento a una nueva subred (usando algún tipo de evento lanzado desde las capas inferiores de la pila de protocolo) y comenzará a recibir los paquetes tan rápido como se enlace con un nuevo MAG. Con esta aproximación se incrementa el rendimiento con respecto a las comunicaciones en tiempo real y las dependientes de diversos parámetros de QoS. Como contra, el coste de actualización de los registros en FH PMIPv6 pueden ser excesivos, especialmente en nodos con una alta movilidad.

En este artículo, se presenta un nuevo esquema de gestión de la movilidad para PMIPv6 basado en Fast Handovers y MPLS. Combina los beneficios de ambos (PMIP y Fast Handovers) con las capacidades de MPLS. La propuesta es la mejora del protocolo Fast Handover PMIPv6 reduciendo el coste de las actualizaciones de registro e incluyendo parámetros de QoS a la comunicación usando MPLS para crear túneles en la red de acceso.

III. Esquema propuesto: E-FH PMIPv6 MPLS

En esta sección se presenta el esquema propuesto que se llama Enhanced Fast Handover Proxy Mobile IPv6 MPLS (E-FH PMIPv6 MPLS). Este esquema se basa en la integración de los protocolos Fast Handover Proxy Mobile IPv6 (FH PMIPv6) [9] y MPLS [6]. La arquitectura de la propuesta se muestra en la Figura 1.

Se ha de tener en cuenta que la red de acceso existente entre el Entrance Label Edge Router Gateway (E-LER) y los Label Edge Router/Mobility Anchor Gateway (LER/MAG) es MPLS. La arquitectura de red está basada en una jerarquía de tres niveles. En el nivel superior se encuentra el E-LERG que realiza la labor de Label Switching Router (LSR) del borde filtrando la señalización inter-intra dominio. El segundo nivel está formado por el Local Mobility Anchor (LMA) y el tercer nivel es LER/MAG conectado a diversos puntos de acceso (APs) que ofrecen conectividad a nivel de enlace. En este punto se ha de distinguir entre funcionalidades del nivel de enlace de la interfaz aérea, que son controladas por los AP, y las funcionalidades de movilidad de la capa IP (handoff de nivel 3), que se producen cuando el MN se mueve entre diversas subredes creadas por diferentes LER/MAGs. Por lo tanto, el LER/MAG es el primer elemento de la red que gestiona paquetes IP visto por el MN.

En este artículo, la latencia introducida por el handoff es definida como el tiempo que transcurre desde el momento que se detecta el evento del handoff al momento en el que el primer paquete se recibe desde la nueva subred. Hay dos tipos de handoff en una red de comunicación móvil: Intra-MAG e Inter-MAG. Un handoff Intra-MAG se produce cuando el MN se mueve entre dos APs gestionados por el mismo LER/MAG. Este tipo de handoff son gestionados por completo por la capa del nivel de enlace (handoff de nivel 2).

Por otro lado, se produce un Inter-MAG handoff cuando el nuevo AP y el antiguo están bajo diferentes LER/MAGs. Este tipo de handoff son típicos handoff del nivel 3 (capa de red).

Este trabajo se va a centrar en los Inter-MAG handoff ya que introducen importantes efectos en el rendimiento de la red. La mejora realizada por el protocolo FH PMIPv6 está basada en la gestión de la comunicación cuando el nodo móvil se mueve entre diferentes LER/MAG.

La propuesta usa el mecanismo de túnel incluida en FH PMIPv6, la cual reenvía los paquetes recibidos por el LER/MAG que se abandona (pLER/MAG) al nuevo LER/MAG (nLER/MAG). En este artículo se propone una nueva técnica basada en un conjunto de caminos de reenvío. Un camino de reenvío es un túnel LSP entre el LER/MAG que se abandona y el nuevo cada vez que el MN se mueve a una nueva subred. Para ello, envía un identificador del nodo móvil (MN ID) y el identificador el nuevo punto de acceso (AP ID). Estos dos elementos serán registrados en el LER/MAG que se abandona en vez de en el LMA, como muestra la Figura 2.

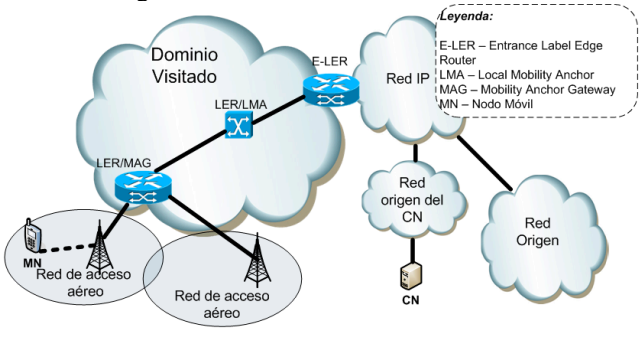


Fig. 1. Arquitectura de una red E-FH PMIPv6 MPLS

Gracias a este procedimiento, el túnel-LSP existente entre el LMA y el LER/MAG anterior puede ser extendido a los nuevos LER/MAGs si fuera necesario. Los paquetes que se envíen hacia el MN, serán enviados a través del LMA mediante el túnel usando para ello, el LSP creado entre el LMA y el primer LER/MAG, y será reenviado a través del conjunto de LSP extendidos a través de los diferentes LER/MAGs que haya visitado el MN. Esta aproximación introduce un importante delay a la comunicación debido a la extensión excesiva de caminos por los que reenviar los paquetes.

Para limitar esto último, los routers LER/MAG deberán medir la QoS de la comunicación y decidir cuándo la QoS está por debajo del umbral límite aceptable. El LMA detecta esta situación y crea un nuevo túnel-LSP desde un LMA al nuevo LER/MAG donde el MN se conectará.

Con esta aproximación, se reduce el coste de la actualización de registros enviado desde el LER/MAG al LMA. Para ello se añade un nuevo camino de reenvío hasta el nuevo LER/MAG que va a ser visitado por el MN (registro local). Esta aproximación es apropiada para MNs con una alta movilidad, donde los paquetes de datos deben ser reenviados rápidamente a la nueva localización.

La Figura 2 muestra la operación básica del protocolo E-FH PMIPv6. En este caso, el MN se mueve desde la subred 1 a la subred 2. Todos los movimientos del MN son detectados a través del nivel de enlace (Trigger de nivel 2). En el ejemplo, el primer movimiento del MN hacia la subred 2 reporta la tupla [MN ID, nuevo AP ID] al LER/MAG1 que va a abandonar (paso 1).

En ese momento, los paquetes destinados hacia el MN serán enviados desde el E-LERG al LMA y reenviados al LER/MAG1 usando el $LSP_{LMA,\ LER/MAG1}$, (paso 2 y 3 respectivamente). Cuando el LER/MAG1 recibe la notificación de handover, mandará un mensaje Handover Initiate al siguiente LER /MAG. En ese momento, todos los paquetes serán reenviados hacia el nuevo LER/MAG2 cuando la extensión del túnel-LSP se complete. Esto será anunciado por el nuevo LER/MAG2 usando un mensaje Handover ACK (paso 4). Cuando el MN se enlaza, todos los paquetes almacenados en el LER /MAG2 son entregados al MN (paso 5).

Este mecanismo podrá ser repetido mientras la QoS en la nueva extensión del túnel sea al menos la requerida para la comunicación.

De esta manera, el coste de actualización de los registros se reduce de manera drástica ya que la distancia entre dos LER/MAG vecinos es normalmente menor que la distancia entre el LER/MAG y el LMA.

La Figura 3 presenta el mecanismo que evita la latencia en la propuesta presentada. En la figura se muestra un escenario en el que el túnel original se ha extendido tres veces desde LER/MAG1 hasta LER/MAG3. En un momento dado, el nodo móvil se mueve a la subred 4. Como se presentó al principio de la sección, el MN manda la tupla [MN ID, nuevo AP ID] al LER/MAG que se va a abandonar, en este caso es LER/MAG3 (paso 1).

Este LER/MAG usando IGP busca el mejor túnel-LSP para extender el camino. Si la QoS ofrecida por cualquiera de los posibles túneles-LSP es menor que la QoS requerida, el LER/MAG que se va a abandonar, elige el túnel LSP que ofrece una QoS cercana a la requerida como una extensión provisional del LSP (paso 2).

Este método es empleado para mejorar el ratio de pérdida de paquetes. La creación del túnel y el reenvío de los paquetes es el mismo que se expuso en el funcionamiento básico de E-FH PMIPv6 MPLS.

Al mismo tiempo, el LER/MAG que se abandona envía un mensaje extendido de RSVP al LMA (paso 3). En este mensaje se indica que el umbral de QoS ha sido alcanzado y el LMA comienza a crear un nuevo camino desde el LMA al LER/MAG donde el MN se va a enlazar; en el ejemplo es el LER/MAG4 (paso 4).

Cuando el nuevo camino es creado el LMA envía al LER/MAG que se abandona (el cual ha mandado el mensaje RSVP extendido) un mensaje ACK. Este será el último mensaje enviado por el camino anterior desde el LMA hasta el LER/MAG4 (paso 5). Este paquete será interceptado por el LER/MAG3 ya que está esperando el último mensaje de la comunicación para cerrar el túnel-LSP.

El LER/MAG3 enviará un mensaje extendido de RSVP para cerrar el camino a su LER/MAG anterior y esto será replicado hasta el primer LER del camino, el cual será el LMA.

Además, un mensaje RESV ERR será enviado al LER/MAG4 para cerrar el túnel-LSP provisional desde el LER/MAG3 hasta el LER/MAG4. Con este método se cierra el camino de reenvío. El funcionamiento de E-FH PMIPv6 MPLS es descrito por el seudocódigo de la Tabla I.

IV. Evaluación del Funcionamiento y Análisis

En esta sección, se va a desarrollar los modelos analíticos para calcular el coste de actualización de los registros, la pérdida de paquetes y el tamaño de buffer de diferentes protocolos.

Se va a comparar la propuesta presentada en la sección anterior con respecto a PMIP [3], PMIP-MPLS[8], FH PMIP [9].

Seguidamente se introducirán los parámetros usados en el análisis.

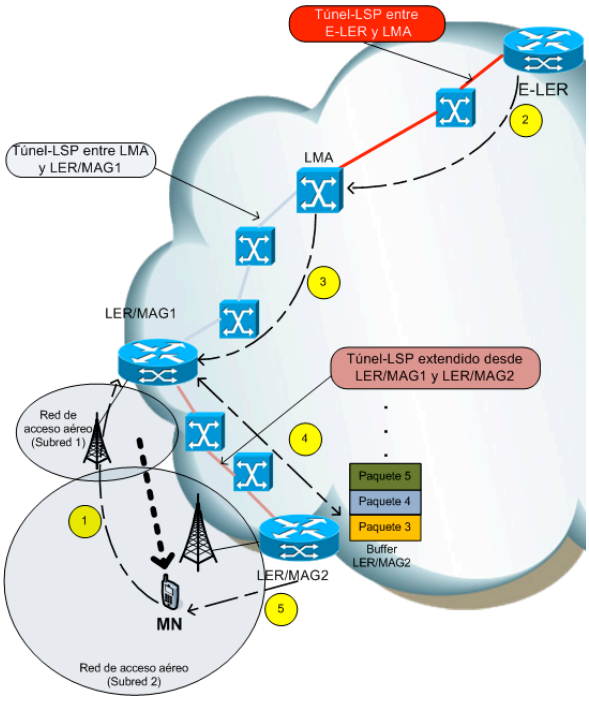


Fig. 2. Funcionamiento básico de E-FH PMIPv6 MPLS

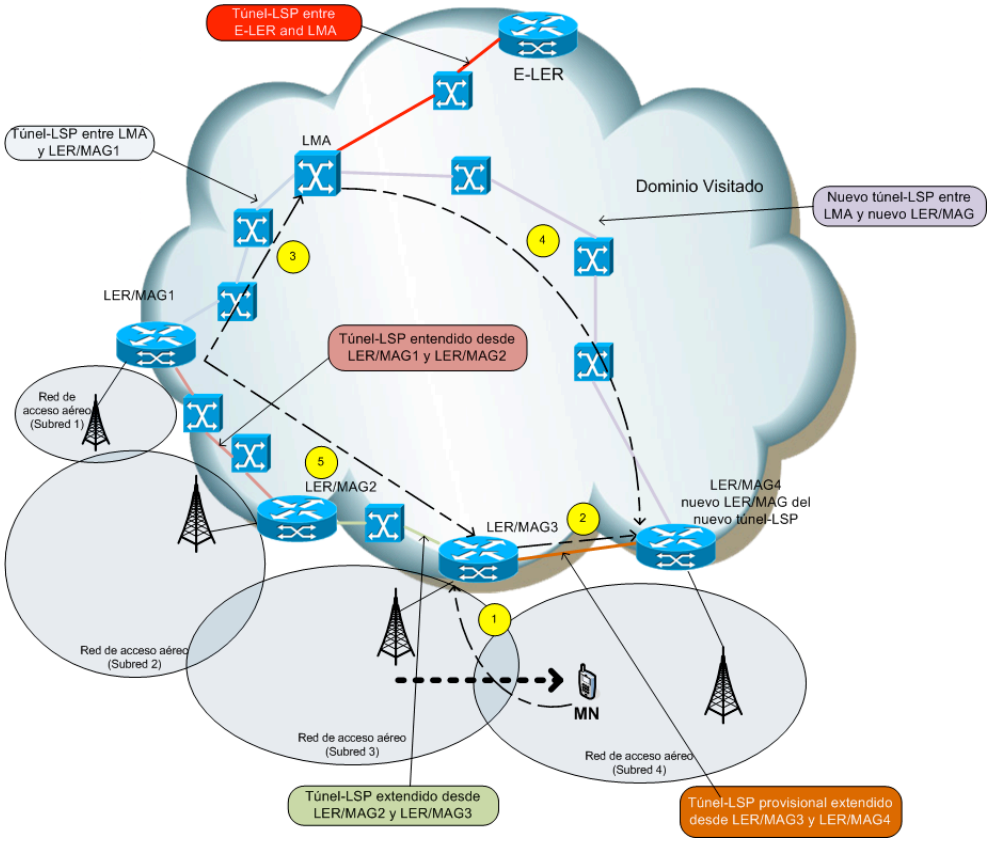


Fig. 3. Mecanismo para mantener la QoS requerida por la comunicación

TABLA I. E-FHPMIPv6 MPLS

```
%procedimiento de extension del túnel-LSP
Inicializar;
IF (MN entra en una nueva subred)
    Comunicar al LER/MAG que se abandona la dirección del nuevo
LER/MAG;
    Comparar la dirección del MN en la table de conmutación;
    IF (Camino_cspf ≥ QoS_requerida por el Nodo Móvil)
        Extender el túnel-LSP desde el LER/MAG que se abandona
        hasta el nuevo LER/MAG;
    ELSE
        Elegir el Camino_cspf cercano a la QoS_requerida;
        Extender el túnel-LSP desde el LER/MAG que se abandona
        hasta el nuevo LER/MAG;
        SEND mensaje RSVP_extendido al LMA para crear un nuevo
        camino entre el LMA y el nuevo LER/MAG;
    ENDIF
ENDIF

%Procedimiento de llegada de paquete
El E-LER envía los paqutes correspondientes al LMA del MN;
Conmuta los paquetes al primer LER/MAG usando un procedimiento de label
swapping;

IF (el LER/MAG ha enviado un mensaje RSVP_extendido y el mensaje es
la respuesta de ese mensaje)
        Limpiar el túnel-LSP y todas las extensions: Este es el ultimo
        paquete de datos del túnel LSP;
    ELSE
    IF (el LER/MAG no es el LER/MAG que actualmente está
        dando conectividad al MN)
        Conmuta los paquetes al siguiente LER/MAG;
    ELSE
        El LER/MAG elimina la etiqueta MPLS del paquete y lo reenvia
      al MN;
    ENDIF
ENDIF
```

Parámetros:

- t_s Tiempo medio de conexión por sesión;
- t_r Tiempo medio de estancia en una red visitada;
- T_{ad} Intervalo de tiempo entre mensajes de Agent Advertisements;
- N_h Número medio de handovers de nivel 3 en una session ($N_h = t_s/t_r$);
- N_f Número medio de extensiones del camino de reenvio durante una session ($N_f = N_h/número\_extensiones$);
- s_u Tamaño medio de un mensaje de señalización para actualización de los registros;
- s_l Tamaño medio de un mensaje para el establecimiento de un LSP;
- h_{x-y} Número medio de saltos entre x e y en la red cableada;
- B_w Ancho de banda de la red cableada;
- B_{wl} Ancho de banda en el enlace inalámbrico;
- L_w Latencia del enlace cableado (delay de propagación);
- L_{wl} Latencia del enlace inalámbrico (delay de propagación);
- P_t Routing o búsqueda de etiquetas y procesado;
- λ_d Ratio de transmisión para un paquete;
- T_{inter} Tiempo entre llegadas de paquetes de datos consecutivos.

Sea $t(s, h_{x-y})$, el tiempo usado por un paquete con tamaño s en ser reenviado desde x a y a través de un enlace alámbrico y un enlace inalámbrico. $t(s, h_{x-y})$ puede ser expresado de la forma:

$$t(s,h_{x-y}) = c + h_{x-y}\cdot\left(\frac{s}{B_w} + L_w\right) + (h_{x-y}+1)\cdot P_t \quad (1)$$

$$donde \qquad c = \begin{cases} \frac{s}{B_{wl}} + L_{wl} & si \quad x = MN \\ 0 & si \quad x \neq MN \end{cases}$$

A. Coste de actualización de los registros

El coste total de la señalización para la actualización de registros durante una sesión es denotado por C_u. El coste de señalización es la carga de tráfico acumulado en el intercambio de mensajes de señalización (salto x tamaño de mensaje) durante la sesión de comunicación de un MN. Para cada movimiento en la nueva subred, el envío del mensaje Proxy Binding Update al LMA será realizado por PMIP y PMIP-MPLS. En FH PMIP, un camino de reenvío será creado entre el MAG anterior y el nuevo MAG para reenviar los paquetes hacia el nodo móvil antes de que se envíe el mensaje Proxy Binding Update al LMA. En la propuesta presentada, se extiende el camino de reenvío si los requerimientos de QoS son satisfechos. El registro con el LMA será realizado cuando esos requerimientos no puedan ser satisfechos (por ejemplo, de ancho de banda o de delay). Seguidamente se expone las expresiones para el coste de actualización de registros de todos los protocolos expuestos anteriormente:

$$C_u(PMIP) = 2\cdot s_u\cdot h_{MAG-LMA}\cdot N_h + \\ +2\cdot s_u\cdot h_{nMAG-LMA}\cdot N_h \quad (2)$$

$$C_u(FHPMIP) = 2\cdot s_u\cdot h_{nMAG-pMAG}\cdot N_h + \\ +2\cdot s_u\cdot h_{nMAG-LMA}\cdot N_h \quad (3)$$

$$C_u(PMIP-MPLS) = 2\cdot s_u\cdot h_{pMAG-LMA}\cdot N_h + \\ +2\cdot s_l\cdot h_{nMAG-LMA}\cdot N_h \quad (4)$$

$$C_u(E-FHPMIPM) = 2\cdot s_u\cdot h_{MN-pLER/MAG}\cdot N_h + \\ +2\cdot s_l\cdot h_{pLER/MAG-nLER/MAG}\cdot N_h + \\ +2\cdot(N_f)\cdot s_l\cdot h_{pLER/MAG-LMA} + 2\cdot s_l\cdot h_{LMA-nLER/MAG}\cdot N_f \quad (5)$$

B. Paquetes perdidos durante una sesión

El conjunto de paquetes perdidos P_{loss} durante una sesión es definido como la suma de los paquetes perdidos durante todos los handoffs mientras que el MN recibe paquetes de datos.

En PMIP y PMIP-MPLS, todos los paquetes en vuelo se perderán durante el handoff debido a la falta de mecanismo de almacenamiento de los paquetes. En FH PMIP y E-FH PMIP MPLS, los paquetes en vuelo se perderán hasta que se activa el mecanismo de buffering. Como se mencionó anteriormente, los lanzadores de nivel 2 son usados tanto en FH PMIP como en E-FH PMIP MPLS. Se asume por lo tanto que la pérdida de paquetes comienza cuando el handoff de nivel 2 es detectado.

$$P_{loss}(PMIP) = \left[\left(\frac{1}{2}T_{ad}\right) + T_c(PMIP)\right]\cdot\lambda_d\cdot N_h \quad (6)$$

$$P_{loss}(FHPMIP/E-FHPMIPM) = \\ t(s_u,h_{MN-pMAG[pLER/MAG]})\cdot\lambda_d\cdot N_h \quad (7)$$

$$P_{loss}(PMIP-MPLS) = \\ \left[\left(\frac{1}{2}T_{ad}\right) + T_c(PMIP-MPLS)\right]\cdot\lambda_d\cdot N_h \quad (8)$$

C. Tamaño de buffer requerido

Los buffer son usados para almacenar los paquetes en vuelo, estos buffers se encuentran en los MAG y los LER/MAG de los protocolos FH PMIP y E-FH PMIP MPLS respectivamente. El mecanismo de buffering se activa en el momento que se recibe un mensaje Handover Initiate hasta el momento en el que se realiza la actualización de registros.

Los requerimientos de tamaño del buffer para FH PMIP y E-FH PMIP MPLS se muestran a continuación. Cabe observar que ambos esquemas requieren el mismo tamaño de buffer:

$$B_{size}(FHPMIP) = \left(\frac{1}{2}T_{ad}\right) + t(s_u,h_{MN-pMAG} + \\ +h_{pMAG-nMAG})\cdot\lambda_d \quad (9)$$

$$B_{size}(E-FHPMIPM) = \left(\frac{1}{2}T_{ad}\right) + (t(s_u,h_{MN-pLER/MAG}) + \\ +t(s_l,h_{pLER/MAG-nLER/MAG}))\cdot\lambda_d \quad (10)$$

V. RESULTADOS

En esta sección, se comparan los protocolos presentados anteriormente usando una aproximación analítica. La configuración de los parámetros usados en los experimentos está listada en la Tabla II.

TABLA II: CONFIGURACIÓN DE PARÁMETROS

Parámetro	Valor
t_s	1000 seg
Tr	5 ~ 50 seg (por defecto 20)
Tad	1 seg
su	48 Bytes
sl	28 Bytes
B_w	100 Mbps
B_{wl}	11 Mbps
L_w	1 mseg
L_{wl}	2 mseg
P_t	10^{-6} seg
λ_d	64 Kbps

La figura 4 muestra el escenario en el que se puede observar el camino que sigue un paquete involucrado en el mecanismo de handoff. Hay 8 saltos desde el LMA hasta el pMAG o pLER/MAG y la distancia entre el LMA y el nMAG o nLER/MAG es de 9 saltos.

La figura 5 muestra la comparativa de los costes de actualización de registro. Tanto FH PMIP como E-FH PMIP MPLS introducen un decremento significativo del coste de actualización comparado con PMIP y PMIP-MPLS. Esto es producido por el método de extensión de túneles aplicado en FH PMIP. La propuesta presentada mejora el coste de actualización de registros comparado con FH PMIP extendiendo el túnel mientras que los requerimientos de QoS sean satisfechos. Con este método se reduce el 50% de los mensajes extra de señalización enviados desde la antigua subred para reenviar los paquetes en vuelo desde el pMAG. La reducción comparada con PMIP es de alrededor del 75% usando E-FH PMIP MPLS.

La figura 6 muestra la cantidad de paquetes perdidos durante toda la sesión usando las diferentes aproximaciones. PMIP y PMIP-MPLS no usan mecanismo de buffering, esto conlleva que tengan la mayor cantidad de paquetes perdidos. Sin embargo, la aproximación de FH PMIP y E-FH PMIP MPLS, que inician el mecanismo de buffering cuando el handover es detectado, tienen la misma cantidad de paqetes perdidos, la cual es cercana a 0.

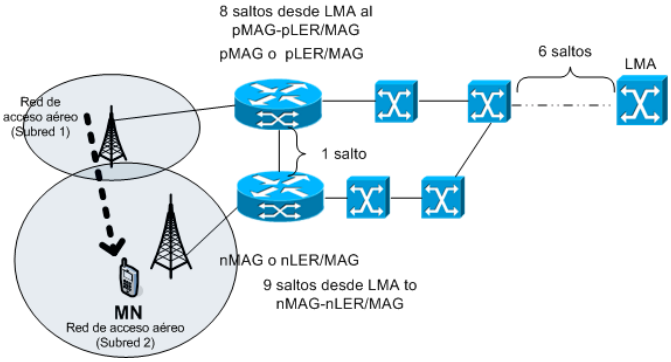


Fig. 4. Distancias relativas en saltos en la red simulada

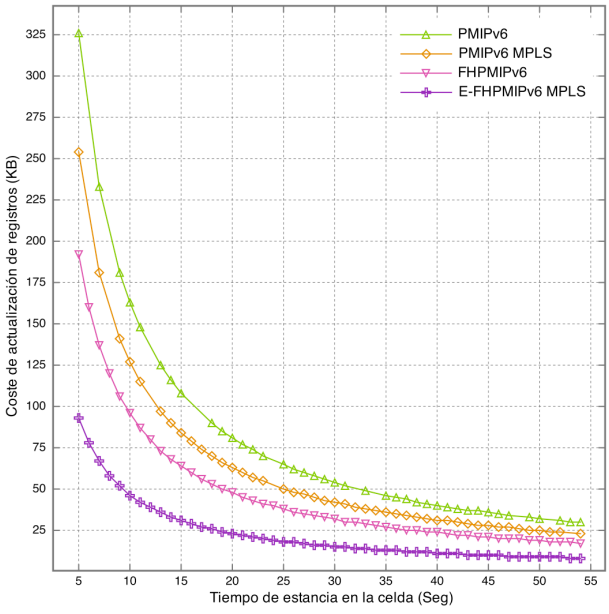


Fig. 5. Coste de actualización de registros

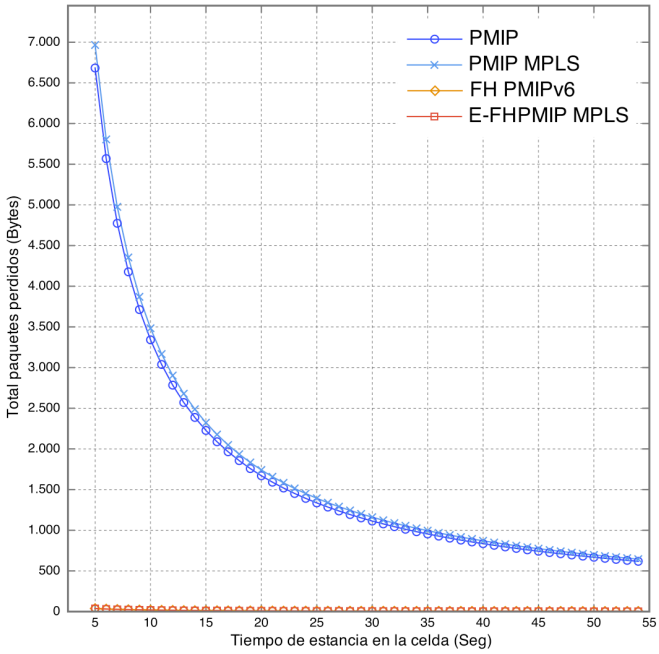


Fig. 6. Total de paquetes perdidos durante una sesión

Los requerimientos de tamaño de buffer por cada MN son de aproximadamente 2,7 KB en FH PMIP y E-FH PMIP MPLS. Los protocolos PMIP y PMIP-MPLS no se contemplan en esta figura ya que no implementan ningún mecanismo de buffering.

Esto significa que cerca de 40.000 nodos móviles pueden ser gestionados con el mecanismo de buffering usando una memoria con un tamaño de 100 MB. Esto se muestra en la Figura 7.

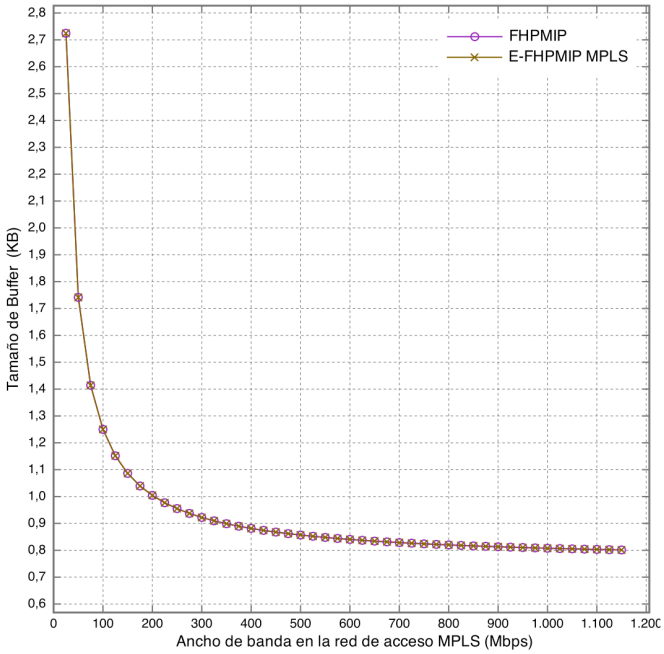


Fig. 7. Tamaño de Buffer en términos de ancho de banda en la red de acceso

VI. Conclusiones

Este trabajo presenta una mejora sobre el protocolo de gestión de la movilidad Fast Handover Proxy Mobile IP, llamado Enhaced-Fast Handover Proxy Mobile IP MPLS, que soporta tanto la gestión de la movilidad como la gestión de la Calidad de Servicio (QoS) en la red de acceso a la red inalámbrica.

Se ha desarrollado el mecanismo básico de operación para crear un LSP antes de que el MN se mueva a la nueva subred usando funcionalidades de nivel 2 para reducir el tiempo de desconexión. Se ha propuesto la extensión del túnel-LSP para seguir el movimiento del nodo. Con las extensiones, se reduce los mensajes de señalización del handoff, el coste de actualización de registros y provee de una mejor latencia de handoff y una menor tasa de pérdida de paquetes.

Para conseguir esto, se ha realizado una comparativa entre la propuesta desarrollada y las soluciones existentes (PMIP, PMIP-MPLS y FH PMIP).

Se ha modelado analíticamente el coste de actualización de los registros, los paquetes perdidos durante el handoff y el tamaño de los buffer implementados en los nodos. También se ha descrito la arquitectura para implementar esta aproximación. Se ha probado, a través de análisis, que la aproximación propuesta consigue una sustancial mejora en la cantidad de señalización enviada y mejora la QoS ofrecida por la red, únicamente aumentando ligeramente el uso del enlace.

Agradecimientos

Este trabajo ha podido ser realizado gracias a la Junta de Extremadura y fondos FEDER, con referencia GR10116.

Referencias

[1] Frattasi S., Fathi H., Fitzek F., Prasad R., Katz M.: 'Defining 4G technology from the users perspective', IEEE Network, 2006

[2] C. Perkins, IP Mobility Support for IPv4, RFC 3220, January 2002.

[3] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, B. Patil, "Proxy Mobile IPv6" RFC 5213, August 2008.

[4] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview", Internet Engineering Task Force, RFC 1633, June 1994.

[5] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services", Internet Engineering Task Force, RFC 2475, December 1998.

[6] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture", Internet IETF RFC 3031, January 2001.

[7] P. Francesco, "An MPLS-based architecture for scalable QoS and traffic engineering in converged multiservice mobile IP networks," Comput. netw., vol. 47, pp. 257-269, 2005.

[8] F. Xia, B. Sarikaya. "MPLS Tunnel Support for Proxy Mobile IPv6". IETF Draft (work in progress), October 2008

[9] H. Yokota, K. Chowdhury, R. Koodli, B. Patil, F. Xia, "Fast Handovers for Proxy Mobile IPv6", RFC 5959, September 2010

[10] M. Wenchao and F. Yuguang, "Dynamic hierarchical mobility management strategy for mobile IP networks," Selected Areas in Communications, IEEE Journal on, vol. 22, pp. 664-676, 2004.

[11] Z. Hairong, Y. Chihsiang, and H. T. Mouftah, "DHMM: A QoS Capable Micro-Mobility Management Protocol for Next Generation All-IP Wireless Networks," in Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE, 2007, pp. 4989-4993.

[12] F. M. Chiussi, D. A. Khotimsky, and S. Krishnan, "Mobility management in third-generation all-IP networks," Communications Magazine, IEEE, vol. 40, pp. 124-135, 2002.

[13] R. Langar, N. Bouabdallah, R. Boutaba, " A Comprehensive Analysis of Mobility Management in MPLS-Based Wireless Access Networks," Transactions on Networking, vol 16 no 4, August 2008

Gestión de políticas y precios en entornos de acceso heterogéneos

Javier Baliosian
Facultad de Ingeniería, Instituto de Computación Universidad de la República (Uruguay)
javierba@fing.edu.uy

Javier Rubio-Loyola
Pablo Salazar
Lab. de Tecnologías de la Información CINVESTAV Tamaulipas
jrubio@tamps.cinvestav.mx
psalazar@tamps.cinvestav.mx

Ramón Agüero
Dpto. Ingeniería Comunicaciones Universidad de Cantabria
ramon@tlmat.unican.es

Joan Serrat
Dpto. Teoría de Señal y Comunicaciones Universidad Politécnica de Catalunya
serrat@tsc.upc.edu

***Resumen*- La irrupción de nuevos servicios, y el imparable aumento del número de tecnologías de acceso a disposición de los usuarios, hace que los diferentes actores involucrados en el ámbito de las comunicaciones tengan que replantearse sus estrategias tradicionales. Es necesario afrontar el diseño de una arquitectura capaz de asumir los retos que van apareciendo, siendo clave romper con algunas de las soluciones tradicionales, que no son capaces de responder a las nuevas exigencias que aparecen. Uno de los aspectos más importantes es afrontar el diseño desde un punto de vista global, haciendo que las diferentes entidades cooperen entre sí de manera abierta y flexible, lo que no es posible (en muchas ocasiones) con parches a las alternativas que actualmente existen. Este marco es en el que se sitúa el proyecto Comunicaciones Cognitivas, Cooperativas y Gestión Autónoma de Servicios (C3SEM), que se fundamenta en la cooperación e integración del substrato de comunicación subyacente con la arquitectura de gestión de servicios. En este trabajo se plantea una de las líneas de trabajo actuales, en la que se analiza diferentes políticas de gestión de precios, ya que parece razonable que en el medio plazo los operadores se vean obligados a redefinir sus estrategias actuales, principalmente basadas en tarifas planas.**

I. INTRODUCCIÓN

Es innegable que estamos viviendo un cambio sustancial en la manera de entender las comunicaciones móviles. La proliferación de dispositivos capaces de utilizar múltiples tecnologías, la aparición de nuevos operadores y modelos de negocio, son algunas de las piedras angulares de lo que se ha venido a llamar como sistemas 4G. A pesar de que es innegable que se han producido una serie de avances tecnológicos, quedan todavía numerosos retos por afrontar.

A pesar de que dichos retos se podrían afrontar de manera individual (y así se ha hecho en numerosos trabajos existentes en la literatura), es cada vez más evidente que se hace necesario acometer el análisis desde un punto de vista global, favoreciendo la interacción y cooperación entre los diferentes elementos del sistema. Este es precisamente la aproximación que se sigue en el proyecto C3SEM, en el que se distinguen dos líneas de trabajo, diferenciadas entre sí, pero con un alto grado de cooperación/integración entre ambas.

El primer grupo de líneas de investigación se centra en la propia red, en el sustrato de comunicación, proponiendo elementos funcionales, algoritmos, mecanismos y protocolos para mejorar el comportamiento que las tecnologías actuales ofrecen. Las tres áreas principales en las que se está trabajando vienen marcadas por: la heterogeneidad en la red de acceso, el uso de técnicas cognitivas, y el empleo de topologías malladas.

El segundo bloque en el que se centra el proyecto C3SEM vendría dado por los servicios. Las mayores posibilidades que, a día de hoy, ofrece la tecnología existente tienen que ser aprovechada por los servicios que demandarán los usuarios finales. Para que esto sea una realidad, es necesario mejorar los procedimientos de gestión de servicios que se utilizan en la actualidad.

Como primer nexo de unión entre los dos bloques previamente descritos, se presentan, en este artículo, los primeros resultados obtenidos en parte de las líneas de investigación antes citadas, en el marco de optimizar políticas de gestión y de precios para entornos de acceso inalámbricos y heterogéneos. Se asume que se dispone de un conjunto de elementos de red, que cooperan entre sí, pudiendo ser de un único operador o de varios y que ofrecen acceso a los usuarios. Estos elegirán el acceso que mejor les convenga, en función de una serie de políticas de uso y de los requerimientos de los servicios. La red, por su parte, debe optimizar sus políticas de gestión de servicios y de precios, con el objetivo de maximizar su beneficio, respetando las garantías de servicio con los usuarios finales.

Para responder adecuadamente las cuestiones que se han planteado anteriormente, este artículo se ha estructurado en las siguientes secciones: la Sección II resume el estado del arte más relevante en los diferentes ámbitos en los que se trabaja en el marco de este artículo; la Sección III presenta un escenario típico de aplicación y una arquitectura de alto nivel que emana de él y que sirve para integrar los mecanismos que se detallarán posteriormente; así, en la Sección IV se describen los mecanismo de gestión de políticas y su aplicación en entornos de comunicaciones móviles (más concretamente en el ámbito de la tecnología UMTS). La Sección V presenta un mecanismo, basado en un conjunto de reglas sencillas, que permiten encontrar el precio óptimo que un operador debería establecer en sus conexiones, para maximizar su beneficio. Finalmente, la Sección VI concluye el artículo, presentando una serie de líneas futuras de actuación, que se abren a raíz del trabajo presentado.

ISBN: 978-84-694-5948-5

II. Estado del Arte y Antecedentes

Durante la segunda mitad de la pasada década, se observó una actividad investigadora muy relevante en arquitecturas/mecanismos/algoritmos que trataban de asegurar una elección de acceso óptimo en entornos altamente heterogéneos de redes. El punto de partida puede datarse en 2003, cuando Gustafsson y Johnnson acuñan el término Always Best Connected (ABC) [1]. A partir de ese momento, aparecen numerosas iniciativas y propuestas que proponen arquitecturas para conseguir alcanzar este objetivo; algunas de las más importantes serían: el Multi-Radio Resource Management (MRRM) [2], que se acuña en el proyecto europeo Ambient Networks, o el Joint Radio Resource Management (JRRM) [3] y el Common Radio Resource Management (CRRM) [4], resultados de los proyectos Everest y AROMA, respectivamente. Además, se tendría que destacar el papel jugado por los organismos de estandarización más relevantes, que también identifican la necesidad de trabajar es este tipo de escenarios; así, el grupo de trabajo 802.21 del IEEE, especifica el Media Independent Handover Framework (MIHF) [5,6], en el que se describe la señalización necesaria para acometer procesos de selección de acceso (handover) en entornos de red heterogéneos. Asimismo, en el ámbito del grupo de trabajo 3GPP [7] también se han especificado mecanismos para favorecer la interconexión de redes celulares y accesos más "locales" (WiFi).

A pesar de la existencia de las arquitecturas y procedimientos que se han descrito previamente, quedan aún un número relevante de aspectos y retos a cubrir. Algunos de ellos son consecuencia directa de los avances que se han ido produciendo (como podría ser el uso cognitivo de los recursos radio, espectro radioeléctrico, o la eclosión de las topologías multi-salto, en la que la gestión de los recursos conlleva un número notablemente mayor de dificultades [8]). Por otro lado, también es cierto que sigue siendo necesario analizar con un mayor nivel de detalle cuáles son las posibilidades que se abren con este tipo de funcionalidad, tanto desde un punto de visto analítico [9] (para entender cuáles son los límites en su comportamiento) como más realista (para analizar cuáles son los límites impuestos por los protocolos de señalización necesarios, por los acuerdos de cooperación entre las diferentes entidades, etc).

Uno de los elementos que mayor relevancia han adquirido en esta nueva filosofía de red es la de que el usuario (de manera automática) tenga un mayor grado de responsabilidad a la hora de tomar la decisión respecto a la red a utilizar. Hay varios criterios que se pueden emplear para tomar una decisión "óptima" [9], entre los que podríamos mencionar: la carga de los elementos de acceso, la calidad del enlace radio, la conexión con un operador con el que se tenga un acuerdo, etc. Otro parámetro fundamental es, evidentemente, el precio que tendría que pagar para acceder al servicio. A día de hoy las compañías ofrecen tarifas planas a sus clientes, pero la aparición de nuevos competidores, con estrategias mucho más agresivas y competitivas, puede que cambie esta tendencia. Además, los operadores de la red, que como se ha dicho anteriormente, van a tener menos grado de responsabilidad en las decisiones, pueden emplear sus tarifas como elementos para disuadir (encarecer) o favorecer (abaratar) las conexiones de los usuarios finales.

Nos encontramos con un problema con dos vertientes claramente relacionadas entre sí; por un lado se analizarán los procedimientos de gestión de políticas de servicio (incluyendo el precio) por parte de los operadores, con el objetivo de maximizar su beneficio y, además, se estudiarán los procedimientos de selección de acceso por parte del usuario utilizando como criterio fundamental el precio a pagar por la conexión.

III. Escenario y Arquitectura de Referencia

Uno de los objetivos del proyecto C3SEM es proponer una arquitectura integrada de red de acceso vía radio y de gestión de servicios ofrecidos a través de dicha red de acceso. Entre las características de dicha arquitectura cabe citar su flexibilidad y autonomía de gestión. En cuanto a la primera, significa integrar un conjunto de funcionalidades para aprovechar las posibilidades de un entorno de acceso heterogéneo, junto al uso de técnicas cognitivas y técnicas de reenvío cooperativo. En cuanto a la segunda, decir que debe desarrollar el paradigma de gestión autónoma de servicios, haciendo uso del contexto de los dominios relevantes. En este ámbito entenderemos por servicio el que permite al usuario acceder a contenidos u otras aplicaciones de la red.

Para delimitar el marco del problema que tratamos, supóngase un escenario en el que se ofertan en régimen competitivo distintos servicios de información por parte de varios proveedores a los mismos colectivos de usuarios. Estos servicios exigen la concurrencia de diferentes componentes, algunos de ellos creados ad hoc y otros ya desplegados en la red, y que constituían el soporte de otras aplicaciones (servicios de acceso, de transporte, de localización, de seguridad, etc.). En suma, tenemos un conjunto de servicios y componentes de servicio, de los que queremos destacar el de acceso a red constituido por tecnologías como Wi-Fi, WiMAX, UMTS , ofertados por distintos proveedores, y que debidamente federados permiten ofrecer productos a los usuarios finales

Nuestro escenario parte de la base que el producto final no está ligado a la tecnología de acceso. Por consiguiente, uno de los aspectos en que pivota nuestra arquitectura es facilitar el acceso de forma flexible. Para ello supondremos la existencia de un flujo de información inter-dominios (proveedores de servicios, proveedores de recursos y componentes de servicios principalmente) que facilitará los procesos de auto-reconfiguración del acceso. Por supuesto, que un elemento clave para ejecutar este proceso de reconfiguración serán las técnicas de red cognitiva que harán uso de la información anterior y de la información local del proveedor de red para recomendar un eventual handover vertical entre tecnologías de acceso a una parte de los usuarios finales, para adecuar los recursos de la red de acceso a las necesidades del servicio. El tipo de información que se manejará para determinar el acceso óptimo será de naturaleza diversa pero cabe resaltar que entre otras se hará uso de la oferta y de la demanda y que en definitiva se traducirá en el precio que finalmente se aplicará al producto o servicio.

Todavía en este escenario, es de destacar su naturaleza dinámica en el sentido que contempla una demanda que varía con el tiempo. Los usuarios acceden al servicio de forma aleatoria; nuevos usuarios se suscriben al servicio y otros causan baja. Por otro lado, los proveedores de recursos

pueden aportar o retirar recursos también en función de la demanda que perciben de los proveedores de servicios. Para mantener la calidad de todos los servicios cabe aumentar la capacidad de los recursos asignados a la aplicación (más servidores y/o memoria y capacidad de proceso en algún servidor, y/o aumentar la capacidad de los recursos de acceso a la red en alguna área geográfica).Todo esto es harto complejo si se pretende resolver con técnicas de mejor esfuerzo y/o manuales. La complejidad del proceso hace necesaria la aplicación de sistemas de gestión autónomos. Una aplicación y servicios gestionados con principios de autonomía es consciente (self-aware) de esta situación y se adaptará a la misma (auto-reconfiguración). Esta auto-reconfiguración provocará la asignación de "recursos apropiada", en los "nodos apropiados" y/o a los "usuarios apropiados" del servicio final. Los sistemas autónomos de gestión ejecutarán parte de este proceso en el dominio local del proveedor de la aplicación, mientras que otra parte se realizará en dominios remotos.

En resumen, por un lado transparencia entre dominios para facilitar el flujo de información imprescindible aunque respetando siempre la no injerencia, de forma que cada dominio pueda aplicar las políticas de gestión que considere más apropiadas. Por otro lado, empleo de técnicas adaptativas y algoritmos de optimización eficientes. Todo ello nos lleva a la conclusión que dar con una arquitectura cerrada válida para este escenario tan abierto es prácticamente inalcanzable. Sin embargo, la existencia de una arquitectura marco genérica que sintetice los principios de diseño y sirva para instanciarse en casos concretos es fundamental.

La idea que se acaba de exponer se concreta en el diagrama de la Figura 1. Los clientes y sus usuarios gozan de servicios ofrecidos por diferentes proveedores de servicio. Estos últimos cuentan con la concurrencia de varios proveedores de red con diversas tecnologías de acceso. Nótese que proveedores de servicios y de red son en realidad roles que pueden ser adoptados por la misma o por distintas entidades administrativas. En todo caso, asumimos que las distintas tecnologías de acceso de los proveedores de red se gestionan mediante principios de gestión autónoma y que la coordinación de las mismas corre a cargo del proveedor de servicios que en un momento dado haga uso de ellas. A tal efecto, el proveedor de servicios cuenta con mecanismos apropiados para coordinar las tecnologías de acceso subyacentes. Al conjunto de todos estos mecanismos de coordinación lo denominamos Unidad de Orquestación del

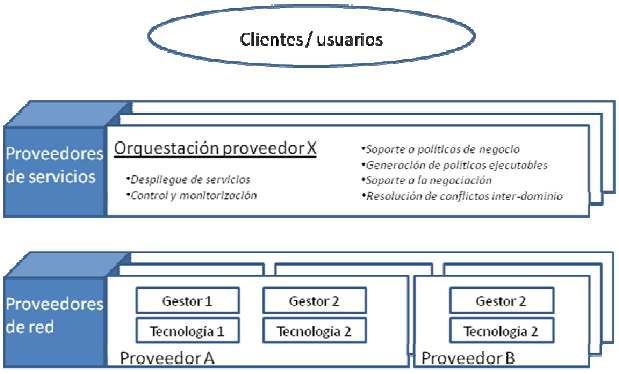


Figura 1. Arquitectura genérica de base en C3SEM

proveedor. La concepción y diseño de esta unidad es una de las tareas principales de este proyecto.

IV. Gestión de políticas para redes móviles

El Sistema de Telecomunicaciones Móviles Universal UMTS (del inglés *Universal Mobile Telecommunications System*) es hoy por hoy una alternativa apropiada para soportar los nuevos servicios de alta velocidad en redes móviles. Diversos grupos de estandarización han desarrollado los elementos necesarios para que la tecnología UMTS pueda transportar la información con garantías suficientes de calidad de servicio. Estos elementos incluyen [10]:

- Mapeo de servicios punto-a-punto con los propios de la arquitectura UMTS, que incluyen el UE (*User Equipment*), la UTRAN (*UMTS Terrestrial RAN – Radio Access Network*), la red central y redes IP externas.
- Clases de tráfico asociadas con parámetros de calidad de servicio.
- Negociación y localización de funciones de calidad de servicio.
- Multiplexado de flujos hacia los recursos de red.
- Un modelo que relaciona los servicios (subyacentes) necesarios para soportar los servicios punto a punto.

Un aspecto clave que permite a la tecnología UMTS soportar servicios multimedia basados en IP, y otros con garantías de calidad de servicio, es la arquitectura de control basada en políticas dentro del subsistema multimedia IP (IMS por sus siglas en inglés *IP Multimedia Sub-system*) [11]. Mediante un marco de trabajo basado en políticas, los operadores cuentan con un mecanismo de configuración dinámica de los dispositivos de red, que les permite alterar los servicios multimedia ofrecidos de una manera dinámica y en tiempo real.

La posibilidad de controlar el acceso a los servicios proporcionados por la red UMTS y la utilización eficiente de los recursos que los soportan es requerimiento necesario para dotar de ciertas garantías de calidad de servicio. Así, en los últimos años se han estudiado diversos esquemas de control de acceso, de ajuste de tasas de transmisión de usuarios, mecanismos de control de congestión y de gestión de clases de servicio soportadas por la tecnología UMTS. Aunque estas soluciones han demostrado parcialmente la posibilidad de proveer servicios de alta velocidad con calidad de servicio, su utilización en escenarios en los que los recursos se tienen que gestionar de manera eficiente no ha sido relevante, a pesar de que el objetivo final de cualquier infraestructura de redes y servicios ha de ser la maximización de la eficiencia económica.

El proyecto C3SEM desarrolla una metodología de gestión de servicios, similar a la presentada en [12], y soportados por tecnología UMTS, considerando aspectos de negocio y parámetros de calidad de servicio (ver Figura 2).

La metodología considera la definición de indicadores de negocio claves, tales como pérdidas económicas por rechazo en el acceso a servicio, por degradación del mismo, y la satisfacción de los usuarios. Se definen métricas para cada uno de estos indicadores de negocio, que permiten analizar las estrategias de los operadores, para su posterior retroalimentación y, además, para posibilitar la (re)definición de la estrategia de gestión.

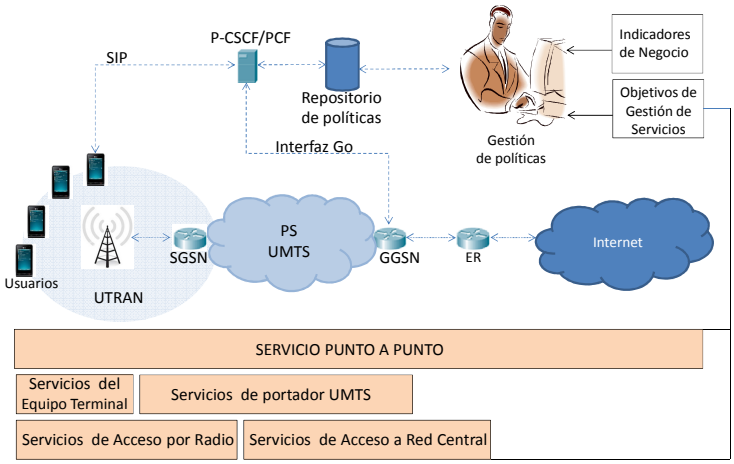


Figura 2. Escenario de optimización de políticas de configuración de servicios en entornos UMTS

Además, se contempla la correlación de los indicadores de negocio con objetivos de gestión de los servicios subyacentes: servicios del equipo terminal, servicios de portador UMTS, servicios de acceso por radio, servicios de acceso a la red central, tal y como se refleja en la Figura 2.

Finalmente, la metodología diseñada contempla diseñar las políticas de gestión que controlarán los servicios mencionados anteriormente, para ser implementadas en los elementos de red encargados de soportar dichos servicios: P-CSCF (de las siglas en inglés *Proxy Call State Control Function*), PCF (de las siglas en inglés *Policy Control Function*), y el GGSN (de las siglas en inglés *Gateway GPRS Serving Node*).

Como se puede ver, el objetivo final de la metodología que se propone son las políticas de gestión que controlarán e implementarán las estrategias de negocio del proveedor (o conjunto de proveedores). Dichas políticas definirán, por ejemplo, los parámetros y granularidad en la monitorización, umbrales de varios parámetros técnicos de la red y los servicios subyacentes, y las acciones correctivas, alineadas a las estrategias de negocio. Las acciones estarán destinadas, por ejemplo, a permitir rechazar usuarios de un perfil determinado, o el ajuste de las tasas de transmisión de ciertos grupos de usuarios en algún canal de radio en concreto, etc.

Para llevar a cabo la validación de esta metodología, el proyecto C3SEM desarrolla una plataforma basada en OPNET [13], que implementará las políticas de gestión y los mecanismos propuestos a lo largo de esta investigación. Con esta plataforma se pretende ampliar el enfoque actual hacia la definición de diversas estrategias de maximización de beneficio económico ante diferentes patrones de comportamiento de los usuarios en lo que respecta a la invocación de servicios, movilidad, y utilización de recursos de red.

V. GESTIÓN DE PRECIOS

En el contexto expresado anteriormente, en el que diferentes operadores compiten por un usuario y diferentes tecnologías a veces compiten y otras colaboran para mejorar la calidad global de un determinado servicio de acceso, las estrategias de establecimiento de precios juegan un papel fundamental. Fijando el precio correcto para un determinado servicio, un operador intentará obtener la mayor ganancia posible, mientras que los usuarios intentarán obtener un servicio que cumpla con sus requerimientos al menor precio posible. Así, el precio de un servicio debe ser elegido de forma tal que ambos se beneficien, es decir, para maximizar la ganancia del operador y la satisfacción de los usuarios.

En nuestro modelo hay dos factores principales que influyen sobre el precio de un servicio: la demanda de los usuarios y la competencia entre los operadores. Precio y demanda depende mutuamente entre sí, de tal manera que si la demanda es alta, el operador puede subir el precio de un servicio para incrementar su ganancia. Sin embargo, si la demanda es baja, el precio debe reducirse para atraer a más usuarios. La competencia entre proveedores de servicios de acceso también influye al precio de un servicio. Normalmente, si estos son sustituibles (aunque sean diferentes), los usuarios comprarán el que proporcione la mayor satisfacción al menor precio posible.

Sin embargo, estas reglas, bien conocidas en un mercado de servicios substituibles, no pueden aplicarse directamente a los servicios de telecomunicaciones actuales, en los que el usuario suele estar obligado contractualmente con un único operador por un período considerablemente largo de tiempo (meses o años).

El trabajo realizado en este proyecto presupone que esta situación podría revertirse y que un modelo más libre de competencia y demanda, como el descrito anteriormente, es viable, dando lugar a un beneficio mayor, tanto para los usuarios como para los operadores.

Nuestro escenario es el de un sistema distribuido, basado en reglas, que fija precios de servicios de acceso dinámicamente, implementando exactamente las mismas ideas de demanda y mercado que se han mencionado anteriormente, expresadas en forma de políticas de gestión de precios para ser incorporadas en cada punto de acceso (estación base) de un determinado operador. Las políticas, expresadas como reglas, que se han diseñado buscan mejorar la calidad de servicio e incrementar la ganancia global de un proveedor en un escenario en el que los usuarios son libres de elegir, de manera dinámica (esto es, sin estar atados por un contrato), el operador con el que se conectan.

A. Arquitectura General de una Solución Dinámica para Establecer Precios

La arquitectura del sistema (puede verse en más detalle en [14]) tiene tres tipos de actores: usuarios, proveedores y un regulador (ver Figura 3).

Los usuarios son personas que poseen algún dispositivo inalámbrico móvil y que desean establecer una conexión con determinados requerimientos de calidad de servicio y al menor coste posible. Estos usuarios no están unidos contractualmente a ningún proveedor en concreto, sino que pagan, bajo demanda, por los servicios de un proveedor por medio de, por ejemplo, una tarjeta de crédito o algún tipo de sistema de pre-pago. En nuestro modelo, frente a dos servicios sustituibles, los usuarios elegirán los servicios del operador más barato, y establecerán una conexión por un tiempo determinado con éste, que mantendrá el mismo precio durante el período contratado.

Los proveedores tienen una red de acceso conformada por un conjunto de dispositivos, llamados *Access Points* (AP), que ofrecen un servicio a un precio variable en el tiempo e independiente de los otros APs del mismo proveedor. De esta forma, un AP en particular se podrá adaptar a la cantidad y perfil de los usuarios en su área de cobertura en un momento dado. Por ejemplo, en horario de oficina, un AP en una localización céntrica de un entorno urbano puede permitirse ofrecer un precio más alto por el mismo servicio que un AP del mismo proveedor en un barrio alejado. El objetivo de un proveedor es, evidentemente, maximizar sus ganancias globales.

Finalmente, la tercera entidad, el regulador, es una entidad neutral, que, en una situación real, se podría ver como un organismo gubernamental, encargado de asegurar que los diferentes proveedores conozcan, en tiempo real, el precio de sus competidores en una determinada región; se asume que estos compartirán información de precios, con el objetivo de habilitar una competencia de mercado adecuada.

B. Políticas de Gestión de Precios

Las políticas de gestión de pecios del sistema tienen el objetivo de permitir a cada AP decidir el precio óptimo al que cobrar sus servicios, dado su contexto, la demanda y sus competidores potenciales. Estas decisiones son tomadas de forma autónoma por un "*Policy Decision Point*" [15], que se integra en los AP siguiendo un conjunto de políticas definidas por el operador correspondiente, y que se expresan como reglas, que modelan los criterios económicos de demanda y competencia mencionados.

En particular, los AP utilizan 15 reglas basadas en criterio de demanda [4], siendo las dos siguientes un ejemplo ilustrativo de las mismas.

```
if few_users & users_steady then
decrease_price_slow

if lots_users & users_increasing_fast then
increase_price_fast
```

Además, también hacen uso de 9 reglas que se basan en la competencia [4], de las que se muestran, a modo de ejemplo, las dos siguientes.

```
if competitor_price_lower &
competitor_price_decreasing_slow then
decrease_price_slow

if competitor_price_higher &
competitor_price_increasing_fast then
increase_price_fast
```

La idea detrás de esos dos conjuntos de reglas es simple, el precio de un servicio sube, se mantiene o baja dependiendo del número de usuarios a los que se les da servicio en un momento determinado, y la tendencia correspondiente, así como de la relación del propio precio con el de la competencia y la dinámica correspondiente.

Si se presta atención a las reglas, y se considera que se imponen independientemente en cada AP, puede inferirse que las reglas basadas en demanda operan creando una cierta competencia entre APs de un mismo proveedor. Así, si un AP tiene usuarios en exceso subirá su precio pero si otro en su cercanía (y del mismo operador) tuviera recursos libres, podría bajar su precio, por lo que aquellos usuarios que en la cobertura de ambos optarán por el segundo, generándose así un balanceo de carga natural entre APs de un mismo operador. Esta situación puede observarse en las figuras 4 y 5, generadas a partir de los resultados de simulaciones descritas en [14]. En la Figura 4 se muestra un área geográfica cuadrada cubierta por nueve celdas de un

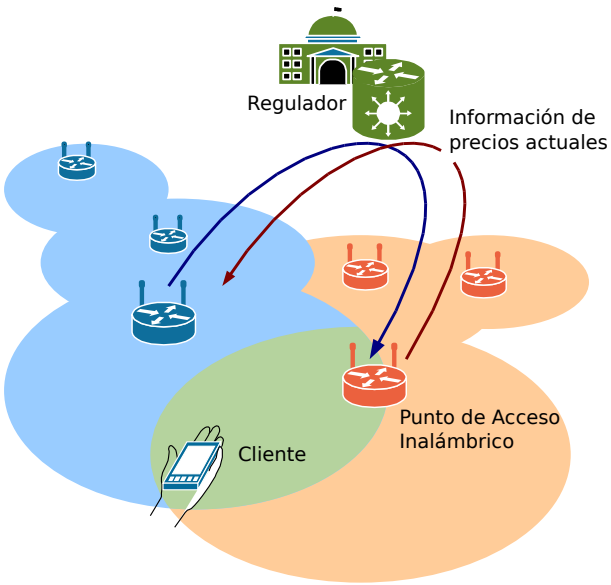


Figura 3. Escenario de Fijación de Precios Dinámica

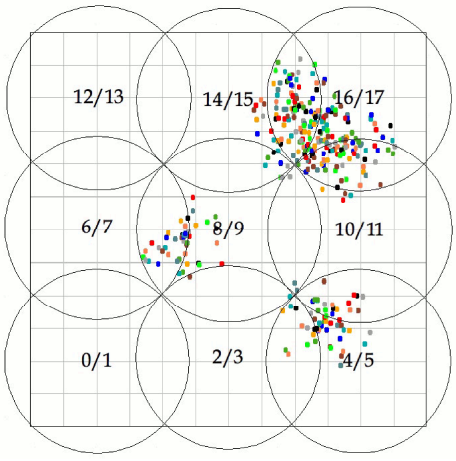


Figura 4 Distribución de Celdas y Usuarios en Grupos

operador tradicional que utiliza una estrategia de precios de tarifa fija (celdas impares) y nueve celdas, en las que el AP fija el precio siguiendo reglas como las mencionadas anteriormente (celdas pares). Se trata de una situación ideal, en la que ambos operadores compiten en igualdad de condiciones geográficas y de recursos. En dicha figura también puede verse una distribución de usuarios no homogénea, lo que origina una demanda mayor de recursos en algunas de las celdas, dejando recursos libres en otras.

En la Figura 5, se aprecia la utilización de recursos como porcentaje del ancho de banda disponible contratado por los usuarios, según las simulaciones presentadas en [14]. Como puede apreciarse, la distribución de carga del proveedor que utiliza una estrategia de precios dinámica es mucho más homogénea que la del operador tradicional.

En [14] puede verse el ejemplo anterior con mayor nivel de detalle, así como una serie de resultados que ponen de manifiesto que la estrategia de precios basada en reglas también maximiza ganancias. Estos resultados preliminares ponen de manifiesto que un modelo dinámico de asignación de precios (y de libre competencia entre operadores) puede impactar positivamente, tanto en la calidad de los servicios obtenidos por los usuarios como en las ganancias de los proveedores de acceso.

A raíz de los resultados preliminares que se presentan en [14] y de la herramienta que se describe en [9] se pueden analizar cuáles son las estrategias de selección de acceso que pueden asegurar un comportamiento óptimo en cuanto a los beneficios de los operadores – para una política de precios dada. Por otro lado, se está analizando, utilizando teoría de juegos [16], cuál sería la estrategia que daría lugar a un beneficio óptimo y, a diferencia del trabajo presentado en [16], con escenarios más dinámicos.

VI. Conclusiones

El proyecto C3SEM recoge algunos de los retos que aparecen con el incremento notable de la heterogeneidad en las redes de acceso (no sólo tecnológica, sino también en lo que se refiere a los operadores que están detrás de ellas), planteando una arquitectura con técnicas innovadoras, en la que se establece una fuerte cooperación entre el substrato de comunicación subyacente y los servicios que soporta. A partir de las posibilidades que se abren con la solución propuesta, aparece un extenso conjunto de líneas de investigación o de actuación que pueden ser exploradas.

Dentro de dicho abanico de posibilidades, este trabajo esboza en primer lugar una metodología, en un escenario basado en la tecnología UMTS, en la que el operador se podría beneficiar de un conjunto de políticas de gestión de servicios basada en parámetros de negocio y elementos de calidad de servicio, con el objetivo de incrementar los beneficios. Seguidamente, el trabajo se centra en las políticas y estrategias de asignación de precios de los operadores, que a medio plazo (y debido a la aparición de nuevos servicios y modelos de negocio) tengan que replantearse las soluciones actuales (habitualmente basadas en tarifas planas). Entre sus aspectos más innovadores cabe citar el uso de mecanismos de auto-aprendizaje y un conjunto de reglas sencillas, para que los operadores puedan adaptar sus tarifas en función de la dinámica del escenario en el que se encuentran. La arquitectura propuesta se ha aplicado en un escenario con dos operadores y una entidad reguladora, poniendo de manifiesto un notable incremento de los beneficios que se pueden obtener.

Los trabajos presentados en este artículo están en clara fase de desarrollo. De forma inmediata esperamos obtener los primeros resultados de la metodología de optimización de políticas de gestión en entornos de acceso UMTS con objetivos tecno-económicos. Por otro lado, se ampliarán los escenarios en los que se han analizado las políticas de asignación de precios, utilizando herramientas más analíticas que permitan obtener comportamientos óptimos, para ser contrastados posteriormente con los obtenidos con las herramientas de simulación que se han desarrollado. En este sentido se plantea la utilización de técnicas de optimización lineal, así como de la teoría de juegos, que pueden aportar un gran valor a la investigación que se está llevando a cabo.

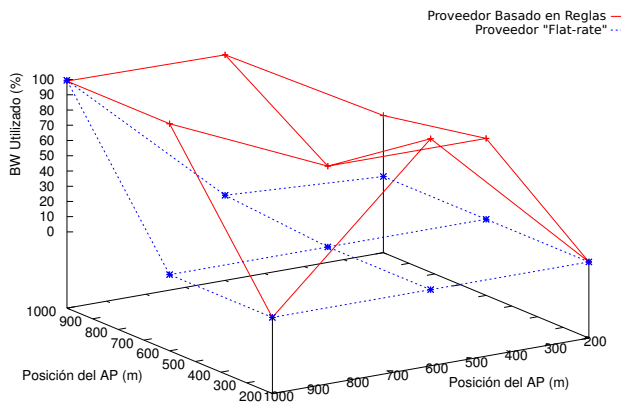


Figura 5 Balanceo de Carga de la Estrategia Basada en Precios

Agradecimientos

Los autores desean expresar su agradecimiento al Ministerio de Ciencia e Innovación, por su financiación en el proyecto "Comunicaciones Cognitivas, Cooperativas y Gestión Autónoma de Servicios", C3SEM (TEC2009-14598-C02-01/02). Javier Rubio-Loyola y Pablo Salazar agradecen la colaboración del OPNET University Program.

Referencias

[1] E. Gustafsson, A. Jonsson, "Always best connected," Wireless Communications, IEEE , vol.10, no.1, pp. 49- 55, Feb. 2003

[2] J. Sachs et al., "Generic abstraction of access performance and resources for multiradio access management" Mobile and Wireless Communications Summit, 2007. 16th IST (julio 2007)

[3] L. Giupponi, R. Agusti, J. Perez-Romero, O. Sallent. "Joint radio resource management algorithm for multi-RAT networks". Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE. vol. 6 (Diciembre 2005)

[4] J. Perez-Romero, et al. "Common radio resource management: functional models and implementation requirements." Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on (2005)

[5] IEEE Standard for Local and Metropolitan Area Networks; Part 21: Media Independent Handover (2009)

[6] E. Piri, K. Pentikousis, "IEEE 802.21: Media Independent Handover Services" The Internet Protocol Journal 12(2), 7 – 27 (Junio 2009)

[7] 3rd Generation Partnership Project 3GPP TS 23.402. 3GPP System Architecture Evolution: Architecture Enhancements for non-3GPP accesses (Release 8), Enero 2007.

[8] L. Ferreira, M. De Amorim, L. Iannone, L. Berlemann, y L. Correia, "Opportunistic management of spontaneous and heterogeneous

wireless mesh networks," Wireless Communications, IEEE, vol. 17, no. 2, pp. 41 –46, abril 2010.

[9] R. Agüero, J. Choque, E. Hortigüela, L. Muñoz. "Aplicación de técnicas de programación lineal en la asignación óptima de recursos en redes inalámbricas heterogéneas". Actas de las IX Jornadas de Ingeniería Telemática. JITEL2011 (Septiembre 2010).

[10] S. Dixit et al. "Resource Management and Quality of Service in Third-Generation Wireless Networks". IEEE Communications Magazine, February 2001

[11] W. Zhuang et al. "Policy-based QoS Architecture in the IP Multimedia Subsystem of UMTS". IEEE Network Magazine. May/June 2003

[12] J. Rubio-Loyola et al. "Business-driven Management of Differentiated Services" 12th IEEE/IFIP Network Operations and Management Symposium NOMS 2010 , 19-23 April 2010 Osaka, Japan

[13] OPNET Network Simulator http://www.opnet.com

[14] J. Baliosian, J. Serrat et al. "Policy-based Pricing for Heterogeneous Wireless Access Networks", in proceedings of the 5th International Conference on Autonomous Infrastructure, Management and Security (AIMS 2011), June 13-17, 2011, Nancy France.

[15] B. Moore, E. Ellesson, J. Strassner, andA. Westerinen, "Policy Core Information Model, RFC3060" (ProposedStandard), Feb. 2001. [Online]. Available: http://www.ietf.org/rfc/rfc3060.txt

[16] D. Niyato, E. Hossain, "A game theoretic analysis of service competition and pricing in heterogeneous wireless access networks," Wireless Communications, IEEE Transactions on, vol.7, no.12, pp.5150-5155, Diciembre 2008

A Consensus Policy Based Protocol for Multi-Agent Negotiation

Miguel A. López-Carmona and Iván Marsá-Maestre
Departamento de Automática, Universidad de Alcalá
{miguelangel.lopez,ivan.marsa}@uah.es

***Abstract*—Multiagent negotiation may be understood as a consensus based group decision-making process which ideally should seek the agreement of all the participants. However, there exist situations where an unanimous agreement is not possible or simply not desired. We propose to use a Consensus Policy based Mediation Framework (CPMF) to perform multiagent negotiations. This proposal fills a gap in the literature where protocols are in most cases biased to search for an unanimous agreement. The mechanisms proposed to perform the exploration of the negotiation space are derived from the Generalized Pattern Search (GPS) non-linear optimization technique. The mediation process is guided by the aggregation of the agent preferences on the set of alternatives the mediator proposes in each negotiation round. Considerable interest is focused on the implementation of the mediation rules where we allow for a linguistic description of the type of agreements needed. We empirically show that CPMF efficiently manages negotiations following predefined consensus policies.**

I. Introduction

Most research in multiparty automated negotiation has been focused on building efficient mechanisms and protocols to reach agreements among multiple participants, optimizing some type of social welfare measurement like the sum or product of utilities [1], [2], [3], [4], [5]. These proposals have been mainly focused on obtaining unanimous agreements, not considering situations where unanimous agreements are not possible or not desired.

We propose CPMF, a Consensus Policy based Mediation Framework for Multi-Agent Negotiation. CPMF relies on a novel mediation protocol based on the Generalized Pattern Search (GPS) non-linear optimization technique [6] and on the use of Ordered Weighted Averaging (OWA) operators [7]. This framework allows a mediator to search for agreements following predefined consensus policies, which may take the form of linguistic expressions. It is worth noting, however, that our negotiation protocol is not a multi-objective centralized optimization process. Agents will only reveal their preferences for the contracts proposed by a mediator, not their utility functions.

Next section presents the basic operation of the negotiation protocol. Section 3 focuses on the mechanisms used by the mediator to aggregate agents' preferences and Section 4 presents the agreement search process. Section 5 describes the experimental evaluation and the last section summarizes our conclusions.

II. The Mediation Protocol

We shall assume a set of n agents $A = \{A_i | i = 1, \ldots, n\}$ and a finite set of issues $X = \{x_l | l = 1, \ldots, s\}$, where each issue x_l can be normalized to a continuous or discrete range $d_l = [x_l^{min}, x_l^{max}]$. Accordingly, a *contract* is a vector $x = \{x'_l | l = 1, \ldots, s\}$ defined by the issues values. Furthermore, we assume that each agent A_i has a real or virtual mapping $V_i : X \rightarrow \mathbb{R}$ function that associates with each contract x a value $V_i(x)$ that gives the payoff the agent assigns to a contract. The preference function can be described as any mapping function between the negotiation space contracts and the set of real numbers, and it can be non-monotonic and non-differentiable. The aim of the agents will be to reach an agreement on a contract x maximizing their individual payoff and minimizing the revelation of private information.

A. The GPS Optimization Algorithm

In GPS [6], the optimization process of a function begins with the generation of an initial random point (*reference point*). The algorithm generates a set of points (*mesh*) around the reference point at a predefined distance, and evaluates the function at each of these points. If the evaluations do not improve the evaluation of the reference point, a new mesh is generated at half the current distance and the process is repeated. If one or more evaluations in the mesh improve the evaluation of the reference point, we move to the point with the highest improvement, and this point becomes the current reference point. In this case, a new mesh is generated increasing by a factor of 2 the current distance. The optimization process may finish when the distance falls below a predefined threshold or when a deadline or function evaluations count expires.

Formally, the maximization problem can be defined as $\max f(x)$, where $f : \mathbb{R}^m \rightarrow \mathbb{R}$, $x \in \mathbb{R}^m$. At an iteration k of the protocol, we have an iterate $x(k) \in \mathbb{R}^m$ and a step-length parameter $\triangle_k > 0$. We successively look at the points in the *mesh* $x^+(k) = x(k) \pm \triangle_k e_j$, $j \in \{1, \ldots, m\}$, where e_j is the jth standard basis vector, to search for a contract $x'(k)$ in $x^+(k)$ for which $f(x'(k)) > f(x(k))$. We will use the notation $x^{+o}(k)$ to designate the mesh at round k plus the current point $x(k)$ (see Figure 1). This set of points or mesh is an instance of what we call a *pattern*. If we find no $x'(k)$ such that $f(x'(k)) > f(x(k))$, then we reduce $\triangle_k$ by half and continue; otherwise, we increase by a factor of 2 the step-length parameter, setting $\triangle_{k+1} = 2 \cdot \triangle_k$ and $x(k+1) = x(k)$. We repeat the iteration just described until $\triangle_k$ is deemed

ISBN: 978-84-694-5948-5

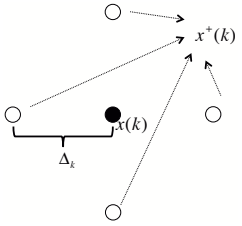


Figure 1. An illustration of a mesh for $m = 2$ at round k. The reference point is $x(k)$.

sufficiently small. One important feature of pattern search that plays a significant role in a global convergence analysis is that we do not need to have an estimate of the derivative of f at $x(k)$ so long as included in the search is a sufficient set of directions to form a positive spanning set for the cone of feasible directions, which in the unconstrained case is all of $\mathbb{R}^m$.

The set e_i is defined by the number of independent variables in the objective function m and the positive standard basis set. A commonly used positive basis is the maximal basis, with $2m$ vectors. For example, if there are two independent variables in the optimization problem, the default for a $2m$ positive basis consists of the following pattern vectors: $e_1 = \{1, 0\}$, $e_2 = \{0, 1\}$ and $-e_1 = \{-1, 0\}$, $-e_2 = \{0, -1\}$. We will use the notation $x^{e_j}(k)|j = 1, \ldots, 2m$ to describe each point in a mesh, and $x(k)$ or $x^{e_0}(k)$ to designate the current point. For example, $x^{e_1}(k)$ specifies the contract generated by the current contract $x(k)$ and the vector e_1 for the current step-length $\triangle_k$, while $x^{e_{m+1}}(k)$ points to the negative version of $x^{e_1}(k)$.

B. Basic Operation of the Negotiation Protocol

The basic protocol of the negotiation process is as follows:

1) The mediator proposes a mesh from an initial random contract $x^{ini}(1)$ for a step-length parameter $\triangle_1$.
2) Each agent provides the mediator their preferences for the contracts in the current mesh x^{+o}, in terms of a mapping $S_i : X \rightarrow [0, 1]$ such that for example $S_i(x^{e_j}(k))$ indicates agent i's support for the alternative $x^{e_j}(k)$. An agent does not know the other agents' support for the contracts. Though agents are free to provide support values which are coincident or not with the corresponding private valuation function $V_i(x^{e_j}(k))$, in this work we will assume a perfect correspondence between both values.
3) The individual preferences for each contract are aggregated by the mediator to obtain the **group preferences**.
4) The mediator obtains the **preferred contract** $x^{e_*}(k)$ according to the group preferences.
5) Based on the **the preferred contract**, the mediator decides to **expand** or **contract** the mesh. Should a contraction make $\triangle_k$ small enough negotiation ends, otherwise go to step 2.

We assume that the negotiation process is such that a solution from X is always obtained. Negotiation may end when $\triangle_k$ is below a predefined threshold value or when a deadline expires.

III. The Aggregation of Preferences

Here we look at the process where the mediator aggregates the individual support for the contracts in the mesh at round k. Our point of departure is a collection of n agents and a mesh $x^+(k)$. We assume each agent A_i has provided her preference $S_i(x^{+o}(k))$ over the set $x^{+o}(k)$ such that it indicates the degree to which each agent A_i supports each contract. The mediator's objective is to obtain a group preference function $G : x^{+o} \rightarrow [0, 1]$ which associates with each alternative $x^{e_j}(k) \in x^{+o}(k)$ a value $G(x^{e_j}(k)) = M(S_1(x^{e_j}(k)), \ldots, S_n(x^{e_j}(k)))$.

The form of M is called the *mediation rule*, it describes the process of combining the individual preferences. The form of M can be used to reflect a desired mediation imperative or *consensus policy* for aggregating the preferences of the individual agents to get the mesh group preferences. M will guide the mediator in the expansion-contraction decisions in order to meet the desired type of agreements for the negotiation process.

The most widespread consensus policy found in the automated negotiation literature suggests using as an aggregation imperative a desire to satisfy *all* the agents. We propose to use application dependent mediation rules to manage the negotiation processes. The idea is to use a *quantifier guided aggregation*, which allows a natural language expression of the quantity of agents that need to agree on an acceptable solution. As we shall see, the OWA operators [8] will provide a tool to model this kind of softer mediation rule.

A. OWA Operators

An aggregation operator $M : S^n \rightarrow G, (S, G \in [0, 1])$ is called an OWA operator of dimension n, if it has an associated weighting vector $W = [w_1 w_2 \ldots w_n]$ such that $w_t \in [0, 1]$ and $\sum_{t=1}^{n} w_t = 1$ and where $M(S_1, \ldots, S_n) = \sum_{t=1}^{n} w_t b_t$, where b_t is the tth largest element of the aggregates $\{S_1, \ldots, S_n\}$. In the OWA aggregation the weights are not directly associated with a particular argument but with the ordered position of the arguments. If ind is an index function such that $ind(t)$ is the index of the tth largest argument, then we can express $M(S_1, \ldots, S_n) = \sum_{t=1}^{n} w_t S_{ind(t)}$.

The form of the aggregation is dependent upon the associated weighting vector. We have a number of special cases of weighting vectors. The vector W^* defined such that $w_1 = 1$ and $w_t = 0$ for all $t \neq 1$ gives us the aggregation $Max_i[S_i]$. Thus, it provides the largest possible aggregation. The vector W_* defined such that $w_n = 1$ and $w_t = 0$ for all $t \neq n$ gives the aggregation $Min_i[S_i]$. An interesting family of OWA operators are the E-Z OWA operators. There are two families. In the first family we have $w_t = 1/q$ for $t = 1$ to q, and $w_t = 0$ for $t = q + 1$ to n. Here we are taking the average of the q largest arguments. The other family defines $w_t = 0$ for $t = 1$ to q, and $w_t = \frac{1}{n-q}$ for $t = q + 1$ to n. We can see that this operator can provide a softening of the original min and max mediation rules by modifying q.

B. Quantifier Guided Aggregation

Our aim is to define consensus policies in the form of a linguistic agenda for our mediation mechanisms. For example,

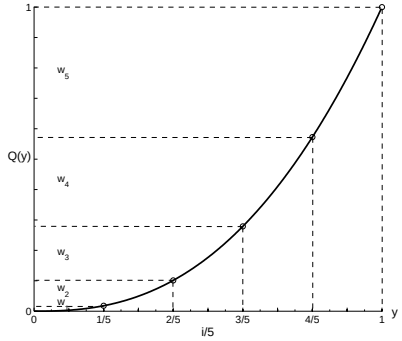


Figure 2. Example of how to obtain the weights from the quantifier for $n = 5$ agents.

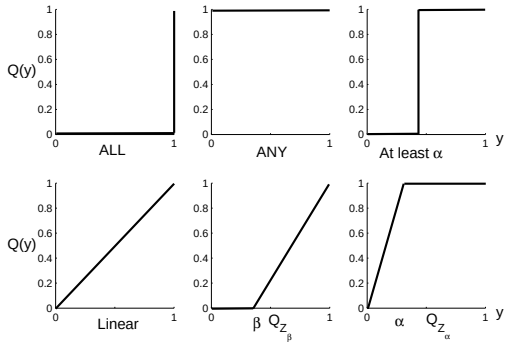


Figure 3. Functional form of typical quantifiers: all, any, at least, linear, piecewise linear Q_{Z_β} and piecewise linear Q_{Z_α}.

the mediator should make decisions regarding the exploration of the negotiation space following mediation rules like *"Most* agents must be satisfied by the contract", *"at least* α agents must be satisfied by the contract", *"many* agents must be satisfied", ...

The above statements are examples of *quantifier guided aggregations*. Zadeh [9] suggested that any relative linguistic quantifier can be expressed as a fuzzy subset Q of the unit interval $I = [0, 1]$. In this representation for any proportion $y \in I$, $Q(y)$ indicates the degree to which y satisfies the concept expressed by the term Q. Formally, these quantifiers are characterized in the following way: 1) $Q(0) = 0$, 2) $Q(1) = 1$ and 3) $Q(x) \geq Q(y)$ if $x > y$. Examples of this kind of quantifier are all, most, many, at least α. Two examples of quantifiers are *all* which is represented by Q_* where $Q_*(1) = 1$ and $Q_*(x) = 0$ for all $x \neq 1$, and *any* which is defined as $Q^*(0) = 0$ and $Q^*(x) = 1$ for all $x \neq 0$.

Under the quantifier guided mediation approach a group mediation protocol is expressed in terms of a linguistic quantifier Q indicating the proportion of agents whose agreement if necessary for a solution to be acceptable. The basic form of the mediation rule in this approach is "Q agents must be satisfied by the contract", where Q is a quantifier. The formal procedure used to implement the mediation rule is as follows:

1) Use Q to generate a set of OWA weights, $w_1, \ldots, w_n$.
2) For each contract $x^{e_j}(k)$ in $x^{+o}(k)$ use these weights to calculate the overall group support $G(x^{e_j}(k)) = M(S_1(x^{e_j}(k)), \ldots, S_n(x^{e_j}(k)))$.

The procedure used for generating the weights from the quantifier is to divide the unit interval into n equally spaced intervals and then to compute the length of the mapped intervals using Q

$$w_t = Q(\frac{t}{n}) - Q(\frac{t-1}{n}) \,\text{for}\, t = 1, \ldots, n\,.$$

In Figure 2 we show an example of a linguistic quantifier and illustrate the process of determining the weights from the quantifier. The weights depend on the number of agents as well as the form of Q. In Figure 3 we show the functional form for the quantifiers *all*, *any*, Q_*, Q^*, *at least* α *percent*, *linear quantifier*, *piecewise* Q_{Z_β} and *piecewise* Q_{Z_α}. The quantifiers *all*, *any* and *at least* α describe the consensus policy using a natural language verbal description. However, more generally any function $Q : [0, 1] \rightarrow [0, 1]$ such that $Q(x) \geq Q(y)$ for $x \geq y$, $Q(1) = 1$ and $Q(0) = 0$ can be seen to be an appropriate form for generating mediation rules or consensus policies.

One feature which distinguishes the different types of mediation rules is the power of an individual agent to eliminate an alternative. For example, in the case of *all* this power is complete, and any agent could force an alternative to be rejected by voting zero. In order to capture this idea the *Value Of Individual Disapproval* (VOIDNESS) [7]

$$VOIDNESS(Q) = 1 - \int_0^1 Q(y)dy$$

measures this power. For the *all*, *any, at least* α and *linear* quantifiers the VOIDNESS measures are respectively 1, 0, α and 0.5. For the Q_{Z_β} quantifier $VOIDNESS(Q_{Z_\beta}) = \frac{1}{2} + \frac{\beta}{2}$ and therefore $VOIDNESS(Q_{Z_\beta}) \in [0.5, 1]$. The Q_{Z_α} quantifier gets $VOIDNESS(Q_{Z_\alpha}) = \frac{\alpha}{2}$ and $VOIDNESS(Q_{Z_\alpha}) \in [0, 0.5]$. Another family of quantifiers are those defined by $Q_p(y) = y^p$ for $p > 0$. In this case $VOIDNESS(Q_p) = 1 - \int_0^1 r^p dr = \frac{p}{p+1}$. For Q_p we see that as p increases we get closer to the min and that as p gets closer to zero we get the max.

IV. The Search Process

Everytime the mediator determines the group preferred contract $x^{e_*}(k)$ within a mesh, it selects among three possible alternatives:

1) Move to the group preferred contract $x(k+1) = x^{e_*}(k)$ in $x^+(k)$ and expand the mesh by a factor of two $\triangle_{k+1} = 2 \cdot \triangle_k$.
2) Keep the current contract $x(k + 1) = x(k)$ and reduce by half the mesh step-length $\triangle_{k+1} = \triangle_k/2$.
3) Finish the negotiation process.

The alternative 1 is selected if the preferred contract is in $x^+(k)$, i.e., $x^{e_*}(k) \in x^+(k)$. If the preferred contract is $x(k)$ then the mediator selects alternative 2. Finally, we define two stopping rules, one which bounds the maximum number of rounds k, and a second one which stops negotiation when the step-length $\triangle_k$ is below a predefined threshold γ. We assume that in both cases the agreement reached is the preferred group contract in the last negotiation round.

A. Preferred Contract Selection in the Search Process

Here are described in detail the mechanisms used to select the preferred contract. The point of departure is the set of final group preferences for the contracts in $x^{+o}(k)$. We propose a probabilistic selection process to select the winner contract. We associate with each contract $x^{e_j}(k) \in x^{+o}(k)$ a probability

$$P(x^{e_j}(k)) = \frac{G(x^{e_j}(k))^\sigma}{\sum_{l=1}^{2m} G(x^{e_l}(k))^\sigma} .$$

The process selects the winner contract using a biased random experiment with these probabilities. The parameter $\sigma > 0$ works as an indication of the significance we give to the final group preferences. If $\sigma \to \infty$ we select the contract with the maximum support, which means that the mediator is given the higher significance to the group preferences. If $\sigma = 1$ then the probability of selecting $x^{e_j}(x)$ would be proportional to its group support. The rationale behind using this probabilistic process is to introduce randomness and avoid local optima in the following way.

With G, the mediator is able to select a contract within the mesh. However, this selection is based on a relative measurement and it is not considering how good is the selection made. The mediator must consider both the G value and the relative values to make the decision of expansion and contraction. Thus, we make σ vary as a function of G and the number of rounds k. If G is high, σ must be high, favouring a deterministic mesh movement, i.e. with a high probability the contract with a higher G is selected. Otherwise, if G is low, σ must be low to induce randomness and avoid local optima. More specifically, for $\sigma = 0$ the selection of contracts is equiprobable, making such selection independent of G. For $\sigma = 1$ the selection probability is proportional to G. Higher values for σ increases the probability of choosing the contract with a higher G. To control σ we define

$$\sigma(k, G) = \sigma_{min} + (\sigma_{max} - \sigma_{min}) \cdot G^{(1-\frac{k}{k_{max}})\cdot\alpha} ,$$

where σ depends on the negotiation round k and the maximum number of rounds k_{max}. The function is bounded by σ_{max} and σ_{min} given $G = 0$ and $G = 1$ respectively. The parameter $\alpha > 0$ determines the initial curvature of $\sigma(k, G)$. As the number of rounds k increases, the function increases its concaveness, which means that G induces higher values for σ, favouring convergence. Figure 4 shows the evolution of $\sigma(k, G)$ for k_{max} =50, $\alpha = 6$, $\sigma_{max} = 200$ and $\sigma_{min} = 1$. The principle of this approach is analogous to the simulated annealing technique without reannealing.

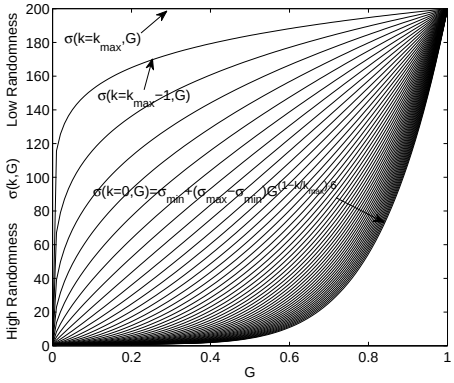


Figure 4. Sigma function.

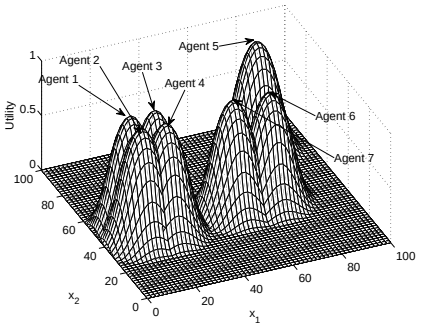


Figure 5. Testbed Utility Functions

V. Experimental Evaluation

In this section, we show that the mechanisms proposed provide the mediator the tools to efficiently conduct multiagent negotiations following different consensus policies. In the first experimental setup we have considered 7 agents, 2 issues and 2 different types of negotiation spaces: a negotiation space where agents' utility functions are strategically built to define a *proof of concept negotiation scenario*, and a *complex negotiation scenario* where utility functions exhibit a more complex structure. In both cases utility functions are built using an aggregation of *Bell functions*. This type of utility functions capture the intuition that agents' utilities for a contract usually decline gradually with distance from their ideal contract. Bell functions are ideally suited to model, for instance, spatial and temporal preferences and to simulate different levels of complexity.

A *Bell* is defined by a center c, height h, and a radius r. Let $\parallel s - c \parallel$ be the euclidean distance from the center c to a contract s, then the *Bell function* is defined as

$$fbell(s, c, h, r) = \begin{cases} h - 2h\frac{\|s-c\|^2}{r^2} & \text{if } \parallel s - c \parallel < \frac{r}{2} \\ \frac{2h}{r^2}(\parallel s - c \parallel - r)^2 & \text{if } r > \parallel s - c \parallel \geq \frac{r}{2} \\ 0 & \parallel s - c \parallel \geq r \end{cases}$$

and the *Bell utility function* as $U_{b,s}(s) = \sum_{i}^{nb} fbell(s, c_i, h_i, r_i)$ where nb is the number of generated bells. The complexity of the negotiation space can be modulated by varying c_i, h_i, r_i and nb.

In the *proof of concept negotiation scenario* each agent has a utility function with a single optimum. Figure 5 shows in the same graph the agents' utility functions in the bidimensional negotiation space $[0, 100]^2$. Four agents (Agent 1, 2, 3, 4) are in weak opposition (i.e. their preferences are quite similar), Agents 6 and 7 are in weak opposition and in very strong opposition with respect the other agents, and Agent 5 is in very strong opposition with respect the rest of the agents. In the *complex negotiation scenario* each agent's utility function

is generated using two randomly located bells. The radius and height of each bell are randomly distributed within the ranges $r_i \in [20, 35]$ and $h_i = [0.1, 1]$. The configuration of parameters in the mediator is: $k_{max} = 50$ rounds, mesh tolerance 10^{-6}, and $\alpha = 2$, $\sigma_{min} = 1$, $\sigma_{max} = 200$ for the preferred contract selection process.

We tested the performance of the protocol for 5 different consensus policies with VOIDNESS degrees: 0, 0.25, 0.5, 0.75 and 0.95, using the quantifier $Q_p(y) = y^p$, $p = 2$. We also define a contrast experiment where the consensus policy based mediation process is deactivated such that the mediator uses the pattern search based process but there is no randomness and the group preference evaluation is limited to compute the sum of agents' valuations for a given contract (i.e. the winner contract is that with the highest sum of valuations).

Each experiment consist of 100 negotiations where we capture the utilities achieved by each agent. To analyze the results we first build a $7\,\text{agents} \times 100\,\text{negotiations}$ utility matrix where each row provides each agent's utilities and each column is a negotiation. The matrix is then reorganized such that each column is individually sorted from higher to lower utility values. Note that after this transformation the association row/particular-agent disappears. Given the matrix, we form 7 different utility groups: a first group named *group level 1* where we take the highest utility from each negotiation (i.e. the first row), a second group named *group level 2* with the two first rows and so on. In order to show the performance of the protocol we have used the Kaplan-Meier estimate of the cumulative distribution function (*cdf*) of agents' utilities for each group. Thus, we compute the *cdf* for the highest utilities, for the two highest utilities and so on. The *cdf* estimates the probability of finding agent's utilities below a certain value. The rationale behind using grouping in the analysis is to evaluate the ability of the protocol to find solutions which satisfy groups of agents.

In the proof of concept scenario (see Figure 5) it can be seen that when an unanimous is needed, the best alternative is to get satisfied agents 1, 2, 3 and 4. If it is enough to have one agent satisfied, any of the utility peaks would be a good solution. In Figure 6 we show the results for the proof of concept scenario. Each line shows the *cdf* for a group level and the number above each line identifies the corresponding level. For instance, for the reference experiment and the group level 1 there is a 98% probability of having agents with utility 0.7, and a 2% probability of having agents with utility 0. In the group level 7 case, there is a 50% probability of having agents with utility 0.7, and a 50% probability of having agents with utility 0. For a VOIDNESS=0 and group level 1, however, there is a 98% probability of having agents with utility 1, which means that the mediator is applying efficiently the consensus policy which states that the main objective is to have at least one agent highly satisfied. As VOIDNESS increases (i.e. the policy is to have more agents satisfied) the *cdf* for group level 1 performs worse, though better than in the reference scenario, and for higher group levels the performance increases.

In Figure 7 are shown the results for the complex negotiation scenario. The results also show that as VOIDNESS increases, the mediator biases the search for agreements where more

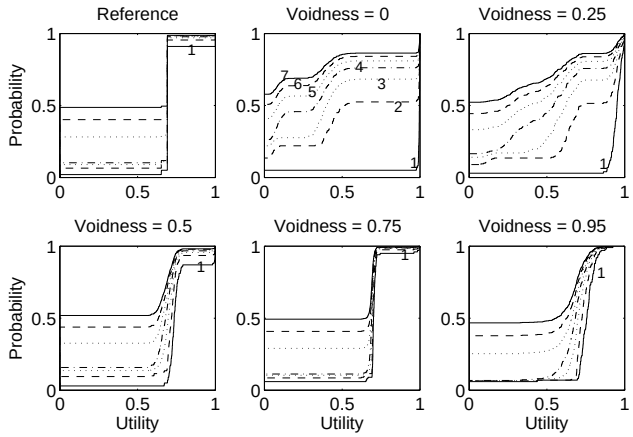


Figure 6. Cumulative distributions of utilities for the *proof of concept scenario*.

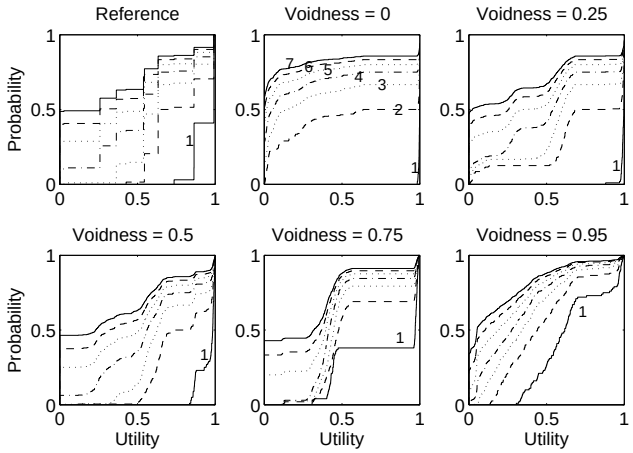


Figure 7. Cumulative distributions of utilities for the *complex negotiation scenario*.

agents are satisfied at the expense of the individual satisfaction level. In general, it is worth noting that the application of a consensus policy may incur in a cost in terms of social welfare. In a second experimental setup we have considered 7 agents, 2 issues and 5 different types of negotiation spaces in increasing complexity to evaluate this issue. Figure 8 shows the social welfare measurements (sum of utilities) for different VOIDNESS degrees. Social welfare is normalized to its optimal value. VOIDNESS ranges from 0 to 0.95 and the last tick (Ref) shows the social welfare results for the contrast experiment (the mediator explicitly maximizes social welfare). In the proof of concept scenario (the simplest one), social welfare increases with VOIDNESS. It makes sense that when there is no a strong oposition among the agents, a consensus policy which aims at satisfying as many agents as possible also maximizes social welfare. However, we can see how as complexity increases, the application of consensus policies come at a cost in terms of social welfare, both for low and for high VOIDNESS values. For example, in scenarios where there exist a strong opposition among the agents, if we want to have many agents satisfied, individual utilities cannot be simultaneously large for all the agents, and therefore social welfare decreases. Also note

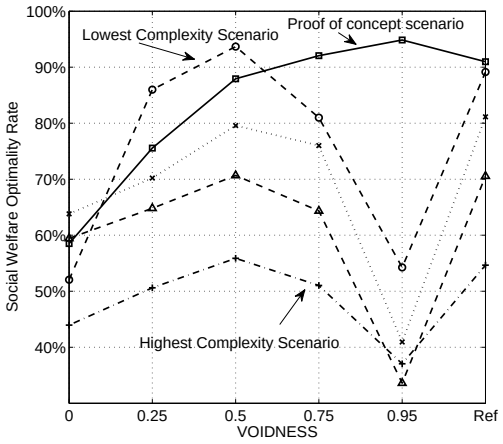


Figure 8. Social Welfare Optimimality Rate vs VOIDNESS.

that there exists a VOIDNESS value which maximizes social welfare and that this social welfare value is not statistically different from the value obtained when using the contrast mediation approach. For complex scenarios, there will be a trade-off between VOIDNESS and social welfare.

VI. Conclusion

We argue that there exist situations where an unanimous agreement is not possible or simply the rules imposed by the system may not seek such unanimous agreement. Thus, we developed a consensus policy based mediation framework (CPMF) to perform multiparty negotiations. The mediation mechanisms proposed to perform the exploration of the negotiation space are derived from the GPS non-linear optimization technique. The exploration performed in the mediator is guided by the aggregation of the agents' preferences on the set of alternatives the mediator proposes in each negotiation round. The mediation rules at the mediator may take the form of a linguistic description of the type of agreements needed. We showed empirically that CPMF efficiently manages negotiations following predefined consensus policies.

We believe that the negotiation framework presented opens the door to a new set of negotiation algorithms where consensus criteria may play an important role. However, the strategical issue remains opened. We have assumed that agents reveal their true valuations. It is expected that the performance of the protocol deviates from the optimal if agents act strategically. Thus, the strategic issue needs to be evaluated, and mechanisms need to be implemented to avoid or mitigate the incentive compatibility problem.

Acknowledgements

This work has been supported by the Spanish Ministry of Education and Science grant TIN2008-06739-C04-04, T2C2 research project.

References

[1] M. Klein, P. Faratin, H. Sayama, and Y. Bar-Yam, "Protocols for negotiating complex contracts," *IEEE Intelligent Systems*, vol. 18(6), pp. 32–38, 2003.

[2] U. Endriss, N. Maudet, F. Sadri, and F. Toni, "Negotiating socially optimal allocations of resources," *Journal of Artificial Intelligence Research*, vol. 25, pp. 315–348, 2006.

[3] G. Lai and K. Sycara, "A generic framework for automated multi-attribute negotiation," *Group Decision and Negotiation*, vol. 18, pp. 169–187, 2009.

[4] M. Li, Q. B. Vo, and R. Kowalczyk, "Searching for fair joint gains in agent-based negotiation," in *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)* (Decker, Sichman, Sierra, and Castelfranchi, eds.), (Budapest, Hungary), pp. 1049–1056, May, 10-15 2009.

[5] M. A. Lopez-Carmona, I. Marsa-Maestre, M. Klein, and T. Ito, "Addressing stability issues in mediated complex contract negotiations for constraint-based, non-monotonic utility spaces," *Journal of Autonomous Agents and Multiagent Systems*, vol. Published online, pp. 1 – 51, 2010.

[6] R. M. Lewis, V. Torczon, and M. W. Trosset, "Direct search methods: then and now," *Journal of Computational and Applied Mathematics*, vol. 124, pp. 191–207, 2000.

[7] R. Yager and J. Kacprzyk, *The Ordered Weighted Averaging Operators: Theory and Applications*. Kluwer, 1997.

[8] R. Yager, "Quantifier guided aggregation using owa operators," *International Journal of Intelligent Systems*, vol. 11, pp. 49 – 73, 1996.

[9] L. Zadeh, "A computational approach to fuzzy quantifiers in natural languages," *Computing and Mathematics with Applications*, vol. 9, pp. 149 – 184, 1983.

Sesión 3.A
Modelado y análisis de prestaciones (I)

Cross-layer optimization of AMC/ARQ-based wireless networks with channel-aware multiuser scheduling protocols

Loren Carrasco , Guillem Femenias , Jaume Ramis .
Mobile Communication Group, Universitat de les Illes Balears, Ctra. de Valldemossa Km. 7,5.
loren.carrasco@uib.es, guillem.femenias@uib.es,jaume.ramis@uib.es.

***Abstract*—This paper proposes a novel framework for the cross-layer design and optimization of wireless networks combining adaptive modulation and coding (AMC) at the physical layer with automatic repeat request (ARQ) and channel-aware multiuser scheduling protocols at the data-link control (DLC) layer. The proposed framework is based on the use of first-order two-dimensional discrete time Markov chains (DTMC) jointly modeling the AMC scheme and the amplitude and rate-of-change of the wireless channel fading envelope. The behavior of the scheduleris embedded into the multidimensional physical layer Markov model through the use of a service-vacation process. Using this PHY-MAC Markov model the QoS performance at the DLC layer is discussed considering two different approaches. The first one relies on an analytical framework that is based on the multidimensional DTMC jointly describing the statistical behavior of the arrival process, the queuing system and the physical layer. The second one is rooted in the use of the effective bandwidth theory to model the packet arrival process and the effective capacity theory to model the PHY/MAC behavior. Both the DTMC-based and the effective bandwidth/capacity-based approaches are analyzed and compared in a cross-layer design aiming at maximizing the average throughput of the system where constraints on the maximum tolerable average packet loss and delay are to be fulfilled.**

***Index Terms*—Adaptive modulation and coding, automatic repeat request, channel aware scheduling, Markov models, effective capacity, cross-layer design.**

I. Introduction

Scheduling and automatic repeat request (ARQ) error control protocols at the data-link control (DLC) layer and adaptive modulation and coding (AMC) strategies at the physical (PHY) layer, are some of the key technologies underpinning state-of-the art and next-generation wireless communication systems. They are used to optimize the resource utilization while providing support to a wide range of multimedia applications with heterogeneous quality of service (QoS) requirements. However, owing to the strong dependencies between DLC and PHY layers in wireless networks, efficiency in system performance cannot be warranted using a strictly layered optimization approach. Consequently, cross-layer designs able to jointly optimize the scheduling, ARQ and AMC functions should be devised.

Although many recent works focus on cross-layer designs that combine AMC schemes with ARQ error control protocols (see, e.g., [1]–[12]), proposals also incorporating the multiuser scheduling process at the MAC sublayer are much less common (see, e.g., [13], [14]). Poggioni et al. in [13] develop a theoretical framework based on a finite state Markov chain modeling a heterogeneous multiuser scenario where groups of users with different QoS requirements coexist. In this analysis, it is assumed that the Markov chain steady state probabilities of any user can be considered independent from the steady state probabilities of all the other users in the system. Futhermore, it is assumed that the steady state probabilities of different users belonging to the same QoS class are identical. These assumptions restrict the possible application scenarios of this approach as they imply that the traffic and channel characteristics are exactly the same for all users belonging to the same QoS class. The first-order amplitude-based finite state Markov chain (AFSMC) model developed by Le at al. in [5], including both the AMC and ARQ procedures, was extended by the same authors in [14] to incorporate the multiuser scheduling process. The max-rate multiuser scheduler was included in the model through a service-vacation process allowing a manageable number of system states irrespective of the number of users sharing the channel. Nevertheless, the analysis in [14] suffers from an inaccurate modelling of the flat fading wireless channel caused by the use of a first-order AFSMC (see [9]–[12] for an in-depth discussion). Moreover the approach in [14] does not define a cross-layering scheme as a means to optimize the system performance and, on top of this, users are assumed to operate in channels with equal characteristics, thus restricting the usefulness of the presented results.

In this paper, capitalizing on the approach described in [14], a service-vacation process is used to embed the channel-aware scheduling protocol behavior into the AMC/ARQ multidimensional discrete time Markov chain model described in some of our previous contributions [9]–[12]. Our approach is based on a first-order two-dimensional Markov model for the wireless flat-fading channel that, as was shown in [9]–[12], solves most of the drawbacks of the AFSMC model used in [1]–[8], [13], [14]. In addition to the max-rate scheduling algorithm discussed in [14], our approach can be extended to the analysis of more sophisticated scheduling algorithms, including the proportional fairness multiuser scheduler. Moreover, it is not constrained by assumptions on the users' traffic and/or channel characteristics. Furthermore, as in [12] two of the principal approaches used in the technical literature to model the DLC layer behavior, namely, the DTMC model [4], [5] and the effective capacity and effective bandwidth theories [15], are compared in this paper. Both schemes are used to jointly characterize the effects of the multiuser scheduler, the ARQ error control protocol and the AMC strategies. Finally, another contribution of this paper is the proposal of a cross layer

ISBN: 978-84-694-5948-5

optimization design that, by tuning selected system parameters such as the average target packet error rate and/or the average packet arrival rate, is able to coordinate the behavior of AMC, ARQ and scheduling procedures. The main objective is to optimize the global system performance in terms of average throughput, delay, queue length and packet loss ratio.

The organization of this paper is as follows. The system model is introduced in Section II including subsections describing the AMC scheme, the physical layer first-order two-dimensional Markov model and the joint MAC-PHY Markov model. Sections III and IV are used to describe the max-rate and the proportional fair schedulers, respectively. Section V is devoted to discuss the different approaches that have been used to analyze the interactions between PHY and DLC layers, namely, the embedded DTMC approach and the effective bandwidth/capacity theory-based approach. The PHY-MAC cross-layer designs for max-rate and proportional fair schedulers are described in Section VI. In Section VII, analytical and Monte-Carlo simulation results are used to validate our model and to establish a fair comparison between DTMC-based and effective bandwidth/capacity-based cross-layer approaches. Finally, the paper concludes in Section VIII with a summary of the main results and contributions.

II. System model and assumptions

The downlink scenario of a wireless system with a base station (BS) serving N_s users is considered. At the BS there are N_s separate radio link level buffers that are used to queue the packet arrivals corresponding to every user connected to the BS. The scheduler, based on channel state information (CSI) collected from the N_s users and using a time division multiplexing (TDM) scheme, takes scheduling decisions to allocate transmission opportunities to active users. Adaptive transmission is performed by using an ARQ error control scheme at the DLC layer and an AMC strategy at the PHY layer. The processing unit at the DLC layer is a packet and the processing unit at the PHY layer is a frame. The link is assumed to support QoS-guaranteed traffic characterized by a maximum average packet delay $D_{l\max}$ and a target link layer packet loss rate $P_{l\max}$. The radio link level buffers of the different users, concerning the acknowledged packets, operate in a first-in-first-out (FIFO) fashion and can store up to $\overline{\boldsymbol{Q}} = \{\overline{Q}^1, \ldots, \overline{Q}^{N_s}\}$ packets, where $\overline{Q}^u$ is the queue length of user u. A block diagram of the system under consideration is shown in Fig. 1.

The AMC scheme is assumed to have a set $\boldsymbol{\mathcal{M}}_p = \{0, \ldots, M_p - 1\}$ of M_p *possible* transmission modes, each of which corresponding to a particular combination of modulation and coding strategies, including the case in which the transmitter does not transmit. It is assumed that when the system uses transmission mode $n \in \boldsymbol{\mathcal{M}}_p$, it transmits $p_n = bR_n$ packets per frame, where R_n denotes the number of information bits per symbol used by TM n and b is a parameter that determines the number of transmitted packets per frame, which is up to the designer's choice. For convenience, we consider that $p_0 < \cdots < p_{M_p-1}$, with $p_0 = 0$ (i.e., transmission mode 0 corresponds to the absence of transmission) and $p_{M_p-1} \triangleq \mathcal{C}_p$. As it was shown in [9], depending on the channel conditions and the QoS requirements of the different users, some of these M_p *possible* TMs may be deemed *useless* and thus, only a set $\boldsymbol{\mathcal{M}} = \{0, \ldots, M^u - 1\}$ of M^u *useful* TMs will be available to the AMC scheme for user u. It will be assumed that when the user u uses *useful* transmission mode $n \in \boldsymbol{\mathcal{M}}^{\boldsymbol{u}}$, the system transmits c_n packets and, for convenience, we also consider that $c_0 < \cdots < c_{M^u-1}$, with $c_0 \geq 0$ and $c_{M^u-1} = \mathcal{C}^u \leq \mathcal{C}_p$.

A Rayleigh block-fading channel model [16] is adopted for the propagation channel, according to which the channel is assumed to remain invariant over a time frame interval T_f and is allowed to vary across successive frame intervals[1]. Perfect channel state information (CSI) is assumed to be available at the receiver side and, thus, an ideal frame-by-frame TM selection process is performed at the AMC controller of the receiver. Furthermore, an error-free and instantaneous ARQ feedback channel is assumed. As in [5], [9]–[12] we assume that the packet generation adheres to a discrete batch Markovian arrival process (DBMAP). We refer to [12] for the description of the transition probability matrices $\{\boldsymbol{U}^1, \ldots, \boldsymbol{U}^{N_s}\}$ and average arrival rates $\boldsymbol{\lambda} = \{\lambda^1, \ldots, \lambda^{N_s}\}$ describing these arrival processes.

A. Adaptive modulation and coding (AMC)

Let γ_ν^u denote the instantaneous received SNR of user u at time instant $t = \nu T_f$. For the assumed Rayleigh block-fading channel model, γ_ν^u is an exponentially distributed random variable with mean $\overline{\gamma}^u = E\{\gamma_\nu^u\}$. Given γ_ν^u, the objective of AMC is to select the TM that maximizes the data rate while maintaining an average PER less or equal than a prescribed value P_0^u. To this end, and according to [3], the entire SNR range is partitioned into a set of nonoverlapping intervals defined by the partition $\boldsymbol{\Gamma}^{u,m} = \left\{0, \gamma_1^{u,m}, \gamma_2^{u,m}, \ldots, \gamma_{M^u-1}^{u,m}, \infty\right\}$ and mode n will be selected when $\gamma_\nu^u \in \left[\gamma_n^{u,m}, \gamma_{n+1}^{u,m}\right]$. In this paper, the partition $\boldsymbol{\Gamma}^{u,m}$ is obtained by using the threshold searching algorithm described in [12]. This searching algorithm has the capability to discriminate between *useful* and *useless* transmission modes, while guarantying that the average PER fulfils the prescribed constraint. We also assume, without loss of generality, that convolutionally coded M-QAM, adopted from IEEE 802.11a standard [17], are used in the AMC pool. All possible TMs are listed in [8, Table I].

B. Two-dimensional Markov channel modeling

Let us define $\delta_\nu^u = \gamma_{\nu-1}^u - \gamma_\nu^u$. Let us also partition of the ranges of γ_ν^u and δ_ν^u into sets of nonoverlapping two-dimensional cells defined by the partitions $\boldsymbol{\Gamma}^{u,c} = \left\{0, \gamma_1^{u,c}, \gamma_2^{u,c}, \ldots, \gamma_{K-1}^{u,c}, \infty\right\}$ and $\boldsymbol{\Delta} = \{-\infty, 0, \infty\}$, respectively. A first-order two-dimensional Markov channel model can now be defined where each state of the channel corresponds to one of such cells. That is, the Markov chain state of the channel at time instant $t = \nu T_f$ can be denoted as $\boldsymbol{\zeta}_\nu^u = (\chi_\nu^u, \Delta_\nu^u)$, $\nu = 0, 1, \ldots, \infty$, where $\chi_\nu^u = k$ if and only if $\gamma_k^{u,c} < \gamma_\nu^u \leq \gamma_{k+1}^{u,c}$ and $\Delta_\nu^u = 0$ (or $\Delta_\nu^u = 1$) if and only if $\delta_\nu^u < 0$ (or $\delta_\nu^u \geq 0$).

In our approach the partition $\boldsymbol{\Gamma}^{u,c}$ is designed assuming that the observable dummy output of our improved first-order two-dimensional Markov model at time instant $t = \nu T_f$

[1] It is assumed in this paper that the frame duration is smaller than the coherence time of the channel.

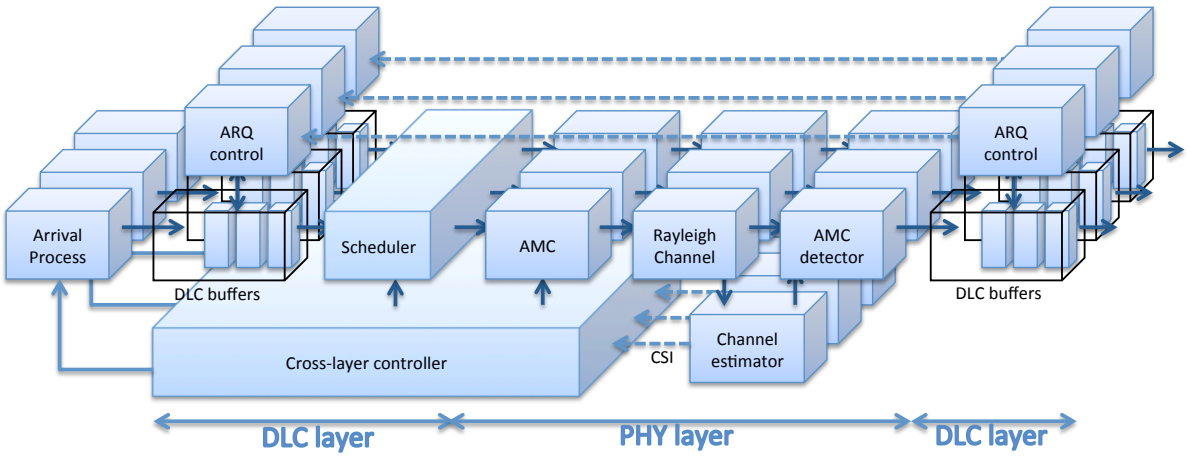


Fig. 1. System model

belongs to a codebook of nominal values of SNR $\mathbf{\Psi}^{u,c} = \{\Psi_1^{u,c}, \Psi_2^{u,c}, \ldots, \Psi_K^{u,c}\}$. The Max-Lloyd algorithm [18], [19], developed for the optimum design of non-uniform quantizers, is then used to determine the partition and (dummy) codebook minimizing the mean square error between γ_ν^u and the (dummy) quantizer output.

C. Physical layer 2-D Markov model

Based on the TM selection process used by the AMC scheme (which is defined by the partition $\mathbf{\Gamma}^{u,m}$) and the first-order 2-D Markov channel model (which is characterized by the partitions $\mathbf{\Gamma}^{u,c}$ and $\mathbf{\Delta}$), the range of γ_ν^u is partitioned into the set of nonoverlapping intervals defined by $\mathbf{\Gamma}^{u,m,c}=\left\{[\gamma_0^{u,m,c},\gamma_1^{u,m,c})\ldots\left[\gamma_{N_{\text{PHY}}^u\text{-}1}^{u,m,c},\gamma_{N_{\text{PHY}}^u}^{u,m,c}\right)\right\}$ where $\left\{\gamma_1^{u,m,c}\ldots\gamma_{N_{\text{PHY}}^u\text{-}1}^{u,m,c}\right\}=\text{sort}\left(\{\gamma_1^{u,m}\ldots\gamma_{M^u\text{-}1}^{u,m}\}\cup\{\gamma_1^{u,c}\ldots\gamma_{K\text{-}1}^{u,c}\}\right)$, N_{PHY}^u denotes the number of user u PHY states, $\gamma_0^{u,m,c}=0$, and $\gamma_{N_{\text{PHY}}^u}^{u,m,c}=\infty$. Each partition interval $\left[\gamma_k^{u,m,c},\gamma_{k+1}^{u,m,c}\right)$ is characterized by a particular combination of TM and channel state. As in Subsection II-B, let us also consider the partition of δ_ν^u into the set of nonoverlapping intervals $\mathbf{\Delta}=\{-\infty,0,\infty\}$. Using this two-dimensional partitioning, a first-order two-dimensional Markov model for the physical layer of user u can be defined where each state corresponds to one of such two-dimensional rectangular-shaped cells. Furthermore, the physical layer Markov chain state at time instant $t = \nu T_f$ is denoted by $\boldsymbol{\varsigma}_\nu^u = (\varphi_\nu^u, \Delta_\nu^u)$, $\nu = 0, 1, \ldots, \infty$, where $\varphi_\nu^u \in \{0, \ldots, N_{\text{PHY}}^u - 1\}$ represents the combination of TM and channel state in this frame interval and $\Delta_\nu^u \in \{0, 1\}$ is used to denote the *up* or *down*[2] characteristic of the instantaneous SNR over the time frame interval $t = (\nu - 1)T_f$. At any time instant $t = \nu T_f$ the physical layer state can be univocally identified by an integer number $y_\nu^u = 2\varphi_\nu^u + \Delta_\nu^u$ with, $y_\nu^u \in \{0, \ldots, 2N_{\text{PHY}}^u - 1\}$, which can be characterized by an steady-state probability $P_{\text{PHY}}(y_\nu^u)$ and a corresponding conditional average packet error rate $\overline{PER}_{\text{PHY}}(y_\nu^u)$. Additionally, the physical layer FSMC will be characterized by a transition probability matrix $\boldsymbol{H}_s^u = \left[H_{i,j}^u\right]_{0\le i,j\le 2N_{\text{PHY}}^u-1}$ where $H_{i,j}^u = Pr\{y_{\nu+1}^u = j | y_\nu^u = i\}$.

[2]If $\gamma_\nu^u < \gamma_{\nu-1}^u$ then the instantaneous SNR is descending and it can be tagged as *down*; on the contrary, if $\gamma_\nu^u \ge \gamma_{\nu-1}^u$ then the instantaneous SNR is ascending and it can be tagged as *up*.

D. Joint PHY-MAC layer Markov model

Channel-aware only schedulers can be incorporated to the joint PHY-MAC Markov model by means of a service-vacation process [14]. When a particular user u is selected for transmission in a given time slot it is said that this user physical layer is in service, otherwise it is said to be on vacation. The parameter $z^u \in \{0, 1\}$ is used to denote the service $(z^u = 0)$ or vacation $(z^u = 1)$ state. The decision wether a user u will be in service or vacation during the next time slot will depend on the possible physical layer states of all users in the next time slot and on previous scheduling decisions. A D-step memory in the service-vacation process represents the scheduling dependence on D previous decisions and accounts for an increased degree of fairness between users. The joint MAC-physical layer Markov chain state for user u at time instant $t = \nu T_f$ is denoted by $\boldsymbol{\iota}_\nu^u = (z_\nu^u, z_{\nu-1}^u, \ldots, z_{\nu-D+1}^u y_\nu^u)$. At any time instant $t = \nu T_f$ the joint PHY-MAC layer state can be univocally identified by an integer number n_ν^u with $n_\nu^u \in \{0, \ldots, 2^{D+1}N_{\text{PHY}}^u - 1\}$. The joint MAC-PHY layer will be in state $n_\nu^u \in \{0, \ldots, 2^{D+1}N_{\text{PHY}}^u - 1\}$ with a steady-state probability $P_{\text{PHY-MAC}}^u(n_\nu^u)$. Taking into account that the user u transmits only when it is in service, the different PHY-MAC states will have a transmission rate, measured in packets per slot, of $\widehat{c}_{n_\nu^u} = c_{y_\nu^u}(1 - z_\nu^u)$, where $c_{y_\nu^u}$ is the transmission rate characterizing PHY layer state y_ν^u. Furthermore, the PHY-MAC layer FSMC will be described by a transition probability matrix $\boldsymbol{P}_s^u = \left[P_{i,j}^u\right]_{0\le i,j\le 2^{D+1}N_{\text{PHY}}^u-1}$, with state-transition probabilities

$$P_{i,j}^u = Pr\{n_{\nu+1}^u = j | n_\nu^u = i\} \quad (1)$$

that can be analytically calculated for a significant number of scheduling schemes. In the following sections these probabilities are derived for the max-rate and proportional fair algorithms, which, in both cases, can be modeled by a service-vacation process with one-step memory ($D = 1$).

III. The Max-Rate Scheduling Example

In the max-rate scheduling, the physical layer states of all active users are assumed to be available at the sched-

uler without delay. It is further assumed that the physical layer processes for all users are independent. The max-rate scheduler grants the transmission opportunity to the user that can achieve the highest transmission rate. If more than one user can attain this maximum rate, the scheduler chooses one of them randomly. Although this case was covered in [14], several modifications are included in our analysis in order to adapt it to the two-dimensional physical layer channel model and, also, to generalize its application to more realistic scenarios with users experiencing heterogeneous average SNRs.

In one-step memory service-vacation processes, scheduling decisions only rely on the actual system state and thus, the state transition probabilities in (1) can be simplified to $P^u_{i,j} = Pr\{z^u_{\nu+1}, y^u_{\nu+1} | z^u_\nu, y^u_\nu\}$. The transition probability matrix can be expressed as

$$\boldsymbol{P}^u_s = \begin{pmatrix} \boldsymbol{S}^u_{0,0} & \boldsymbol{S}^u_{0,1} \\ \boldsymbol{S}^u_{1,0} & \boldsymbol{S}^u_{1,1} \end{pmatrix}$$

where $\boldsymbol{S}^u_{i,j}$ is a $(2N^u_{\text{PHY}}) \times (2N^u_{\text{PHY}})$ matrix with elements $S^u_{i,j}(k,l) = Pr\{z^u_{\nu+1}{=}j, y^u_{\nu+1}{=}l | z^u_\nu{=}i, y^u_\nu{=}k\}$. Without loss of generality, user $u = 1$ is considered to be the user of interest and, for notation simplicity, it is assumed that $z^1_\nu = z_\nu$, $y^1_\nu = y_\nu$. Taking into account that the physical layer and service-vacation processes are independent, the elements of the $\boldsymbol{S}^1_{i,j}$ matrices can be written as

$$Pr\{z_{\nu+1}, y_{\nu+1} | z_\nu, y_\nu\} = Pr\{z_{\nu+1} | z_\nu, y_{\nu+1}, y_\nu\} Pr\{y_{\nu+1} | y_\nu\},$$

the latter term being an element of the physical layer state transition probability matrix $\boldsymbol{H}^1_s$. Moreover, since $z_{\nu+1} \in \{0, 1\}$ it holds that

$$\begin{aligned} &Pr\{z_{\nu+1}{=}1 | z_\nu{=}i, y_{\nu+1}{=}l, y_\nu{=}k\} \\ &\qquad = 1 - Pr\{z_{\nu+1}{=}0 | z_\nu{=}i, y_{\nu+1}{=}l, y_\nu{=}k\} \end{aligned}$$

and therefore only the case $z_{\nu+1} = 0$ needs to be discussed. Considering now that the service state at time ν depends only on the physical layer state at time ν it holds that

$$\begin{aligned} &Pr\{z_{\nu+1}{=}0 | z_\nu{=}i, y_{\nu+1}{=}l, y_\nu{=}k\} \\ &\qquad = \frac{Pr\{z_{\nu+1}{=}0, z_\nu{=}i | y_{\nu+1}{=}l, y_\nu{=}k\}}{Pr\{z_\nu{=}i | y_\nu{=}k\}}. \end{aligned} \tag{2}$$

A. Calculation of $Pr\{z_\nu{=}i | y_\nu{=}k\}$

Assuming $z_\nu = 0$, the denominator of (2) can be calculated as

$$\begin{aligned} &Pr\{z_\nu{=}0 | y_\nu{=}k\} \\ &\quad = \sum_{y^2_\nu=0}^{\widehat{N}^2_{\text{PHY}}} \sum_{y^3_\nu=0}^{\widehat{N}^3_{\text{PHY}}} \cdots \sum_{y^{N_s}_\nu=0}^{\widehat{N}^{N_s}_{\text{PHY}}} Pr\{z_\nu{=}0, y^2_\nu, y^3_\nu, \ldots, y^{N_s}_\nu | y_\nu{=}k\}, \end{aligned}$$

where $\widehat{N}^u_{\text{PHY}} \triangleq 2N^u_{\text{PHY}} - 1$. At time slot ν, user 1 (the user of interest), whose physical layer is in state k, can only be in service if the rest of users have a physical layer state with a lower or equal transmission rate. When a users (including user 1) can transmit at maximum transmission rate, then user 1 is chosen for transmission with a probability $1/a$. Thus,

$$\begin{aligned} &Pr\{z_\nu{=}0, y^2_\nu, y^3_\nu, \ldots, y^{N_s}_\nu | y_\nu{=}k\} \\ &\qquad = \begin{cases} 0, \text{ if } \exists u \in \{2, \ldots, N_s\} : c_{y^u_\nu} > c_k \\ \frac{1}{a} \prod_{u=2}^{N_s} P^u_{\text{PHY}}(y^u_\nu), \text{ otherwise} \end{cases} \end{aligned} \tag{3}$$

where c_k is the transmission rate corresponding to $y_\nu{=}k$. The case with $z_\nu = 1$ simply gives $Pr\{z_\nu{=}1 | y_\nu{=}k\} = 1 - Pr\{z_\nu{=}0 | y_\nu{=}k\}$.

B. Calculation of $Pr\{z_{\nu+1}{=}0, z_\nu{=}i | y_{\nu+1}{=}l, y_\nu{=}k\}$

The numerator of (2) can be written as shown in (4) (at the bottom of next page). In order to obtain the terms inside the summations of this expression, two different cases should be considered:

1) Case 1 ($z_{\nu+1} = 0$, $z_\nu = 0$):: In this case user 1 is in service during time slots ν and ν+1, given that its PHY state in these time slots is k and l, respectively. This will only happen when the potential transmission rates of the other $N_s - 1$ users are smaller or equal than the transmission rates of user 1 in PHY states k and l during time slots ν and ν+1, respectively. If a and b users (including user 1) can transmit at maximum transmission rate during the ν and ν+1 time slots, respectively, then user 1 will be granted transmission for both time slots with probability $1/(ab)$. Therefore, in this case the probabilities in (4) can be calculated as in (5) (shown at the bottom of next page), where $Pr\{y_\nu{=}k, y_{\nu+1}{=}l\} = H^1_{k,l} P^1_{\text{PHY}}(k)$.

2) Case 2 ($z_{\nu+1} = 0$, $z_\nu = 1$):: In this case user 1 goes from the vacation state during time slot ν to the service state at ν+1. The service state in time slot ν+1 can occur if b users (including user 1) can transmit at maximum transmission rate and user 1 is selected for transmission with probability $1/b$. A vacation state during time slot ν can happen as a result of two different situations, either there are users with higher transmission rates than user 1, or a users (including user 1) can transmit with the maximum transmission rate and user 1 is not selected with probability $(1 - \frac{1}{a})$. Then, in case 2, the probabilities in (4) can be obtained using (6) (shown at the bottom of next page).

IV. The proportional fair scheduling example

Originally proposed in the wired network scheduling context, a Proportional Fair (PF) scheduler promises a trade off between the maximization of average throughput and system fairness. At each time instant, the user experiencing the highest instantaneous rate with respect to its average rate is scheduled. That is, user q is selected for transmission during time slot ν if

$$q = \arg \max_{u \in \{1, \ldots, N_s\}} \frac{c_{y^u_\nu}}{T^u_\nu}, \tag{7}$$

where T^u_ν is the average rate of user u. It can be proved [20] that the scheduler defined in (7) maximizes the logarithmic sum of system throughput. The average rate can be computed as a moving average over a time window of length W, that is,

$$T^u_{\nu+1} = \left(1 - \frac{1}{W}\right) T^u_\nu + (1 - z^u_\nu) \frac{1}{W} c_{y^u_\nu}.$$

We define $\hat{\mathcal{T}}^u \triangleq \lim_{W \to \infty} T^u$ as the stationary throughput of user u and $C^u = \sum_{y=0}^{2N^u_{\text{PHY}}-1} c_y (1 - \overline{PER}_{\text{PHY}}(y)) P_{\text{PHY}}(y)$ as the user u channel average rate. Using the results of Holtzman in [21] and assuming that the fast fading component of all users in the system is identically distributed it can be demonstrated that,

$$\frac{\hat{\mathcal{T}}^u}{\hat{\mathcal{T}}^v} = \frac{\hat{\mathcal{C}}^u}{\hat{\mathcal{C}}^v}$$

then the proportional fair weight of user u in time slot ν can be defined as $F_\nu^u = \frac{c_{y_\nu^u}}{\overline{C}^u}$. The transition probability matrix can be construed as in the max-rate example. Expressions (3), (5) and (6) can be rewritten by substituting the transmission rates $c_{y_\nu^u}$ with the corresponding proportional fair weights F_ν^u. Now a and b will denote the number of users with the maximum PF weights at time slots ν and ν+1, respectively.

V. Queueing model and analysis

Once the PHY layer and MAC sublayer have been properly modeled, the queuing behavior of the DLC layer has to be introduced in the analysis. Two different techniques are proposed in this paper:

A. Queuing Markov model-based approach

Following the work described in [9]–[12], the queueing process induced by both the ARQ protocol and the AMC scheme can be formulated in discrete time with one time unit equal to one frame interval. Each user's subsystem states are observed at the beginning of each time unit. Let $\sigma_\nu^u = (q_\nu^u, a_\nu^u, \boldsymbol{\iota}_\nu^u)$ denote the user u subsystem state at time instant $t = \nu T_f$, where $q_\nu^u \in \{0, \ldots, \overline{Q}^u\}$ denotes the queue length at this time instant, $a_\nu^u \in \{0, \ldots, \mathcal{A}^u - 1\}$ represents the phase of the DBMAP and $\boldsymbol{\iota}_\nu^u \in \{0, \ldots, 2^{D+1} N_{\text{PHY}}^u - 1\}$ represents the combination of PHY layer state and scheduling decision for user u during this frame interval. Focusing on the set of time instants $t = \nu T_f$, $\nu = 0, 1, \ldots, \infty$, the transitions between states σ_ν^u are Markovian. Therefore, an embedded Markov chain can be used to describe the underlying queueing process for each user u and expressions for the packet loss rate P_l^u, average throughput η^u, average queue length L_q^u and average packet delay D_p^u can be derived.

In previous work we have developed the embedded Markov chains describing the underlying queueing process for a variety of AMC schemes, such as the ones described in 802.11 and 802.16 proposals, and ARQ protocols comprising infinitely persistent [9], [12], truncated [10] and hybrid ARQ [11] schemes obtaining for all of them expressions for P_l^u, η^u, D_p^u and L_q^u. All these results can be directly combined with the PHY-MAC Markov model to obtain a global multiuser system model. In this paper, without loss of generality, we have used the model developed in [12] for an IEEE 802.11a AMC scheme and an infinitely persistent SR-ARQ procedure.

B. Effective bandwidth/capacity-based approach

The DLC layer can also be modelled by applying the effective bandwidth/capacity-based approach [15]. The effective capacity and effective bandwidth allow the analysis of the so-called packet loss rate bound probability. The analysis is analogous to the one developed in [12]. The effective bandwidth of the DBMAP arrival process of user u, characterized by a transition matrix $\boldsymbol{U^u}$, can be calculated as [22], $E_B^u(\psi) = \psi^{-1} \log(\Upsilon_U^u(\psi))$, where Υ_U^u is the Perron-Frobenius eigenvalue of the matrix $\boldsymbol{U}_{\boldsymbol{\varpi}}^u = \mathcal{D}(\boldsymbol{\varpi}^u)\boldsymbol{U}^u$, with $\boldsymbol{\varpi} \triangleq \left(e^{\lambda_0^u \psi}, \ldots, e^{\lambda_{\mathcal{A}^u-1}^u \psi}\right)$, where λ_n^u denotes the number of packets per frame generated when the source of user u is in state n and $\mathcal{D}(x)$ denotes a diagonal matrix with the elements of x on its main diagonal. The effective capacity of the service process that models the behavior of the MAC and PHY layers for the user of interest, which is characterized by a transition probability matrix $\boldsymbol{P}_s^u$, can be obtained as $E_C^u(\psi) = -\psi^{-1} \log(\Upsilon_P^u(-\psi))$. In this expression, Υ_P^u denotes the Perron-Frobenius eigenvalue of the matrix $\boldsymbol{P}_{\upsilon_\nu}^u = \mathcal{D}(\boldsymbol{\upsilon}_\nu^u)\boldsymbol{P}_s^u$, with $\boldsymbol{\upsilon}_\nu^u \triangleq \left(e^{-\check{c}_0^u \psi}, \ldots, e^{-\check{c}_{(4N_{\text{PHY}}^u-1)}^u \psi}\right)$, where $\check{c}_n^u$ denotes the number of packets per frame leaving the queue when the PHY-MAC for user u is in state n, which for an SR-ARQ infinitely persistent scheme can be calculated as $\check{c}_\nu^u = \sum_{k=0}^{\widehat{c}_\nu^u} k \; p_{k,\widehat{c}_\nu^u}^{(\nu)}$ with $p_{k,\widehat{c}_\nu^u}^{(\nu)}$ defined in [12, (16)].

VI. Cross layer design

Given a maximum allowed queue length $\overline{\boldsymbol{Q}} = \{\overline{Q}^1, \ldots, \overline{Q}^{N_s}\}$, average SNRs $\bar{\boldsymbol{\gamma}} = \{\bar{\gamma}^1, \ldots, \bar{\gamma}^{N_s}\}$, a normalized maximum Doppler frequencies $\boldsymbol{f_d} T_f$ with $\boldsymbol{f_d} = \{f_d^1, \ldots, f_d^{N_s}\}$, and the use of the max-rate algorithm in the MAC layer, the derived PHY-MAC-DLC layer model basically depends on the prescribed average PER of each user $\boldsymbol{P_0} = \{P_0^1, \ldots, P_0^{N_s}\}$ where P_0^u is a real number in the range $\boldsymbol{\Phi} = [0, 1]$ and the measured or estimated arrival packet rate of each user $\lambda^u \in \boldsymbol{\Theta}$ where $\boldsymbol{\Theta}$ is the range of feasible arrival rate values. Thus, if the users in the system are to support QoS-guaranteed traffic characterized by a maximum packet loss rate $P_{l\max}$ and a maximum average packet delay $D_{l\max}$, the proposed cross-layer design must aim at optimally determining the prescribed average PER vector $\boldsymbol{P_0}$ and average packet arrival rate vector $\boldsymbol{\lambda} = \{\lambda^1, \ldots, \lambda^{N_s}\}$ that maximizes the total system throughput, that is,

$$\left(\boldsymbol{P_0}^{\text{opt}}, \boldsymbol{\lambda}^{\text{opt}}\right) = \arg \max_{\boldsymbol{P_0} \in \boldsymbol{\Phi}, \boldsymbol{\lambda} \in \hat{\boldsymbol{\Theta}}} \sum_{u=1}^{N_s} \eta^u(\boldsymbol{P_0}, \boldsymbol{\lambda}) \tag{8}$$

subject to the constraints $P_l^u(\boldsymbol{P_0}, \boldsymbol{\lambda}) \leq P_{l\max} \quad \forall u$ and/or $D_l^u(\boldsymbol{P_0}, \boldsymbol{\lambda}) \leq D_{l\max} \quad \forall u$ depending on the traffic type.

A similar cross layer design can be derived for the proportional fair algorithm, but, taking into account that this

$$Pr\{z_\nu{=}z, z_{\nu+1}{=}v | y_\nu{=}k, y_{\nu+1}{=}l\} = \sum_{y_{\nu+1}^2=0}^{\widehat{N}_{\text{PHY}}^2} \cdots \sum_{y_{\nu+1}^{N_s}=0}^{\widehat{N}_{\text{PHY}}^{N_s}} \sum_{y_\nu^2=0}^{\widehat{N}_{\text{PHY}}^2} \cdots \sum_{y_\nu^{N_s}=0}^{\widehat{N}_{\text{PHY}}^{N_s}} Pr\{z_\nu{=}z, z_{\nu+1}{=}v, y_\nu^2 \ldots y_\nu^{N_s} y_{\nu+1}^2 \ldots y_{\nu+1}^{N_s} | y_\nu{=}k, y_{\nu+1}{=}l\}, \tag{4}$$

$$Pr\{z_\nu{=}0, z_{\nu+1}{=}0, y_\nu^2 \ldots y_\nu^{N_s}, y_{\nu+1}^2 \ldots y_{\nu+1}^{N_s} | y_\nu{=}k, y_{\nu+1}{=}l\} = \begin{cases} 0, \text{ if } \exists u \in \{2 \ldots N_s\} : c_{y_\nu^u} > c_k \text{ or } c_{y_{\nu+1}^u} > c_l \\ \frac{1}{ab} \prod_{u=2}^{N_s} Pr\{y_\nu^u, y_{\nu+1}^u\}, \text{ otherwise.} \end{cases} \tag{5}$$

$$Pr\{z_\nu{=}1, z_{\nu+1}{=}0, y_\nu^2 \ldots y_\nu^{N_s}, y_{\nu+1}^2 \ldots y_{\nu+1}^{N_s} | y_\nu{=}k, y_{\nu+1}{=}l\} = \begin{cases} 0, \text{ if } c_{y_\nu^u} < c_k \forall u \in \{2 \ldots N_s\} \text{ or } \exists u \in \{2 \ldots N_s\} : c_{y_{\nu+1}^u} > c_l \\ \frac{1}{b} \prod_{u=2}^{N_s} Pr\{y_\nu^u, y_{\nu+1}^u\}, \text{if } \exists u \in \{2 \ldots N_s\} : c_{y_\nu^u} > c_k \\ \frac{a-1}{ab} \prod_{u=2}^{N_s} Pr\{y_\nu^u, y_{\nu+1}^u\}, \text{ otherwise.} \end{cases} \tag{6}$$

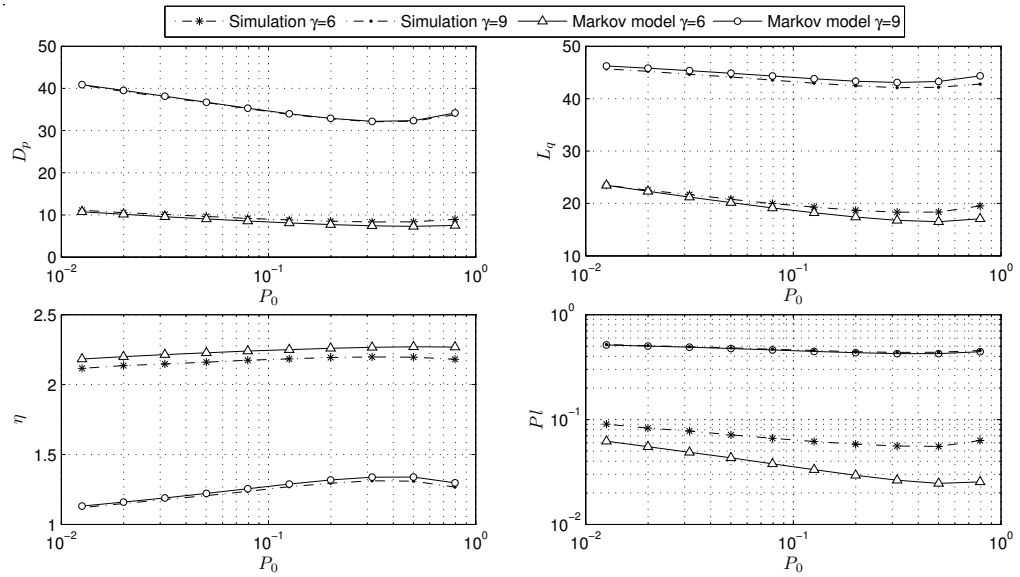


Fig. 2. Max rate algorithm. Average delay (D_p), queue length (L_q), throughput (η) and packet loss rate (P_l) vs. target PER.

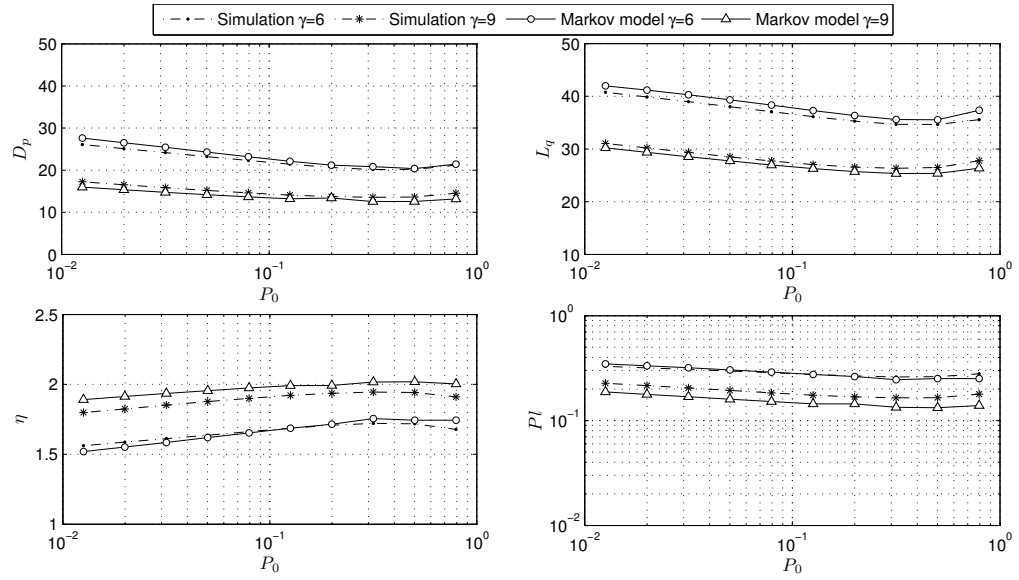


Fig. 3. Proportional Fair algorithm. Average delay (D_p), queue length (L_q), throughput (η) and packet loss rate (P_l) vs. target PER.

algorithm maximizes the logarithmic sum of capacity, the optimization function must be designed accordingly as,

$$\left(\boldsymbol{P_0}^{\text{opt}}, \boldsymbol{\lambda}^{\text{opt}}\right) = \arg \max_{\boldsymbol{P_0} \in \boldsymbol{\Phi}, \boldsymbol{\lambda} \in \hat{\boldsymbol{\Theta}}} \sum_{u=1}^{N_s} \log(\eta^u(\boldsymbol{P_0}, \boldsymbol{\lambda})) \quad (9)$$

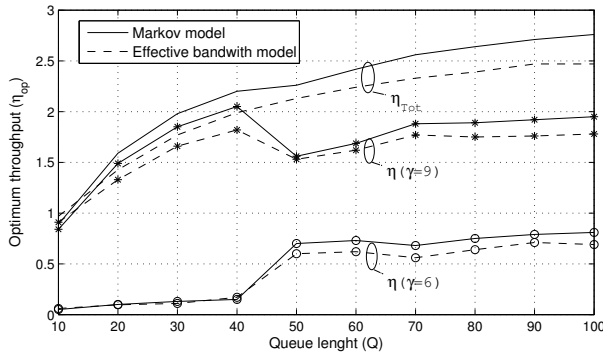
subject to the constraints $P_l^u(\boldsymbol{P_0}, \boldsymbol{\lambda}) \leq P_{l\max} \quad \forall u$ and/or $D_l^u(\boldsymbol{P_0}, \boldsymbol{\lambda}) \leq D_{l\max} \quad \forall u$ depending on the traffic type.

In both cases, the analytical expressions for η^u, P_l^u and D_l^u do not leave much room for developing efficient algorithms in solving our constrained optimization problem. However, considering that $\boldsymbol{P_0}$ and $\boldsymbol{\lambda}$ lie in a bounded space $\boldsymbol{\Phi}^u \times \boldsymbol{\Theta}^u$, a multidimensional exhaustive search can be used to numerically solve the proposed cross-layer optimization problem.

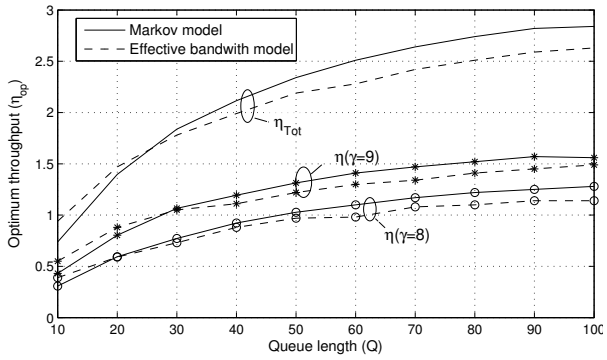
VII. Numerical results

In order to verify the validity of the proposed cross-layer framework, analytical results will be confronted with computer simulation results obtained using Clarke's statistical Rayleigh fading model of the wireless flat-fading channel [23]. Unless otherwise specified, numerical results are presented for the following default parameters: a normalized maximum Doppler frequency $f_d T_f = 0.02$, a queue length $\overline{Q} = 50$, a number of channel states $K = 10$, a parameter $b = 2$ and a DBMAP parameterized to obtain a truncated Poisson process with a variable arrival rate λ. These parameters apply to all users in the system.

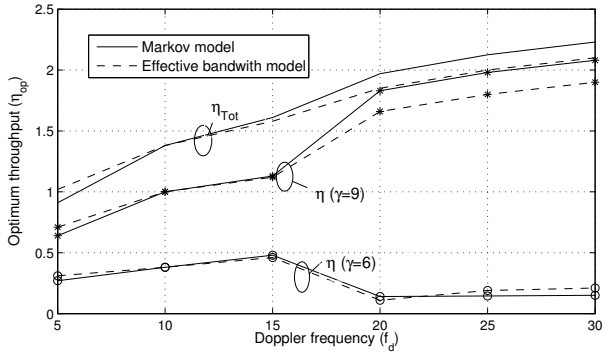
Figures 2 and 3 show the dependence of the average delay D_l^u, queue length L_q^u, throughput η^u and packet loss rate P_l^u on the target average PER P_0^u of the two users in the system. In these figures P_0^1 and P_0^2 have been set to a common value P_0 in order to show the analytical and simulation results of both users simultaneously. As it can be observed, in all cases the behavior of the simulation of the FIFO queuing system under an infinitely persistent ARQ protocol, with



(a) Average Optimum Throughput vs. Queue length.



(b) Average Optimum Throughput vs. Queue length.

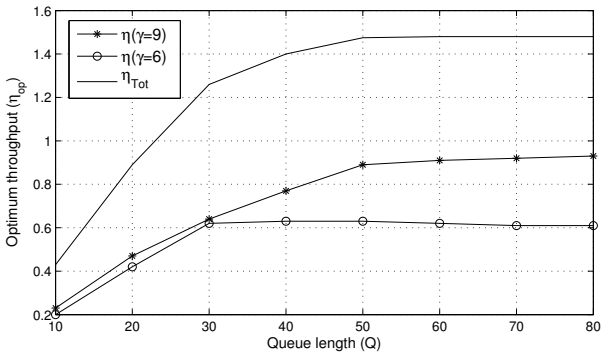


(c) Average Optimum Throughput vs. Max. Doppler Frequency.

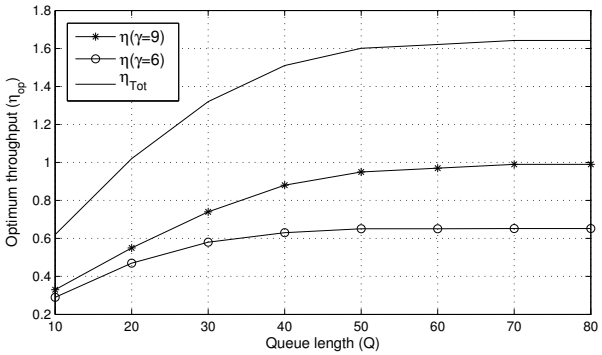
Fig. 4. Max-rate algorithm example

a PHY layer based on Clarke's model and with different scheduling algorithms in the MAC sublayer, namely the max-rate algorithm (fig. 2) and the proportional fair algorithm (fig. 3), is faithfully reproduced by our analytical PHY-MAC-DLC layer model. In particular, it is interesting to note how the shape and location of the minimum/maximum of the curves obtained by simulation (Clarke's model) coincide with those obtained using the analytical framework, even for a small number of channel states K. The accuracy in determining the location of the maximum of the throughput and the minimum of the average packet loss rate or the average packet delay is particularly important in order to ensure an optimal cross-layer design. Regarding the scheduling performance, it can be observed in figs. 2 and 3 that, as expected, PF attains higher fairness at the expense of a global throughput loss.

Figures 4(a), 4(b) and 4(c) show the total (aggregated) and per-user throughputs of the system when applying the cross layer optimization defined in (8) for the max-rate algorithm.



(a) Average Optimum Throughput vs. Queue length (Markov model-based approach)



(b) Average Optimum Throughput vs. Max. Normalized Doppler Frequency (Effective bandwidth/capacity-based approach).

Fig. 5. Proportional fair example.

These figures have been obtained applying a maximum affordable packet loss rate $P_{l_{max}} \leq 0.01$ and using either a Markov model or the effective bandwidth/capacity-based approach to model the data link buffer behavior as described in Section V. The optimization process further increases the aggregate throughput of the max-rate policy, while maintaining a desired level of QoS in the form of a maximum packet loss rate. This is accomplished by tuning the PHY layers of the users through the P_0^u parameters and shaping the users average arrival rate λ^u. Figure 4(a) reveals that, for short queue lengths ($\overline{Q} < 50$), the higher aggregated throughput is obtained by assigning a very low P_0 to the user with a lower average SNR ($\gamma = 6$ dB), which results in a very low throughput for that user. The same behavior is observed in Fig. 4(c) for high Doppler frequencies ($f_d > 20$ Hz). When the queue length increases or the Doppler frequency decreases, the system achieves higher capacity by assigning similar P_0 values to both users and the throughput of the lower SNR user increases accordingly. Logically, when the two users have similar average SNR values, as shown in Fig. 4(b), the maximum sum-throughput is always achieved by assigning similar P_0 values.

Figures 5(a) and 5(b) show the total (aggregated) and per-user throughputs in the system when applying the cross-layer optimization defined by (9) for the proportional fair algorithm. Figure 5(a) has been obtained using a maximum packet loss constraint $P_{l_{max}} \leq 0.01$ and a maximum average delay constraint $D_{l_{max}} \leq 10$. As expected, for low values of the queue length $\overline{Q} < 40 - 50$, the constraint limiting the throughput is the packet loss, that is mainly caused by the buffer overflow and therefore, the throughput increases

with $\overline{Q}$. For higher queue lengths $\overline{Q} > 40 - 50$, the limiting constraint is the maximum average delay and, in this case, additional increases in the queue length have a negligible effect over throughput. Figure 5(b) depicts results obtained when using an optimization performed using the effective bandwidth model formulated in (9). In this case the constraint in (9) has been modified to $Pr\{D_l^u(\boldsymbol{P_0}, \boldsymbol{\lambda}) \geq Dl_{\max}\} \leq \epsilon \quad \forall u$, as the effective bandwidth theory only offers an approximation of the probability of a system to exceed a certain maximum delay. The specific values of the constraint used to generate this figure are $Pr\{D_l^u \geq 50\} \leq 0.01 \quad \forall u$. Results presented in Fig. 5(b) show a similar behavior as those in Fig. 5(a). For a queue length below $\overline{Q} = 50$, the limiting constraint is the packet loss rate and, therefore, the throughput increases with $\overline{Q}$. In contrast, for $\overline{Q} > 50$ the active constraint is the maximum affordable delay, which does not depend on the queue length causing the throughput to remain nearly constant with respect to the queue length.

VIII. Conclusion

This paper extends the analytical framework presented in [12] to include the MAC sublayer in the proposed analytical model that now includes a multiple user shared channel scenario. Channel-aware only schedulers have been embedded in a joint PHY-MAC Markov model by using a service-vacation process to model the scheduling decisions. Two widely used scheduling rules have been considered, the max-rate and the proportional fair algorithms. As in [12], two different approaches have been used to model the data link level queueing behavior: an analytical Markov model and an approach based on the effective bandwidth theory. Results show that the use of the effective bandwidth approach significantly simplifies the global model and is therefore an interesting technique to use by the resource allocation algorithms. Numerical examples confirm that the derived performance metrics obtained with the PHY-MAC-DLC analytical model faithfully reproduce simulation results. It is important to point out that the multiple user model obtained in this paper can be easily adapted to include truncated or hybrid ARQ techniques in the data link layer as it was proposed in [10], [11]. Finally a cross layer design interrelating the physical, MAC and data link layers has been described. The obtained results show the potential of cross layer resource allocation designs where slot-by-slot decisions are assigned to simple and PHY efficient schedulers, such as the max-rate or proportional fair algorithms, while QoS control is performed at a higher level by well selected optimization functions. These optimization functions enhance and complement the scheduling algorithms while maintaining an adequate QoS performance by modifying average parameters of the different layers in the system. The proposed cross-layer approach fits in the radio resource management (RRM) framework proposed for state-of-the-art networks such as LTE that define a division between fast dynamic layer 1 and layer 2 RRM functions working at the transmission time interval level and semi dynamic layer 3 RRM procedures.

Acknowledgments

This work has been partially funded by MEC and FEDER through project COSMOS (TEC2008-02422).

References

[1] Q. Liu, S. Zhou, and G. B. Giannakis, "Cross-layer combining of adaptive modulation and coding with truncated ARQ over wireless links," *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1746–1755, Sept. 2004.

[2] ——, "Queuing with adaptive modulation and coding over wireless links: cross-layer analysis and design," *IEEE Trans. Wireless Commun.*, vol. 4, no. 3, pp. 1142–1153, May 2005.

[3] ——, "Cross-layer scheduling with prescribed QoS guarantees in adaptive wireless networks," *IEEE Journal on Selected Areas in Commun.*, vol. 23, no. 5, pp. 1056–1066, May 2005.

[4] L. B. Le, E. Hossain, and A. S. Alfa, "Service differentiation in multirate wireless networks with weighted round-robin scheduling and ARQ-based error control," *IEEE Trans. Commun.*, vol. 54, no. 2, pp. 208–215, Feb. 2006.

[5] ——, "Radio link level performance evaluation in wireless networks using multi-rate transmission with ARQ-based error control," *IEEE Trans. Wireless Commun.*, vol. 5, no. 10, pp. 2647–2653, Oct. 2006.

[6] F. Ishizaki and G. U. Hwang, "Cross-layer design and analysis of wireless networks using the effective bandwidth function," *IEEE Trans. Wireless Commun.*, vol. 6, no. 9, pp. 3214–3219, Sept. 2007.

[7] M. Poggioni, L. Rugini, and P. Banelli, "Analyzing performance of multi-user scheduling jointly with AMC and ARQ," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, Nov. 2007, pp. 3483–3488.

[8] X. Wang, Q. Liu, and G. B. Giannakis, "Analizing and optimizing adaptive modulation coding jointly with ARQ for QoS-guaranteed traffic," *IEEE Trans. Veh. Technol.*, vol. 56, no. 2, pp. 710–720, March 2007.

[9] J. Ramis, L. Carrasco, and G. Femenias, "A two-dimensional Markov model for cross-layer design in AMC/ARQ-based wireless networks," in *Proc IEEE GLOBECOM*, Dec. 2008, pp. 4637–4642.

[10] ——, "Diseño intercapas en redes inalámbricas basadas en AMC y ARQ truncado," in *Proc Jitel 2009*, Sep. 209, pp. 260–267.

[11] J. Ramis, G. Femenias, F. Riera-Palou, and L. Carrasco, "Cross-layer optimization of adaptive multi-rate wireless networks using truncated chase combining HARQ," in *Proc IEEE GLOBECOM*, Dec. 2010.

[12] G. Femenias, L. Carrasco, and J. Ramis, "Using two-dimensional markov models and the effective-capacity approach for cross-layer design in amc/arq-based wireless networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 8, pp. 4193–4203, Oct. 2009.

[13] M. Poggioni, L. Rugini, and P. Banelli, "QoS analysis of a scheduling policy for heterogeneous users employing AMC jointly with ARQ," *IEEE Trans. Communications*, vol. 58, p. 9, Sep. 2010.

[14] L. B. Le, E. Hossain, and A. S. Alfa, "Delay statistics and throughput performance for multi-rate wireless networks under multiuser diversity," *IEEE Trans. Wireless Commun.*, vol. 5, no. 11, pp. 3234–3243, Nov. 2006.

[15] D. Wu and R. Negi, "Effective capacity: a wireless link model for support of quality of service," *IEEE Trans. Wireless Commun.*, vol. 2, no. 4, pp. 630–643, July 2003.

[16] E. Biglieri, G. Caire, and G. Taricco, "Limiting performance of block fading channels with multiple antennas," *IEEE Trans. Inf. Theory*, vol. 47, no. 4, pp. 1273–1289, May 2001.

[17] IEEE, *802.11: Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. New York: IEEE, 1997.

[18] J. Max, "Quantization for minimum distortion," *IRE Trans. Information Theory*, vol. IT-6, pp. 7–12, March 1960.

[19] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Information Theory*, vol. IT-28, pp. 129–137, March 1982.

[20] K. K. H. Kim and Y. Han, "An efficient scheduling algorithm for QoS provision in wireless packet data transmission," in *Proc. IEEEPIMRC02*, Sept. 2002, pp. 73–77.

[21] J. Holtzman, "Asymptotic analysis of the proportional fair algorithm," in *Proc. IEEEPIMRC01*, Aug. 2001, pp. F33–F37.

[22] C. Chang, Ed., *Performance guarantees in communication networks*, 1st ed. Springer-Verlag, 2000.

[23] R. H. Clarke, "A statistical theory of mobile radio reception," *Bell System Tech. Journal*, vol. 47, no. 6, pp. 957–1000, Sept. 1968.

Performance analysis of fast link adaptation-based 802.11n basic and RTS/CTS access schemes

Gabriel Martorell, Felip Riera-Palou y Guillem Femenias
Grupo de Comunicaciones Móviles,
Universitat de les Illes Balears
{gabriel.martorell, felip.riera, guillem.femenias}@uib.es

***Abstract*—This paper presents a comprehensive performance study of closed-loop fast link adaptation (FLA) in the context of IEEE 802.11n, spanning the physical (PHY) and medium-access control (MAC) layers. In particular, a semi-analytical model is derived for Basic and RTS/CTS access scheme of the distributed coordination function (DCF), that applies to both, open- and closed-loop strategies. Numerical results serve to demonstrate the accuracy of the proposed model and the superiority of FLA in terms of MAC goodput in comparison to open-loop policies. Realistic operating conditions such as outdated feedback information and the use of statistical packet length distributions, issues not treated in previous studies, have also been considered. Moreover, it is shown how the inclusion of a time-out mechanism in the FLA scheme that weighs down the influence of channel information as this becomes outdated is a useful strategy to counteract its deleterious effects in Basic Access.**

***Keywords*— FLA, DCF, 802.11n, AMC, Basic Access mechanism, RTS/CTS access mechanism.**

I. Introduction

Over the last decade the IEEE 802.11 standard for wireless local area network (WLAN) has become the prevalent technology for indoor wireless Internet access. More recently, and in response to the growing demands for higher capacity, the IEEE standards committee has published the final version of IEEE 802.11n [1] as a new amendment of IEEE 802.11. Compared to previous specifications, this new norm allows much higher throughputs to be achieved while being able to fulfill more stringent quality of service (QoS) requirements. This amendment specifies enhancements to the IEEE 802.11 physical layer (PHY) and the medium access control (MAC) sublayer, most notably, the use of multiple-antenna technology (so-called MIMO) and frame aggregation, respectively. Additionally, it incorporates a feedback control channel from the receiver (Rx) to the transmitter (Tx) that enables the implementation of closed-loop adaptive mechanisms.

Adaptation plays a crucial role in dealing with the time varying nature of the wireless channel. Adaptive mechanisms allow the reconfiguration of system parameters in order to exploit the available instantaneous channel capacity while satisfying QoS constraints. One of the most widely used reconfiguration techniques is adaptive modulation and coding (AMC), which selects an appropriate modulation and coding scheme (MCS) in response to changes in the environment or system behaviour. AMC algorithms can be broadly categorized as closed- or open-loop, depending on whether an explicit feedback channel between Rx and Tx is used or not. Open-loop setups operate in an heuristic manner and their rate of adaptation tends to be slow with respect to channel changes, thus compromising the fulfilment of QoS constraints. In contrast, closed-loop mechanisms track more accurately the channel behavior and they are more reactive to rapid channel variations.

Most IEEE 802.11-based systems employ the distributed coordination function (DCF) at the MAC sublayer and adopt open-loop AMC policies such as automatic rate fallback (ARF) [2] or one of its variants (e.g. CARA [3], SARA [4]). Owing to its simplicity, ARF is by far the most popular algorithm in use. However, the DCF scheme does not differentiate between collisions and transmission failures caused by poor channel conditions. Consequently, when the system experiences a high collision probability, ARF tends to use the lowest transmission rate even if the channel conditions are favorable to use much higher transmission modes (see for example, [3], [5], [6], [7]). Other adaptive strategies have been proposed to solve this issue, but they may require frame format modifications [8], modifications to the medium access technique [3], or the use of channel quality indicators (e.g. signal strength indicator) [6], [8] and, in fact, none of them has achieved widespread use in current WLAN systems [9].

Analytical models for DCF-based WLANs that do not make use of AMC have been known since long ago [10], however, the inclusion of AMC in the theoretical framework has only been recently addressed [11], [12]. These studies have demonstrated that, in the context of IEEE 802.11n, the use of closed-loop techniques such as fast link adaptation (FLA) offers important benefits in terms of physical layer throughput. Nevertheless, current literature does not consider how this improvement reflects on the MAC goodput of a FLA-based system. This paper presents a semi-analytical model that can be used to assess the goodput performance at the MAC layer of both, open- and closed-loop adaptive schemes targeting IEEE 802.11n. The proposed model expands the one presented by the authors in [13] by modelling the retry limits and the anomalous slot performance reported in [14]. Additionally, issues that may affect very significantly the practical implementation of closed-loop strategies such as having to cope with delayed (possibly outdated) feedback information and the possible utilization of different packet lengths are also considered in this study. Lastly, a novel strategy for the FLA-based scheme that minimises the effects of delayed feedback is presented and validated.

The rest of the paper is structured as follows. Section II

ISBN: 978-84-694-5948-5

describes the system model under consideration. Section III briefly reviews the two adaptive schemes covered in this work. In Section IV the analytical framework used to analyze the system goodput is presented. In section V the simulation tool features are described and the numerical results comparing the performance of open- and closed-loop schemes are presented under different configurations and access techniques. Finally, in Section VI, the main conclusions of this study are summarized.

II. System Overview

A. Physical layer description.

Without loss of generality, our study focuses on the IEEE 802.11n standard [1], whose PHY layer is based on MIMO-OFDM. The MIMO component enables the use of different transmission techniques (e.g., space-time block coding (STBC), space division multiplexing (SDM), cyclic delay diversity (CDD) and/or combinations of them) in order to increase the system capacity and/or reliability [15]. At the transmitter side, information bits are first encoded with a rate $C = \frac{1}{2}$ convolutional encoder with generator polynomials $[133, 171]$ and then punctured to one of the possible coding rates $C_m \in \{1/2, 2/3, 3/4, 5/6\}$. Depending on the selected MIMO configuration, the resulting bits are demultiplexed into N_s spatial streams. For each stream, the coded bits are interleaved and then assigned to symbols from one of the allowed signal constellations (BPSK, QPSK, 16-QAM or 64-QAM). According to the chosen MIMO mode, the symbols are then either STBC encoded or antenna mapped on the available N_T transmit antennas. The resulting symbols are finally supplied to a conventional OFDM modulator consisting of an IFFT (inverse fast fourier transform) and the addition of a guard interval. For simplicity of exhibition, this paper focuses on a 2×2 MIMO system ($N_T = N_R = 2$), implying that MCSs with $N_s = 1$ and $N_s = 2$ spatial streams employ STBC [16] and SDM [17], respectively.

At the receiver side, Alamouti decoding or Minimum Mean Square Error (MMSE) detection is applied depending on whether STBC or SDM has been employed. In either case, the detector extracts soft information in the form of log-likelihood ratios (LLRs) that, after suitable de-interleaving/de-parsing, can be exploited by a soft Viterbi decoder [12].

B. MAC layer description.

The IEEE 802.11 standard specifies three different MAC mechanisms for WLANs, namely the DCF, the point coordination function (PCF) and the hybrid coordination function (HCF). The DCF is the mandatory MAC mechanism for the IEEE 802.11 standard [1]. It is a random access scheme based on the carrier sense multiple access with collision avoidance (CSMA/CA) protocol that incorporates a binary exponential backoff (BEB) algorithm to manage the retransmission of collided and erroneous packets.

DCF defines two access techniques, the Basic Access and the RTS/CTS. The Basic Access technique is the mandatory DCF mechanism for 802.11 and the most extensively used [3]. In Basic Access, a station (STA) transmits one data packet at its BEB scheduled slot and waits for its packet-acknowledgement (ACK-control frame) from the receiver. If no reply arrives during a predefined time interval, the STA interprets the transmission as erroneous and the packet is retransmitted in the next BEB scheduled slot or discarded if the number of packet retransmissions exceeds the maximum number of allowed retransmissions, which will be denoted by R.

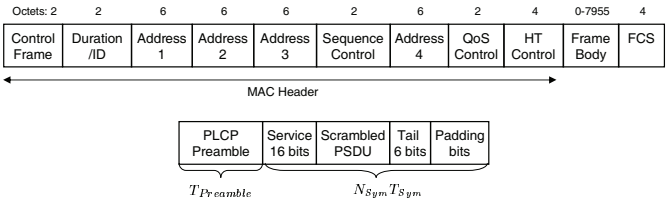


Figure 1. MAC and PPDU frame format

In contrast the implementation of RTS/CTS is optional and its utilization is not common in real systems [3]. This mechanism is enabled once the packet length is larger than a predefined threshold (S_{thres}). Unlike Basic Access, prior to the data transmission, RTS/CTS exchanges two control frames (request to send (RTS) and clear to send (CTS)) between source and destination. This frame exchange allows the reservation of the channel for the current data transmission and drastically reduces the collision duration to that of two collided RTS frames. However, RTS/CTS considerably increases the system overhead lowering in this way the system performance when short packet lengths or high transmission rates are used. Nevertheless, thanks to its reduced collision duration, it outperforms Basic Access in dense user scenarios where the collision probability is high.

C. Frame format novelties of IEEE 802.11n.

IEEE 802.11n has introduced some frame format changes with respect to previous amendments that allow transmissions techniques mostly available thanks to the presence of a feedback channel. Note that often these enhancements are at the cost of some overhead increment and the backward device compatibility. At the MAC layer, the MPDU (MAC protocol data unit) frame format (Fig. 1) incorporates, among others, the high throughput (HT) control field that possibilities the MCS feedback between transmission entities. Additionally, two extra PPDU frame formats are defined at the physical layer with longer preamble fields improving the accuracy of the channel estimation and enabling other techniques (e.g., beamforming, SDM or CDD). In this work, the PPDU 802.11n frame format presented in Fig. 1 has been employed.

D. Timing of DCF events.

1) Basic Access technique: According to the Basic Access of DCF, the elapsed time for a successful transmission of an L-bit MAC packet data unit (MPDU) using MCS m is

$$T_s^{bas}(m, L) = T_{Tr}(m, L) + t_{SIFS} + T_{ACK+HTC}(m) + t_{DIFS}, \quad (1)$$

where t_{SIFS} (short interframe space) and t_{DIFS} (distributed interframe space) are 802.11n time constants defined in [1]. The time elapsed in the MPDU transmission, $T_{Tr}(m, L)$, is defined as

$$T_{Tr}(m, L) = t_{Preamble} + N_{Sym}(m, L)t_{Sym}, \quad (2)$$

with $t_{Preamble}$ representing the PLCP preamble duration, t_{Sym} denoting the OFDM symbol period and

$$N_{Sym}(m, L) = m_{STBC} \left\lceil \frac{L + 22}{m_{STBC} N_{DBPS}(m)} \right\rceil, \quad (3)$$

being the number of OFDM symbols involved in the transmission of a complete MPDU, where $N_{DBPS}(m)$ is the quantity of bits forming each OFDM symbol as defined by MCS m, $\lceil z \rceil$ denotes the smallest integer greater than or equal to z, and $m_{STBC} = 2$ if STBC is used and $m_{STBC} = 1$ otherwise. Similarly, the time required for the transmission of an ACK+HTC frame[1] using PHY mode m is given by

$$T_{ACK+HTC}(m) = t_{Preamble} + N_{Sym}(m, 20 \cdot 8) t_{Sym}. \quad (4)$$

A collision occurs whenever two or more STAs transmit on the same slot, finishing t_{EIFS} (extended interframe space) after the end of the longest transmission of the collided STAs. That is, its duration depends on the MCS and MPDU length corresponding to the longest transmission, denoted by m^* and L^*, respectively. Therefore, the collision duration can be mathematically expressed as

$$T_c^{bas}(m^*, L^*) = T_{Tr}(m^*, L^*) + t_{EIFS}, \quad (5)$$

where

$$t_{EIFS} = t_{SIFS} + T_{ACK}(\text{m=0}) + t_{DIFS}. \quad (6)$$

The MPDU error transmission duration $T_e(m, L)$ is the time elapsed in a transmission that experiences errors without collisions, and it can be expressed as

$$T_e^{bas}(m, L) = T_{Tr}(m, L) + t_{EIFS}. \quad (7)$$

2) RTS/CTS access technique: Compared to Basic Access, the RTS/CTS access mechanism incorporates a frame exchange prior to the data transmission, thus, increasing the elapsed time during successful and error transmissions and fixing to small value the collision duration from any STA.

The RTS/CTS time elapsed for a successful transmission of a L-bit MPDU using MCS m is

$$\begin{aligned} T_s^{rts}(m, L) =& T_{RTS+HTC}(m) + T_{CTS+HTC}(m) + \\ &+ T_{Tr}(m, L) + T_{ACK+HTC}(m) + \\ &+ 3 * t_{SIFS} + t_{DIFS}, \end{aligned} \quad (8)$$

where

$$T_{RTS+HTC}(m) = t_{Preamble} + N_{Sym}(m, 20 \cdot 8) t_{Sym}, \quad (9)$$

and

$$T_{CTS+HTC}(m) = t_{Preamble} + N_{Sym}(m, 26 \cdot 8) t_{Sym}, \quad (10)$$

are control wrapper frames encapsuling RTS and CTS frames, respectively.

Similarly, the RTS/CTS elapsed time in a transmission error of an L-bit MPDU using MCS m is

$$\begin{aligned} T_e^{rts}(m, L) =& T_{RTS+HTC}(m) + T_{CTS+HTC}(m) \\ &+ T_{Tr}(m, L) + 2 * t_{SIFS} + t_{EIFS}. \end{aligned} \quad (11)$$

[1]Wrapper control frame that encapsules the ACK and the HT control field required to feedback the MCS selection.

In this case, collisions only involve RTS frames, which are independent of the data packet length. Consequently their duration is drastically reduced when compared to the Basic Access. The collision duration can be defined as

$$T_c^{rts}(m^*) = t_{RTS+HTC}(m^*) + t_{EIFS}. \quad (12)$$

In this model, due to its negligible probability of occurrence, we have not considered the possibility of an error in the ACK, RTS and CTS transmissions. The ACK transmission takes place under the same system conditions than the packet being acknowledged, i.e., using the same MCS and suffering similar channel conditions. However, its packet size is considerably smaller than that of the information packets and therefore, its error probability can be safely considered insignificant. Similarly, packet sizes for RTS and CTS transmissions are also small and the use of the most reliable MCS warrants a negligible error probability.

III. Adaptive modulation and coding strategies

A. ARF

This algorithm adapts the transmission rate according to the number of consecutive transmission failures and successes, both reported by the ACK mechanism. The transmission rate is decreased after two consecutive transmission failures and increased after either ten consecutive successful packet transmissions or a timeout. In order to improve the system adaptation during long intervals of inactivity, this timeout counter is reset after a transmission rate change or after a transmission failure [2]. Acceptable timeout values lie in the range of 50-200 ms [18]. Note that, following a rate increase, the next data transmission is deemed as a probing transmission for the new mode. If an ACK is not received for this probing packet the system falls back to the previous data rate.

In order to implement ARF in IEEE 802.11n it is necessary to determine the available rates in the MCS set, denoted by $\boldsymbol{\mathcal{M}}$. In contrast to previous IEEE 802.11 standards, in 802.11n different MCSs $\in \boldsymbol{\mathcal{M}}$ can provide the same transmission rate, but only one of them can be used by the ARF algorithm. For this reason, the MCSs in $\boldsymbol{\mathcal{M}}$ are reordered according to their transmission rate [12]. For those rates that can be attained using either SDM or STBC, only the STBC MCS is kept as it can be shown to be more robust against channel variations [16]. The reordered and pruned MCS set will be denoted by $\overline{\boldsymbol{\mathcal{M}}}$.

B. FLA

Fast link adaptation is a closed-loop technique that relies on the availability of a feedback channel from the receiver to the transmitter. The main idea behind FLA is that the receiver, thanks to an accurate knowledge of the channel response, can compute a reliable prediction of the error rate for all available MCSs and choose the one maximising the instantaneous throughput while satisfying QoS constraints in the form of outage packet error rate probability. The selected MCS can then be communicated to the transmitter via the feedback channel. In this work we assume the use of the methodology presented in [12], where link performance prediction for each MCS is based on the exponential effective SNR mapping (EESM). Using this approach, the EESM for a given MCS can

be easily associated to packet error rate (PER) using look-up tables that have been previously computed during an off-line calibration phase.

1) Basic Access: In crowded scenarios using Basic Access, the delay between the MCS selection at the receiver and its use at the transmitter can be very large, very often exceeding the channel coherence time and significantly affecting the FLA operation. In DCF, all STAs have an equal long term probability of accessing the medium. Therefore, successive transmissions from a given STA are intertwined with transmissions from the other contending STAs with the time between successive transmissions increasing with the MCS feedback delay. This delay becomes critical for FLA when it approaches the channel coherence time, indicating that the provided MCS has been determined for a channel response that is almost uncorrelated to the current channel response. This mismatch between current and prior channel state can increase the error probability due to a mistakenly selected or expired MCS, causing several consecutive errors in the next retransmissions prior to packet discard.

In order to counteract the effects of using stale feedback MCSs, we propose that the STA decreases the transmission mode when the MCS feedback delay exceeds a fixed timeout. The STA will decrease again the MCS in all the subsequent packet retransmissions (if any) until the packet is successfully transmitted. The timeout is configured to a value close to the channel coherence time, assuring in this way that the current channel response is similar to the one that the receiver has used to determine the feedback MCS. As it will be shown in the numerical results section, this strategy reduces the error probability without considerably affecting the goodput and fulfilment of QoS constraints.

2) RTS/CTS: Unlike Basic Access, the provided MCS is calculated with the channel response affecting the RTS frame and is returned to the sender in the CTS frame response, just before its use for the data transmission. Therefore, the MCS feedback delay is almost negligible compared to the channel coherence time, and independent of the time elapsed since the last successful transmission.

IV. Goodput analysis

Following the model presented in [10] and then refined in [14], the goodput analysis presented in this paper is suitable for both access techniques, Basic Access and RTS/CTS, and focuses on the saturation region, defined as the operation point where each STA has always new packets to transmit. The system saturation goodput S can be defined as

$$S = \frac{E\{\text{payload information in a slot}\}}{E\{\text{duration of a slot}\}}, \tag{13}$$

where $E\{\cdot\}$ denotes statistical expectation. The duration of a slot refers to the time interval between two consecutive backoff counter decrements.

In any given slot, one out of four events can occur: a successful packet transmission (s), an error packet transmission (e), a collision (c) or an idle slot (i). From the point of view of the BEB algorithm, error transmissions and collisions are undistinguishable. The conditional probability of the union of these events can be computed as

$$p = 1 - (1 - \zeta_u)(1 - \tau)^{n-1}, \tag{14}$$

where n is the number of active STAs in the scenario, ζ_u is the user error transmission probability for the considered AMC algorithm, averaged on a per-user basis, and τ is the stationary probability that a particular STA transmits in a given slot. This transmission probability can be obtained as

$$\tau = \frac{1}{1 + \frac{1-p}{2(1-p^{R+1})}[\sum_{j=0}^{R} p^j \cdot (2^{\tilde{j}}W - 1) - (1 - p^{R+1})]}, \tag{15}$$

where $W = CW_{min} + 1$ and $\tilde{j} = \min(j, m_{max})$ with m_{max} denoting the maximum backoff stage. Notice that p and τ can be obtained by solving the nonlinear system formed by eqs. (14) and (15).

Using τ, the probability that only one STA transmits on a given slot is

$$P_s = n\tau(1 - \tau)^{n-1}. \tag{16}$$

Furthermore, the probability that a given slot is idle is given by

$$P_i = (1 - \tau)^n. \tag{17}$$

Among all possible events, only the successful packet transmission increases the payload information while any other event leads to a goodput degradation. Consequently, combining the goodput expression in [19, eq. (50)] with [14, eq. (18)], and taking into account the use of multiple transmission modes, the system goodput can be expressed as

$$S = \frac{(1 - \zeta_s)P_s\overline{L_p}\left[\frac{W}{W-1}\right]}{P_i\sigma + (1 - \zeta_s)P_s\overline{T_s^{(n,L)}} + \zeta_s P_s\overline{T_e^{(n,L)}} + (1 - P_s - P_i)\overline{T_c^{(n,L)}}}, \tag{18}$$

where $\overline{L_p} = E\{L_p\}$, with $L_p = L - L_h$ representing the packet payload length and L_h denoting the MAC sublayer overhead, ζ_s denotes the average system packet error probability for a given slot, σ is the idle slot duration [1], and the time values $\overline{T_s^{(n,L)}}$, $\overline{T_c^{(n,L)}}$ and $\overline{T_e^{(n,L)}}$ represent the average elapsed time for successful, colliding and error transmissions, respectively. Notice that $\overline{L_p}$ is multiplied by $\left[\frac{W}{W-1}\right]$ in order to account for the additional information transmitted in anomalous slots[2] [14] and $\overline{T_s^{(n,L)}}$ includes the anomalous slot duration.

V. Numerical Results

In order to validate our semi-analytical model and compare the performance of FLA and ARF under different system configurations, an IEEE 802.11n system-level Matlab simulator has been implemented using the link-level parameters derived in [12]. It should be stressed that this model is considerably more realistic than the one proposed by Tinnirello et al. in [14], since it allows the treatment of AMC, statistical packet length distribution and non-ideal closed-loop FLA strategies, at the expense of relying on some semi-analytic parameters. In this paper we concentrate on the performance evaluation of the uplink scenario where, nevertheless, MAC control frame transmissions from access point (AP) to STA are also accounted for. Different scenarios have been generated by uniformly distributing n static users in a circular area of radius R_{max} centered around the AP and then determining

[2]Slot with a lower probability to be accessed than the average (see [14] for more details).

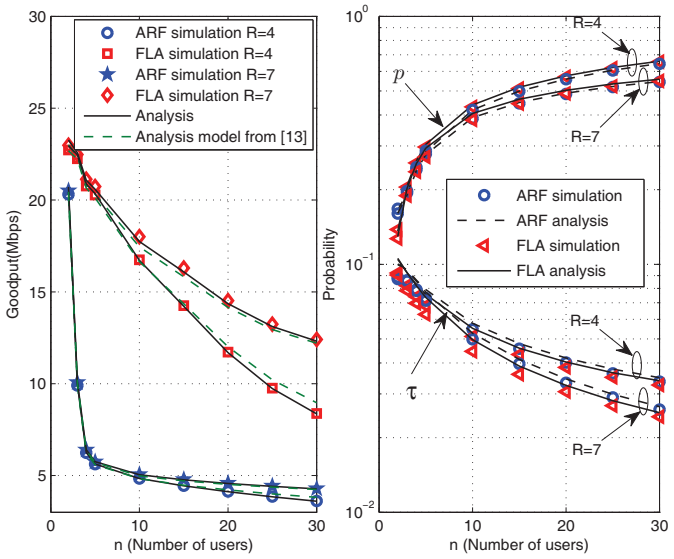


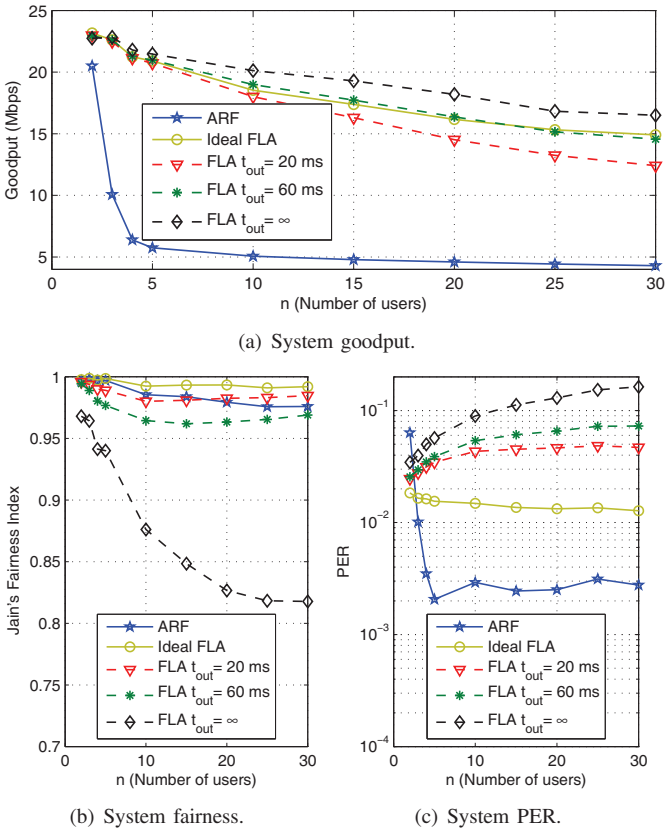
Figure 2. Semi-analytic and simulated system performance of goodput, τ and p using $R = 4$ and $R = 7$ retransmissions.

the individual channel response from each user to the AP. To this end, the MIMO channel generation tool presented in [20], parameterized with each user's distance to the AP, has been employed. The maximum radius R_{max} has been set to 30 m, a value that ensures the avoidance of the hidden terminal problem and precludes the utilization of the no transmission mode (available in FLA). For all STAs, transmit power has been set to 20 dBm and receiver noise power to -80 dBm. The physical layer uses only the first 16 MCS modes of IEEE 802.11n (MCS0-MCS15), achieving date rates of up to 130 Mbps [1]. It should be stressed that all the users inside the scenario use the same DCF access technique, Basic Access or RTS/CTS. The ARF timeout has been set to 60 ms and the FLA outage constraint for a PER objective (not including collisions) of 10^{-1} has been configured to 10%. The corresponding CSI feedback overhead has also been taken into account in FLA. In order to obtain an accurate estimate of the average system performance $N_{sim} = 100$ simulation runs of duration $t_{sim} = 22$ seconds have been generated for each value of n.

A. Basic Access

The left and right plots of Fig. 2 show the goodput performance and the conditional probabilities p and τ, respectively, as a function of the number of STAs and for a fixed packet length of $L_p = 1500$ bytes. A very accurate match between the semi-analytical and simulated system performance metrics for FLA- and ARF-based schemes can be appreciated. The left plot of Fig. 2 also reports the goodput performance obtained using the proposed semi-analytical model compared to the previous proposal[3] presented in [13], where the anomalous slot performance and the packet retry limit were not considered. Although the previous model provides valuable approximations to the simulation performance, the new semi-analytical model results in improved modelling accuracy, especially when the system uses $R = 4$. The left plot in Fig. 2

[3]Configured to $m_{max} = 4$ or $m_{max} = 6$ in order to be compared to the new model using $m_{max} = 6$ with $R = 4$ or $R = 7$, respectively.



(a) System goodput.

(b) System fairness.

(c) System PER.

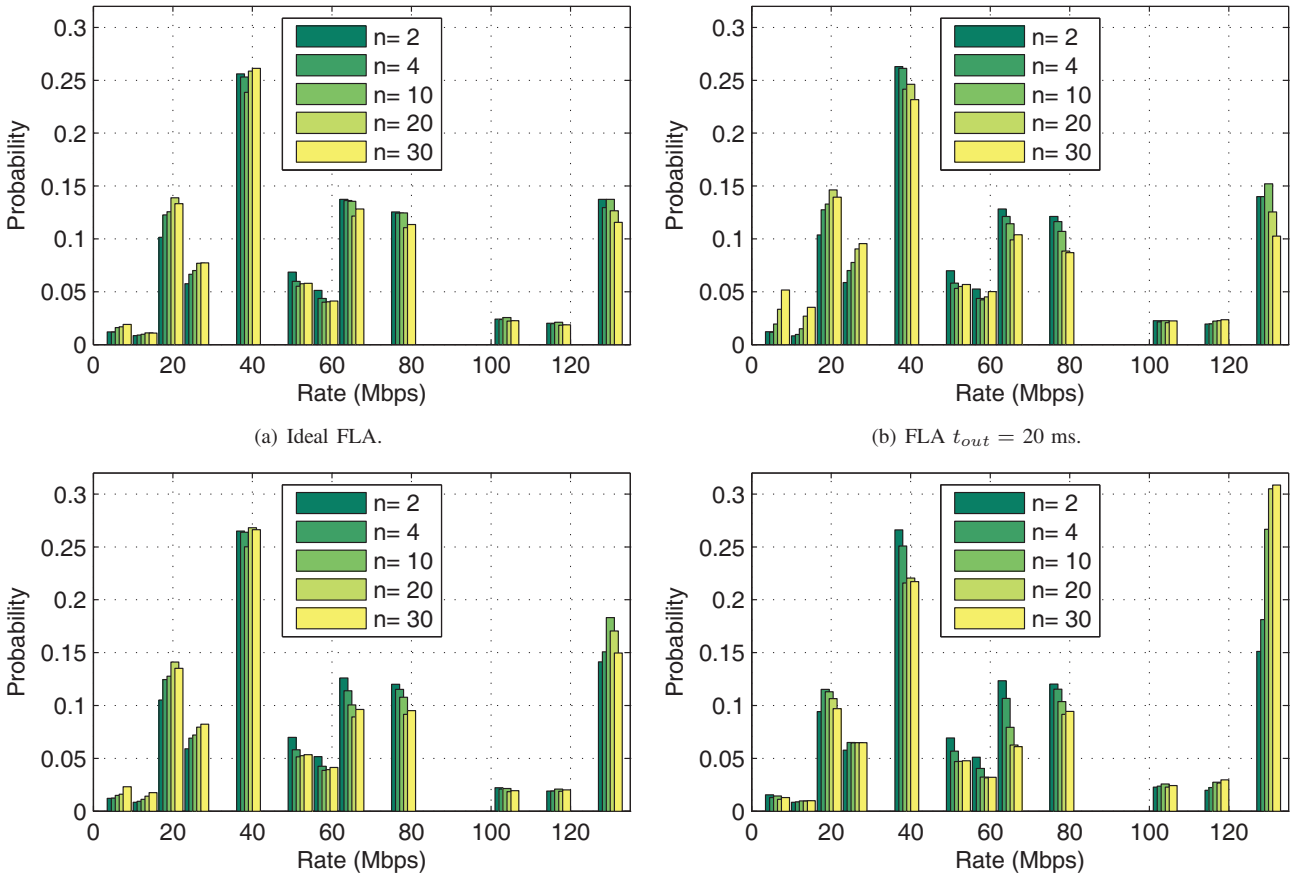
Figure 3. Goodput, Jain's fairness index and PER of ARF and FLA strategies using $R = 7$.

also illustrates that regardless of the retry limit, FLA-based schemes outperform ARF-based strategies in terms of goodput performance. This is because a lower retry limit leads to a lower average backoff contention window, thus increasing the transmission probability (τ) and, consequently, the collision and error probability (p).

The Fig. 3a shows the goodput performance of FLA under the assumptions of ideal channel state information (CSI) and non ideal CSI with different timeout values (t_{out}) when using $R = 7$. A priori, ideal FLA could be expected to provide the maximum goodput, however it is outperformed by FLA with $t_{out} = \infty$. Remarkably, as it can be observed in Fig. 3b, this goodput improvement is at the expense of a loss in the Jain's fairness index measured in terms of the per-STA transmission opportunity[4]. This loss in fairness is mainly due to two facts:

1) The STAs with very good channel conditions (high SNR) mostly use the highest throughput MCS and, on average, they are granted the channel more frequently than the other STAs. This is because their conditions are so favorable that their probability of error is very small, regardless of the MCS feedback delay.
2) Due to the MCS feedback delay, the rest of STAs experience an increased error rate and consequently, the DCF mechanism reduces their probability of accessing the medium.

[4]The Jain's fairness measure used in this paper is calculated as $I = \frac{(\sum_i^n \beta_i)^2}{n \sum_i^n \beta_i^2}$ where β_i denotes the number of transmissions for STA i. Note that $I = \frac{1}{n}$ implies an unfair system and $I = 1$ reflects a completely fair system.



(a) Ideal FLA.

(b) FLA $t_{out} = 20$ ms.

(c) FLA $t_{out} = 60$ ms.

(d) FLA $t_{out} = \infty$.

Figure 4. Rate probability of successful transmissions for different FLA settings using $R = 7$ system configuration.

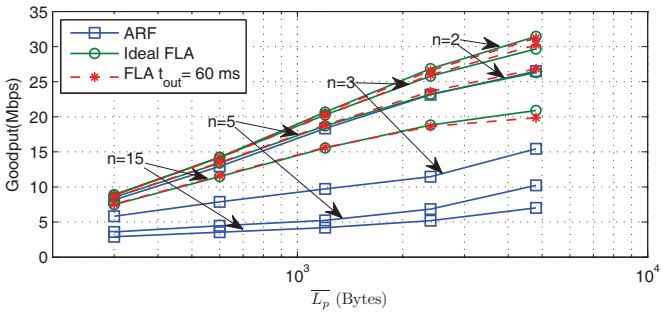
The combination of these two facts results in an overall system goodput improvement due to a more frequent use of the highest rate MCS (see Fig. 4 where the MCS rate probability of successful transmissions is depicted for different FLA configurations), and despite the higher error probability for configurations employing a finite time-out value (see Fig. 3c). Under the constraint of maximum system fairness, the ideal FLA can be considered as the benchmark system from a goodput point of view. Nevertheless, ideal FLA is not implementable due to the 802.11n MCS feedback mechanism, which invariably introduces some delay in its transmission. In order to improve the FLA performance for those STAs that experience large MCS delays, the FLA algorithm proposed in this paper lowers the MCS rate after the expiration of a finite timeout. Although the timeout should be set according to the channel coherence time, experimental results show that FLA with $t_{out} = 60\ ms$ performs similarly to ideal FLA in terms of goodput, while preserving a high fairness index and satisfying PER-based QoS constraints (see Fig. 3b and Fig. 3c, respectively).

In Fig. 5, the performance of FLA and ARF is shown for a packet length (L_p) modelled as a doubly truncated exponential distribution between 40 and 10.000 bytes. When using FLA, it is assumed that the receiver knows the length of the next packet to be transmitted when determining the most suitable MCS for the next packet transmission. This assumption is quite realistic since there exists a high L_p correlation between consecutive packet lengths sent from the same STA in typical WLAN environments.

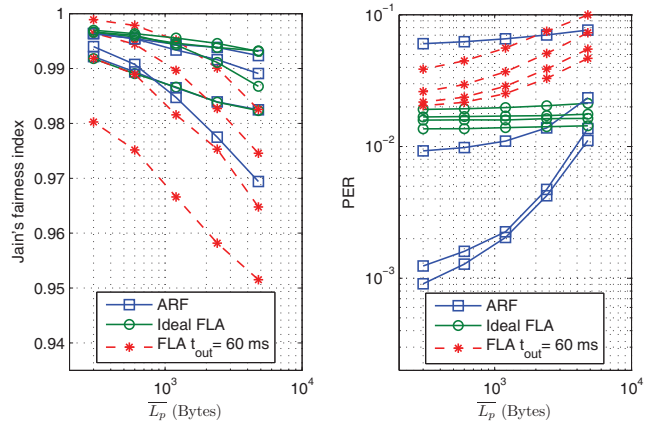
The Fig. 5a presents the goodput performance of ARF, ideal FLA and FLA (FLA with $t_{out} = 60ms$), for different average packet sizes ($\overline{L_p}$). Due to the large overhead introduced by the DCF mechanism, the adoption of long $\overline{L_p}$ values improves the DCF protocol efficiency and consequently, the system goodput increases, especially for the FLA cases. Note that FLA is still outperforming ARF for any $\overline{L_p}$ and number of users, most notably for those cases where more than two users are contending for the medium. As previously observed, the goodput performance of FLA with $t_{out} = 60$ ms is similar to ideal FLA for the whole range of $\overline{L_p}$ values and number of users under consideration (see Fig. 5a). Furthermore, it keeps Jain's fairness index high (see Fig. 5b) and fulfills the PER QoS constraint for all the considered configurations (see Fig. 5c). Note that the system PER performance of FLA increases for large packets as a consequence of the obvious increment of the average MCS feedback delay. For completeness, Fig. 5b and Fig. 5c, also present fairness and PER performance, respectively, for ARF and ideal FLA.

B. RTS/CTS

In Fig. 6, the system performance of FLA and ARF with Basic and RTS/CTS access techniques using $L_p = 1500$ bytes is presented. The ARF with RTS/CTS studied in this paper is equivalent to the CARA-RTS algorithm defined in [3] setting $P_{th} = 0$ (probing activation threshold) and $N_{th} = 1$

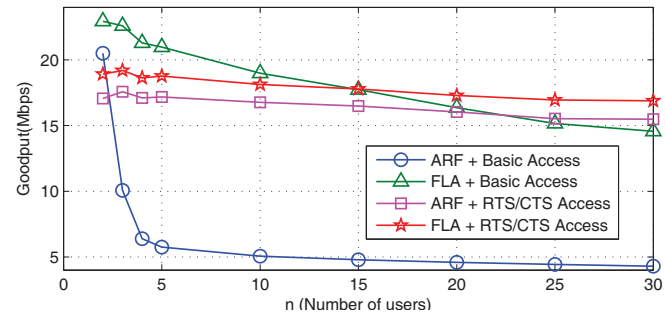


(a) System goodput.

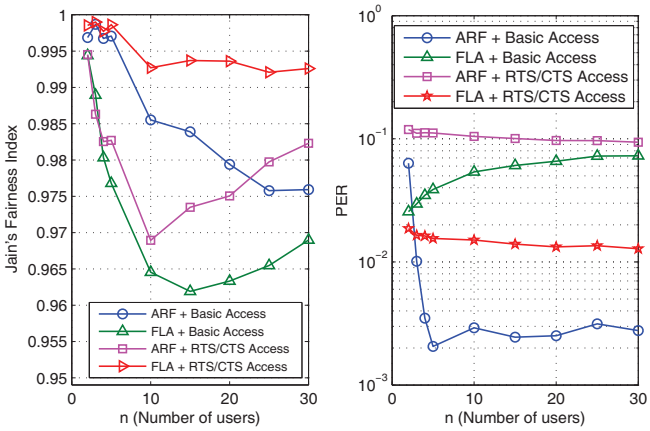


(b) System fairness. (c) System PER.

Figure 5. Goodput, Jain's fairness index and PER system performance as a function of $\overline{L_p}$ and n.



(a) System goodput.



(b) System fairness. (c) System PER.

Figure 6. FLA and ARF performance using RTS/CTS and Basic Access with fixed $L_p = 1500$ Bytes.

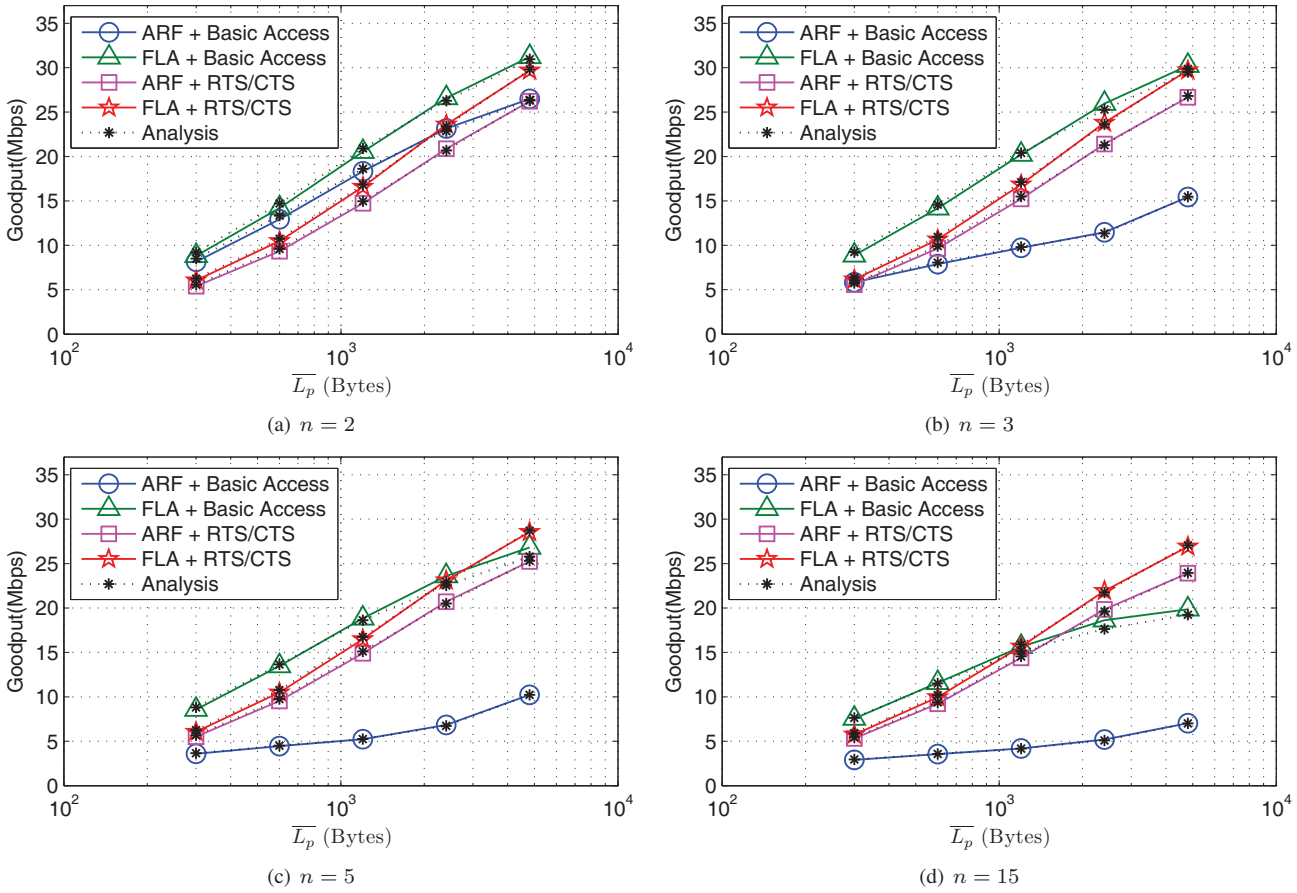
(consecutive failure threshold) parameters. In RTS/CTS, FLA is still outperforming ARF in terms of goodput irrespective of the number of users. However, the goodput improvement is significantly lower than in the Basic Access case (see Fig. 6a), as now ARF is capable of distinguishing the source of errors (collisions or channel errors). Note that, thanks to the reduced MCS feedback delay when employing RTS/CTS, FLA optimally selects the MCS and results in PER values similar to those obtained when using ideal FLA (see Fig. 6c and Fig. 3). Remarkably, FLA with Basic Access is outperformed by FLA with RTS/CTS for $n > 15$, due to the better performance of RTS/CTS in denser collision environments. Finally, it should be mentioned that, as in Basic Access, RTS/CTS for both AMCs maintains a high degree of fairness for any n (see Fig. 6b).

Figure 7 shows analytical and simulation goodput performance for different n and $\overline{L_p}$, demonstrating the accuracy of the semi-analytical model and the clear superiority of FLA with respect to ARF when compared over the same access technique. Moreover, FLA on RTS/CTS, thanks to the reduction in the collision duration, outperforms FLA on Basic Access for long packet lengths and $n > 2$ (see Fig. 7a). Lastly, it should be stressed the goodput performance enhancement of ARF with RTS/CTS over ARF with Basic Access for any $\overline{L_p}$ and $n > 2$.

VI. Conclusions

This paper has presented a semi-analytical framework for the performance modelling of MIMO-OFDM WLANs when using the Basic and RTS/CTS access scheme of DCF at the MAC layer. Unlike previous works, the proposed model is able to incorporate the effects of channel errors, the possibility of using open- or closed-loop transmission mode adaptation, the effect of the retry limit at the MAC layer and the use of outdated MCS feedback information. A complete study of FLA over 802.11n PHY/MAC in terms of goodput, fairness and system PER performance for a wide range of number of users and packet sizes has been presented and contrasted to those obtained using ARF. Noteworthy, the influence of feedback delay in FLA has been assessed. In Basic Access scheme, It has been found that the degradation caused by an outdated MCS information can be largely compensated with the use of a time-out strategy that weighs down the influence of the received feedback. Numerical results clearly show that as the number of users in the system grows, the FLA-based adaptation proves to be much more robust to collisions than ARF even when employing outdated MCS information. This effect is clearly demonstrated by the fact that whereas ARF-based schemes suffer a dramatic reduction in goodput for more than 2 users, the FLA-based strategy exhibits a very graceful degradation thanks to a more accurate rate selection in the presence of collisions.

When using the RTS/CTS access scheme, FLA still improves over ARF irrespective of the number of users and packet size, although the difference is not as significant as in Basic Access. Nevertheless, it should be remarked that the RTS/CTS frame exchange allows a delay-free selection of the MCS and results in a system PER similar to the one of the ideal FLA case. Overall it can be concluded that FLA yields a goodput that significantly outperforms the one of ARF for



(a) $n = 2$ (b) $n = 3$

(c) $n = 5$ (d) $n = 15$

Figure 7. Goodput using different $\overline{L_p}$.

most system loads and access schemes, while keeping a large degree of fairness and satisfying prescribed PER-based QoS constraints.

Acknowledgements

This work has been partially funded by MEC and FEDER through project COSMOS (TEC2008-02422) and Conselleria d'Economia, Hisenda i Innovació del Govern de les Illes Balears through a PhD grant.

References

[1] IEEE, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput," *IEEE Std 802.11n-2009*, 2009.

[2] A. Kamerman and L. Monteban, "WaveLAN®-II: a high-performance wireless LAN for the unlicensed band," *Bell Labs technical journal*, vol. 2, no. 3, pp. 118–133, 1997.

[3] S. Kim, L. Verma, S. Choi, and D. Qiao, "Collision-aware rate adaptation in multi-rate wlans: Design and implementation," *Computer Networks*, vol. 54, no. 17, pp. 3011 – 3030, 2010.

[4] T. Joshi, D. Ahuja, D. Singh, and D. Agrawal, "Sara: stochastic automata rate adaptation for IEEE 802.11 networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 11, pp. 1579–1590, 2008.

[5] J. He, D. Kaleshi, A. Munro, and J. McGeehan, "Modeling Link Adaptation Algorithm for IEEE 802.11 Wireless LAN Networks," in *IEEE ISWCS*, Valencia, Spain, Sept. 2006.

[6] J. Zhang, K. Tan, J. Zhao, H. Wu, and Y. Zhang, "A Practical SNR-Guided Rate Adaptation," in *IEEE INFOCOM*, Phoenix, AZ, April 2008.

[7] H. Jung, T. Kwon, Y. Choi, and Y. Seok, "A scalable rate adaptation mechanism for IEEE 802.11e wireless," in *IEEE FGCN*, vol. 1, Jeju-Island, Korea, Dec. 2007.

[8] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-Hop wireless networks," in *ACM MobiCom*, 2001, pp. 236–251.

[9] J. Choi, J. Na, Y. sup Lim, K. Park, and C. kwon Kim, "Collision-aware design of rate adaptation for multi-rate 802.11 WLANs," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 8, pp. 1366 –1375, 2008.

[10] 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, March 2000.

[11] G. Martorell, F. Riera-Palou, G. Femenias, "Cross-layer link adaptation for IEEE 802.11n," in *IEEE IWCLD*, Palma, Spain, June 2009.

[12] G. Martorell, F. Riera-Palou, and G. Femenias, "Cross-layer fast link adaptation for MIMO-OFDM based WLANs," *Springer Wireless Personal Communications*, vol. 56, no. 3, pp. 599–609, 2011.

[13] G. Martorell, F. Riera-Palou, G. Femenias, "DCF performance analysis of open- and closed-loop adaptive IEEE 802.11n networks," in *IEEE ICC*, Kyoto, Japan, June 2011.

[14] I. Tinnirello, G. Bianchi, and Y. Xiao, "Refinements on IEEE 802.11 Distributed Coordination Function Modeling Approaches," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 3, pp. 1055 –1067, 2010.

[15] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[16] Y.-S. Choi and S. Alamouti, "A pragmatic PHY abstraction technique for link adaptation and MIMO switching," *IEEE Journal of Selected Areas in Communications*, vol. 26, no. 6, pp. 960–971, 2008.

[17] G. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs Technical Journal*, vol. 1, no. 2, pp. 41–59, 1996.

[18] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-hop wireless networks," in *ACM MobiCom*, Rome, Italy, 2001.

[19] B. Bing, *Emerging Technologies in Wireless LANs: Theory, Design, and Deployment*. Cambridge Univ Press, 2007.

[20] J. Kermoal, L. Schumacher, K. Pedersen, P. Mogensen, and F. Frederiksen, "A stochastic MIMO radio channel model with experimental validation," *IEEE Journal of Selected Areas in Communications*, vol. 20, no. 6, pp. 1211–1226, 2002.

Evaluación de prestaciones de diferentes variantes de TCP en un entorno satelital DVB-S2

Elizabeth Rendon-Morales[1], Jorge Mata-Díaz[1], Juanjo Alins[1], José Luis Muñoz[1], Oscar Esparza[1]
Departamento de Ingeniería Telemática
Universidad Politécnica de Cataluña (UPC)
Jordi Girona 1-3, 08034, Campus Nord, Barcelona, España.
{elizabeth.rendon, jmata, juanjo, jose.munoz, oscar.esparza}@entel.upc.es

Resumen- La arquitectura de Servicios Diferenciados (DiffServ por sus siglas en inglés) permite que el tráfico IP sea clasificado en un número finito de clases con diferentes niveles de prioridad. El protocolo de Control de Transporte (TCP) es empleado para garantizar el transporte extremo a extremo del tráfico en Internet, gracias a sus mecanismos de control de flujo y control de congestión. El propósito de este trabajo es presentar un análisis de desempeño del protocolo TCP considerando la arquitectura DiffServ para proporcionar Calidad de Servicio (QoS) sobre un sistema satelital basado en el estándar Digital Video Broadcasting - Segunda Generación (DVB-S2). El análisis se realiza mediante la herramienta de simulación NS-2 y las variantes de TCP consideradas son: Sack, Cubic y Hybla. El objetivo es evaluar el rendimiento del protocolo TCP tomando en cuenta los problemas más comunes presentes en un enlace satelital DVB-S2, como lo son el retardo, la perdida de paquetes y las variaciones de ancho de banda. Mediante la evaluación de las variantes de TCP propuestas en diferentes escenarios de simulación, se observa que TCP Cubic es la variante que muestra un mejor rendimiento, comparado con otras variantes de TCP al ser evaluado en un sistema satelital DVB-S2 con QoS.

Palabras Clave- DiffServ, TCP, Evaluación de desempeño, DVB-S y QoS.

INTRODUCCIÓN

Actualmente, debido a la creciente demanda de usuarios de Internet, las conexiones basadas en la pila de protocols TCP/IP (Transport Control Protocol/Internet Protocol por sus siglas en inglés) han experimentado un constante aumento y su impacto será aún mayor en un futuro próximo [1]. En particular, el uso del protocolo TCP en entornos satelitales ha cobrado relevancia en los últimos años. Sin embargo, la implementación del protocolo TCP en entornos satelitales requiere considerar los problemas actuales que enfrentan los sistemas de satélites geoestacionarios (GEO), ya que estos pueden afectar seriamente el rendimiento del protocolo TCP [2]. Los problemas mas importantes son: la disponibilidad de ancho de banda en el canal satelital, la cual puede variar debido a las condiciones atmosféricas, además de los efectos que genera el retardo de propagación, y la presencia de pérdidas de paquetes, ya sea debido a la corrupción de datos por errores en la transmisión o a la congestión de paquetes.

Para resolver esto, los estándares Digital Video Broadcasting Primera y Segunda Generación (DVB-S y DVB-S2) [3] han sido propuestos con el fin de contrarrestar los errores de transmisión presentes en las redes satelitales.

Particularmente, el estándar DVB-S2 se beneficia de los avances más recientes basados en la codificación de canal (códigos de baja densidad LDPC), que al combinarse con diferentes esquemas de modulación (QPSK, 8PSK, 16APSK, 32APSK), permiten ajustar la tasa de transmisión dependiendo de las condiciones del canal. Por otra parte, el estándar DVB-S2 hace normativo el uso de la codificación y modulación adaptativa (ACM) con el fin de mantener la tasa de error (BER) acotada.

Para proporcionar Calidad de Servicio (QoS) en un sistema satelital basado en el estándar DVB-S2, el grupo de trabajo denominado Broadband Satellite Multimedia (BSM), recomienda el uso de la arquitectura de Servicios Diferenciados (DiffServ). En esta arquitectura el tráfico IP se clasifica en un número finito de clases diferenciadas por su orden de prioridad con el objetivo de ofrecer diferentes niveles de QoS. Adicionalmente, mediante el uso de la arquitectura DiffServ la complejidad de la red se traslada a los nodos frontera, con el fin de brindar escalabilidad y simplicidad.

La arquitectura DiffServ [4] se puede describir considerando tres componentes principales: los *clasificadores de tráfico* que seleccionan los paquetes y dependiendo de su nivel de prioridad asignan su respectivo identificador DSCP (Differentiated Services Code Point). Los *acondicionadores de tráfico* marcan y fortalecen las especificaciones de velocidad definidas en los Acuerdos de Nivel de Servicio (SLA). Finalmente, los *tratamientos por salto* (PHB), son definidos para garantizar la diferenciación de paquetes mediante una política de QoS. Dicha política se define con el objetivo de gestionar y controlar la distribución proporcinal de ancho de banda entre los tres tipos de PHBs o clases de servicio (nombrados por sus acrónimos en inglés): Expedited Forwarding (EF), Assured Forwarding (AF) y Best-Effort (BE).

En este contexto, el presente trabajo propone analizar el desempeño de un grupo selecto de variantes de TCP trabajando sobre un sistema satelital DVB-S2, teniendo en cuenta la arquitectura DiffServ para el soporte de QoS. El análisis propuesto se lleva a cabo mediante la herramienta de

ISBN: 978-84-694-5948-5

simulación NS-2, en el cual tres variantes de TCP son consideradas: TCP Sack, TCP Cubic y TCP Hybla.

El objetivo de este trabajo es evaluar el desempeño del protocolo de transporte TCP trabajando en una arquitectura con el soporte de QoS y teniendo en cuenta los principales problemas que afectan a un sistema satelital, como lo son el retardo, las pérdidas de paquetes y las variaciones de ancho de banda. Para alcanzar este objetivo, se propone una topología de red satelital con el fin de evaluar las prestaciones de las variantes de TCP seleccionadas. Dicha topología de red considera la arquitectura DiffServ para proporcionar garantías de QoS, la cual ha sido simulada utilizando la herramienta NS-2.

Las variantes de TCP seleccionadas se evalúan considerando diferentes parámetros de desempeño como: el nivel de rendimiento conocido como *goodput* (por sus siglas en inglés), el raparto equitativo de ancho de banda conocido como *fairness* y la capacidad de convivencia del TCP con otras variantes de TCP conocido como *friendliness*. Los resultados de la simulación nos permiten seleccionar de manera preliminar la variante de TCP que presenta un mejor rendimiento trabajando en un sistema satelital DVB-S2 con el soporte de QoS.

Una vez elegida la variante de TCP más adecuada para trabajar en un entorno satelital, se proponen escenarios de simulación típicos en entornos satelitales, que consideren por ejemplo variaciones en las tasas de tráfico y en el ancho de banda disponible. El objetivo de dichos escenarios es analizar el impacto del modelo de calidad de servicio propuesto cuando las variaciones de ancho de banda son experimentadas en el sistema satelital. Por último se propone un escenario en donde se comparan dos técnicas de asignación de recursos (*scheduling*) con el fin de elegir la técnica más adecuada y hacer frente a las variaciones constantes de ancho de banda presentes en los sistemas satelitales GEO.

El presente trabajo se organiza de la siguiente manera: En la sección 2 se presenta un resumen de las variantes de TCP seleccionadas. En la sección 3 se describen las características del entorno satelital DVB-S2 y el modelo de calidad de servicio propuesto, además de la descripción de los parámetros de evaluación de desempeño. En la sección 4, el análisis de las variantes de TCP seleccionadas se realiza mediante el simulador NS-2 con el fin de elegir la variante de TCP que presenta un mejor desempeño en el escenario propuesto. En la sección 5 los resultados de simulación que consideran variaciones de tráfico y de ancho de banda presentes en un ambiente satelital son analizados. Finalmente, en la Sección 6 las conclusiones se presentan así como los trabajos futuros.

RESUMEN DE LAS VARIANTES DE TCP UTILIZADAS

En esta sección se describen de manera general los principales problemas que enfrenta el protocolo TCP para el transporte de datos en una red satelital. Adicionalmente, se presentan las variantes de TCP seleccionadas para realizar la simulaciones en un entorno satelital con soporte de QoS mediante DiffServ.

Las características de un canal satelital que afectan el desempeño del protocolo TCP son principalmente: el retardo de propagación que genera un incremento en el tiempo de ida y vuelta (RTT) de los paquetes, los errores de transmisión que generan una alta tasa de error (BER) en los paquetes y la presencia de las variaciones de ancho de banda causados por las condiciones atmosféricas por ejemplo los eventos de lluvia.

Una gran cantidad de variantes de TCP han sido diseñadas para contrarrestar estos problemas, sin embargo en la mayoría de los casos la ingeniería del diseño se ha enfocado en las características particulares definidas para un determinado tipo de red. Estas variantes de TCP pueden ser clasificadas en cuatro grupos. La primera clasificación inicia con el conjunto de variantes de TCP definidas para atender las necesidades generales. Algunos ejemplos de estas versiones consideradas como "clásicas" son: New Reno y Sack. La segunda categoría está representada por las variantes de TCP diseñadas para trabajar en redes de comunicación inalámbrica, tales como TCP Westwood y TCP Eifel. Las variantes de TCP desarrolladas para redes LFN (Long Fat Networks) son: Bic, Cubic, Speed, Hamilton, Scalable (STCP) y High Speed (HSTCP). La última categoría se propone para redes inalámbricas LFN [5], en donde las redes satelitales se encuentran clasificadas, definiéndose el uso de los TCP Hybla y Peach.

Aunque, el análisis de las variantes de TCP sobre un sistema satelital ha sido estudiado por varios autores [6], [7] y [8], en el mejor de nuestro conocimiento, ningún de estos trabajos se ha enfocado a la evaluación de las variantes de TCP considerando una arquitectura con el soporte de QoS.

En el presente trabajo proponemos evaluar el desempeño de un grupo selecto de variantes de TCP considerando dos aspectos primordiales: los problemas asociados los cambios dinámicos del canal satelital DVB-S2 y el soporte de la QoS a través del marco de referencia DiffServ.

Para lograr esto, se eligen tres variantes de TCP: TCP Sack [9], representando la categoría de las variantes de TCP clasicas. TCP Cubic [10], el cual se caracteriza por ser la versión de TCP seleccionada para trabajar sobre entornos Linux, debido a su rápida respuesta en presencia de eventos de congestión, y finalmente TCP Hybla [11] que se caracteriza por ser diseñado exclusivamente para entornos satelitales. A continuación se presenta una visión general de las variantes de TCP seleccionadas:

El TCP Sack está diseñado siguiendo el mecanismo de control de congestión clásico definido por Van Jacobson. También añade un mecanismo de confirmación de recepción selectiva (SACK) que permite al emisor tener información sobre de los segmentos recibidos, generando así la retransmisión únicamente de los segmentos que se han perdido.

El TCP Cubic está basado en su predecesor, el TCP BIC (Binary Increase Congestion control), y se caracteriza por calcular el crecimiento de su ventana de congestión considerando una función cúbica. El principal objetivo a cubrir en el diseño de TCP Cubic fue lograr independencia al retardo. Como resultado de esto, TCP Cubic muestra un desempeño mejorado cuando se emplea en redes satelitales.

Finalmente, TCP Hybla está diseñado para contrarrestar el deterioro en el desempeño del TCP provocado por los largos RTT presentes en los enlaces satelitales. TCP Hybla se

compone de ciertos procedimientos que mejoran el desempeño de su algoritmo de control de congestión. Dichos procedimientos son: la opción SACK obligatoria, la estimación del ancho de banda del canal y el uso de marcas de tiempo (timestamps).

ESCENARIO DE SIMULACIÓN EN EL ENTORNO SATELITAL DVB-S2

En esta sección se describe el entorno de simulación de una red satelital DVB-S2. Así mismo se presenta una breve descripción de los parámetros de evaluación de desempeño con los que se comparará la eficiencia de las variantes de TCP seleccionadas. Adicionalmente, se presenta el modelo de calidad de servicio basado en la arquitectura DiffServ.

El entorno de simulación evaluado considera un sistema satelital DVB-S2 con el soporte de la arquitectura DiffServ detallado en [12]. Dicho entorno se desarrolla utilizando la herramienta de simulación NS-2 [13]. El modelo de calidad de servicio sigue las especificaciones definidas en [14].

La Fig. 1 muestra las características de la gestión activa de colas (AQM) utilizada en la simulación. En esta figura se observan tres fuentes de datos enviando información, mediante una red satelital, a un destino remoto.

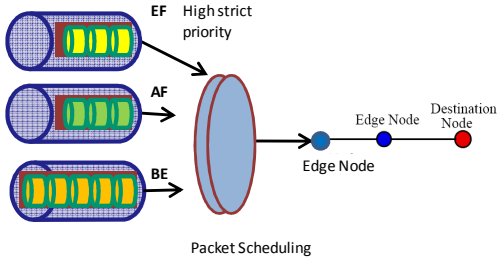


Fig. 1. El modelo de QoS

En este escenario, cada fuente de datos tiene un PHB asociado, considerándose tres clases de servicio: EF, AF y BE. La clase de servicio EF está representada por un servidor de Voz sobre IP (VoIP). Este tipo de trafico es transportado por el protocolo UDP, con el fin de reservar y garantizar de manera estricta el ancho de banda solicitado. La clase de servicio AF tiene como aplicación asociada un servidor con transferencia de archivos (FTP). Esta clase de tráfico tiene mayor nivel de prioridad que la clase de tráfico BE. Por último, la clase de servicio BE también tiene asociado un servidor de transacciones FTP, sin embargo, esta clase de tráfico no tiene asignada garantías de QoS. En otras palabras, la clase BE será capaz de utilizar el ancho de banda restante siempre que las clases de alta prioridad no utilicen dichos recursos. Cada clase de servicio tiene asociada una cola física de longitud fija definida por la operación resultado del producto Retardo - Ancho de banda (BDP). Los parámetros utilizados en el ambiente de simulación propuesto son los siguientes:

- La capacidad de canal satelital se considera como el cuello de botella y se fija a 2Mbps.
- Las simulaciones se llevan a cabo considerando que el 20% de la reserva de ancho de banda se asigna al tráfico EF, el 40% se asigna al tráfico AF y el restante 40% para el tráfico BE. De esta manera, las velocidades de transmisión se establecen considerando 400 Kbps para la clase EF y 800 Kbps para cada una las clases AF y BE.
- Cada clase de servicio tiene su propia cola física con una longitud fija a 90 paquetes, igual al BDP.
- La tasa de error de paquete (PER) configurada es el valor típico presente en redes satelitales igual a 1*10^-7
- El mecanismo configurado para descartar paquetes cuando los buffers se sobrecargan es el mecanismo llamado d*rop tail.*
- En la simulación se agrega un *token bucket* como limitador de trafico para cada una las clases de servicio con alta prioridad (EF y AF). Este elemento se agrega con el fin de descartar los paquetes en el caso de que la tasas de transmisión no estén de acuerdo con los valores previamente definidos en el SLA.
- En el caso especifico de la clase de servicio AF, estos paquetes no serán descartados, por el contrario, se propone enviarlos a la cola de BE, de menor prioridad.

Los parámetros de evaluación de desempeño utilizados para analizar los resultados en el escenario satelital DVB-S2 con el soporte de QoS propuesto son: el nivel de *goodput*, el nivel de *fairness* y el nivel de *friendliness,* los cuales se detallan a continuación:

- El nivel de g*oodput* es similar al rendimiento o *througput* del TCP (tasa de salida alcanzado en el sistema). En este caso particular, el *goodput* sólo toma en cuenta los datos útiles que le llegan al receptor sin considerar la tasa de retransmisión. Sin embargo se incluyen los retardos asociados a las retransmisiones y al tiempo de espera (timeout).
- Para nuestro caso especifico, el nivel de *fairness* se evaluará como la capacidad que tiene una variante de TCP, de compartir el mismo ancho de banda, de manera equitativa, entre todas las conexiones que empleen la misma versión de TCP en un enlace determinado.
- Similarmente, en nuestro caso específico, el nivel de *friendliness* se evaluará como la capacidad que tiene una variante de TCP, de convivir y compartir el ancho de banda disponible entre todas las conexiones que empleen diferentes versiones de TCP en un enlace determinado.

EVALUACIÓN DEL DESEMPEÑO DE LAS VARIANTES DE TCP

En esta sección se presentan los resultados de simulación obtenidos considerando la arquitectura satelital DVB-S2 propuesta y el modelo de DiffServ. Con el fin de realizar el análisis de las variantes de TCP seleccionadas, se proponen los siguientes escenarios:

El *escenario 1* se define con el objetivo de seleccionar la variante de TCP que muestre el mejor desempeño en el ambiente satelital DVB-S2 propuesto. Especialmente, en este escenario, se propone evaluar los efectos de la adopción de la arquitectura DiffServ para proporcionar servicios con QoS.

El *escenario 2* se propone con el fin de evaluar el modelo de QoS propuesto, teniendo en cuenta las variaciones en las tasas de tráfico y en el ancho de banda, que son situaciones comunes en un canal satelital DVB-S2. Adicionalmente, se propone evaluar la importancia del esquema de asignación de

recursos (*scheduler*) a nivel IP frente a dichas variaciones de ancho de banda.

Escenario 1: Resultados de simulación

Este escenario considera los tres tipos de tráfico (EF, AF y BE) y las tres variantes de TCP (Sack, Cubic y Hybla) previamente definidas. Adicionalmente, se considera el modelo de QoS basado en la arquitectura DiffServ detallado en la figura 1.

El comparativo de las variantes de TCP se lleva a cabo entre las clases de servicio AF y BE, teniendo en cuenta los parámetros de evaluación de desempeño *goodput*, *fairness* y *frienliness* previamente definidos.

Es importante mencionar, que debido a que la tasa de transmisión de la clase de servicio EF es constante (ya que es transportada usando el protocolo UDP para garantizar su estricta reserva de ancho de banda), esta clase no se mostrara en las gráficas obtenidas de las simulaciones, con el fin de simplificar el comparativo entre las clases de servicio AF y BE que emplean el protocolo TCP.

Como primer comparativo, la Tabla I muestra los resultados numéricos de la simulación considerando diferentes variantes de TCP trabajando en conjunto, ya sea para la clase AF o para la clase BE. En particular, la Tabla I muestra el número de paquetes recibidos considerando la interacción entre las clases AF y BE.

Num. Prueba	Variante de TCP usada para la clase AF	Variante de TCP usada para la clase BE	Total de paquetes AF recibidos	Total de paquetes BE recibidos	Total de paquetes recibidos (incluye EF)
1ª	SACK	SACK	99197	100115	250682
1b	CUBIC	CUBIC	99909	99113	250392
1c	HYBLA	HYBLA	87619	67266	206255
2a	SACK	CUBIC	99533	100044	250947
2b	CUBIC	SACK	99885	98253	249508
3a	SACK	HYBLA	99117	100779	251266
3b	HYBLA	SACK	106812	20359	178541
3c	CUBIC	HYBLA	99904	99492	250766
3d	HYBLA	CUBIC	93616	56510	201496

Tabla 1. Número de paquetes recibidos considerando la interacción entre las clases de servicio AF y BE.

Los resultados mostrados en la Tabla 1, nos permiten observar que, cuando la variante de TCP empleada es la misma tanto para la clase AF como para la clase BE (número de prueba: 1a, 1b y 1c), el ancho de banda compartido tiene un nivel de *fairness* aceptable (ya que se observa un número de paquetes recibidos muy parecido en ambos tipos de tráfico). También se observa que cuando los TCPs Sack y Cubic comparten el ancho de banda (número de prueba: 2a y 2b), estas dos variantes de TCP muestran un nivel adecuado de *friendliness*, ya que el número de paquetes recibidos son prácticamente similares en ambos casos.

Finalmente, es importante mencionar, que para los casos en que se utiliza la variante TCP Hybla, el total de paquetes recibidos para la clase BE es bastante desproporcionado en comparación con los casos anteriores (número de prueba: 3b y 3d). Esta comportamiento se debe principalmente a que el TCP Hybla trata de enviar una gran cantidad de paquetes para garantizar la clase AF, lo que genera un comportamiento poco equitativo, reduciendo su nivel de *friendliness* al compartir el ancho de banda con otras variantes de TCP. Esta diferencia es muy notoria cuando el TCP Hybla es empleado para el transporte de la clase AF. Aquí se puede observar que la clase de mayor prioridad AF limita el desempeño de la clase de menor prioridad BE.

Estos resultados son causados principalmente por la política de encolamiento que se ha implementado. Esta política, definida para nuestro caso especifico, establece que los paquetes de la clase de servicio AF que no cumplan con los requerimientos del SLA no sean descartados si no por el contrario, sean enviados a la cola de BE, de menor prioridad. Esta situación afecta considerablemente el desempeño de la variante de TCP Hybla.

Tomando en cuenta este mismo escenario de simulación se obtienen los resultados de la evolución de la ventana de congestión en función del tiempo, como se muestra en la Fig. 2. En este caso se consideran las variantes de TCP Sack y Cubic compartiendo el canal satelital. La simulación se realiza con el fin de comparar las ventanas de congestión en cuando el modelo de QoS propuesto es adoptado y cuando no hay garantías de QoS. El gráfico de la izquierda (Fig. 2) muestra el escenario que no considera la arquitectura DiffServ, mientras que el gráfico de la derecha muestra el escenario que si considera la arquitectura DiffServ.

Observando la grafica derecha de la Fig. 2, cuando el modelo DiffServ es adoptado, los diferentes niveles de prioridad por clase de servicio se mantienen, permitiendo que ambas ventanas de congestión alcancen sus valores determinados. La evolución de la ventana de congestión de la AF (usando TCP Sack) es capaz de alcanzar valores mayores a 100 paquetes, mientras que la ventana de congestión de la clase BE (usando TCP Cubic) alcanza valores mayores a 130 paquetes. El clásico "*diente de sierra*", característico de TCP Sack también se puede observar. Con este resultado se confirma el nivel adecuado de *frienliness* que experimentan los TCPs Sack y Cubic al compartir el ancho de banda disponible en el enlace satelital.

Por el contrario, cuando la arquitectura DiffServ no es empleada (Fig 2 - gráfico izquierdo), los niveles de prioridad de las clases de servicio no son respetados. Esta situación {on resulta es un resultado esperado al no considerar el modelo con garantías de QoS. Como se observa en esta gráfica el TCP Cubic (transportando la clase BE) utiliza el ancho de banda disponible casi es su totalidad, alcanzando valores de más de 150 paquetes en su ventana de congestión, mientras que el TCP Sack (transportando la clase AF) es completamente penalizado. Esta situación resulta no deseable en un modelo con QoS ya que BE (que es la clase con ninguna prioridad) obtiene el mayor ancho de banda.

Finalmente, y con el fin de evaluar el comportamiento del TCP Hybla, se realiza la misma prueba pero ahora evaluando el nivel de *goodput.* Los resultados se muestran en la gráfica 3. En este caso, se propone comparar TCP Hybla y Cubic cuando se usan ambas para el transporte de la clase de servicio AF, mientras que la clase BE será transportada usando TCP Sack.

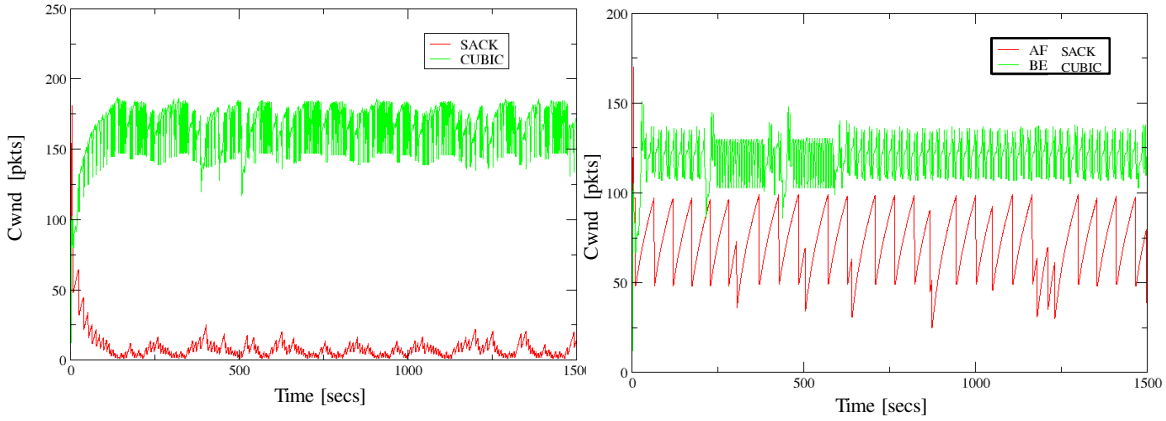


Fig. 2. Evolución de la ventana de congestión en el tiempo considerando la arquitectura DiffServ (gráfico derecho) y sin garantías de QoS (gráfico izquierdo). Las variantes usadas son TCP Sack para transportar la clase AF y TCP Cubic para transportar la clase BE.

La Fig. 3 muestra el nivel de goodput alcanzado por ambas variantes de TCP considerando la arquitectura DiffServ. El gráfico de la izquierda muestra la interacción entre TCP Cubic y Sack empleados para transportar la clase AF y BE respectivamente. Como se observa, cuando estas dos variantes de TCP trabajan simultáneamente, ambas son capaces de alcanzar sus 800 Kbps de *goodput,* compartiendo equitativamente el ancho de banda disponible mostrando un nivel aceptable de *friendliness*.

Es importante mencionar que se han realizado varias pruebas de simulación (en este articulo solo se muestran las más representativas) que nos permiten confirmar que la combinación de los TCPs Sack y Cubic no sólo muestran un nivel aceptable de *friendliness* sino también, su capacidad destacada de compartir equitativamente el ancho de banda disponible (nivel de *fairness*).

Por el contrario, como se puede observa en la Figura 3 (derecha), la interacción entre los TCPs Hybla y Sack (cuando son utilizados para el transporte de las clases de servicio AF y BE respectivamente), muestran su capacidad limitada de compartir de manera equitativa el ancho de banda disponible. Ya que el TCP Hybla trata de enviar el mayor numero de paquetes disponibles penalizando completamente al TCP Sack. Con ello se puede confirmar el comportamiento agresivo que muestra TCP Hybla al trabajar en conjunto con el modelo de DiffServ propuesto. Este resultado lo atribuimos principalmente al uso de la política de encolado previamente definida para la clase AF.

Lo que se puede concluir en este primer conjunto de simulaciones (escenario 1) es que cuando se considera la arquitectura DiffServ, las variantes de TCP Sack y Cubic presentan mejores resultados en cuanto a su nivel de *friendliness* y *fairness* en comparación con el TCP Hybla.

Dado que en la siguiente prueba se considera un escenario satelital DVB-S2 con variaciones en las tasas de tráfico y en el ancho de banda disponible, es importante seleccionar la variante de TCP más adecuada para trabajar en este entorno. Sin embargo, a pesar de que en nuestras simulaciones el TCP Sack muestra un comportamiento similar en comparación con el TCP Cubic trabajando sobre una arquitectura DiffServ, hemos decidido trabajar sólo con el TCP Cubic para transportar ambos tipos de tráfico (AF y BE). Esta decisión se respalda con el trabajo desarrollado en [8], en donde se emplea la variante TCP BIC (que es la versión predecesora de TCP Cubic) en un sistema satelital sin soporte de QoS, mostrando mejores resultados en comparación con otras versiones de TCP.

Escenario 2: Resultados de la simulación

En este escenario se evalúa el modelo de calidad de servicio (definido previamente) teniendo en cuenta las variaciones de tráfico y de ancho de banda comunes en un sistema satelital DVB-S2. Adicionalmente, en este escenario se considera dos TCPs Cubic para el transporte de las clases de servicio AF y BE (basado en los resultados demostrados en el escenario 1).

De acuerdo con el modelo de calidad de servicio propuesto (Fig.1), la clase EF debe garantizarse con alta prioridad sobre las clases AF y BE. De la misma forma, se define que la clase de servicio AF debe garantizarse, al tener mayor prioridad que la clase BE. Finalmente, la clase BE deberá ser capaz de emplear el ancho de banda restante (es decir el ancho de banda que las clases de alta prioridad no utilicen).

La prueba de simulación del escenario 2 considera que las tres clases de tráfico (EF, AF y BE) están presentes en diferentes instantes de tiempo durante la transmisión. La simulación inicia cuando sólo una clase de servicio envía información (ya sea la clase AF o la BE), con el objetivo de utilizar en su totalidad el ancho de banda disponible (2Mbps).

Una vez transcurridos los 500 segundos de simulación, la clase de servicio EF (con una tasa de transmisión de 400 Kbps) es agregada en conjunto con la otra clase de servicio faltante (ya sea la clase BE o AF). Una vez alcanzados los 1000 segundos se regresa al estado original con una sola clase de servicio.

Los resultados de los niveles de *goodput* alcanzados en presencia de variaciones de tráfico se muestran en la Figura 4. El gráfico de la derecha muestra el escenario iniciando la transmisión con la clase de servicio AF. Por otro lado el gráfico de la izquierda muestra el escenario iniciando con la clase de servicio BE. En ambos casos, la clase de tráfico EF está presente en el intervalo de 500 y 1000 segundos.

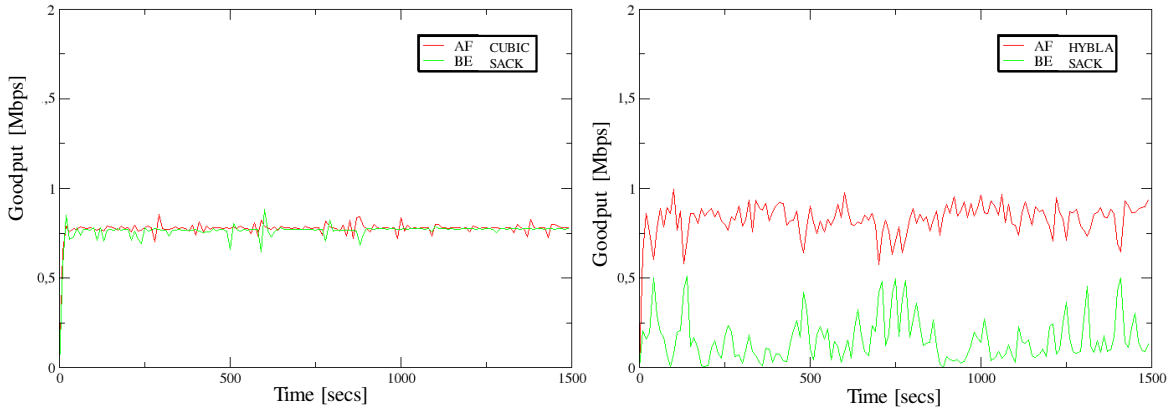


Fig. 3. Resultados del nivel de goodput alcanzado considerando la arquitectura DiffServ. Las variantes usadas para transportar la clase AF son TCP Cubic (gráfico izquierdo) y TCP Hybla (gráfico derecho). En ambos casos se emplea la versión de TCP Sack para transportar la clase BE.

Como se puede observar en ambos gráficos (Fig. 4), cuando se inicia la simulación, el ancho de banda disponible es utilizado en su totalidad por la variante de TCP seleccionada. En particular, en el gráfico de la izquierda se observa que el modelo de QoS propuesto mejora el nivel de *goodput* de la clase de servicio BE, permitiéndole utilizar todo el ancho de banda disponible (2 Mbps) cuando ninguna de las clases de alta prioridad (EF y AF) están presentes. Adicionalmente, en esta misma grafica se observa, que al agregar las clases EF y AF (transcurridos los primeros 500 segundos), el *goodput* de la clase BE se reduce considerablemente con el fin de liberar el ancho de banda requerido por las clases de mayor prioridad.

Similarmente en el gráfico de la derecha (Fig. 4) se muestra la clase de servicio AF iniciado la transmisión. Después de transcurridos los primeros 500 segundos, las clases EF y BE son inyectadas. En este caso es posible observar como las clases AF y BE reducen el ancho de banda utilizado, con el fin de dar prioridad y garantizar la clase de más alta prioridad, en este caso la clase EF. Este resultado es un resultado esperado teniendo en cuenta el soporte de QoS con la arquitectura DiffServ.

Con los resultados de esta prueba, hemos visto que el modelo DiffServ propuesto alcanza un nivel adecuado de adaptación frente a las variaciones del tráfico presentes en un sistema satelital. Del mismo modo, se ha mostrado que la clase de servicio BE es capaz de aprovechar el ancho de banda disponible que no está siendo utilizado por las clases de mayor prioridad (AF y EF).

Finalmente, para analizar la respuesta del modelo DiffServ propuesto frente a las variaciones de ancho de banda, se propone simular un evento de lluvia como una de las variaciones de ancho de banda más comunes en los sistemas satelitales. Para simular dicho evento se propone realizar una aproximación, mediante una señal senoidal, del evento de lluvia caracterizado por los autores en [15].

La onda senoidal propuesta oscilará entre 1,2 y 2,8 Mbps. También se especifica que para los valores superiores a 1600 Kbps se considerará que el sistema satelital experimenta condiciones de cielo despejado, es decir que el ancho de banda estará completamente disponible. Por otro lado para valores inferiores a 1200 Kbps se considerará que el sistema satelital experimenta un fuerte evento de lluvia reduciendo la disponibilidad de ancho de banda.

En este contexto, se evaluaran dos esquemas de asignación de recursos (*scheduling*): la primera simulación se desarrolla utilizando el esquema denominado Round Robin scheme (RR), mientras que la segunda simulación se efectúa empleando el mecanismo Weighted Round Robin (WRR).

Los resultados de esta evaluación se muestran en la Figura 5. En particular, en esta grafica se observa los niveles de *goodput* alcanzados considerando el canal satelital DVB-S2 con la adopción de la arquitectura DiffServ frente a las variaciones de ancho de banda provocadas por un evento de lluvia. Es importante recordar que los tres tipos de tráfico (EF, AF y BE) están presentes en esta prueba y que la variante de TCP usada para el transporte de AF y BE es la variante TCP Cubic.

En la Figura 5 (gráfico izquierdo) se muestra los resultados de la simulación cuando el mecanismo de asignación de recursos empleado es el RR. En esta figura, se puede observar que cuando el ancho de banda disponible aumenta (ver la onda sinodal arriba de 2 Mbps), la clase de servicio AF mantiene su nivel de prioridad de acuerdo a la definición del modelo de QoS (donde la clase AF tiene mayor prioridad sobre la clase BE). Adicionalmente, la clase de servicio EF es completamente garantizada (por simplicidad los resultados de EF no se muestran en la gráfica).

En el caso contrario, cuando el ancho de banda disponible se reduce (ver la onda senoidal en los 1200 Mbps), el mecanismo de asignación de recursos RR no es capaz de garantizar los niveles de prioridad de las clase EF y AF. Ya que, tanto la clase AF como la BE comparten el mismo ancho de banda, alcanzando los 400 Kbps. Este comportamiento se debe principalmente a que el mecanismo RR asigna el mismo valor de pesos (1, 1, 1) sin importar la clase de servicio que este atendiendo, dando como resultado que los niveles de prioridad sean omitidos completamente.

Finalmente, en la Figura 5 (gráfico derecho) se muestran los resultados cuando el mecanismo de asignación de recursos WRR es empleado. En este caso los valores especificados para los pesos de la clase EF, AF y BE son 7, 7 y 1 respectivamente. Estos valores se definen considerando la especificación [16], con el fin de dar mayor ponderación a las clases de más alta prioridad (EF y AF dándole el mismo valor alto de ponderación) y así diferenciarlas de la clase de menor prioridad (BE dándole el menor valor de ponderación).

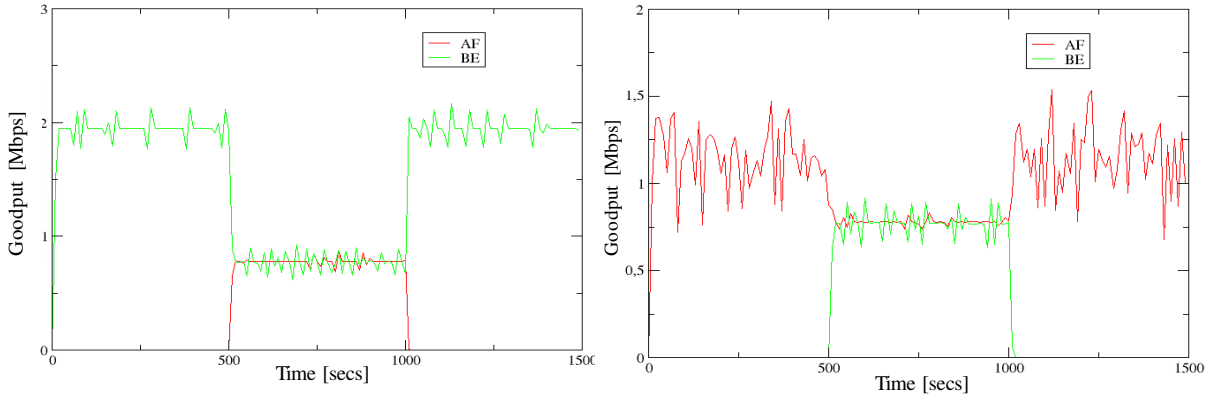


Fig. 4. Resultados del nivel de *goodput* alcanzado considerando sólo la variante TCP Cubic. La transmisión inician con la clase BE (gráfico izquierdo), y con la clase AF (gráfico derecho). En ambos casos la clase EF se activa sólo entre los 500 y 1000 segundos.

Como se observa en la Figura 5 (gráfico derecho), la propuesta de modelo DiffServ en conjunto con el mecanismo de asignación de recursos WRR, permite a ambas clases de tráfico (AF y BE) compartir equitativamente el ancho de banda disponible (alcanzando sus 800 Kbps de *goodput*), mientras que la clase de servicio EF es totalmente garantizada.

Con estos resultados se puede concluir que las clases con mayor nivel de prioridad (EF y AF) son totalmente garantizadas aún cuando la capacidad del sistema es afectada seriamente por presencia de un evento de lluvia. Adicionalmente, la clase de tráfico AF mantiene su utilización de ancho de banda con un valor constante (800 Kpbs) frente al evento de lluvia simulado, mientras que la clase BE se adapta completamente a la onda senoidal, siendo capaz de aprovechar el ancho de banda restante que las clases de alta prioridad no utilizan.

CONCLUSIONES

En este artículo se ha evaluado la implementación de un modelo de QoS, basado en la arquitectura DiffServ, bajo un entorno satelital DVB-S2.

Se han propuesto diferentes escenarios de simulación con el fin de evaluar diferentes variantes de TCP sobre el sistema satelital con QoS. Las variantes de TCP analizadas han sido Sack, Cubic y Hybla.

Los resultados nos permiten concluir que la variante de TCP que ha demostrado una mayor capacidad de adaptación en un contexto satelital DVB-S2 con QoS, es la variante: TCP Cubic. Al emplear esta variante de TCP se alcanza un nivel mejorado de *goodput, friendliness* y *fairne*s en comparación con el alcanzado por TCP Hybla cuando la transmisión de información es extremo-a-extremo.

El TCP Hybla mostró un desempeño extremadamente agresivo al trabajar con el modelo de DiffServ propuesto, ya que el mecanismo de encolado definido afecta severamente el desempeño de este TCP. Como medida preventiva, el establecimiento de mecanismos de doble encolado se debe analizar detalladamente, ya que algunas variantes de TCP como Hybla son afectadas severamente, impactando directamente el desempeño de las clases de servicio con las que comparte el enlace.

Finalmente, el mecanismo de WRR con valores de ponderación fijos (priorizando las clases con mayor nivel) resulto el más adecuado con el fin de garantizar los niveles de prioridad de la clase EF y AF, cuando en el sistema satelital experimenta una reducción de ancho de banda provocado por un evento de lluvia. El trabajo futuro estará enfocado al estudio de la arquitectura DiffServ en un entorno satelital considerando el algoritmo Random Early Detection (RED).

AGRADECIMIENTOS

Este trabajo está financiado por el Gobierno de España mediante el Ministerio de Ciencia y Educación bajo los proyectos CONSOLIDER-ARES (CSD2007-00004), ITACA (TSI2007-65393-C02-02) y P2PSEC (TEC 2008-06663-C03-01). E. Rendón agradece el apoyo de FPI-UPC para la realización de la tesis doctoral.

REFERENCIAS

[1] J. Scott Marcus, Dieter Eichmann, “The future of IP connexion”, Study for the European Commission, 2008.

[2] M. Allman, D. Glover, L. Sanchez,. “RFC 2488 - Enhancing TCP Over Satellite Channels using Standard Mechanisms”. January 1999

[3] Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2). Specification ETSI: EN 302 307

[4] S. Blake, D. Black, M. Carlson, “An Architecture for Differentiated Services”. RFC 2475, December 1998.

[5] Yee-Ting Li, Douglas Leith and Robert N. Shorten. “Experimental Evaluation of TCP Protocols for. High-Speed Networks.

[6] Juan Carlos Nattero Valentin. “PFC- Estudio de la conexión TCP/IP a través de un satélite geoestacionario”. Universidad Politecnica de Cataluña. 2006.

[7] Carlo Caini, Rosario Firrincieli, Mario Marchese, Tomaso de Cola, Michele Luglio, Cesare Roseti, Nedo Celandroni, Francesco Potortí. “Transport layer protocols and architectures for satellite networks”. International Journal of Satellite Communications and Networking. October 2006

[8] C. Caini, , R. Firrincieli, , , D. Lacamera, , T. de Cola, , M. Marchese, , C. Marcondes, , M.Y. Sanadidi, and M. Gerlad, “Analysis of TCP live experiments on areal GEO satellite testbed”. 2009

[9] E. Blanton, M. Allman, K. Fall, L. Wang, RFC 3517 A Conservative Selective Acknowledgment (SACK)-based. Loss Recovery Algorithm for TCP

[10] Rhee, L. Xu, “CUBIC: A new TCP-friendly High Speed TCP variant”, 2005

[11] C. Caini, R. Firrincieli,. TCP Hybla: a TCP enhancement for heterogeneous network. 2004

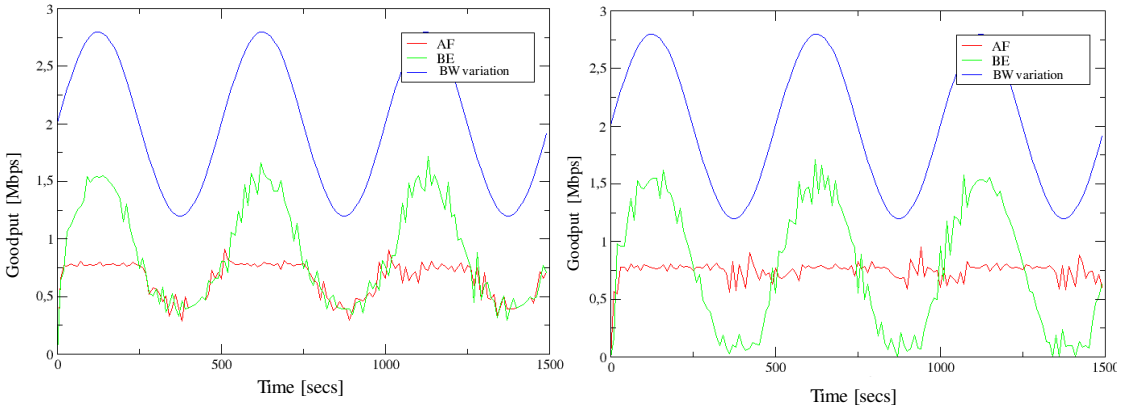


Fig 5. Resultados del nivel de *goodput* alcanzado frente a la presencia de un evento de lluvia. Los mecanismos de asignación de recursos empleados son RR (gráfico izquierdo) y WRR (gráfico derecho).

[12] R. J. Mort, S. Combes , H. Cruickshank. QoS architecture for Broadband Satellite Multimedia (BSM) networks.

[13] P.Pieda, J.Ethridge, M. Bainess, F. Shallwani A Network Simulator Differentiated Services Implementation. 2000

[14] B. Mathus, V.Robert, "La qualité, de service dans les réseaux. Étude de Differentiated Services". 2004

[15] S. Cioni, R.Gaudency, and R.Rinaldo."Channel estimation a physical layer adaptation techniques for satellite networks exploiting adaptive coding and modulation". International Journal of Satellite communication and networking. 26(2): 157-188, 2008.

[16] V. Jacobson, K. Nichols, Bay Networks, "RFC 2598-An Expedited Forwarding PHB" June 1998.

Análisis del comportamiento de TCP sobre modelos de canal a *ráfagas*

Roberto Fernández-Cueto, Ramón Agüero, Marta García-Arranz, Luis Muñoz
Departamento de Ingeniería de Comunicaciones
Universidad de Cantabria
Plaza de la Ciencia s/n, 39005 Santander
{ramon, marta, luis}@tlmat.unican.es

***Abstract*—La presencia de ráfagas de errores, que pueden ser comunes en entornos de propagación en interiores, tiene un efecto dramático sobre el rendimiento de los protocolos de capas superiores, especialmente si se trata de TCP. La mayoría de los modelos de canal que se utilizan para simular este tipo de entornos de propagación no ofrecen un comportamiento muy cercano al real. Frente a ellos, ha adquirido un creciente interés la utilización de Cadenas de Markov Ocultas; sin embargo, no existen muchos trabajos que analicen las prestaciones que ofrece esta alternativa cuando se utilizan con el protocolo TCP. El trabajo se centra en este aspecto, comparando el comportamiento de este tipo de modelos de canal (que se configuran en base a un conjunto de medidas reales) con una propuesta diferente, que se basa en un modelado de la relación señal a ruido mediante un filtro auto-regresivo, y que también es capaz de reflejar el comportamiento *a ráfagas* que caracteriza los canales reales.**

I. Introducción

La proliferación de las tecnologías de comunicaciones inalámbricas ha propiciado un sustancial incremento de la investigación sobre su rendimiento y operación bajo diferentes supuestos. A pesar de que la experimentación real aporta una serie de ventajas muy importantes, tiene asimismo algunos inconvenientes (como el de la escalabilidad), lo que deriva en una necesidad, cada vez mayor, de llevar a cabo análisis y estudios mediante técnicas de simulación.

Tradicionalmente uno de los principales puntos en contra de la simulación viene dado por la escasa fiabilidad de los modelos de propagación que se emplean. A pesar de que existen mecanismos que pueden aportar una gran precisión (modelado de rayos, técnicas electromagnéticas, etc.), éstos suelen requirir de unos tiempos de simulación muy elevados, lo que les incapacita para ser integrados en plataformas de simulación de redes, donde el objetivo principal se sitúa habitualmente en los protocolos, algoritmos y mecanismos de capas superiores. Se afronta, por tanto, el reto de dotar a los modelos de una precisión adecuada, sin que impongan un incremento sustancial del tiempo de simulación necesario.

Desde una perspectiva más concreta, es bien conocido que una de las limitaciones más relevantes en el comportamiento de los protocolos más habituales sobre redes inalámbricas (específicamente en entornos de interiores) es que los errores tienden a aparecer a ráfagas, lo que damnifica de manera notable el rendimiento que los protocolos de capas superiores pueden alcanzar. Esto es especialmente relevante para el caso del protocolo TCP, ya que su diseño se realizó para adaptarse adecuadamente a las principales fuentes de errores en redes tradicionales (saturación de nodos -*routers*- intermedios), por lo que su adaptación a otros escenarios no es muy apropiada.

En la literatura se han postulado diferentes aproximaciones para modelar el comportamiento de los canales inalámbricos. Uno de los más empleados (a pesar de que sus prestaciones son bastante limitadas) es el llamado *Gilbert-Elliot*, que se basa en una cadena de Markov de dos estados. Recientemente, se ha empezado a explorar las posibilidades de utilizar cadenas más complejas [1], denominadas *Hidden Markov Models* (HMM), aunque todavía no se ha analizado exhaustivamente el comportamiento de TCP sobre los mismos.

En este artículo se analiza de manera detallada el comportamiento del protocolo de transporte TCP utilizando modelos de canal *HMM*. Se parte de varias configuraciones que se han realizado a partir de un conjunto de trazas reales (obtenidas empíricamente) y se comparan las prestaciones de dicho modelo de canal con las de otra alternativa descrita en [2], y que mantiene una filosofía completamente diferente. El trabajo se ha estructurado tal y como sigue; la Sección II recoge los trabajos existentes en la literatura que guardan relación con éste, identificando las diferencias más importantes con ellos; la Sección III resume las características más importantes de los modelos de canal basados en *HMM*. La Sección IV presenta los resultados obtenidos en una exhaustiva campaña de medidas llevada a cabo en un escenario real, que servirán para corroborar el buen comportamiento de los diferentes modelos de canal analizados. La Sección V describe los principales resultados obtenidos, comparando las prestaciones del modelo de canal *HMM* con las de otras alternativas. Finalmente, la Sección VI concluye el artículo, presentando una serie de líneas de trabajo que quedan abiertas.

II. Antecedentes y estado del arte

Como ya se ha adelantado anteriormente, la gran actividad investigadora en el ámbito de las comunicaciones inalámbricas ha favorecido la aparición de numerosos trabajos que se basan en la utilización de diferentes herramientas de simulación. En ellos se pone de manifiesto, entre otros aspectos, la gran diversidad de modelos para los canales inalámbricos, así como la falta de precisión de algunos de los más empleados.

Con el principal objetivo de solventar algunos de los inconvenientes más importantes, en [3] se propuso el modelo de canal *Bursty Error channel model based on Auto-Regressive filter* (BEAR). Se trata de una alternativa que se centra en modelar adecuadamente la relación señal a ruido obtenida en la realidad, con el principal valor añadido de que se intenta reflejar la *memoria* que se observa en la realidad, a partir de un filtro AR. Se compara el rendimiento de BEAR con el de otras alternativas ampliamente empleadas en la literatura, como un

ISBN: 978-84-694-5948-5

modelo sencillo, que no incorpora ninguna memoria, y el tradicional *Gilbert-Elliot*. Se pone de manifiesto que, a pesar de su empleo masivo por parte de la comunidad científica, las prestaciones de algunos de los modelos de canal más empleados en la actualidad son limitadas. Sin embargo, en dicho artículo no se incorporan al análisis modelos de canal más complejos que podrían dar lugar a comportamientos más cercanos a los que se podrían observar en entornos reales.

Para profundizar en el análisis se llevó a cabo un primer estudio en el que se analizó el comportamiento de los modelos *HMM* utilizando tráfico UDP [4]. En dicho trabajo se comprueba la mejora que supone el incorporar varios estados a las cadenas de Markov, ya que consigue reflejar de manera muy precisa la estadísitica de ráfagas que se observaron empíricamente, proporcionando un comportamiento incluso mejor que el del modelo *BEAR* que, a pesar de todo, sigue ofreciendo un comportamiento apropiado. El trabajo que se presenta en este artículo complementa y extiende el anterior, ya que se centra en analizar la influencia adversa de las ráfagas de errores en el protocolo TCP.

El uso de cadenas de Markov (en el modelo del canal) para analizar el comportamiento de TCP es un aspecto que aparece en numerosas ocasiones en la literatura. Sin embargo, en la mayoría de las ocasiones se utiliza un modelo de dos estados (*Gilbert-Elliot*) [5], [6] que, como se ha visto anteriormente, no refleja de manera adecuada el comportamiento observado en la realidad. Además en varios de los trabajos existentes, la configuración de la cadena subyacente se realiza a nivel de trama [7], [1], a diferencia de la aproximación que se sigue en este trabajo, en el que la configuración se lleva a cabo temporalmente, con lo que se pretende reflejar, de manera más adecuada, la dinámica que podría darse sobre un canal real.

III. Modelo de canal HMM

Una cadena de Markov oculta es un modelo matemático no determinista, que se corresponde con sendos procesos estocásticos, uno relacionado con la transición entre estados y otro con la decisión que se toma en cada uno de ellos.

Para describir adeacuadamente el modelo se considera un sistema de N estados independientes entre sí: $\{S_1, S_2, \dots S_N\}$. La transición entre ellos se lleva a cabo mediante procesos estocásticos, a través de probabilidades de transición $a_{i,j}$, donde i y j se corresponden con los estados origen y destino, respectivamente; así, $a_{i,j}$ representa la probabilidad de cambio entre dos estados, considerándose asimismo el suceso de permanencia en el mismo estado $a_{i,i} \neq 0$. El segundo conjunto de parámetros que es necesario para definir completamente una cadena oculta de Markov son las probabilidades de decisión. En un HMM, las salidas observables no se corresponden directamente con el estado en el que se encuentre la cadena, sino que para cada uno de ellos se define las probabilidad de que la salida sea una de las que forman parte del conjunto de observables. Dichas probabilidades se definen como $b_i(k)$, donde i es el estado al que se asocia la probabilidad de decisión y k el observable correspondiente. Además, es necesario determinar asimismo las probabilidades iniciales de cada uno de los estados, representadas por las variables π_i, siendo i el índice del estado correspondiente.

Con todo lo anterior, una cadena oculta de Markov se define atendiendo a los siguientes elementos.

1) El número de estados del modelo, N.
2) El número de símbolos observables, M.
3) La matriz $\mathbf{A}$ de transición, de dimensión $N \times N$, que recoge las diferentes probabilidades de transición $a_{i,j}$.
4) La matriz $\mathbf{B}$ de decisión, con dimensión $N \times M$, con todas las combinaciones $b_i(k)$.
5) La distribución inicial de probabilidades para cada uno de los estados, $\mathbf{\Pi} = \{\pi_i\}$.

En la teoría clásica de las HMM se establecen tres problemas fundamentales relacionados con ellas [8]: (1) determinar la probabilidad de que se obtenga una secuencia dada de observables, a partir de un modelo completamente conocido; (2) establecer la secuencia de estados más plausible, en función de una serie de observables y conocido el modelo; (3) estimar de manera óptima los parámetros del modelo, únicamente a partir de una secuencia de observables.

Es bien conocido que el último de los problemas que se han planteado anteriormente es el que mayor complejidad conlleva. Un algoritmo que ofrece un comportamiento adecuado en su resolución es *Baum-Welch*. Como se ha adelantado previamente, para configurar las cadenas de Markov subyacentes en este trabajo se hará uso de un conjunto de trazas obtenidas en un entorno real, en la que un 0 se corresponde con una trama errónea y un 1 con una recepción correcta. A partir de dichos observables se estimarán los parámetros más apropiados del modelo de Markov oculto correspondiente[1].

Para dotar de una amplitud mayor al trabajo, se utilizarán dos instancias posibles del modelo *HMM*; en la primera se emplearán 4 estados, permitiendo la transición libre entre cualquier pareja de estados, mientras que en la segunda se incrementará el número de estados hasta $N = 16$, aunque únicamente se permitarán las transiciones entre estados consecutivos (proceso de nacimiento y muerte). En ambos casos, las salidas observables del HMM son sólo dos: un 0 si la trama es errónea, y un 1 si se corresponde con una recepción libre de errores.

Uno de los aspectos más importantes es que, a la hora de configurar el modelo, se utilizarán unidades temporales. Como se ha comentado previamente, en varios de los trabajos existentes en la literatura que hacen uso de cadenas de Markov se utilizan estadísticas a nivel de trama. Cardoso et al [1], por ejemplo, así lo hacen, manteniendo un tiempo constante de $5\ ms$ entre tramas consecutivas; si la aplicación tuviera una frecuencia de transmisión diferente, la configuración del modelo dejaría de ser válida. En el caso del protocolo TCP (que es en el que se centra este trabajo), la temporización de los segmentos depende fuertemente de la evolución de la conexión, por lo que se entiende que una configuración a nivel de tramas no sería correcta.

Para establecer la configuración temporal, se empleará la función densidad de probabilidad de la permanencia (en tiempo) en un estado. En las cadenas de Markov, dicha *fdp* es exponencial negativa: $f_{T_i}(t_i) = \lambda e^{-\lambda \cdot t_i}$, en la que $\overline{t}_i = \frac{1}{\lambda}$, es el tiempo medio de permanencia en el estado i. Para estimar $\overline{t}_i$, se calcula el número medio de tramas consecutivas para

[1]Para ejecutar el algoritmo de *Baum-Welch*, se ha empleado la función de Matlab `hmmtrain`, limitando el número de iteraciones a 1000.

cada estado ($\overline{N_i}$), multiplicando posteriormente dicho valor por la duración media de cada trama, ψ.

$$\overline{N_i} = \sum_{i=1}^{\infty} n p_i(n) = \sum_{i=1}^{\infty} n a_{i,i}^{n-1} (1 - a_{i,i}) = \frac{1}{1 - a_{i,i}} \quad (1)$$

donde $p_i(n)$ es la probabilidad de que el sistema haya estado en el estado i durante n instantes consecutivos, que se corresponde con que haya $n-1$ transiciones consecutivas al mismo estado i ($a_{i,i}^{n-1}$) y que en la *n-énesima* transición el sistema vaya a un estado diferente a i ($1 - a_{i,i}$).

Por tanto, se tiene finalmente que:

$$t_i = \psi \overline{N_i} = \frac{\psi}{1 - a_{i,i}} \quad (2)$$

Tal y como se ha mencionado, uno de los aspectos más relevantes (y novedosos) del enfoque que se sigue en este trabajo es que la configuración de las cadenas de Markov ocultas se lleva a cabo utilizando unidades temporales. Para ello, se necesita contar con una caracterización del canal real, sin existir dependencia alguna con la temporalización entre tramas consecutivas. Así, se parte de un conjunto de medidas realizadas con el protocolo de transporte UDP, en las que se satura el canal inalámbrico, asegurando que el transmisor siempre tiene datagramas para ser enviados. Como sucede con el caso del protocolo TCP, el canal inalámbrico real se caracteriza principalmente por tener un carácter muy variable, por lo que se seleccionaron un conjunto de tres medidas puntuales, que representaban todo el abanico de comportamientos (*BAD, AVeraGe, GOOD*) que se observaron en la realidad. A partir de dichas medidas, que se resumen en la Tabla I, se obtuvieron las configuraciones de los canales HMM. En [4] se pone de manifiesto que la elección de una configuración u otra tiene una influencia directa sobre las prestaciones (en aquel caso utilizando tráfico UDP) que se obtienen al usar dicho modelo. Como continuación y complemento de dicho artículo, se pretende analizar en este caso cuál es el impacto de dichas configuraciones en el comportamiento de TCP.

IV. Prestaciones de TCP sobre un canal real "a ráfagas"

Para poder corroborar la validez de la propuesta presentada en este trabajo, es fundamental disponer de un conocimiento profundo acerca del comportamiento real que se podría esperar sobre un escenario real. En este sentido, esta sección presenta un número de medidas que se realizaron sobre un canal de propagación en interiores, dentro de un entorno típico de oficinas, en el que los extremos de la comunicación estaban separados aproximadamente 15 m, con obstáculos metálicos y personas moviéndose libremente entre ambos. La tasa binaria de las tarjetas inalámbricas IEEE 802.11b se fijó a su valor máximo (11 Mbps). Además, el controlador de las mismas se modificó, para poder monitorizar la llegada tanto de tramas erróneas como correctas.

Tabla I
Medidas, obtenidas en un canal inalámbrico real con tráfico UDP, utilizadas para entrenar el modelo *HMM*

Canal	FER	PER	Tput_{UDP}	Ráfagas tramas error		
			Mbps	Media	Varianza	Máxima
Bad	0.517	0.179	2.33	6.22	983.66	821
Avg	0.331	0.058	3.58	2.60	79.53	258
Good	0.163	0.025	4.79	2.63	57.63	144

Sobre dicho canal de comunicaciones se llevaron a cabo 15 experimentos independientes, transmitiendo en cada caso un fichero de 10 MByte, utilizando el protocolo FTP. Se hizo uso de la versión Reno del protocolo TCP, con la opción de reconocimientos selectivos (SACK) activada en todas las conexiones. En cada una de las medidas se recogieron una serie de métricas para analizar el comportamiento del protocolo TCP y del canal inalámbrico.

- **Throughput.** Rendimiento que se alcanzó en cada uno de los experimentos; sobre un canal de propagación ideal (sin errores) las prestaciones de TCP se sitúan en torno a los 5 Mbps.
- **Tasa de error de tramas (FER).** Porcentaje de tramas erróneas que llegan al receptor, frente al total de tramas recibidas.
- **Tasa de error de paquetes (PER).** La tecnología IEEE 802.11b emplea un esquema de retransmisión a nivel MAC, de manera que un datagrama se transmite hasta en cuatro veces (en la configuración particular empleada durante la campaña de medidas) antes de descartarlo; en este sentido, para que un datagrama se pierda es necesario que se produzca, al menos, la recepción de cuatro tramas consecutivas con error[2].
- **ACK Duplicados**. Número de reconocimientos duplicados que llegan a la entidad TCP que transmite; de acuerdo a la especificación del propio protocolo, cada vez que se recibe un segmento fuera de orden, se envía automáticamente un reconocimiento.
- **Triple ACK.** Este supuesto tiene un interés especial, ya que la recepción de un ACK triplicado implica la inmediata retransmisión de un segmento, de acuerdo con el algoritmo *Fast Retransmit*.
- **Inactividad máxima.** Uno de los aspectos que en mayor medida perjudican el comportamiento del protocolo TCP es la presencia de periodos de inactividad en el transmisor. Teniendo en cuenta que fue originalmente diseñado para superar situaciones de congestión de elementos intermedios en la red, un transmisor TCP reduce la tasa a la que genera segmentos en el momento en el que detecta un comportamiento hostil del canal. Debido a los algoritmos que emplea, es posible que se den situaciones con una inactividad elevada, lo que conlleva una reducción relevante del rendimiento TCP.
- **Retransmisiones.** Número de segmentos que el transmisor TCP tiene que retransmitir, ya sea por recepción de un Triple ACK o tras la expiración del temporizador de retransmisión.
- **Número máximo de retransmisiones por segmento.** Se corresponde con el máximo número de veces que el mismo segmento debe ser retransmitido; habitualmente, un valor alto da lugar a la presencia de periodos de inactividad relevantes.

[2]Hay que tener en cuenta, sin embargo, que esta tasa de pérdida no se refiere, en ningún caso, a la que afecta a la aplicación, ya que el protocolo TCP emplea un esquema de retransmisión para recuperarse ante cualquier eventual pérdida de segmentos; la PER se refiere, de manera más correcta, a la tasa de pérdida a nivel IP.

Tabla II
COMPORTAMIENTO DEL PROTOCOLO TCP SOBRE UN ENLACE IEEE 802.11B CON ERRORES

#	*Tput* *Mbps*	*FER*	*PER*	*Dup ACK*	*Trip ACK*	*Max Inac*	*Rtx*	*Max Rtx*
1	4.85	0.025	0.000	0	0	0.0	0	0
2	4.36	0.052	0.000	45	1	0.2	1	1
3	3.67	0.105	0.016	277	17	0.8	138	3
4	3.55	0.186	0.023	341	30	0.6	177	3
5	3.50	0.090	0.015	303	21	1.7	120	4
6	3.23	0.153	0.013	415	36	2.2	103	4
7	3.17	0.143	0.011	313	23	2.1	84	4
8	2.86	0.171	0.000	0	0	0.8	1	1
9	2.43	0.318	0.022	920	108	1.9	185	5
10	2.39	0.255	0.029	577	61	1.3	217	5
11	2.23	0.279	0.033	811	99	2.1	278	5
12	1.31	0.212	0.038	474	39	8.0	321	7
13	1.19	0.292	0.041	586	68	8.3	314	6
14	0.67	0.360	0.034	732	103	39.7	264	9
15	0.55	0.418	0.071	1123	154	28.2	620	9

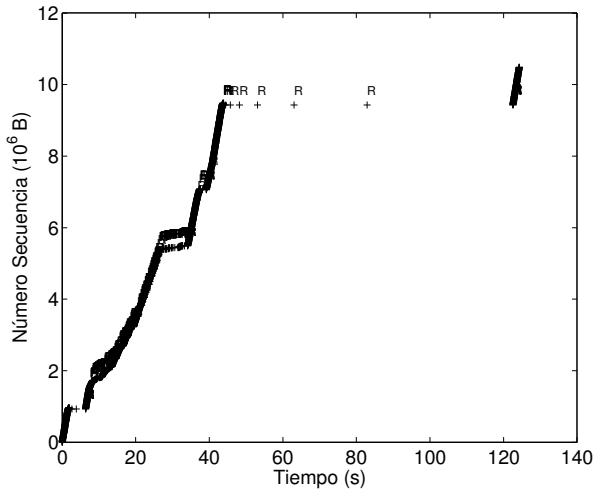


Fig. 1. Comportamiento temporal de una conexión TCP sobre un canal 802.11 con errores (medida #14)

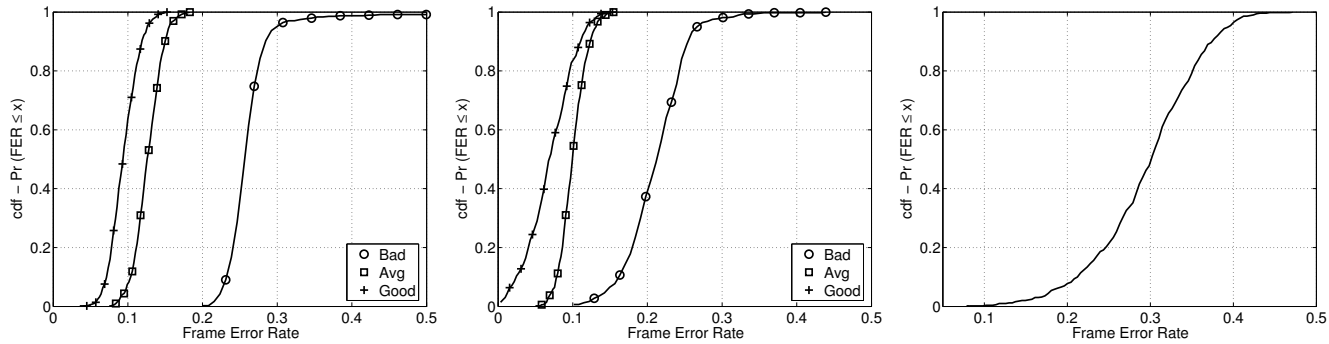
Como se puede ver a la vista de los resultados que se presentan en la Tabla II, el comportamiento del protocolo TCP es muy poco predecible, a pesar de que todas las medidas se han llevado a cabo en la misma posición. Por ejemplo, el rendimiento va desde prácticamente 5 Mbps, que es el que caracteriza un canal ideal, libre de errores, a valores mucho menores (inferiores incluso a 1 Mbps) [9], [10]. Se puede ver asimismo que uno de los aspectos que tiene una influencia mayor en el comportamiento de TCP es, como se ha discutido previamente, la presencia de periodos de inactividad, que habitualmente se asocian a aquellas situaciones en las que un mismo segmento se retransmite un gran número de veces. En este sentido, es interesante la comparación entre las medidas #11 y #14, ya que la PER es, en ambos casos, similar, pero la influencia de los periodos de inactividad se refleja en una disminución del rendimiento (que se divide casi por 4) en el segundo de los casos; se ve que la retransmisión de un mismo segmento, hasta en 9 ocasiones causa una inactividad de prácticamente 40 segundos, lo que perjudica claramente el rendimiento de la conexión. El receptor no recibe ningún segmento de datos durante aproximadamente 80 segundos (ver Figura 1). Otro ejemplo que merece la pena ser resaltado es la medida #8; aunque la PER es nula, se puede ver como el rendimiento es bastante bajo; esto se debe a que en este experimento en particular, la FER asociada al sentido de los reconocimientos TCP no podía despreciarse (siendo cercana al 20 %). Sin embargo, en el resto de experimentos, se comprueba que las tasas de error que afectan a los reconocimientos TCP (más cortos que los segmentos de datos) son notablemente menores; de hecho se asumirá que no hay errores en el sentido correspondiente durante la campaña de simulaciones, a pesar de que el modelo *BEAR* se puede configurar alternativamente.

V. RESULTADOS Y DISCUSIÓN

En esta sección se presentan los resultados obtenidos al analizar el comportamiento de las diferentes configuraciones del modelo *HMM* que se han presentado anteriormente, comparándolas con las obtenidas al utilizar *BEAR*. En todos los casos se utilizó el mismo procedimiento. Se realizaron 500 experimentos independientes para cada una de las configuraciones del canal. En cada una de ellas se lleva a cabo la transferencia FTP de un fichero de 10 `MBytes`; posteriormente, los resultados que devuelve el simulador *ns-2* se procesan para obtener las gráficas que se muestran seguidamente. Hay que destacar que, en todos los casos, se ha utilizado la misma versión/configuración del protocolo TCP y que, además, se ha supuesto que no hay errores en el sentido de los reconocimientos[3]. Finalmente, desde el punto de vista del protocolo MAC, se asume que no hay más nodos en el escenario que los propios transmisor y receptor, y que el número máximo de tramas que se transmiten antes de dar el datagrama por perdido es 4, de manera que las condiciones reflejan de la manera más precisa posible, las que caracterizaban el escenario en el que se llevaron a cabo las medidas reales descritas anteriormente. Finalmente, es pertinente comentar que para el canal *BEAR* se ha decidido utilizar un tiempo de coherencia de 5 s, ya que como se comprueba en [11], [2] consigue reflejar de manera adecuada el comportamiento observado sobre el canal real.

En primer lugar, la Figura 2 presenta la función de distribución (*cumulative distribution function*, cdf) de la probabilidad de error de las tramas para los tres modelos de canal. Las dos configuraciones del canal *HMM* tienen un comportamiento similar, aunque la FER tiende a ser ligeramente inferior con el modelo de 16 estados, que pone de manifiesto, asimismo, una variabilidad algo mayor. En cualquier caso, usando tanto *HMM-4* como *HMM-16* se obtienen unas tasas de error de trama inferiores a las que se utilizaron para entrenar las cadenas de *Markov* correspondientes (ver Tabla I); este es un comportamiento completamente esperado, ya que al emplear el protocolo TCP, la FER suele ser menor que al utilizar UDP, dado que en aquel caso el transmisor TCP reacciona ante una situación hostil del canal reduciendo la ventana de transmisión, de acuerdo con los algoritmos de control de flujo que emplea, lo que no sucede para el caso de UDP (que es el protocolo que se empleó para obtener las trazas con las que se estimaron los parámetros de las

[3]Esta suposición se realiza teniendo en cuenta que el tamaño de las tramas 802.11 correspondientes es notablemente menor y que, por tanto, la probabilidad de error en las mismas será asimismo más reducida



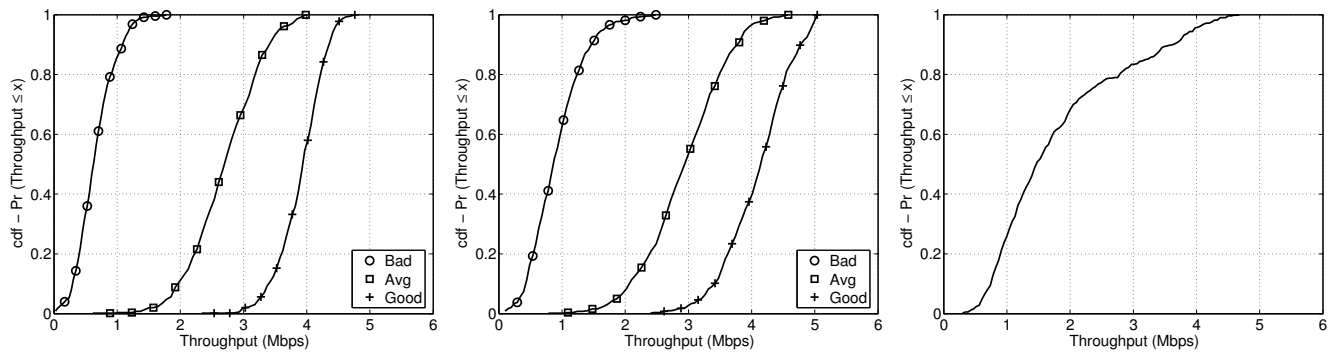
(a) Modelo *HMM* de 4 estados (b) Modelo *HMM* de 16 estados (c) Modelo *BEAR*

Fig. 2. cdf de la FER para los diferentes modelos de canal

cadenas de *Markov*). Es, por ejemplo, interesante el hecho de que la FER obtenida para la configuración *Avg* se acerca considerablemente al del canal *Good*, a pesar de que en las trazas empleadas para entrenar la cadena de *Markov* su comportamiento era sensiblemente diferente. De alguna manera, la mayor precisión de *HMM-16* a la hora de reflejar la estadística de las ráfagas de tramas erróneas, puede justificar la ligera reducción de la FER que se observa en este caso. A pesar de que sí que se observa cierta variabilidad para las diferentes configuraciones, en ningún caso esta llega a ser tan relevante como la que consigue reflejar el canal *BEAR*, en la que se pueden ver FERs que van desde el 5% al 45%; sin embargo, *BEAR* nunca refleja situaciones con FERs inferiores al 5%, que sí que se observan para la configuración *Good* del modelo *HMM-16*.

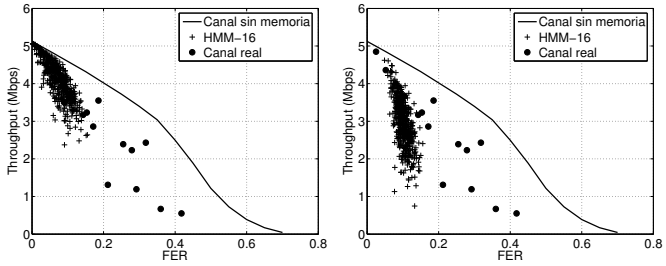
En cuanto al rendimiento observado, la Figura 3 muestra las cdf para todos los modelos de canal analizados. Respecto al comportamiento de los canales basados en cadenas de *Markov*, se vuelve a poner de manifiesto que el incremento en el número de estados tiene como consecuencia una variabilidad ligeramente superior en los resultados. Un aspecto interesante es que, a pesar de que se vio que en la FER, las configuraciones *Good* y *Avg* tienen comportamientos muy similares, en el rendimiento hay una diferencia notable, aproximadamente de 1 Mbps. Por otro lado, al igual que sucedía con la FER, en ningún caso la variabilidad que se consigue con cualquiera de las configuraciones de *HMM* es comparable a la que se logra al utilizar *BEAR*. Como también se pone de manifiesto al emplearlos con tráfico UDP [4], con las diferentes configuraciones de *HMM* sí que se logra cubrir un amplio rango de rendimientos (similar al observado empíricamente), pero en este sentido *BEAR* ofrece unas prestaciones más interesantes, ya que para una única configuración del modelo, se consigue una variabilidad en los rendimientos acorde a las medidas reales.

Para complementar los resultados anteriores, la Figura 4 representa la relación que existe entre la FER y el rendimiento, utilizando únicamente las configuraciones analizadas para el modelo *HMM* de 16 estados, que se comparan con los valores observados en el escenario real, así como con los resultados que ofrece el modelo *BEAR*. En todas las gráficas se ha añadido una línea que marca una cota superior, que se corresponde con el rendimiento que se obtendría al emplear un canal sin memoria. En este caso se puede ver que, a pesar de que es innegable la variabilidad que logra reflejar *HMM*, el canal *BEAR* muestra un comportamiento más cercano al observado empíricamente, aunque no es capaz de mimetizar situaciones en las que el rendimiento se aproxima al ideal (≈ 5 Mbps), que sí que se pueden ver para las configuraciones *Avg* y *Good* del canal basado en cadenas de *Markov* ocultas. En concreto, se puede decir que para FERs inferiores al 15%, el modelo *BEAR* se comporta prácticamente como un canal sin memoria, ya que el rendimiento que se obtiene es prácticamente igual al de la cota superior. Sin embargo (para FERs superiores al 15%, el comportamiento de *BEAR* es claramente superior al de cualquiera de las configuraciones de



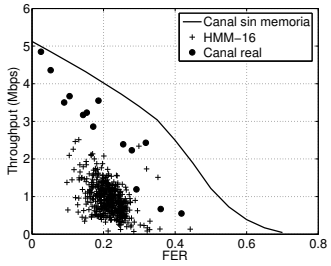
(a) Modelo *HMM* de 4 estados (b) Modelo *HMM* de 16 estados (c) Modelo *BEAR*

Fig. 3. cdf del Throughput para los diferentes modelos de canal

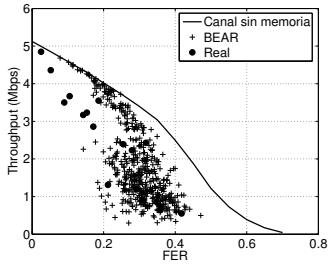


(a) Modelo *HMM-16 - Good*

(b) Modelo *HMM-16 - Avg*



(c) Modelo *HMM-16 - Bad*



(d) Modelo *BEAR*

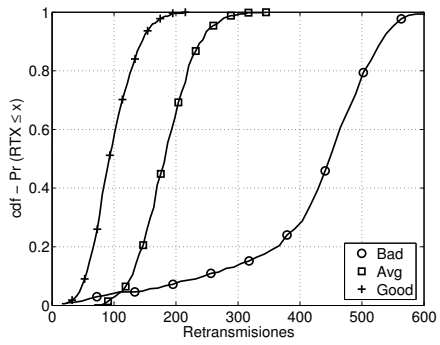
Fig. 4. Rendimiento frente a la FER para diferentes modelos de canal

HMM-16. Además, sería razonable pensar que, al configurar *BEAR* para que tuviera una mayor memoria (aumentando la potencia de ruido blanco de entrada al filtro AR [2]), se podrían reflejar aquellas situaciones que actualmente quedan fuera de su rango de resultados.

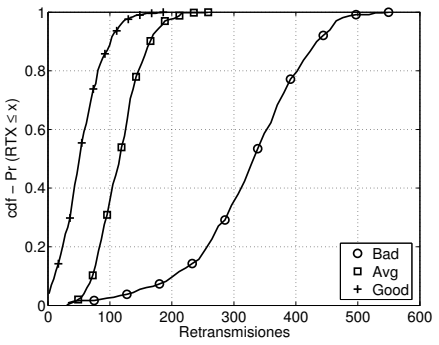
Como ya se comentó anteriormente, uno de los principales aspectos a la hora de determinar el comportamiento del protocolo TCP son las retransmisiones que la entidad transmisora tiene que llevar a cabo para compensar las pérdidas debidas al comportamiento hostil del canal inalámbrico. En primer lugar, la Figura 5 muestra la cdf del número de segmentos retransmitidos para los diferentes modelos de canal y configuraciones empleadas. Comparando los resultados obtenidos con los dos canales basados en cadenas de *Markov* se puede ver que sí que hay diferencias relevantes entre *HMM-4* y *HMM-16*, ya que en éste el número de retransmisiones es sensiblemente menor. Otro aspecto a destacar es que para las retransmisiones, a diferencia de lo que sucedía para los parámetros analizados anteriormente, el modelo de canal *HMM* sí que permite recoger una variabilidad notable, aunque únicamente para la configuración *Bad*, ya que en el resto los resultados se mantienen, de manera bastante predecible, en torno a su valor medio. De todas maneras, se puede ver que nuevamente, el modelo de canal *BEAR* vuelve a ofrecer unas prestaciones muy interesantes, ya que se refleja un número de retransmisiones en un rango amplio, que abarca el observado empíricamente.

En TCP existen principalmente dos posibles tipos de retransmisiones; en primer lugar están aquellas que se realizan al ejecutar el algoritmo *Fast Retransmit*, tras la recepción de un *TripleACK*; por otro lado, transcurrido un tiempo determinado sin recibir confirmación de un segmento (*Retransmission Time Out, RTO*), éste también se retransmitiría. Sin ninguna duda, estas últimas son las que causan una reducción mayor en el rendimiento del protocolo, ya que dan lugar a periodos de inactividad (en los que el canal no se utiliza); teniendo en cuenta que en TCP el *RTO* se va incrementando paulatinamente (algoritmo exponencial binario) a medida que se realizan retransmisiones del mismo segmento, estos periodos de inactividad pueden originar un descenso notable de las prestaciones de TCP, tal y como se vio anteriormente. Es por tanto, interesante analizar el porcentaje de retransmisiones que derivan de la ejecución de *Fast Retransmit*. La Figura 6 permite analizar dicho parámetro, comparando los resultados obtenidos al emplear las tres configuraciones del modelo *HMM-16* con los del canal *BEAR*, el escenario real y los de un modelo sin memoria, que también establece una cota superior (como se vio anteriormente al relacionar la FER y el rendimiento). De nuevo, el modelo *BEAR* ofrece un rango de comportamientos más acorde al visto empíricamente (aunque el porcentaje de retransmisiones por *Fast Retransmit* es algo mayor que el observado en la realidad); además, en este caso, la configuración *Bad* del *HMM-16* consigue reflejar, de manera más precisa, el porcentaje de retransmisiones por expiración de RTO que se observó en las medidas reales.

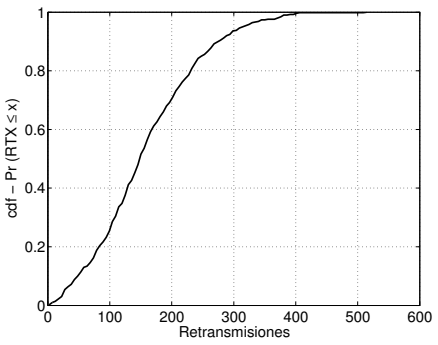
Uno de los aspectos que en mayor medida damnifica el rendimiento que se puede alcanzar al emplear el protocolo TCP es la existencia de periodos de inactividad elevados en el transmisor, consecuencia de los algoritmos que incorpora TCP para combatir la congestión de los elementos intermedios de la red y que tienen un efecto muy negativo al ser utilizados en entornos inalámbricos. La Figura 7 muestra la cdf de los tiempos de inactividad máxima que se observaron al analizar los diferentes modelos de canal. En este caso, el comportamiento de los modelos *HMM* es algo peor al ma-



(a) Modelo *HMM* de 4 estados

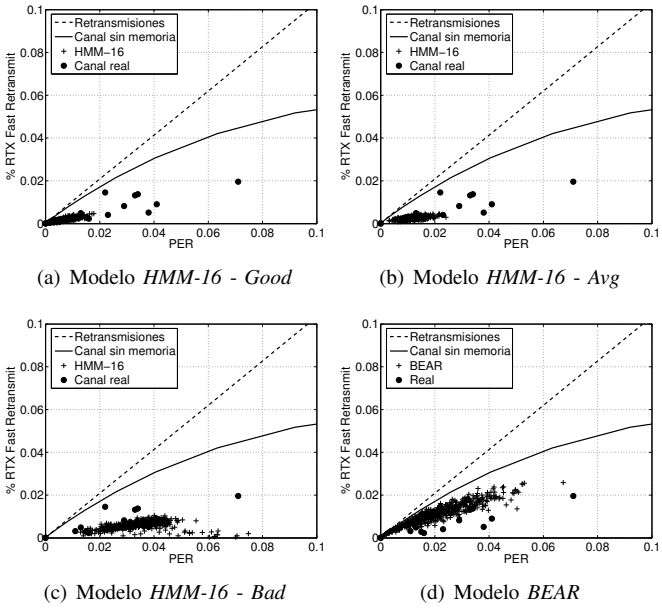


(b) Modelo *HMM* de 16 estados



(c) Modelo *BEAR*

Fig. 5. cdf de las retransmisiones observadas con los diferentes modelos de canal



(a) Modelo *HMM-16 - Good* (b) Modelo *HMM-16 - Avg*

(c) Modelo *HMM-16 - Bad* (d) Modelo *BEAR*

Fig. 6. Retransmisiones *Fast Retransmit* frente a la PER. La línea a trazos se corresponde con el total de retransmisiones, que coincide con la PER.

nifestado con los parámetros anteriores ya que únicamente la configuración *Bad* origina periodos de inactividad superiores a 5 segundos (en los otros dos casos, la probabilidad de tener paradas más largas es prácticamente insignificante), pero a costa de causar inactividades que posiblemente no reflejen (por ser demasiado elevadas) las que se originan en el canal real. Para el resto de configuraciones (entre las que no existen diferencias apreciables), el comportamiento se aleja del que se obtuvo empíricamente. Por su parte, el modelo *BEAR* consigue una variabilidad adecuada, que mimetiza apropiadamente el comportamiento real.

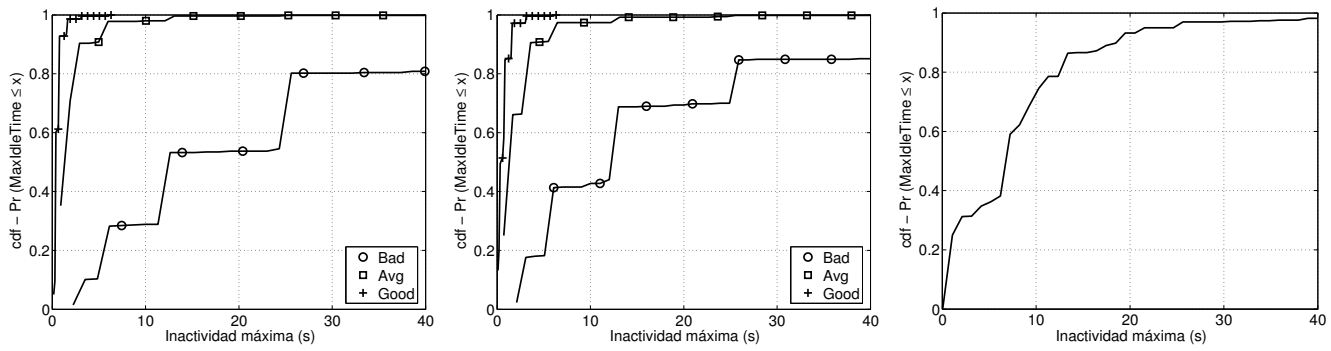
Hasta ahora se ha analizado el comportamiento *global* de los diferentes modelos de canal, analizando las funciones de probabilidad acumulada de varios parámetros fundamentales en la operación del protocolo TCP. Es asimismo interesante estudiar la evolución de medidas individuales, para lo que se utilizarán las gráficas *tiempo-secuencia* tradicionales. Se utilizará únicamente la configuración que mejor representa el comportamiento *medio* del canal real, esto es *Avg*. De las 500 medidas realizadas anteriormente, se seleccionan dos (con *throughputs* elevado y bajo). La Figura 8 muestra los resultados obtenidos; en primer lugar cabe destacar que con los tres modelos de canal es posible reflejar conexiones TCP con una dinámica diametralmente opuesta entre ellas; a la hora de comparar los dos modelos de canal basados en *HMM* se puede ver que el hecho de usar 16 estados supone una mejora, ya que se cubre un rango mayor de comportamientos (permite reflejar situaciones con un rendimiento superior), aunque hay que tener en cuenta que se trata de medidas puntuales.

VI. Conclusiones

Uno de los aspectos que en en mayor medida perjudican el comportamiento del protocolo TCP sobre entornos de propagación inalámbricos en interiores es la presencia de errores a ráfagas, fruto de la *memoria* que exhiben dichos canales. A pesar de la gran actividad investigadora en esta área, no existen muchos modelos de canal capaces de reflejar dicho comportamiento y que gocen de un uso masivo por parte de la comunidad investigadora. Una de las aproximaciones más empleadas tradicionalmente, el canal *Gilbert-Elliot*, que está basado en una cadena de *Markov* de 2 estados, es poco apropiado para mimetizar dicho comportamiento *a ráfagas*, por lo que recientemente se han propuesto otras alternativas más complejas, en las que se introduce la utilización de cadenas de *Markov* ocultas.

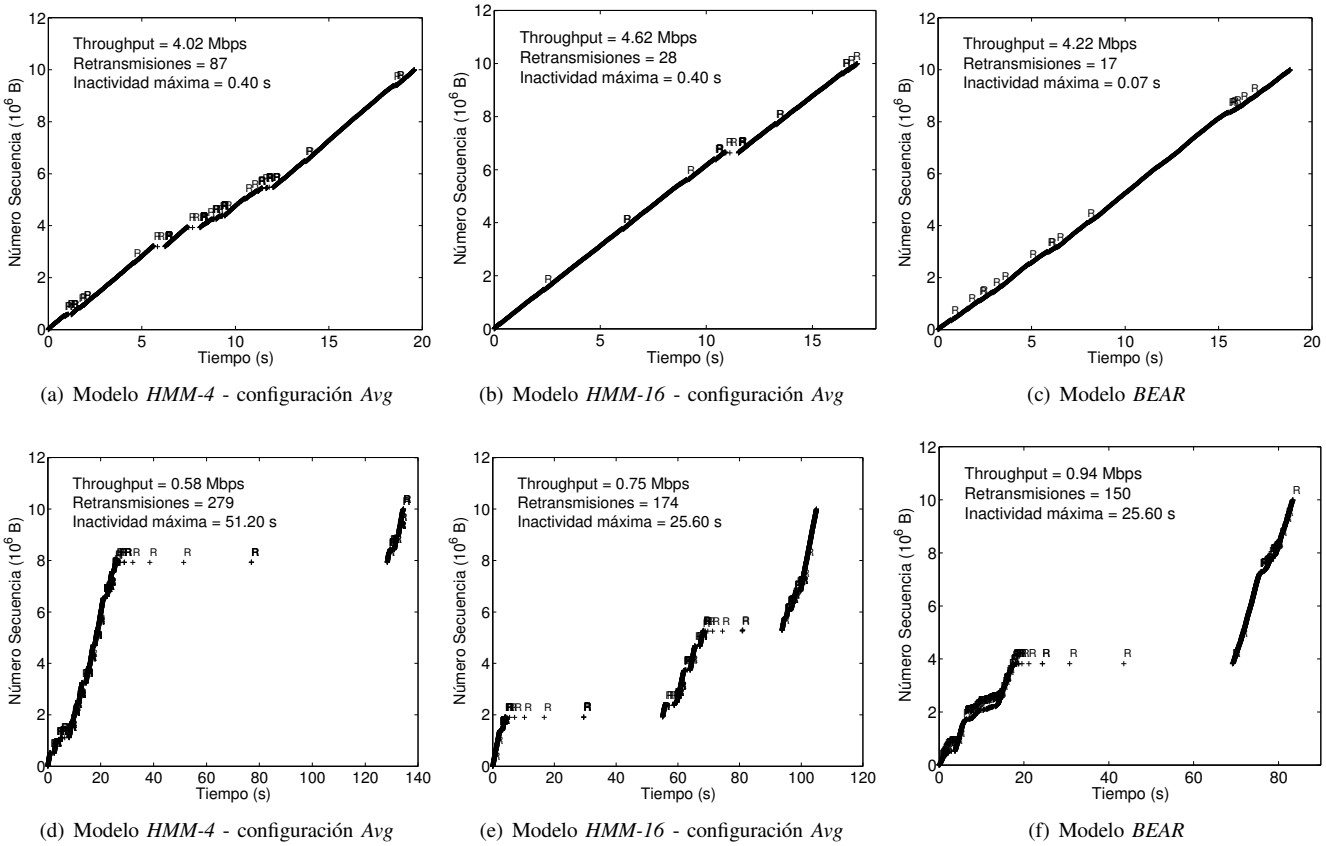
Los modelos *HMM* se han analizado previamente, aunque principalmente utilizando tráfico UDP. En este trabajo se presenta un análisis exhaustivo del comportamiento del protocolo TCP cuando se utilizan canales *HMM*. Una de las características más relevantes es que los parámetros del modelo subyacente se han estimado en base a un conjunto de medidas reales, y la configuración del mismo se ha llevado a cabo de manera temporal, de manera que su comportamiento se independiza de la dinámica del tráfico de capas superiores, lo que es especialmente relevante para el caso de TCP.

Además, se comparan las prestaciones de los canales *HMM* con las de otra alternativa, el modelo *BEAR*, que tiene una filosofía claramente opuesta, ya que se basa en un modelado de la relación señal a ruido *con memoria* y, a partir de la misma, determinar la presencia de error o no en cada trama recibida. Esta comparación se hace en base a un conjunto de medidas reales que permiten conocer cuál es el comportamiento real de TCP en entornos de propagación en interiores. En primer lugar se ha puesto de manifiesto que los modelos *HMM* son capaces de reflejar, de manera adecuada, algunos de los comportamientos observados empíricamente,



(a) Modelo *HMM* de 4 estados (b) Modelo *HMM* de 16 estados (c) Modelo *BEAR*

Fig. 7. cdf de los tiempos de inactividad máxima con los diferentes modelos de canal



(a) Modelo *HMM-4* - configuración *Avg* (b) Modelo *HMM-16* - configuración *Avg* (c) Modelo *BEAR*

(d) Modelo *HMM-4* - configuración *Avg* (e) Modelo *HMM-16* - configuración *Avg* (f) Modelo *BEAR*

Fig. 8. Comportamiento individual de varias conexiones TCP utilizando diferentes modelos de canal

siendo muy superiores en sus prestaciones a las de, por ejemplo, *Gilbert-Elliot*. Sin embargo, su comportamiento depende fuertemente de la configuración concreta del modelo y, a pesar de que se pone de manifiesto cierta variabilidad, esta no es comparable a la que consigue reflejar el modelo *BEAR*, que se corresponde con la observada en la realidad. Sin embargo, se ha visto que en algunas circunstancias (situaciones con una FER inferior al 15%, por ejemplo), algunas configuraciones de *HMM* pueden mejorar el comportamiento de *BEAR*.

Gracias al conocimiento acerca de su comportamiento y a la implementación de los diferentes modelos de canal, se pueden plantear diferentes líneas de investigación futuras. Por ejemplo, en el ámbito de las redes multi-salto se postulan diferentes mecanismos y algoritmos para optimizar su comportamiento (rendimiento, consumo energético, etc); para ello una adecuada interacción con las capas inferiores de la pila de protocolos puede ser fundamental y la utilización de alguno de los modelos de canal presentados en este trabajo puede aportar notables ventajas.

AGRADECIMIENTOS

Los autores querrían expresar su agradecimiento al Gobierno de España por su financiación en el proyecto "Comunicaciones Cognitivas y Cooperativas sobre Entornos Heterogéneos", C3SEM (TEC2009-14598-C02-01)

REFERENCES

[1] K. V. Cardoso y J. F. De Rezende, "Accurate hidden markov modeling of packet losses in indoor 802.11 networks," *IEEE Communication Letters.*, vol. 13, pp. 417–419, Junio 2009.

[2] R. Agüero, M. García-Arranz, y L. Munoz, "Accurate simulation of 802.11 indoor links: A 'bursty' channel model based on real measurements," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, pp. 1–17, 2010.

[3] R. Agüero, M. García, y L. Munoz, "BEAR: A bursty error autoregressive model for indoor wireless environments," en *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*, sept. 2007, pp. 1 –5.

[4] J. R. Santana, R. Agüero, M. García, y L. Munoz, "Modelado de errores a ráfagas en canales WLAN interiores mediate cadenas de markov ocultas," en *IX Jornadas de Ingeniería Telemática, JITEL 2010*, sept. 2010, pp. 110 – 116.

[5] M. Bottigleliengo, C. Casetti, C.-F. Chiasserini, y M. Meo, "Short-term fairness for TCP flows in 802.11b WLANs," en *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 2, marzo 2004, pp. 1383 – 1392 vol.2.

[6] M. Rossi, R. Vicenzi, y M. Zorzi, "Accurate analysis of TCP on channels with memory and finite round-trip delay," *Wireless Communications, IEEE Transactions on*, vol. 3, no. 2, pp. 627 – 640, marzo 2004.

[7] S. Kim y K. Mitchell, "An analytic model of TCP performance over multi-hop wireless links with correlated channel fading," *Performance Evaluation*, vol. 64, no. 6, pp. 573 – 590, 2007.

[8] L. Rabiner, "A tutorial on HMM and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[9] M. García, R. Agüero, y L. Muñoz, "On the unsuitability of TCP RTO estimation over bursty error channels," en *PWC 2004: The IFIP TC6 9th International Conference on Personal Wireless Communications*, Delft, The Netherlands, 2004.

[10] V. Vasudevan, M. Parikh, K. Chandra, y C. Thompson, "TCP and IEEE 802.11b protocol performance in indoor wireless channels," en *IEEE Sarnoff Symposium*, Princeton, New Jersey, USA, 2003.

[11] R. Agüero, M. García, y L. Munoz, "Simulación realista del comportamiento de TCP sobre canales inalámbricos con errores y memoria," en *VII Jornadas de Ingeniería Telemática, JITEL 2008*, sept. 2008, pp. 33 – 40.

AP-CAP framework: Monitorizando a 10 Gb/s en hardware de propósito general

Jaime Fullaondo, Pedro M. Santiago del Río, Javier Ramos,
José Luis García-Dorado y Javier Aracil.
High Performance Computing and Networking group,
Universidad Autónoma de Madrid.
{jaime.fullaondo, pedro.santiago, javier.ramos, jl.garcia, javier.aracil}@uam.es.

***Resumen*—La comunidad investigadora ha propuesto recientemente el uso de hardware de propósito general para tareas antes reservadas a hardware dedicado, típicamente más caro y menos flexible. En este artículo evaluamos la viabilidad de usar este hardware de propósito general para monitorizar enlaces de alta capacidad. Primero, comparamos las distintas propuestas de la literatura para optimizar la capacidad receptora de las tarjetas de red; segundo, se presenta AP-CAP, un framework que permite el acceso a los datos de forma equivalente a pcap pero a alta velocidad; finalmente, se evalúa el rendimiento de AP-CAP en tareas típicas de monitorización como son la identificación de aplicaciones, el filtrado de tráfico y su almacenamiento. Los resultados muestran que AP-CAP es capaz de monitorizar enlaces de 10 Gb/s, velocidad que se ha convertido en el estándar actual de los enlaces de las redes troncales, utilizando hardware de propósito general.**

***Palabras Clave*—monitorización alta velocidad; 10 Gb/s; hardware de propósito general; AP-CAP; motores de captura.**

I. Introducción

La monitorización de Internet se ha convertido en una herramienta vital de los operadores de red para optimizar el rendimiento y calidad de servicio ofrecido por sus redes así como para evitar, en lo máximo posible, el impacto de tráfico malicioso o anómalo. Sin embargo, mientras los esfuerzos en mejorar las inversiones en capacidad para monitorizar las redes actuales ha sido notable [1], la velocidad de transmisión de los enlaces ha crecido más rápidamente. Las causa de este crecimiento es, entre otras razones, el incremento de la popularidad de nuevas aplicaciones demandantes de gran ancho de banda como la televisión y telepresencia en alta definición, copias de seguridad en tiempo real, juegos interactivos, por mencionar algunas de ellas. De este modo, si bien, hace unos años monitorizar enlaces con capacidad de entre 100 Mb/s y 1 Gb/s ya se consideraba un reto, en la actualidad la velocidad típica de un enlace o interfaz de un router en redes de alta velocidad es típicamente de 10 Gb/s, llegando a agregados de 100 Tb/s [2].

Por otro lado, la comunidad científica ha prestado notable atención a la utilización de hardware de propósito general para tareas anteriormente reservadas exclusivamente para sistemas con hardware dedicado. Las ventajas de este planteamiento son principalmente dos: Primero, el uso de sistemas hardware de propósito general facilita una mayor flexibilidad para adaptar cualquier tarea de operación o gestión de red, así como una mayor facilidad para su mantenimiento. Ejemplo de ello es el gran interés que han despertado los routers software [3], [4]. Segundo, la inversión que supone la compra de equipos con hardware dedicado típicamente supera en varios ordenes de magnitud el precio de una solución basada en software y hardware de propósito general. Además, la utilización de hardware de propósito general presenta otras ventajas como el uso de políticas de ahorro de energía ya implementadas en PCs, facilidad en la actualización de ciertas partes del hardware y flexibilidad en la programación de nuevas técnicas de medida [5].

De este modo, este artículo estudia cómo, por un lado, dar respuesta a las demandas de monitorización a alta velocidad (10 Gb/s) pero utilizando equipos baratos y, a la vez, flexibles, esto es, hardware de propósito general. La Fig. 1 muestra cómo estructuramos un sistema de monitorización de alta velocidad usando hardware de propósito general, esencialmente, una pila de cuatro niveles. El primer nivel sería la propia tarjeta de red estándar (típicamente, denominada NIC), a continuación, un driver que accede a ella y pone a disposición del kernel del sistema operativo los datos, este sería el segundo nivel. El tercer nivel consistiría en un conjunto de funciones o framework que permite a las aplicaciones de nivel de usuario acceder a los datos. En general, la combinación de un driver y un framework, genéricamente, puede ser denominada como motor de captura. Por último, tendríamos como cuarto nivel, el propio nivel de aplicación, donde se implementan las distintas funcionalidades, en nuestro caso centradas en monitorización pero que podría tratarse de cualquier otra. Por ejemplo, los ya citados routers software implementan la lógica de enrutamiento de un router con hardware dedicado en este nivel. Sin embargo, esta estructura de 4 niveles puede variar en tanto ciertas herramientas proponen llevar carga que lógicamente debería estar situadas en un nivel dado, a otro nivel inferior por cuestiones de rendimiento.

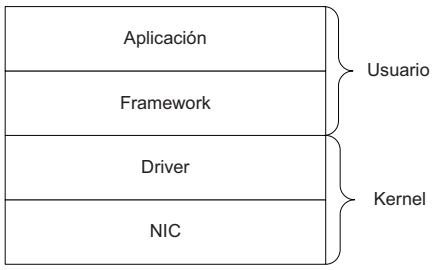


Figura 1. Estructura genérica de un sistema de monitorización de alta velocidad usando hardware de propósito general

El primer paso, por tanto, para optimizar esta pila es estudiar el driver instalado por defecto para la tarjeta de red. Varias han sido las propuestas para optimizar estos drivers [5], [6],

ISBN: 978-84-694-5948-5

[7]. Estas propuestas de optimización, explicadas en mayor detalle en la siguiente sección, consisten esencialmente en la adaptación y modificación del acceso del sistema operativo a la tarjeta de red. Dos son las ideas fundamentales en las que se basan estas propuestas: primero, utilizar las capacidades multicore y multicola de las CPUs y de las tarjetas de red respectivamente, y, segundo, la transmisión de información en bloques de tamaño superior a un paquete.

En este sentido, este artículo, como primera aportación, evalúa las diferentes propuestas de motores de captura en diferentes arquitecturas hardware, esto es, Intel y AMD. A continuación, hemos evaluado el rendimiento y operabilidad del tercer nivel, o framework de acceso, de los propuestos en la literatura. En general se ha encontrado que, o bien el rendimiento es insuficiente [6], [7], o difícilmente usable como [5]. Es por ello, que este trabajo presenta un framework de acceso, denominado AP-CAP (CAPtura de Altas Prestaciones), que aúna, por un lado, rendimiento y, por otro, usabilidad, al seguir el estándar de facto de programación para captura de paquetes que es pcap [8]. Varios casos de uso de esta funcionalidad se incluyen en este trabajo, en concreto, mostramos, como la clasificación de tráfico en aplicaciones (en particular, nos centramos en tráfico Skype [9] y VoIP), la captura y almacenamiento del tráfico en fichero. Nuestro trabajo demuestra, por primera vez, que todas estas funcionalidades se pueden realizar con éxito a 10 Gb/s en hardware de propósito general.

El resto del artículo está estructurado de la siguiente forma. La Sección II presenta una revisión de las distintas propuestas de la literatura en tanto en cuanto a los motores de captura. La Sección III por su parte detalla los distintos escenarios de prueba levantados para la evaluación de prestaciones. En la siguientes dos secciones describimos AP-CAP y mostramos su aplicabilidad a varios problemas típicos en la monitorización. Por último, la Sección VI presenta las principales conclusiones de este trabajo.

II. Trabajos relacionados y fundamentos

En la literatura se pueden encontrar varios artículos que, con objeto de implementar routers software, proponen optimizar el driver de red como uno de los pasos para obtener altas tasas de recepción en tarjetas de red estándar [5], [6] que está muy próximo a los objetivos de este artículo.

En concreto, los autores de [5] presentan RouteBricks, una arquitectura para implementar routers software basada, esencialmente, en tres novedosas ideas. Primero, proponen repartir la carga del proceso de enrutado entre varias máquinas. La segunda idea que se implementa en RouteBricks, consiste en sacar partido de la capacidades multicola de las tarjetas de red y las capacidades multicore de las CPUs actuales. De esta manera, al utilizar varios cores de una CPU es posible repartir la carga y, por tanto, evitar la posibilidad de que un solo core llegue a su límite computacional. De hecho, los autores demuestran que a altas tasas de recepción la carga de un único core, típicamente, alcanza el $100\,\%$ de ocupación, aun cuando el core usado en el experimento tiene una frecuencia de 2.8 GHz. Sin embargo, este planteamiento tienen una limitación. Si son varios los cores que acceden a una única cola de la tarjeta de red, cada core debe, primero, bloquear la cola y, solo entonces, puede acceder a los datos. Este proceso de bloqueo y desbloqueo de la cola tiene un coste en tiempo que perjudica el rendimiento, típicamente, similar al propio beneficio que proporciona el uso de arquitecturas multicore. La solución propuesta por los autores es asignar estrictamente a cada cola de la tarjeta de red un solo core disponible en el hardware, evitando, de este modo, perdidas en la interacción de una cola con varios cores. Consecuentemente, esta aproximación requiere de un algoritmo que reparta los paquetes que llegan a la tarjeta de forma homogénea entre sus colas. Esta funcionalidad, esencialmente una función hash, la proveen las tarjetas de red estándar y está implementado en hardware lo que hace su coste computacional despreciable [10]. Es importante que este algoritmo distribuya el tráfico de forma uniforme, y no sobrecargue a un core mientras otros quedan ociosos. Por último, como tercera idea los autores afirman que la tarjeta de red debe suministrar al sistema operativo los datos recibidos en bloques grandes, esto es, evitando transmisiones paquete a paquete que suele ser la aproximación típica a este problema. Como conclusión, los autores de [5], siguiendo el esquema introducido en la Fig. 1, proponen como nivel 2 un nuevo driver, que denominan RouteBricks driver, que aúna las tres ideas expuestas. Además proponen el uso de Click [11] como framework entre el nivel de aplicación, concretamente un router software, y su driver. Sin embargo, el problema de esta aproximación es que Click, si bien puede ser un framework útil para implementar funcionalidades propios de un router, accede y suministra los paquetes al nivel de aplicación uno a uno, aunque el driver los facilite en bloques más grandes. Esto limita significativamente el beneficio conseguido por el driver RouteBricks. Por consiguiente, en este trabajo presentamos un framework, AP-CAP, que permite que la interacción con el nivel de aplicación se realice también en bloques. De este modo, la Fig. 2 muestra esquemáticamente como encajan cada una de las piezas en el esquema propuesto, nótese que Click se configura como parte del kernel pues se espera que así incremente su rendimiento.

Los autores en [6], por su parte, partiendo de la implementación y mejoras de RouteBricks, presentan su propio driver mejorado y framework, que llamaremos PacketShader driver y PS User Space library, respectivamente. Respecto al driver, los autores proponen paralelizar la transmisión de datos entre la memoria interna de la tarjeta de red y la memoria de la máquina de propósito general con la carga del siguiente bloque a transmitir en la cache de la CPU, mejorando, de forma significativa, el rendimiento. En cuanto al framework, afrontan la limitación en el acceso paquete a paquete de Click, implementando un módulo a nivel usuario que permite al nivel de aplicación acceder a los datos a alta velocidad. En concreto el nivel de aplicación es, de nuevo, un router software, pero esta vez, los autores propone el uso de GPUs para ser capaces de realizar tareas pesadas a la velocidad de recepción alcanzada por el driver PacketShader. Esta velocidad, usando varias maquinas, llega a 40 Gb/s. Sin embargo este framework, PS User Space library, presenta varias limitaciones. Primero, facilita funcionalidades genéricas pero orientadas siempre al enrutado, no a la monitorización, y, segundo, no cumple con la implementación de la librería pcap, limitando así su portabilidad y generalización. De este modo, el presente artículo muestra un framework, AP-CAP,

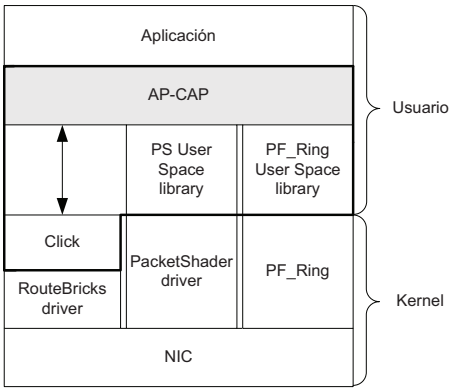


Figura 2. Arquitectura de cada una de la propuestas de la literatura evaluadas y AP-CAP

que proporcionando la velocidad objetivo de 10 Gb/s, sigue el estándar de facto de programación para captura de paquetes que es pcap.

Finalmente, en [7], [12], partiendo de un planteamiento similar a [5], [6], los autores se plantean utilizar tarjetas de red estándar para realizar análisis de tráfico a alta velocidad. Sin embargo, el driver facilitado por los autores, PF_Ring, no ha mostrado resultados satisfactorios. La tasa de este driver, sin pérdidas, con tamaño de paquete pequeño (64 bytes) ha sido de 1.2 millones de paquetes por segundo (M paquetes/s), solo levemente superior a 600 Mb/s.

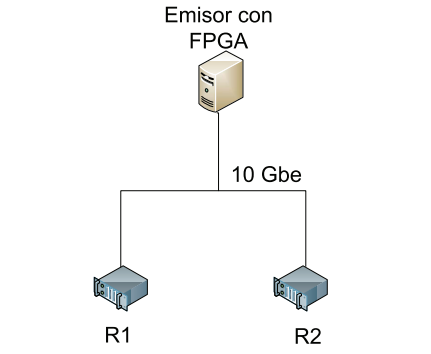
III. Entorno de pruebas

Para la realización de experimentos y comparativas se ha usado un entorno de pruebas controlado que abarca diferentes escenarios. La arquitectura de pruebas se basa en uno o varios emisores y dos receptores. Para los receptores se han seleccionado dos arquitecturas hardware diferentes para analizar el impacto de las soluciones a evaluar. El primer receptor (R1) es un servidor que cuenta con cuatro procesadores AMD Opteron 6128 que trabajan a una frecuencia de 2 GHz. Cada procesador contiene ocho cores reales. En lo referente a la memoria el servidor cuenta con treinta y dos módulos de 4 GB DDR3 con una frecuencia de trabajo de 1333 MHz. Estos componentes se integran sobre una placa base Supermicro H8QG6.

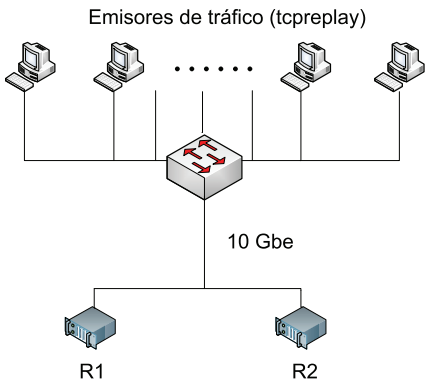
El segundo receptor (R2) se basa en una arquitectura Intel y cuenta con 2 procesadores Intel Xeon E5520 que trabajan a una frecuencia de 2.27 GHz. Cada procesador tiene 4 cores reales que se contabilizan como 8 cores lógicos usando la tecnología Hyperthreading. Para evitar la concurrencia de hilos en un core esta opción ha sido desactivada. En lo referente a la memoria el equipo cuenta con seis módulos de 4 GB DDR3 con una frecuencia de trabajo de 1066 MHz. Todos los componentes se integran en una placa Supermicro X8DTG-D. Ambos receptores hacen uso de tarjetas Intel X520-SR2 10 Gigabit Ethernet de doble puerto para la adquisición de tráfico [10]. Estas tarjetas se basan en el chip de última generación 82599 de Intel.

Respecto al Sistema Operativo, RouteBricks fue instalado en un sistema Linux CentOS 5.3, tal y como se recomienda en su web [13] usando el kernel 2.6.24 para ambos receptores. Por su parte, PacketShader fue ejecutado en un sistema Ubuntu 10.04 Linux Server, siguiendo el despliegue de los autores [6], con kernel 2.6.35. Ambos sistemas operativos fueron instalados en su versión de 64 bits.

Para la emisión de tráfico se han seguido dos aproximaciones diferentes. Para la generación de tráfico sintético se ha utilizado un ordenador de prestaciones medias con una tarjeta PCIe que contiene una tarjeta HTG-V5TXT-PCIE de HitechGlobal basada en una FPGA de la familia Virtex-5 de Xilinx, que dispone de 4 puertos 10 GbE basados en cajas SFP+. Esta FPGA ha sido programada para generar paquetes TCP de diferentes tamaños. Los tamaños configurables son 64 bytes, 512 bytes y 1500 bytes para abarcar los escenarios de caso peor, caso real y caso mejor [14], respectivamente. Tanto las direcciones IP como los puertos de origen y destino de los paquetes generados son incrementales y diferentes para cada paquete. Para la generación de tráfico real se han usado hasta un total de diez máquinas conectadas a un conmutador Cisco Catalyst 2690 que agrega el tráfico de diez interfaces de entrada Gigabit Ethernet en un puerto de salida 10 Gigabit Ethernet. Cada una de las máquinas se encarga de reproducir trazas de enlaces reales usando la herramienta tcpreplay [15] consiguiendo un throughput total de aproximadamente 9.2 Gb/s y 2 M paquetes/s. Las trazas utilizadas han sido obtenidas del conjunto de datos de CAIDA de 2009 del troncal de San José [16]. Estas trazas contienen un minuto de tráfico tomado a las 5 a.m. en diferentes días en los meses de julio a diciembre. Para su correcta reproducción se han unido los sentidos ascendente y descendente y se han añadido cabeceras Ethernet y relleno ya que los datos originales están truncados en captura. En la Fig. 3 se muestran los dos entornos de pruebas.



(a) Tráfico Sintético



(b) Tráfico Real

Figura 3. Entornos de prueba

IV. Evaluación de prestaciones en distintos entornos

Como se ha explicado en la sección anterior, la paralelización en la captura de tráfico requiere aprovechar al máximo las características de las arquitecturas multicore y nos permite distribuir el procesamiento del tráfico entre los mismos. Sin embargo, esto en ocasiones puede no resultar del todo trivial debido a factores como la localidad de memoria (particularmente en arquitecturas *Non-Uniform Memory Acess* -NUMA-), la afinidad de procesos e interrupciones, etc. Por ello hay una serie de medidas que se deben tomar para intentar maximizar la tasa de captura/procesado de tráfico, a saber:

1. Asignar las interrupciones de cada cola RSS (Receive-Side Scaling) a un core. Esta afinidad de interrupciones a cores se puede lograr mediante un script que es proporcionado junto con el driver de Intel. Igualmente, distribuir de forma uniforme los paquetes entre las distintas colas. Merece la pena apuntar que hemos evaluado esta premisa empíricamente encontrando que, efectivamente, el tráfico, tanto real como aleatoriamente generado, es correctamente distribuido entre las colas de la tarjeta.
2. Como consecuencia de establecer una afinidad interrupción/núcleo es importante procurar asignar los procesos de captura para cada cola RSS a un núcleo que comparta la memoria caché y MMU del núcleo al cual se le asignó la afinidad de interrupción para esa cola RSS. Esto se puede lograr mediante los comandos taskset[1] o numactl[2] en linux.

A continuación, explicamos la instalación, configuración y ajuste de parámetros necesaria para obtener el rendimiento óptimo de los motores de captura RouteBricks y PacketShader. Para la búsqueda de esta configuración óptima, que posteriormente será usada como base para el framework APCAP, hemos evaluado el rendimiento en términos de la tasa de pérdidas, la utilización de CPU por cada núcleo de proceso y la tasa de paquetes y bits. Las tasas de pérdidas, paquetes por segundo y bits por segundo se han tomado consultando los registros de la tarjeta mediante el comando Linux ifconfig [3], mientras que la carga de la CPU se ha obtenido con el comando ps [4]. Hay que notar que, debido a la poca frecuencia de refresco de dichos registros (en torno a 1 segundo), se ha tomado el promedio en intervalos de 30 segundos. Las pruebas se han repetido para tres tamaños de paquete distinto (usando el generador de paquetes sintéticos descrito en la Sección III), a saber: 64 bytes, que representa un escenario de caso peor, en el que la tasa de paquetes es máxima ~ 14.8 M paquetes/s; 512 bytes, que representa un caso medio; y 1500 bytes que supone un caso mejor. Merece la pena destacar, que en el caso de 64 bytes no es posible alcanzar la tasa de línea de 10 Gb/s debido a la limitación del estándar IEEE 802.3ae en el número de paquetes por segundo [17]. Por último, como se indica en la sección III, se han considerado dos arquitecturas distintas (AMD Opteron e Intel Xeon).

[1] linux.die.net/man/1/taskset
[2] linux.die.net/man/8/numactl
[3] linux.die.net/man/8/ifconfig
[4] linux.die.net/man/1/ps

El parámetro clave para los dos motores de captura es el número de colas RSS utilizadas. Idealmente, podemos pensar que a mayor número de colas, mejor rendimiento, menos proceso asociado y mayor capacidad de escalabilidad del sistema, permitiéndonos incluso la monitorización, con servidores multicore de propósito general, de enlaces de mayor capacidad a los 10 Gb/s (usando un mayor número de tarjetas de 10 Gb/s o, en un futuro próximo, tarjetas de 40 Gb/s).

Para cada escenario (esto es, fijando el motor de captura, la arquitectura, el tamaño del paquete y el número de colas RSS), se ha repetido el experimento 10 veces, tomando su valor promedio y el intervalo de confianza del $95\,\%$. Por falta de espacio, las gráficas de tasa de paquetes no se han mostrado. En cualquier caso, se obtienen de manera trivial a partir de la tasa en bits y el tamaño del paquete.

IV-A. Evaluación de RouteBricks

A la luz de los resultados publicados en [5] por los autores de RouteBricks, consideramos esta solución no solo para enrutamiento software, sino también para monitorización (captura de tráfico, DPI, etc).

Para poder obtener el tráfico a nivel de usuario, al módulo Click, MQPollDevice, que permite sondear la llegada de paquetes a nivel de cola RSS en el driver e introducir los mismos en el software router, tenemos que sumarle el modulo de Click, ToUserDevice. Este módulo permite situar el tráfico de cualquier interfaz en espacio de usuario mediante un dispositivo de carácter, lo que nos permite leer los paquetes de ese dispositivo de carácter como si se tratara de un fichero. Hay que destacar que el módulo ToUserDevice no solo es independiente del trabajo realizado por el equipo de desarrollo de RouteBricks, sino que además está en fase experimental.

Para llevar a cabo las pruebas aquí mostradas hemos utilizado la última versión disponible de RouteBricks, 0.1.389(alpha) [13]. Esta versión incorpora un instalador que tuvo que ser modificado ligeramente para no incorporar todas las funcionalidades de gestión de nodo. También tuvimos que alterar ligeramente la compilación de Click para que incorporase los elementos experimentales, como ToUserDevice. Además, fue necesario definir unos scripts Click con los cuales se redirige el tráfico desde las colas RSS a los dispositivos de usuario. Click permite asociar sus colas internas a un núcleo determinado, de este modo podemos garantizar la localidad de los datos y maximizar de ese modo el rendimiento. Nosotros sacábamos el tráfico de cada cola RSS, MQPollDevice, para posteriormente introducirlo en una de estas colas con afinidad CPU, CPUQueues, y de estas colas pasábamos a inyectarlo al dispositivo de usuario. Antes de poder hacer ninguna prueba es preciso crear en el árbol dev/ los nodos del dispositivo de carácter asociados a los elementos Click de dispositivo de usuario, ToUserDevice.

Por otra parte, para poder acceder de forma paralela al tráfico resulta necesario definir tantos hilos como colas RSS se hayan definido. El procedimiento consiste en abrir desde cada hilo el dispositivo de carácter de usuario correspondiente a su cola RSS y leer los paquetes, como si se tratara de un fichero, recibiéndose de uno en uno. Es necesario un número de cores del doble del número de colas. Por ejemplo, para recibir de 4 colas RSS, son necesarios 4 cores para la lectura desde la

tarjeta (MQPollDevice) y 4 cores para llevar los paquetes a nivel de usuario (ToUserDevice). Por tanto, en la batería de experimentos realizados para la arquitectura Intel no hemos podido probar mas allá de 4 colas RSS para ser justos en la evaluación. Sin embargo, para la arquitectura NUMA de AMD, disponíamos de 32 cores y se pudieron realizar las pruebas previstas: hasta 8 colas RSS. De cualquier forma, para los experimentos comunes se puede apreciar un rendimiento ligeramente superior para la arquitectura Intel en detrimento de AMD. Es cierto que esto era previsible dado que la tarjeta de red y su driver (en el que se basan tanto RouteBricks como PacketShader) han sido desarrollados por Intel.

Se puede apreciar en las figuras 4 y 6, una tendencia general por la cual la tasa de tráfico capturado crece (y las pérdidas decrecen) con el numero de colas RSS. Sin embargo esa tendencia no es estricta y para ambas arquitecturas se puede apreciar como una mayor cantidad de colas RSS y procesos de captura no siempre garantizan una mejor captura/menor tasa de pérdidas. Estimamos que esto se debe a factores arquitecturales: problemática de la gestión multicore, posibles contenciones a varios niveles como el PCIe, el gestor de interrupciones, etc. Este efecto también se observa en el caso de PacketShader para el escenario de paquete pequeño.

Las pruebas con tráfico sintético de tamaño mínimo inmediatamente reflejan una tasa de captura relativamente lejana a los 7.5 Gb/s (cota superior máxima en la práctica para paquetes de 64 bytes [17]). Aún así, para la arquitectura Intel, con 2 o más colas RSS se tiene una tasa de captura aproximada de 3.2 Gb/s, o dicho de otro modo 6.3 M paquetes/s. La arquitectura AMD, por su parte, ofrece unos resultados ligeramente inferiores a Intel para el mismo número de colas RSS. Sin embargo, con 8 colas y 8 procesos de captura nos ha permitido llegar a una tasa de 5 Gb/s, unos 9.5 M paquetes/s. Sin lugar a dudas, el cuello de botella en cuanto al caudal en paquetes/s se encuentra en el elemento ToUserDevice de Click, aún en fase experimental y no optimizado para su uso en redes de tan elevada tasa de paquetes por segundo: es evidente que realizar una llamada al sistema por paquete resulta altamente ineficiente. La extracción de tráfico a espacio de usuario depende específicamente de este elemento, y por tanto no se pueden aprovechar al máximo las modificaciones de tratamiento de paquetes por lotes que introduce RouteBricks. Teniendo en cuenta que para una distribución de tráfico real el tamaño medio de paquete es de aproximadamente 500-600 bytes [14], podemos ver como para ese tamaño de paquete el volumen de tráfico perdido es nulo para la arquitectura Intel, y muy bajo para AMD (siempre inferiores al 4 %). Parece por tanto aún realista su aplicación para procesado de tráfico en enlaces de 10 Gb/s reales.

Como refleja la Fig. 5, esta solución hace un uso tanto intensivo como extensivo de la CPU y sus cores involucrados en el proceso de captura y procesado de tráfico. El caso peor se manifiesta cuando intentamos procesar los ~ 15 M paquetes/s de 64 bytes del generador de tráfico, en este caso siempre existe tráfico que procesar en el dispositivo de carácter por lo que los procesos de usuario en ningún momento se hallarán ociosos, saturando al $100\,\%$ el consumo de CPU para cada núcleo participando en la captura. Cabía esperar este elevado consumo puesto que al ofrecer el dispositivo de carácter un único paquete por lectura, es necesario hacer peticiones constantes al mismo para extraer el tráfico disponible. A medida que el tamaño de paquete crece, en lineas generales se puede apreciar un descenso en el consumo de CPU. A mayor tamaño, se tiene una menor tasa de paquetes por segundo, lo que posibilita sacar todo el tráfico de cada cola RSS, algo que evidencian las pérdidas prácticamente nulas. Para la arquitectura Intel, sin embargo, se puede apreciar como el consumo de CPU para 4 cores y colas RSS es superior al que se tiene para 3 cores y colas. Esto se puede deber a que el beneficio que aporta el incremento en número de cores se ve perjudicado por la problemática de la localidad de datos (en el caso de 4 colas ningún proceso de captura Click está en el mismo micro que su proceso de nivel de usuario correspondiente). Por el contrario, en el caso de AMD con arquitectura NUMA, siempre podemos garantizar que los procesos de captura y proceso correspondientes comparten el mismo microprocesador.

IV-B. Evaluación de PacketShader

La instalación y configuración de PacketShader está bien documentada en la página de los autores [18]. Los dos parámetros básicos a configurar son, por un lado, el número de colas RSS a utilizar y, por otro, el tamaño del lote de paquetes que son transferidos desde la tarjeta al espacio de usuario en cada lectura. Tras unos experimentos preliminares, pudimos establecer en 128 paquetes el tamaño que menor tasa de pérdidas y menor consumo de CPU generaba para distintos valores de número de colas RSS.

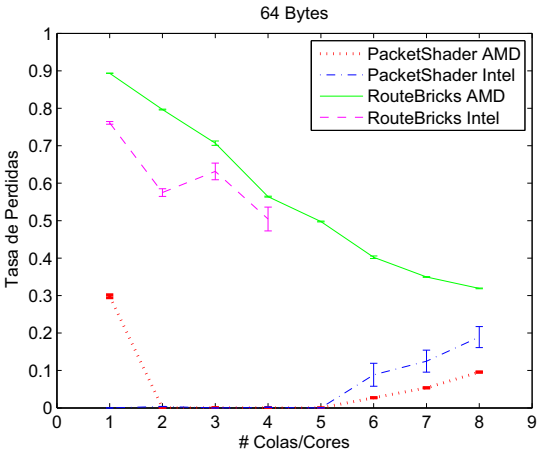
Para poder contar con marcado de tiempo en cada paquete a nivel más bajo y, por tanto, más preciso que desde nivel de usuario, modificamos el driver y la estructura interna que usa PacketShader para guardar los paquetes, de tal modo que cada vez que se recibe un paquete, se marca en el mismo driver con la función del kernel do_gettimeofday[5].

Para evaluar PacketShader, modificamos uno de los scripts de ejemplo que se proveen con el código de PacketShader, para que realizara la captura de paquetes, en lotes de 128, y realizara mínimas labores de conteo (paquetes y bytes por segundo).

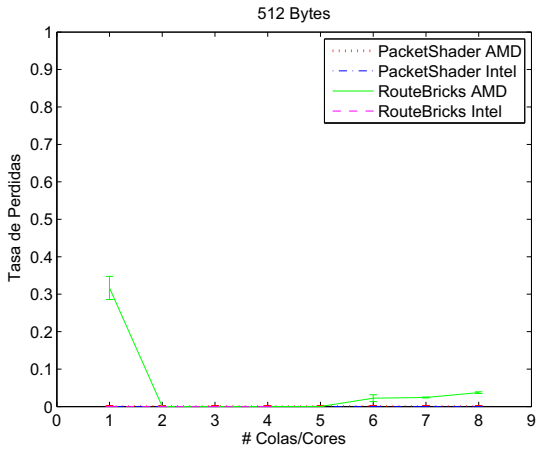
En la Fig. 4(a), en la que se refleja la tasa de pérdidas para el escenario de tamaño de paquete 64 bytes, se observa que para el caso de una cola se obtienen unas pérdidas en torno al $30\,\%$. Esto se debe a que el único core de recepción está saturado (ver la utilización de la CPU en la Fig. 5). Para un número de colas entre 2 y 5, se consigue una tasa de pérdidas prácticamente nula (con valores promedio en torno a 10^{-4}). A partir de 6 colas RSS, se observa un deterioro del rendimiento. Esta situación, es similar a lo observado con RouteBricks para tamaño de paquete 512 bytes. Este efecto puede ser debido a contenciones en el bus PCI, en el controlador de interrupciones o el gestor multihilo. Tanto para el caso de tamaño de paquete de 512 bytes como de 1500 bytes, las prestaciones de PacketShader son excelentes, obteniéndose prácticamente 0 pérdidas (tasas de pérdidas promedio del orden de 10^{-4}) para cualquier número de colas.

Al igual que con RouteBricks, el consumo de CPU es decreciente con el número de colas, como puede observarse en

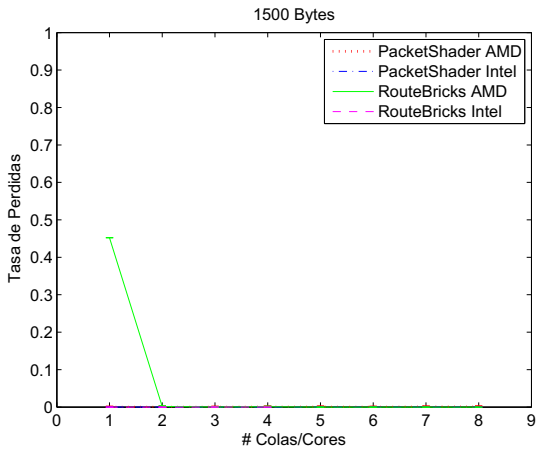
[5] http://linux.die.net/man/2/gettimeofday



(a) Tamaño de paquete 64 bytes

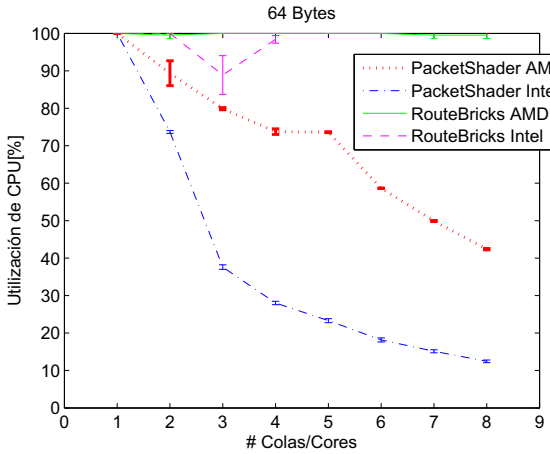


(b) Tamaño de paquete 512 bytes

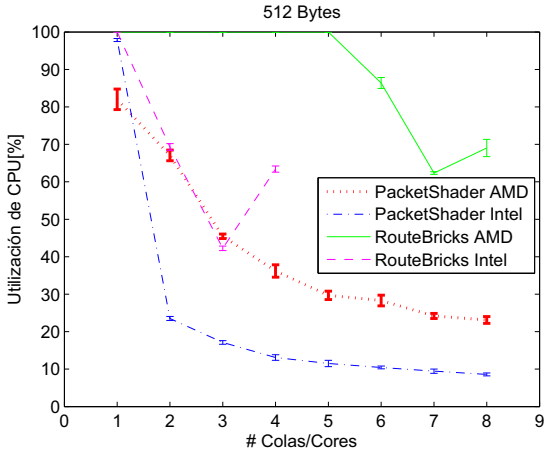


(c) Tamaño de paquete 1500 bytes

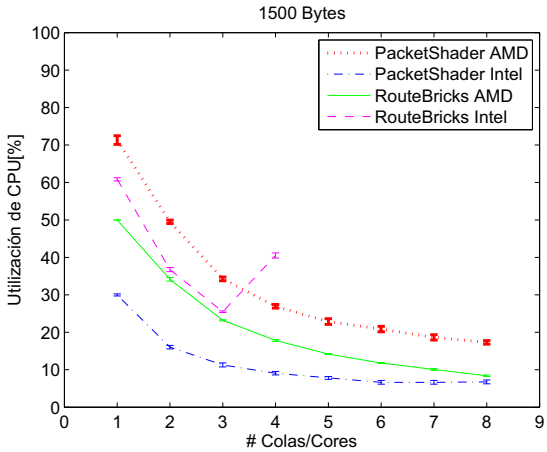
Figura 4. Pérdidas de paquetes



(a) Tamaño de paquete 64 bytes



(b) Tamaño de paquete 512 bytes

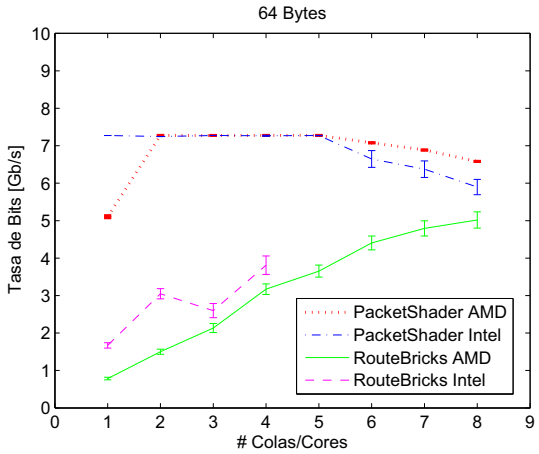


(c) Tamaño de paquete 1500 bytes

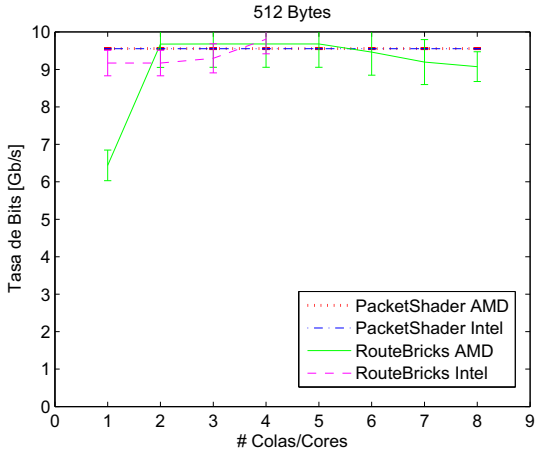
Figura 5. Utilización de CPU

la Fig. 4. Además, aunque con ambas arquitecturas se obtienen resultados casi idénticos en términos de pérdidas y caudal, se puede observar en la Fig. 5 que la arquitectura Intel hace una menor utilización de la CPU que en el caso de AMD. Esto se debe a que al compartir fabricante tanto el micro como la tarjeta de red, el driver hace un uso más eficiente de los recursos.

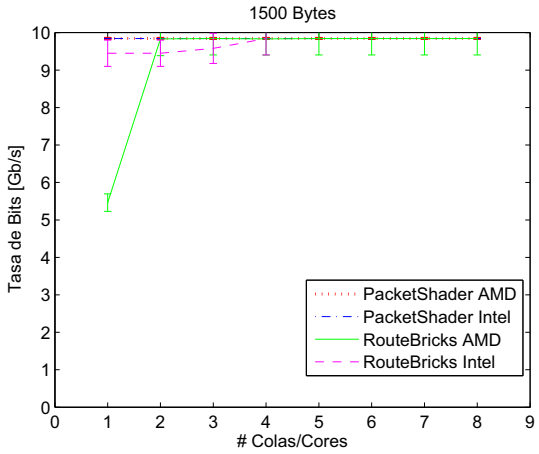
Como conclusión de la batería experimental podemos sacar que PacketShader logra unas prestaciones superiores a RouteBricks debido a que el trasiego de datos desde la NIC hasta el espacio de usuario es más directo y, además, en el caso de PacketShader se agrupan varios paquetes en la misma transferencia. En particular, elegimos como caso óptimo el motor de captura PacketShader, la arquitectura Intel y la configuración con número de colas RSS igual a 4.



(a) Tamaño de paquete 64 bytes



(b) Tamaño de paquete 512 bytes



(c) Tamaño de paquete 1500 bytes

Figura 6. Tasa de bits

V. AP-CAP y casos de uso

Como hemos visto en la sección anterior, configurando convenientemente alguno de los motores de captura propuestos en la literatura (a saber, PacketShader o Routebricks), somos capaces de obtener, desde nivel de usuario y en servidores de propósito general, el tráfico capturado en una interfaz de 10 Gb/s. Sin embargo, estos paquetes que capturamos llegan a nivel de usuario *en crudo*, en un formato no estándar, dependiente del motor de captura que tengamos por debajo.

Por esta razón, hemos desarrollado un framework que nos permite envolver el motor de captura, proveyendo al usuario de una API de programación en lenguaje C similar al estándar de facto pcap. Este framework, lo hemos llamado AP-CAP: CAPtura de Altas Prestaciones. De este modo, las primitivas del API pcap (e.g. de apertura de interfaz, pcap_open, o lectura de paquetes, pcap_next) han sido sustituidas por su versión correspondiente en AP-CAP (e.g. apcap_open o apcap_next).

Para evaluar tanto la integrabilidad del framework como la capacidad del mismo en enlaces de alta velocidad, hemos implementado varias herramientas: por un lado, dos herramientas de monitorización de tráfico de uso generalizado como son un capturador de paquetes en formato pcap (similar a tcpdump [6])y un generador de registros en formato Netflow [7]; por otro lado, hemos integrado dos herramientas de clasificación de tráfico como son Skypeness (un clasificador de tráfico Skype propuesto por los autores en [9]) y un detector de tráfico VoIP (SIP/RTP). Para la evaluación, se ha elegido como motor de captura PacketShader (con 4 colas para captura) y como arquitectura Intel. Se ha elegido este caso por ser el que menos CPU utilizaba consiguiendo una tasa de pérdidas mínima (en torno a 10^{-4}) para cualquier tamaño de paquete. Dicha evaluación se ha llevado a cabo en el entorno de pruebas con tráfico real de la Fig. 3.

La Fig. 7 muestra los resultados en términos de tasa de bits capturados, tasa de paquetes capturados y pérdidas para las cuatro aplicaciones. Se obtienen tasas de captura cercanas a 9.2 Gb/s y 2 M paquetes/s que es el límite de nuestro generador de tráfico. Las pérdidas son exactamente cero en todos los casos, menos en dos puntos donde no superan $5 \cdot 10^{-4}$. Cabe destacar que, en este caso, entre la captura y el proceso ocupan los 8 cores de la máquina, de tal modo que cualquier otro proceso (e.g. del sistema operativo) puede ocasionar un cambio de contexto de alguno de los hilos de AP-CAP. Esta puede ser la causa de las pérdidas espúreas observadas. En cuanto a la utilización de la CPU, ha sido superior a la obtenida en los experimentos de la sección anterior, obteniéndose una carga por core promediada entre los 8 cores utilizados (4 para captura y 4 para proceso) del 60 %. Esto se debe al mayor proceso necesario para la copia y proceso posterior de los paquetes: volcado, separación de flujos o detección.

Hay que notar que el volcado de paquetes, flujos y estadísticas no se hace a disco duro (lo que crearía un cuello de botella al tener capacidad inferior al enlace que estamos monitorizando) sino a memoria, usando el sistema de ficheros tmpfs [8]. Para poder volcar los paquetes a un medio no volátil, sería necesario el uso de tecnologías RAID o discos de estado sólido, en los que la tasa de escritura fuera superior a los 10 Gb/s del enlace de red.

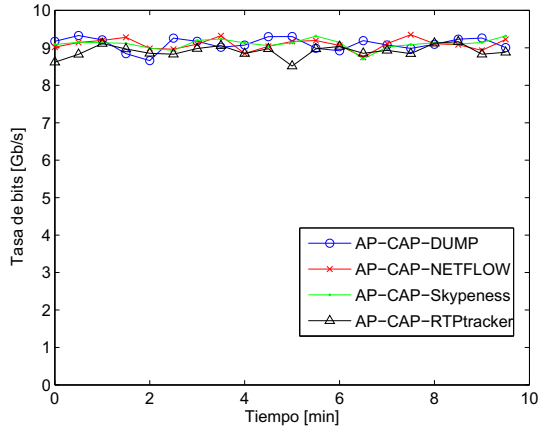
VI. Conclusiones

Este artículo ha evaluado la viabilidad de monitorizar enlaces de red de 10 Gb/s de capacidad utilizando hardware de propósito general. Las ventajas de estos sistemas están

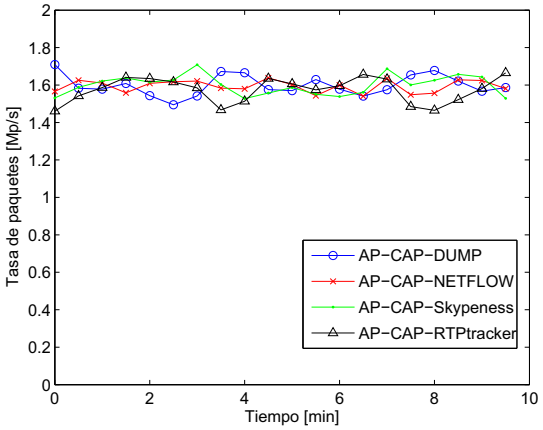
[6] http://www.tcpdump.org

[7] http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html

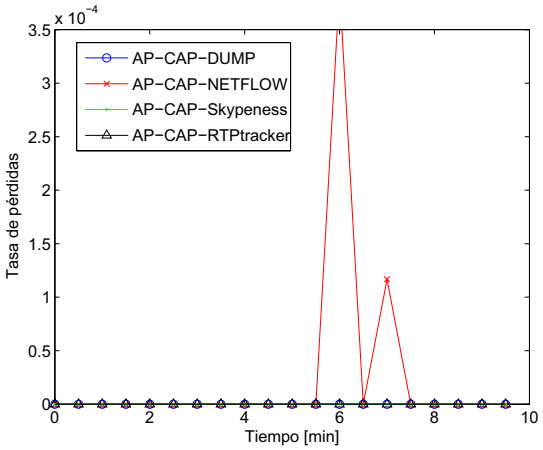
[8] http://compute.cnr.berkeley.edu/cgi-bin/man-cgi?tmpfs+7



(a) Tasa de bits



(b) Tasa de paquetes



(c) Pérdidas

Figura 7. Evaluación del framework con la aplicación de volcado de paquetes

en su menor precio y gran flexibilidad cuando son comparadas con soluciones basadas en hardware dedicado. Como primer paso de esta evaluación hemos estudiado y comparado las opciones ya implementadas por la comunidad científica. Dos han sido las propuestas evaluadas en detalle en dos arquitecturas hardware distintas, esto es, Intel y AMD. Estas propuestas han mostrado resultados satisfactorios en cuanto a la tasa de recepción del driver de la tarjetas de red, sin embargo, presentaban ciertas limitaciones en su interacción con el nivel del aplicación. De este modo, este artículo, además, ha presentado AP-CAP un framework que aúna alto rendimiento mientras respeta el interfaz de programación para captura de paquetes estándar de facto que es pcap. El rendimiento de AP-CAP ha sido evaluado en tareas propias de la monitorización de redes como son la identificación de aplicaciones y el volcado de tráfico. Los resultados han mostrado que estas tareas pueden ejecutarse a tasa de 10 Gb/s con pérdidas de paquetes mínimas, esto es, inferiores a $5{\cdot}10^{-4}$ en el caso peor. Este trabajo, consecuentemente, muestra a los operadores y gestores de red que el uso hardware de propósito general puede resultar de gran interés ante el reto que supone monitorizar enlaces de alta velocidad.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el Ministerio de Educación y Ciencia mediante el proyecto *ANFORA* (TEC2009-13385).

REFERENCIAS

[1] M. Crovella and B. Krishnamurthy, *Internet measurement: infrastructure, traffic and applications*, John Wiley and Sons Inc., New York, USA, 2006.

[2] Cisco, "Cisco carrier routing system," http://cisco.com/en/US/products/ps5763/index.html.

[3] K. Argyraki, S. Baset, B.-G. Chun, K. Fall, G. Iannaccone, A. Knies, E. Kohler, M. Manesh, S. Nedevschi, and S. Ratnasamy, "Can software routers scale?," in *Proceedings of the ACM workshop on Programmable routers for extensible services of tomorrow*, Seattle, USA, Aug. 2008, pp. 21–26.

[4] S. Han, K. Jang, K. Park, and S. Moon, "Building a single-box 100 Gbps software router," in *Proceedings of IEEE Workshop on Local and Metropolitan Area Networks*, New Jersey, USA, May 2010, pp. 1–9.

[5] M. Dobrescu, N. Egi, K. Argyraki, B.-G. Chun, K. Fall, G. Iannaccone, A. Knies, M. Manesh, and S. Ratnasamy, "Routebricks: Exploiting parallelism to scale software routers," in *Proceedings of ACM Symposium on Operating Systems Principles*, Big Sky, USA, Oct. 2009, pp. 15–28.

[6] S. Han, K. Jang, K.S. Park, and S. Moon, "Packetshader: a GPU-accelerated software router," *SIGCOMM Comput. Commun. Rev.*, vol. 40, pp. 195–206, Aug. 2010.

[7] F. Fusco and L. Deri, "High speed network traffic analysis with commodity multi-core systems," in *Proceedings of ACM Conference on Internet measurement*, Melbourne, Australia, Nov. 2010, pp. 218–224.

[8] TCPDUMP & LIBPCAP, "Libpcap," http://www.tcpdump.org/.

[9] P.M. Santiago del Río, J. Ramos, J.L. García-Dorado, J. Aracil, A. Cuadra-Sánchez, and M. Cutanda-Rodríguez, "On the processing time for detection of Skype traffic," in *Proceedings of Workshop on Traffic Analysis and Classification*, Istanbul, Turkey, July 2011.

[10] Intel, "PCIe* GbE controllers open source software developer's manual," http://download.intel.com/design/network/manuals/316080.pdf.

[11] R. Morris, E. Kohler, J. Jannotti, and M.F. Kaashoek, "The Click modular router," *SIGOPS Oper. Syst. Rev.*, vol. 33, pp. 217–231, Dec. 1999.

[12] L. Deri, J. Gasparakis, P. Waskiewicz, and F. Fusco, "Wire-speed hardware-assisted traffic filtering with mainstream network adapters," in *Proceedings of Workshop on Network Embedded Management & Applications*, Niagra Falls, Canada, Oct. 2010.

[13] RouteBricks, "How to build a Routebricks server," http://routebricks.org/code.html.

[14] K. Thompson, G.J. Miller, and R. Wilder, "Wide-area Internet traffic patterns and characteristics," *IEEE Network*, vol. 11, no. 6, pp. 10–23, 1997.

[15] "Tcpreplay suite," http://tcpreplay.synfin.net/.

[16] kc Claffy, D. Andersen, and P. Hick, "The CAIDA anonymized 2009 Internet traces," http://www.caida.org/data/passive/passive_2009_dataset.xml.

[17] IEEE P802.3ae 10Gb/s Ethernet Task Force, "IEEE Std 802.3ae," Oct. 2007, http://grouper.ieee.org/groups/802/3/ae/public/index.html.

[18] PacketShader, "Packet I/0 engine," http://shader.kaist.edu/packetshader/io_engine/index.html.

Sesión 4.A
Protocolos y técnicas para redes no convencionales: MANET, VANET, WSN

Enrutamiento Basado en Conectividad Multi-hop en Redes Ad-hoc Vehiculares

Michele Rondinone y Javier Gozalvez
Ubiquituos Wireless Communications Research Laboratory
Uwicore, http://www.uwicore.umh.es
Universidad Miguel Hernández de Elche
Avenida de la Universidad, s/n 03202 Elche
mrondinone@umh.es, j.gozalvez@umh.es

***Resumen*- El enrutamiento de datos en redes ad-hoc de comunicaciones vehiculares ha sido recientemente abordado con técnicas que eligen dinámicamente los caminos de enrutamiento en base a estimaciones de la densidad de vehículos en distintas calles obtenidas en tiempo real. La mayoría de estas técnicas utilizan métodos de encaminamiento *multi-hop* en los que el poseedor actual del paquete elige el próximo retransmisor (esquemas de retransmisión *sender-based*). Sin embargo, la estimación de la densidad vehicular puede necesitar una considerable sobrecarga de comunicaciones, y las retransmisiones *multi-hop* del tipo *sender-based* pueden aumentar la utilización de enlaces radio poco fiables. En este contexto, este artículo presenta un novedoso protocolo de enrutamiento basado en contención en el que los caminos *multi-hop* se eligen dinámicamente en base a la conectividad que proporcionan. Los resultados obtenidos demuestran que, gracias a su diseño y a la optimización de sus parámetros, la técnica propuesta en este artículo obtiene unos altos niveles de entrega de paquetes al destino utilizando de manera eficiente el canal de comunicaciones inalámbrico.**

***Palabras Clave*- Redes Ad-hoc de Comunicaciones Vehiculares; Enrutamiento; Conectividad Multi-hop; Retransmisión Basada en Contención**

I. INTRODUCCIÓN

Las redes ad-hoc de comunicaciones vehiculares (en inglés *Vehicular Ad-hoc Networks* o VANETs) utilizan comunicaciones radio para mejorar la seguridad vial y la eficiencia del tráfico a través del intercambio dinámico de información entre vehículos y con la infraestructura de comunicaciones. Las comunicaciones vehiculares se basan normalmente en los estándares IEEE 802.11p y WAVE (Wireless Access for Vehicular Environments) que están siendo adaptados a nivel europeo por la ETSI (European Telecommunications Standards Institute) [1]. En las VANETs, la información acerca de situaciones de tráfico locales (p.e. atascos o accidentes) puede ser transmitida a vehículos posicionados en zonas lejanas de la red de carreteras utilizando comunicaciones *multi-hop* a través de vehículos retransmisores intermedios. Sin embargo, la eficacia de las transmisiones *multi-hop* depende considerablemente de los protocolos de enrutamiento que se utilicen, y de los desafíos planteados por la alta movilidad de los vehículos y las adversas condiciones de propagación en entornos vehiculares. Estos desafíos han sido abordados de manera distinta por los diferentes tipos de protocolos de enrutamiento vehiculares presentados en la literatura hasta la fecha. Las propuestas más recientes intentan seleccionar los caminos *multi-hop* de manera dinámica basándose en estimaciones del estado del tráfico en tiempo real. Con este objetivo, las técnicas propuestas seleccionan calles caracterizadas por una alta densidad vehicular para asegurar la presencia de un número suficiente de vehículos retransmisores. Sobre los caminos *multi-hop* seleccionados, estas técnicas utilizan normalmente esquemas de retransmisión denominados en inglés *sender-based*, lo cual representa situaciones en las que el nodo actualmente encargado de transmitir el paquete de datos (*sender*) elige cómo próximo retransmisor entre sus nodos vecinos aquel que más progreso proporciona hacia el destino. En este contexto cabe destacar que la estimación de la densidad vehicular de forma distribuida y en tiempo real puede generar una importante sobrecarga de comunicaciones, comprometiendo así la eficiente utilización del medio inalámbrico [16]. Además, para aumentar su fiabilidad los métodos de retransmisión *sender-based* pueden necesitar la utilización de contramedidas a expensas de un incremento de la sobrecarga de comunicaciones o de la complejidad de la propuesta [12]. Para superar estas limitaciones, este artículo presenta un novedoso esquema que usa un método de retransmisión *broadcast* basado en contención (*contention-based*) y que elige los caminos *multi-hop* de forma dinámica estimando su conectividad. Como se demostrará en el artículo, gracias a una adecuada optimización de sus parámetros, esta propuesta puede proporcionar un buen rendimiento en términos de paquetes de datos entregados al destino, garantizando a la vez una gestión eficiente del canal de comunicaciones.

II. ESTADO DEL ARTE

Los estándares de comunicaciones vehiculares utilizan mensajes *broadcast* periódicos (*beacons*) para informar a los vehículos vecinos sobre la posición geográfica de un vehículo. Los esquemas denominados en inglés *greedy forwarding* aprovechan esta información para seleccionar como retransmisores aquellos vehículos que proporcionan más progreso hacia el destino final de los paquetes de datos. Como ejemplos básicos de este enfoque se pueden citar los protocolos Greedy Perimeter Stateless Routing (GPSR) [3] y Contention-Based Forwarding (CBF) [4]. Estas propuestas pueden sufrir el problema del "máximo local" cada vez que

ISBN: 978-84-694-5948-5

un paquete alcanza un nodo que no tiene vecinos que ofrecen más progreso hacia el destino que él. Este problema tiene una relevancia particular en escenarios urbanos, donde la presencia de edificios puede "ocultar" a los retransmisores óptimos, generando situaciones de máximo local más a menudo [5]. En estos casos, un protocolo podría intentar un proceso de recuperación de la retransmisión, generando un coste adicional de recursos de comunicaciones, o interrumpir el enrutamiento, causando un empeoramiento de la tasa de paquetes entregados. Para superar estos problemas se han propuesto protocolos que, como Spatially Aware Routing [6], gracias a la ayuda de mapas digitales, encaminan los paquetes a través de caminos geográficos fijos constituidos por un conjunto de intersecciones intermedias que conectan origen y destino de la transmisión. Sobre estos caminos, dirigen el paquete de datos hacia los vehículos que se encuentran en intersecciones intermedias para proporcionar una visión más completa de los posibles retransmisores en cada dirección. Sin embargo, la propuesta SAR selecciona el conjunto de intersecciones basándose únicamente en la menor distancia entre origen y destino, lo cual puede empeorar de manera significativa la correcta entrega de paquetes si el camino escogido no proporciona una adecuada conectividad por falta de vehículos.

Para evitar situaciones de este tipo, y facilitar las comunicaciones *multi-hop*, Vehicle-Assisted Data Delivery (VADD) [7] propone encaminar los paquetes a través de caminos geográficos caracterizados por una alta densidad de vehículos. Para detectar dichos caminos, VADD supone la utilización de sistemas GPS (Global Positioning System) capaces de proporcionar una caracterización estadística de la densidad vehicular de las calles. Aunque este enfoque podría resultar valido en término medio, no puede asegurar la necesaria conectividad *multi-hop* en cada momento, especialmente en el caso en que se produzcan cambios inesperados en la distribución de los flujos de tráfico. Propuestas como Landmark Overlays for Urban Vehicular Routing Environments (LOUVRE) [8] reaccionan a estas situaciones gracias al empleo de información de tráfico en tiempo real. En LOUVRE, los vehículos "miden" en tiempo real las densidades vehiculares en entornos locales y diseminan esta información a través de mensajes periódicos para obtener un mapa de la conectividad de la red de carreteras compartido por todos los nodos. Sin embargo, aunque proporcione un buen rendimiento, LOUVRE requiere una considerable sobrecarga de comunicaciones para mantener el mapa de conectividad actualizado. SADV (Static node Assisted adaptive Data dissemination protocol for Vehicular networks) [9] encamina los paquetes de datos a través de nodos estáticos posicionados en cada intersección. Los vehículos diseminan las estimaciones del tiempo que los paquetes necesitan para ser entregados entre dos intersecciones adyacentes, de manera que el protocolo puede adaptar la selección de los caminos de enrutamiento a la distribución del tráfico de vehículos en la red de carreteras. No obstante esta capacidad adaptativa, SADV se basa en la hipótesis inverosímil de que existen nodos estáticos desplegados en cada intersección. El protocolo Improved Greedy Traffic Aware Routing (GyTAR) [10] actualiza dinámicamente los caminos de enrutamiento cada vez que se recibe un paquete en una intersección. GyTAR selecciona la próxima intersección hacia la que el paquete tiene que ser dirigido considerando el progreso que la intersección proporciona hacía el destino final, y la densidad vehicular estimada a través de la técnica IFTIS (Infrastructure-Free Traffic Information System) [11].

La mayoría de los protocolos de enrutamiento mencionados adoptan mecanismos de retransmisión *sender-based* en los que el paquete se transmite de manera *unicast* a los nodos vecinos que más progreso proporcionan hacia el destino. Este tipo de retransmisión puede reducir el número de *hops* y la latencia, pero también puede provocar la utilización de enlaces radio que no siempre garantizan fiabilidad y estabilidad y por lo cual incrementan la sobrecarga debida a retransmisiones [12]. De otro lado, los métodos de retransmisión *contention-based* transmiten los paquetes de manera *broadcast*. Los nodos receptores activan mecanismos de contención según los que el próximo retransmisor se elige entre ellos de forma distribuida. Este enfoque obliga a varios nodos a recibir y procesar el mismo paquete, pero también incrementa la probabilidad de que en cada *hop* por lo menos un nodo retransmita el paquete. Además de CBF, también los protocolos CBRP (Contention-Based Routing Protocol) [13] y CLA-S (Connection-Less Approach for Streets) [14] aplican el método *contention-based*. CBRP supone el uso de nodos estáticos en las intersecciones para informar a los vehículos acerca de los caminos de enrutamiento más fiables para utilizar. Por otro lado, CLA-S introduce el concepto de "*forwarding area*" (area de retransmisión) como el conjunto de calles paralelas e intersecciones en las que el paquete tiene que ser replicado. La existencia de múltiples caminos paralelos aumenta la posibilidad de encontrar, entre ellos, un camino *multi-hop* conectado de origen a destino, pero también incrementa la sobrecarga de comunicaciones. Más recientemente, el protocolo BRAVE (Beacon-less Routing Algorithm for Vehicular Environments) [15] ha sido propuesto para encaminar paquetes de datos de manera dinámica utilizando el método *contention-based*. BRAVE calcula los caminos de enrutamiento más convenientes según una métrica arbitraria, pero permite cambiar estos caminos a lo largo del proceso de enrutamiento en caso de que falten vehículos retransmisores en ellos. Sin embargo, dichos cambios sólo se basan en información acerca del entorno local de un nodo, sin evaluar si existe conectividad *multi-hop* en un rango mayor.

III. DESCRIPCIÓN DE TOPOCBF

Para aprovechar de los beneficios ofrecidos por los esquemas de retransmisión basados en contención y de los protocolos que utilizan información de tráfico en tiempo real en sus decisiones de enrutamiento, este artículo presenta TOPOCBF (Road Topology-Aware Contention-Based mechanism). De manera parecida a GyTAR, TOPOCBF selecciona dinámicamente los caminos de enrutamiento en las intersecciones, pero en vez de estimar la densidad vehicular, basa sus decisiones en la conectividad *multi-hop* estimada gracias al algoritmo DiRCoD (Distributed and Real Time Communications Road Connectivity Discovery mechanism) [16]. Como se demuestra en [16], la estimación de la conectividad *multi-hop* de las calles requiere una cantidad de sobrecarga de comunicaciones considerablemente más baja que la estimación de la densidad vehicular. Además, las

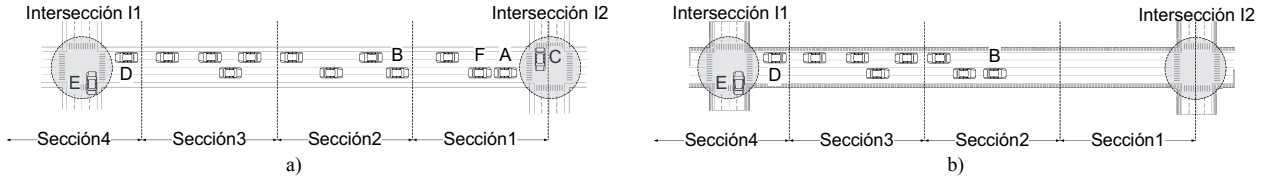


Figura 1. Calle con conectividad *multi-hop* total (a), y con conectividad *multi-hop* parcial (b)

decisiones de enrutamiento basadas en la conectividad *multi-hop* pueden ayudar a distribuir espacialmente y balancear la sobrecarga debida a paquetes de ruteo en la red de carreteras, evitando que los paquetes sean transmitidos siempre a través de las calles más ocupadas, siendo estas las qué más podrían sufrir una congestión del medio inalámbrico.

A. DiRCoD

DiRCoD utiliza comunicaciones vehiculares para calcular la conectividad *multi-hop* de una calle y notificar esta información a las intersecciones que la delimitan. Para su funcionamiento, utiliza mensajes *beacon* periódicos, actualmente en proceso de definición en los estándares internacionales [2]. Para explicar el algoritmo, considérense calles como las de la Figura 1 delimitadas por dos intersecciones I1 e I2. En este escenario un vehículo E que entra en las intersección I1 requiere la información acerca de la conectividad de la calle en la dirección de I2 para decidir si conviene encaminar un paquete en esta dirección o en otra. Con este objetivo, DiRCoD incluye campos adicionales de pocos bits denominados *Connectivity Fields* (CFs) en los *beacons* transmitidos por los vehículos que se encuentran en la calle. Los CFs indican el estado de conectividad *multi-hop* de la calle. Para representar su conectividad, DiRCoD divide la calle en secciones numeradas con valores crecientes según sus distancias de I2 y con una longitud igual al rango de comunicación medio de los vehículos. DiRCoD define la "distancia virtual" que separa I1 de I2 como el numero de secciones (o *hops*) entre I2 y el vehiculo más cercano a I2 que puede ser alcanzado desde I1 a través de transmisiones *multi-hop*. La Figura 1b) representa una calle que ofrece una conectividad *multi-hop* parcial. En este caso, la distancia virtual detectada en I1 para I2 es de 2 *hops*, ya que un paquete transmitido desde I1 sólo puede alcanzar un vehículo que se encuentra a 2 *hops* de distancia de I2. Por el contrario, en la Figura 1a) se representa una calle con conectividad *multi-hop* total. En este caso, la distancia virtual entre I1 e I2 es '0', ya que el paquete puede atravesar toda la calle y alcanzar I2 a través de transmisiones *multi-hop*.

Los CFs de DiRCoD se añaden a los *beacons* sólo en aquellos vehículos situados en la parte interior de la calle, y no en aquellos que se encuentran en el interior de las zonas circulares alrededor de las intersecciones ("*intersection zones"* marcadas en gris en la Figura 1). Un vehículo añade un CF que indica la sección de la calle en la que se encuentra actualmente a menos que no detecte (consultando su tabla de vecinos actualizada a través de recepciones de mensajes *beacons*) la presencia de otros vehículos más cercanos que él a la intersección I2, o dentro de la *intersection zone* I2 misma. Considerando la Figura 1b), el vehículo B indica en su *beacon* una distancia virtual de '2', ya que detecta que no hay otros vehículos más cercanos que él a I2. Por contra, el vehículo F en la Figura 1a) (que en principio añadiría un CF de '1' en su *beacon*) no incluye ningún CF ya que detecta la presencia del vehículo C en I2. El vehículo F incluye un CF que indica '0' sólo tras la recepción de un *beacon* del vehículo C en I2. Asimismo, el vehículo B posicionado en la sección 2 de la Figura 1a) incluye un CF de '0' sólo tras la recepción del *beacon* emitido por F con un CF de este mismo valor. A través de este proceso secuencial, los CFs son retransmitidos hacia I1. Los vehículos posicionados en I1 recibirán un mensaje *beacon* con un CF de '0' que indica conectividad *multi-hop* total en la Figura 1a), y un CF de '2' que indica conectividad *multi-hop* parcial en la Figura 1b).

Sin embargo, si todos los vehículos en la parte interior de la calle incluyeran un CF en sus *beacons*, las estimaciones de conectividad generadas por DiRCoD serían redundantes, lo cual podría comprometer la escalabilidad del algoritmo. Para evitar este problema, DiRCoD define un mecanismo basado en contención en el que sólo uno entre todos los nodos que reciben un CF lo retransmiten hacia I1 [16]. Además, DiRCoD ha sido diseñado para controlar el periodo entre dos evaluaciones consecutivas de la conectividad de la calle. Para ello se define el "*connectivity field generation period*" como el tiempo que los vehículos tienen que esperar antes de que puedan volver a competir para generar o retransmitir nuevos CFs. En este contexto, si el tráfico en la calle no varía con rapidez, el *CF generation period* podrá ser fijado a un valor más alto, de manera que la frecuencia de las medidas de conectividad será menor, y por consiguiente se reducirá la sobrecarga en el canal de comunicaciones.

B. TOPOCBF

TOPOCBF es una evolución del protocolo CBF (Contention-Based Forwarding) [4], diseñado para utilizar la información sobre la conectividad *multi-hop* de las calles proporcionada por DiRCoD en sus decisiones de enrutamiento dinámicas. En CBF, los paquetes de datos se retransmiten a través de transmisiones *broadcast*. Los nodos receptores activan un temporizador cuya duración es inversamente proporcional al progreso proporcionado hacia el destino final. Al caducar este temporizador, el paquete es retransmitido por el nodo más cercano al destino. Al "escuchar" la retransmisión del paquete, los nodos con el temporizador activo suprimen sus intentos de retransmisión. Los resultados contenidos en [12] demuestran que CBF consigue tasas de entrega de paquetes más altas que los protocolos básicos que utilizan esquemas de retransmisión del tipo *sender-based*. Como los autores indican, los esquemas *sender-based* necesitan muchos intentos de transmisión para combatir la perdida de paquetes, y requieren que los *beacons* se intercambien con una frecuencia relativamente alta para detectar los retransmisores fiables.

Aprovechando el buen rendimiento y la sencillez del esquema de retransmisión de CBF, TOPOCBF se ha diseñado para dirigir los paquetes de manera iterativa hacia

puntos intermedios (en este caso intersecciones) posicionados en el camino para alcanzar el destino final, y para escoger de forma dinámica la próxima calle en base a su conectividad *multi-hop*. Una vez se haya elegido la próxima calle, el objetivo de TOPOCBF es alcanzar el vehículo que más esté cerca de la próxima intersección utilizando un esquema de *greedy forwarding*. La selección de dicho vehículo permitirá una mejor visión de la conectividad *multi-hop* de las calles adyacentes y por lo tanto una mejor selección de la próxima intersección. Para alcanzar un vehiculo en una intersección, TOPOCBF utiliza un sistema de retransmisión del tipo *contention-based*, para el que es necesario que los paquetes incluyan las coordenadas geográficas del centro de la intersección en un campo adicional denominado *"next intersection field"*. Para seleccionar la próxima intersección, un vehículo posicionado en la intersección a la que el paquete se dirige actualmente utiliza la información de conectividad *multi-hop* de las intersecciones adyacentes proporcionada por DiRCoD. El proceso de selección de intersecciones consecutivas se repite hasta que se alcanza un vehículo que puede contactar directamente con el destino final del paquete.

Tomando como ejemplo el escenario de la Figura 1, supóngase que el vehículo E en I1 necesita encaminar un paquete hacia un destino D. TOPOCBF selecciona la próxima intersección analizando los siguientes requisitos:

1. Progreso hacia el destino final. Solo se consideran aquellas intersecciones que proporcionan un progreso hacia el destino final D;

2. Validez temporal de la información de conectividad de una calle. Al entrar en la *intersection zone* de I1, el vehículo E procesa los *beacon* recibidos para analizar los CFs que se refieren a las intersecciones adyacentes. A la hora de encaminar un paquete de datos, E comprueba los instantes en los que se recibieron los últimos CFs relativos a las intersecciones que cumplen el requisito 1. En caso de que la información de un CF relativo a una de estas intersecciones sea más antigua que el *CF generation period* definido por DiRCoD, E interpretaría que la calle que conduce a aquella intersección no proporciona conectividad *multi-hop*[1]. Por lo tanto, en su selección de la próxima intersección, E consideraría como posibles candidatas sólo aquellas intersecciones para las que el último CF se haya recibido durante el último período de CET segundos. CET es el acrónimo de "*connectivity expiry time*", definido por TOPOCBF y que debe ser superior al *CF generation period*.

3. Estado de la conectividad multi-hop de la calle. Según DiRCoD, las calles adyacentes a una intersección proporcionan conectividad *multi-hop* total o parcial dependiendo de la distancia virtual contenida en los CF recibidos. Si más de una intersección adyacente cumple las propiedades *1.* y *2.*, el vehículo E selecciona como próxima intersección aquella que proporcione la minima distancia virtual. En el caso en dos (o más) intersecciones ofrezcan la misma distancia virtual, E selecciona la próxima intersección entre ellas de manera aleatoria. Por lo contrario, si ninguna de las intersecciones adyacentes cumple las condiciones mencionadas arriba, el paquete se pierde.

Una diferencia importante entre CBF y TOPOCBF es que, en su esquema de retransmisión *contention-based*, CBF activa temporizadores cuya duración depende del progreso que los nodos receptores proporcionan hacia el destino final del paquete. Por el contrario, TOPOCBF calcula la duración del temporizador de retransmisión t_A en función del progreso proporcionado por un vehículo receptor *A* hacia la próxima intersección *Int* a la que se dirige el paquete:

$$t_A = \begin{cases} t_{\max}\left(1-\left(\frac{p_A}{p_{\max}}\right)\right) & si \quad d_{S-Int} > p_{\max} \\ t_{\max}\left(1-\left(\frac{p_A}{d_{S-Int}}\right)\right) & en \quad otro \quad caso \end{cases} \qquad (1)$$

En la formula (1), p_{max} indica el progreso máximo que un nodo receptor puede proporcionar al vehículo *S* del cual se ha recibido el paquete y por lo tanto corresponde rango de comunicaciones máximo de *S*. t_{max} es la máxima duración que puede tener el temporizador de retransmisión. En el caso en

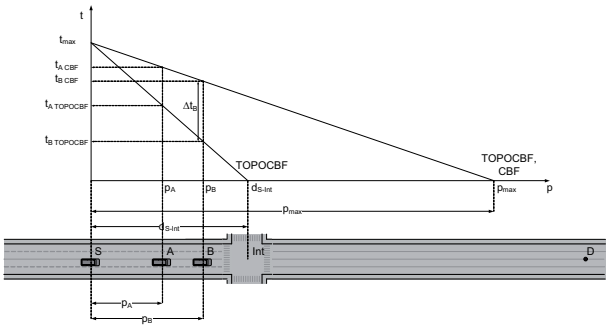


Figura 2. Calculo de la duración del temporizador de retransmisión en TOPOCBF y CBF

que la distancia que separa el transmisor *S* de la intersección Int sea mayor de p_{max}, se adopta la primera línea de la ecuación (1)[2]. Por el contrario, si *Int* se encuentra dentro del máximo rango de comunicación de S, entonces el máximo progreso que un nodo receptor podrá proporcionar hacia la intersección será la distancia $d_{S\text{-}Int}$ que separa *S* de *Int*. Por lo tanto, en la computación de la duración del temporizador de retransmisión se adoptará el valor $d_{S\text{-}Int}$, como se indica en la segunda línea de la ecuación (1). Este último caso se ha representado esquemáticamente en la Figura 2. Como puede apreciarse, la fórmula (1) posibilita que TOPOCBF tenga duraciones del temporizador de retransmisión inferiores cuando son calculadas en las proximidades de las intersecciones a las que se dirige el paquete, y por lo tanto evita cumular retrasos innecesarios.

Cabe también destacar que TOPOCBF reduce el problema sufrido por CBF de crear caminos de retransmisión paralelos entre origen y destino del paquete en escenarios urbanos y

[1] Por lo explicado en el apartado III.A, si la calle proporciona conectividad *multi-hop* parcial o total, el vehiculo E recibe los CFs cada *CF generation period* segundos.

[2] La primera línea de la ecuación (1) también corresponde a la fórmula con la que CBF calcula la duración de su temporizador de retransmisión.

suburbanos cuando se dan condiciones de ausencia de línea de visión directa entre transmisores y receptores [5]. La naturaleza *broadcast* incontrolada de CBF causa que los paquetes se repliquen sobre caminos de enrutamiento paralelos hacía el destino, generando un exceso de carga de comunicación redundante. TOPOCBF elimina este problema ya que los paquetes se dirigen desde una intersección hacia la próxima seleccionada. Por lo tanto, los vehículos que reciben paquetes en calles que no conducen hacia la intersección seleccionada, los descartan por no proporcionar progreso hacía ella.

IV. Evaluación del Rendimiento

A. Entorno de Evaluación

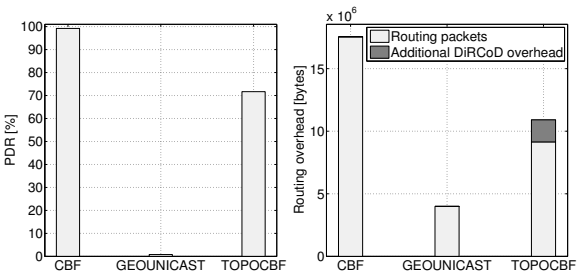
Las prestaciones de TOPOCBF han sido investigadas a través de simulaciones en la plataforma iTETRIS (an Integrated Wireless and Traffic Platform for Real-Rime Road Traffic Management Solutions, http://ict-itetris.eu/) y su simulador de comunicaciones vehiculares implementado en ns-3 (http://www.nsnam.org/). TOPOCBF ha sido comparado con CBF y con el protocolo para transmisiones *geounicast* implementado en iTETRIS por defecto, denominado en lo sucesivo como GEOUNICAST. GEOUNICAST es un ejemplo simple de esquema del tipo *sender-based* especificado por el proyecto GeoNet [17] y que se corresponde con el método *greedy forwarding* utilizado en GPSR. En GEOUNICAST, un nodo elige cómo próximo retransmisor el vecino que más esté cercano del destino final del paquete; los vecinos candidatos son todos aquellos registrados en la tabla de vecinos durante 5 segundos desde la recepción del último *beacon* recibido. La implementación de los paquetes utilizados por CBF y GEOUNICAST se corresponde exactamente al formato definido por ETSI para transmisiones de tipo *geonetworkin*g [2]. Basándose en este formato, TOPOCBF incluye 8 bytes adicionales a sus paquetes para el *next intersection field*. En este artículo, la implementación de DiRCoD supone un *CF generation period* de 2s y un CF con una longitud de 4 bits. Las simulaciones reproducen el enrutamiento de un paquete *geounicast* (mensajes de notificación) entre un vehículo originador y una RSU fija cómo destino. El tamaño de la carga útil de los mensajes de notificación a nivel de aplicación se ha fijado a 300 bytes. Los mensajes de notificación se generan con una frecuencia de 1Hz durante 1000s, lo cual constituye el tiempo simulado.

Utilizando el simulador de tráfico SUMO (Simulation of Urban Mobility http://sourceforge.net/apps/mediawiki/sumo/), se ha creado un escenario urbano del tipo Manhattan con 6 calles horizontales y 6 verticales que se cruzan en intersecciones y forman segmentos de calles de longitud de 300m en las que la máxima velocidad permitida es 50 km/h. Los resultados descritos en este apartado corresponden a una densidad vehicular media de 11 vehículos por kilómetro por carril. Basándose en la clasificación definida por la compañía Skycomp [19], dicha densidad vehicular correspondería a un nivel de servicio (en inglés *Level-of-Service* o LOS) de 'C' correspondiente a condiciones de tráfico moderadas. La métrica LOS ha sido propuesta en el Highway Capacity Manual (HCM) [18], y proporciona una medida para describir las condiciones de funcionamiento de la infraestructura de tráfico. Skycomp extiende la propuesta de HCM para el caso escenarios urbanos y define un sistema para categorizar el LOS en base al número medio de vehículos en los pelotones que se forman en las calles.

En las simulaciones, los vehículos comunican entre ellos utilizando interfaces radio ETSI ITS G5A [1], la adaptación europea del estándar IEEE 802.11p/WAVE. Debido al importante impacto de las de la propagación radio sobre las comunicaciones vehiculares, en este trabajo se han considerado condiciones de propagación realistas (incluyendo atenuación de la señal y desvanecimiento lento y rápido) a través del modelado de propagación urbano en micro-celdas en el ancho de banda de 5 GHz desarrollado en el proyecto europeo WINNER [20], incluido en iTETRIS. La potencia de transmisión utilizada ha sido fijada a 20dBm (100mW). Con esta potencia de transmisión en el modelado de propagación radio utilizado, dos vehículos separados por 200m pueden comunicarse con una tasa de recepción media de paquetes de 70%. Teniendo en cuenta este dato, se ha escogido el valor de 200m como rango de comunicaciones medio para fijar la longitud de las secciones de calles utilizadas por el algoritmo DiRCoD. Esta selección conservadora permite al algoritmo proporcionar estimaciones estables de la conectividad *multi-hop*, que no se ven afectadas por condiciones de propagación esporádicas favorables.

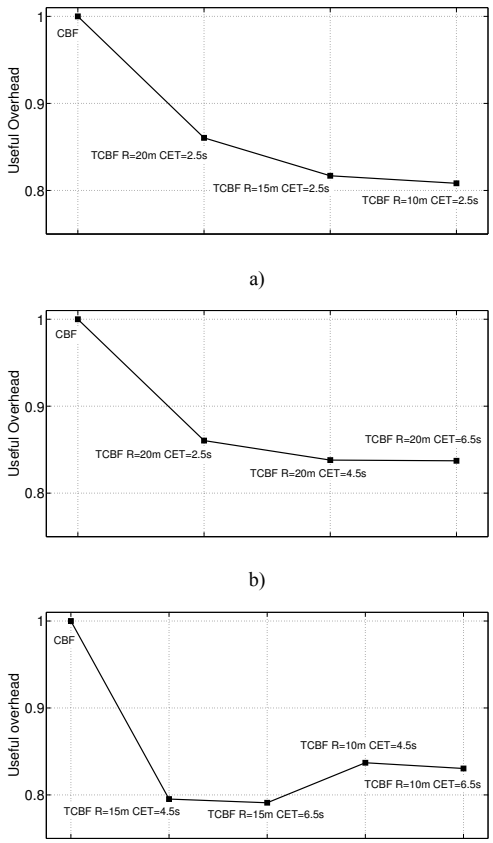
B. Resultados

La Figura 3 compara las prestaciones obtenidas por los tres protocolos de enrutamiento. La Figura 3a) representa la tasa de paquetes entregados (en inglés *Packet Delivery Rate* o PDR) al destino, y la Figura 3b) la sobrecarga de enrutamiento, que se define como la cantidad total de información transmitida en el canal de comunicaciones para encaminar los paquetes (en inglés *Routing overhead*). Como se demuestra en [5], CBF duplica los paquetes en las intersecciones y crea caminos de enrutamiento paralelos. Debido a esto obtiene un excelente PDR que se acerca al 99%. Por lo contrario, GEOUNICAST presenta unas muy bajas prestaciones en términos de PDR debido a la elección de nodos retransmisor con enlaces poco fiables, resultado de su esquema de retransmisión *sender-based*. Los resultados de la Figura 3 para TOPOCBF se han obtenido considerando un radio R de la *intersection zone* de DiRCoD de 20m y un *connectivity expiry time* CET de 2.5s. Aunque no utilice caminos paralelos como CBF, TOPOCBF es capaz de obtener un PDR de 72%, con un 38% de sobrecarga de enrutamiento menos con respecto a CBF. La sobrecarga de enrutamiento de CBF distingue entre la transmisión de paquetes de enrutamiento y los bytes adicionales utilizados por el algoritmo DIRCoD e incluidos en los *beacons*. En este contexto, es interesante analizar el compromiso entre PDR y sobrecarga de comunicaciones representado en la Figura 4 a través de la métrica "sobrecarga útil" (en inglés *useful overhead*), definida como el ratio entre la sobrecarga generada por cada protocolo y la PDR obtenido. Esta métrica mide lo útil que ha sido la sobrecarga utilizada por un protocolo a lo largo del proceso de enrutamiento en términos de PDR conseguido. En principio, cuanto menor sea la sobrecarga útil, más eficiente será la sobrecarga de enrutamiento introducida por un protocolo. La figura 4 sólo representa la sobrecarga útil para los protocolos CBF y TOPOCBF, ya que en el protocolo GEOUNICAST, el nivel



a) b)

Figura 3. Comparación de la PDR (a) y de la sobrecarga de enrutamiento (*routing overhead*) (b)



a)

b)

c)

Figura 4. Sobrecarga útil (*useful overhead*) en CBF y TOPOCBF en función del radio R de la *intersection zone* de DiRCoD (a), del *connectivity expiry time* CET (b), y de valores combinados de R y de CET (c)

muy reducido de PDR conlleva un nivel de sobrecarga útil muy alto. Para facilitar la comparación, la métrica representada ha sido normalizada por la sobrecarga útil de CBF. La figura 4 indica que con su configuración inicial (R=20m y CET=2.5s), TOPOCBF reduce de un 14% la sobrecarga útil de CBF, y por lo tanto se puede considerar más eficiente.

Para mejorar las prestaciones y la operación de TOPOCBF, este artículo analiza la optimización de algunos de sus parámetros de diseño, en particular R y CET. La optimización ha sido efectuada mediante simulaciones motivadas por consideraciones lógicas sobre el impacto que la variación de los distintos parámetros de diseño podían producir en las

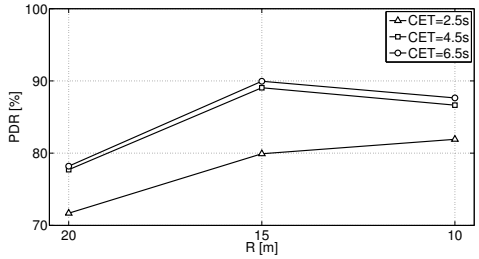


Figura 5: PDR de TOPOCBF en function de R y CET

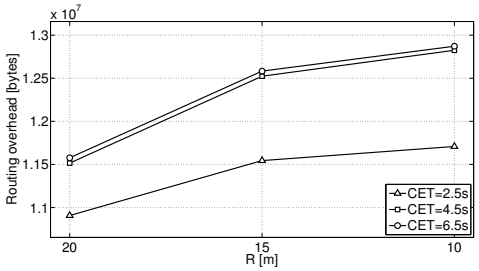


Figura 6. Sobrecarga de enrutamiento (*routing overhead*) de TOPOCBF en función de R y CET

prestaciones del protocolo. Las Figuras 5 y 6 representan los efectos causados al reducir el valor de R e incrementar el valor del CET, sobre la PDR y sobre la sobrecarga de enrutamiento. Los resultados obtenidos demuestran que, si se fija el valor de CET a 2.5s y se incrementa el radio R de la *intersection zone* de DiRCoD, se obtiene un incremento de la PDR (82% frente al 72% de la Figura 3). Esto se debe a que al reducir el radio R, hay más vehículos involucrados en la generación y la retransmisión de los CFs de DiRCoD en la parte interior de la calle. Por consiguiente, se reciben más CFs en las intersecciones y TOPOCBF pierde menos paquetes de enrutamiento en ellas, al no tener medidas de conectividad *multi-hop* de las calles adyacentes. De forma análoga, si se fija el valor de R a 20m, un incremento del CET, también produce un incremento de la PDR (hasta el 78%), ya que permite que TOPOCBF retransmita paquetes en las intersecciones aunque no disponga de información de conectividad que haya sido actualizada recientemente. Sin embargo, en ambos casos el incremento de la PDR se produce a expensas de una mayor cantidad de sobrecarga de enrutamiento (Figura 6). No obstante esto, las Figuras 4a) y 4b) demuestran que la subida de sobrecarga de enrutamiento no se traduce en un incremento de la sobrecarga útil. Estos resultados han motivado el análisis de los efectos que una combinación de valores bajos de R y altos CET pueden producir en las prestaciones de TOPOCBF. Como se muestra en la Figura 5, la mejor combinación de los parámetros de optimización se produce fijando R a 15m y CET a 6s, donde la PDR alcanza un valor de 89%. Cabe destacar que esta PDR, considerablemente cercana a la de CBF, se obtiene con una sobrecarga de enrutamiento mucho menor, como se demuestra observando las Figuras 3 y 6. Al utilizar un valor de R muy reducido combinado con valores de CET muy altos se provocan colisiones entre paquetes de enrutamiento y mensajes *beacon* de DiRCoD, lo cual conduce a una reducción de la PDR. En términos de sobrecarga útil, la Figura 4c) muestra que con una adecuada combinación de los parámetros de diseño, TOPOCBF reduce hasta un 21% la sobrecarga útil de CBF

V. CONCLUSIONES

Los protocolos de enrutamiento en entornos vehiculares han evolucionado hacia enfoques que seleccionan los caminos de enrutamiento de forma dinámica considerando la información de tráfico en tiempo real. Para evitar los costes de implementación y de sobrecarga de comunicaciones necesarios para la estimación de las densidades vehiculares, los autores de este artículo proponen basar las decisiones de enrutamiento en el nivel de conectividad *multi-hop* de las calles. En este contexto, se ha presentado TOPOCBF, un protocolo que utiliza un esquema de retransmisión del tipo *contention-based* y que utiliza estimaciones de la conectividad *multi-hop* de las calles para elegir dinámicamente los caminos de retransmisión. El artículo ha investigado además como configurar eficientemente el protocolo TOPOCBF para maximizar su rendimiento y su eficiencia de comunicaciones. Los resultados obtenidos han demostrado que el esquema propuesto puede obtener unas altas tasas de entrega de paquetes limitando a su vez la sobrecarga generada por el encaminamiento *multi-hop*.

Los autores se plantean como trabajo futuro una evolución de la técnica TOPOCBF para mejorar su eficiencia en la utilización del canal de comunicaciones. También se estudiará la viabilidad de TOPOCBF en diferentes escenarios de tráfico y con distintos parámetros de comunicaciones.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el Ministerio de Industria, Turismo y Comercio a través del proyecto INTELVIA (número de referencia: TSI-020302-2009-90), y por la Comisión Europea a través del proyecto FP7 iTETRIS: An Integrated Wireless and Traffic Platform for Real-Time Road Traffic Management Solutions (número del proyecto: FP7 224644). Los autores desean agradecer el apoyo recibido.

REFERENCIAS

[1] ETSI TC ITS, “Intelligent Transport Systems (ITS); European profile standard on the physical and medium access layer of 5 GHz ITS”, Standard ETSI ES 202 663 v1.1.0, Jan. 2010

[2] ETSI TC ITS, “Intelligent Transport Systems (ITS); Communications; Architecture; Vehicular Communications, Part 4: Geographical Addressing and Forwarding for Point-to-Point and Point-to-Multipoint Communications; Sub-part 1: Media-Independent Functionality”, Draft ETSI TS 102 636-4-1 v0.1.1, Febr. 2011

[3] B. Karp and H. T. Kung, “GPSR: greedy perimeter stateless routing for wireless networks”, Proc. of the ACM/IEEE 6th annual international conference on mobile computing and networking, MOBICOM’00, pp. 243–254, Aug. 2000

[4] H. Fussler, J. Widmer, M. Kasemann, M. Mauve, and H. Hartenstein, “Contention-based forwarding for mobile ad hoc networks”, Ad Hoc Networks, Elsevier, vol. 1, issue 4, pp.351-369, Nov. 2003

[5] J. Gozalvez, M. Sepulcre, and R. Bauza, “Impact of the radio channel modelling on the performance of VANET communication protocols”, Telecommunication Systems, Springer, pp. 1-19, 2010

[6] J. Tian, L. Han, K. Rothermel, “Spatially aware packet routing for mobile ad hoc inter-vehicle radio networks”, Proc. of IEEE Intelligent Transportation Systems 2003, vol.2, pp. 1546- 1551, Oct. 2003

[7] J. Zhao, G. Cao, “VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks”, IEEE Transactions on Vehicular Technology , vol.57, no.3, pp.1910-1922, May 2008

[8] K.C. Lee, M. Le, J. Harri, M. Gerla, “LOUVRE: Landmark Overlays for Urban Vehicular Routing Environments,” Proc. of the 68th IEEE Vehicular Technology Conference VTC Fall 2008, pp.1-5, Sept. 2008

[9] Y. Ding, C. Wang, and L. Xiao, “A static-node assisted adaptive routing protocol in vehicular networks”, Proc. of the 4th ACM international workshop on Vehicular ad hoc networks, pp. 59-68, 2007

[10] M. Jerbi, S.-M. Senouci, T. Rasheed, Y. Ghamri-Doudane, “Towards Efficient Geographic Routing in Urban Vehicular Networks,” IEEE Transactions on Vehicular Technology, vol.58, no.9, pp.5048-5059, Nov. 2009

[11] M. Jerbi, S.-M. Senouci, T. Rasheed, Y. Ghamri-Doudane, “An Infrastructure-Free Traffic Information System for Vehicular Networks”, Proc. of the IEEE 66th Vehicular Technology Conference, VTC-2007 Fall, pp.2086-2090, Oct. 2007

[12] H. Fussler, H. Hartenstein, J. Widmer, M. Mauve, and W. Effelsberg, “Contention-based Forwarding for Street Scenarios”, Proc. Of the 1st International Workshop in Intelligent Transportation, March 2004

[13] Tonghong Li, Yuanzhen Li, Jianxin Liao, “A Contention-Based Routing Protocol for Vehicular Ad Hoc Networks in City Environments”, Proc. of the 29th IEEE International Conference on Distributed Computing Systems, pp.482-487, June 2009

[14] A. Ho, Y.H. Ho, K.A. Hua, “A connectionless approach to mobile ad hoc networks in street environments”, Proc. of Intelligent Vehicles Symposium 2005, pp. 575- 582, June 2005

[15] P. M. Ruiz, V. Cabrera, J. A. Martinez, F. J. Ros, "BRAVE: Beacon-less Routing Algorithm for Vehicular Environments", Proc. Of Second IEEE International Workshop on Intelligent Vehicular Networks (InVeNet 2010), November 2010

[16] M. Rondinone and J. Gozalvez, “Distributed and Real Time Communications Road Connectivity Discovery through Vehicular Ad-hoc Networks”, Proc. of the 13th International IEEE Conference on Intelligent Transportation Systems, pp.1079-1084, Sept. 2010

[17] GeoNet Consortium, “D2.2 Final GeoNet Specification”, Jan. 2010

[18] “Highway Capacity Manual - HCM 2000”, Transportation Research Board, National Research Council, 2000

[19] Skycomp, Inc., Parsons Brinckerhoff Quade & Douglas, Inc., “Performance Ratings of Traffic Flows on Selected New York Metropolitan Area Highways Fall 2007”, 2007

[20] WINNER consortium, “D1.1.2. WINNER II channel models”, WINNER European Research project Public Deliverable, Sept. 2007

Mecanismos de descubrimiento en arquitecturas de gestión para redes malladas

Luis Francisco Díez, José Ángel Irastorza, Ramón Agüero, Luis Muñoz
Departamento de Ingeniería de Comunicaciones
Universidad de Cantabria
Plaza de la Ciencia s/n, 39005 Santander
{angel, ramon, luis}@tlmat.unican.es

***Abstract*—Las tareas de gestión que habitualmente han sido empleadas en entornos de red cableadas son también fundamentales para asegurar el buen funcionamiento de las denominadas *Redes Personales*, con unas características particulares que hacen que su gestión sea compleja. Entre todos los retos que hay que afrontar, uno que destaca sobremanera es el de asegurar su comportamiento autónomo, haciendo que se puedan calificar como redes auto-* gestionables/configurables/.... En este artículo se analiza, sobre una arquitectura de gestión jerárquica/distribuida para redes personales, el comportamiento de un mecanismo de descubrimiento mediante el que los agentes pueden localizar a los gestores, y asociarse con ellos. Para ello se ha llevado a cabo una implementación completa de dicha arquitectura en el marco del simulador *NS-2* (basada en *SNMP*), que incluye además los mecanismos para llevar a cabo el descubrimiento y asociación entre los nodos agentes y gestores.**

***Index Terms*—Gestión, Redes Personales, Mecanismos de Descubrimiento, Simulación**

I. Introducción

La evolución de los dispositivos y periféricos inalámbricos, junto con el desarrollo de las tecnologías de red inalámbricas que los interconectan, han sido cruciales para crear nuevos entornos de comunicación más versátiles, dinámicos y cercanos a la persona que los aportados por los tradicionales entornos de comunicación basados en las redes fijas. Este tipo de escenarios se basan en topologías multi-salto o malladas donde los nodos se caracterizan por su heterogeneidad, desde ordenadores portátiles (*smart-phones*, etc) hasta terminales mucho mas limitados en sus prestaciones, como actuadores o sensores; destacar la creciente relevancia de los despliegues de este tipo de dispositivos, en lo que se ha venido a llamar como *'la Internet de la Cosas'*.

Todos estos nuevos entornos de comunicaciones necesitan ser gestionados. La tarea de gestión debe ser capaz de conseguir una operación efectiva y eficiente de la red, tanto desde el punto de los recursos físicos como de los sistemas distribuidos que la conforman. A pesar de que se han estudiado ampliamente diferentes arquitecturas y modelos de gestión sobre infraestructuras de redes fijas, no ha sucedido lo mismo con redes como las que son objeto de estudio en este artículo, caracterizadas por: topologías de red dinámicas, anchos de banda limitados, enlaces poco fiables, colisiones inherentes al canal inalámbrico, limitaciones energéticas de las baterías en los nodos, descubrimiento de recursos o servicios en dichos entornos dinámicos, etc.

El diseño de un marco de gestión para este tipo de redes tendría que tener en cuenta las particularidades anteriormente expuestas. En particular, sería fundamental el establecimiento de un modelo de organización apropiado, esto es, una adecuada definición de los papeles de gestor-agente y su correcta disposición y selección entre los nodos de la red, de forma que la carga de gestión afecte lo menos posible al tráfico de datos, sin que la tarea de gestión y la disponibilidad/calidad del servicio requerido por los usuarios de la red, se vean penalizadas.

Los modelos de organización mas tradicionales son los modelos centralizados, que no son adecuados para este tipo de redes [1], ya que fueron pensados para ser implementados en redes fijas. Con el propósito de superar estos inconvenientes, se presenta un modelo de organización con una estructura distribuida y jerárquica de tres niveles; un gestor de nivel superior (nivel 1), que puede ser seleccionado entre un número de gestores de segundo nivel. Estos toman un papel de gestor local, controlando un conjunto de nodos, los cuales pueden entenderse como un cluster (caracterizado por algún tipo de conectividad entre sus componentes). Así, los agentes se localizan en el tercer nivel de la jerarquía. Aunque se definen tres niveles, solo existen dos planos de comunicación de gestión: uno conformado por los agentes y su correspondiente gestor (de segundo nivel), y otro que interconecta todos los gestores de segundo nivel entre ellos y el gestor global. Así, el nivel 2 actúa como una red superpuesta de gestores que se comunican de forma colaborativa. Esta propuesta distribuida/jerárquica permite que el subsistema de gestión adquiera una mayor fiabilidad y eficiencia, así como una menor sobrecarga, tanto en las comunicaciones como en los recursos de sistema [1].

Sobre el esquema de organización propuesto se abren dos líneas de investigación: una que estudia la asignación, mediante la definición de unas estrategias, del papel gestor a los nodos de la red; y una segunda que se ocupa de, una vez asignados los papeles de gestor, desarrollar y estudiar el comportamiento de los protocolos de descubrimiento necesarios para que los agentes localicen al mejor gestor y se asocien con él. Este trabajo, que se enmarca principalmente en esta línea de investigación, se organiza en base a la siguiente estructura: la Sección II presenta los antecedentes y motivación del estudio de los protocolos de descubrimiento sobre redes multi-salto o malladas, así como otros trabajos relacionados en este campo; la Sección III describe las estrategias elegidas para el despliegue de los gestores sobre la red, y los parámetros utilizados para comparar el comportamiento de cada una de ellas; la Sección IV describe el funcionamiento de los protocolos de descubrimiento que, son evaluados, mediante técnicas de simulación, en la Sección V. Finalmente la Sección VI presenta las principales conclusiones del trabajo.

ISBN: 978-84-694-5948-5

II. Antecedentes y trabajo relacionado

Es evidente la cada vez mayor relevancia que las topologías multi-salto, o malladas, están adquiriendo. A pesar de que la investigación en el ámbito de las redes multi-salto date ya de comienzos de siglo, canalizada principalmente a través de la actividad del grupo de trabajo Mobile Ad Hoc Networks (MANET) del IETF, en los últimos años se ha observado un cambio paulatino en la concepción que se solía hacer de este tipo de redes. Hay que tener en cuenta que inicialmente se postulaban como topologías que se establecían de manera espontánea (de ahí su nombre), en aquellas situaciones en las que por un motivo u otro no existiera una infraestructura subyacente (ejemplos de aplicación típicos que se solían utilizar era el bélico o un desastre natural), y en la que, además, se suponía que la red era muy dinámica, caracterizándose los nodos por una movilidad elevada. Esta concepción de escenarios y aplicaciones ha ido perdiendo importancia de manera paulatina y, a día de hoy, se piensa que las topologías malladas pueden aportar un conjunto de beneficios muy relevantes. Los operadores de red tradicionales pueden plantearse, por ejemplo, utilizar este tipo de despliegues para extender su área de cobertura de una manera económicamente eficiente. De hecho, los grupos de trabajo del IEEE que trabajan en el ámbito de tecnologías inalámbricas ya incorporan topologías multi-salto en sus especificaciones, como IEEE 802.11s [2] o IEEE 802.16j [3]. Asimismo, el uso de despliegues multi-salto se está analizando en el marco de las redes celulares del futuro, como LTE [4] o WiMax y también ha formado parte de TETRA [5].

Si tradicionalmente la importancia de las tareas de gestión es muy elevada para cualquier tipo de red, en las topologías malladas (inalámbricas) es, si cabe, mayor, ya que es necesario que las redes sean capaces de adaptarse a las condiciones cambiantes del entorno (auto-configurables) y que los recursos se utilicen (gestionen) de una manera eficiente, al ser más limitados que en redes cableadas [6].

Como se ha dicho anteriormente, el principal objetivo de este trabajo es el de analizar el comportamiento del mecanismo de descubrimiento para una arquitectura de gestión de una red mallada. En este sentido, no se trata de que un nodo encuentre un camino/ruta a un destino cualquiera (como es el caso de los protocolos de encaminamiento anteriormente citados), sino que sea capaz de localizar al gestor que mejor le convenga en función de una serie de parámetros. Para ello se parte de un análisis previo en el que se estudiaron diferentes estrategias para el despliegue de los nodos gestores, siendo, en este caso, el principal elemento de análisis el comportamiento específico de los mecanismos y protocolos diseñados para llevar a cabo dicho descubrimiento. Hay que destacar que se utilizará la misma nomenclatura que la empleada en el ámbito de las redes *ad-hoc* para diferenciar los dos procedimientos de búsqueda que se estudiarán: reactivo y preventivo (o proactivo); en el primero de ellos los gestores no anuncian su presencia y los agentes tienen que iniciar un proceso de búsqueda para localizarlos, mientras que en el esquema preventivo los nodos gestores anuncian su presencia a través de mensajes periódicos que se difunden por la red y los agentes utilizan la información obtenida de los mismos para determinar a cuál de ellos tratan de asociarse. De lo dicho anteriormente se desprende que este trabajo es más cercano a los que existen en el ámbito de descubrimiento de servicios y/o de *gateways*. Por ejemplo, en [7] los autores analizan el retraso y rendimiento de las comunicaciones entre los nodos y los *gateways*, pero no se hace mención a la manera en la que estos se desplegaron en la red. En [8], se estudian aspectos de seguridad en el envío de tráfico a un conjunto de *gateways*, que, en cualquier caso, están distribuidos de manera óptima en la red. Se utilizará un mecanismo similar al propuesto en [9] para seleccionar el mejor gestor (en base a una suma ponderada de figuras de mérito); sin embargo, a diferencia de todos los artículos anteriores, el objetivo de este trabajo es el de analizar de manera más concreta el comportamiento de las diferentes estrategias definidas para acometer el despliegue de los gestores, estudiando las ventajas e inconvenientes de las mismas y su influencia en el rendimiento de los mecanismos de descubrimiento (tiempo de asociación, sobrecarga del tráfico de descubrimiento, etc). Otro aspecto que guarda cierta relación con el trabajo que se presenta en este artículo es el de los mecanismos de descubrimiento de servicios (*Service Discovery Protocols*, SDP), especialmente su aplicación sobre redes multi-salto. En [10], [11] se describen las diferentes propuestas que se han hecho, los retos a los que es necesario enfrentarse. A pesar de que la complejidad inherente a los servicios es notablemente mayor, especialmente en términos de su descripción (ontologías) o arquitectura (*overlay*, uso de directorios), los dos trabajos ponen de manifiesto que también es posible distinguir entre modo de descubrimiento reactivo y preventivo. Además, en [11] se destaca el hecho de que la evaluación de los mecanismos de descubrimiento no es lo suficientemente madura. Uno de los pocos trabajos en los que se lleva a cabo una evaluación, de alguna manera, similar a la que se presenta en este artículo es [12], en el que los autores analizan la sobrecarga y el tiempo mínimo necesarios para localizar los servicios; sin embargo, estudian estrategias de mejora basadas en técnicas de *caching* o extensión de la búsqueda en anillo, mientras que en este trabajo se analizan las diferentes estrategias de despliegue de gestores (esto es, el efecto que sobre los mecanismos de descubrimiento tienen la situación particular de los nodos).

III. Estrategias de despliegue de gestores

Tal y como ha quedado de manifiesto, uno de los objetivos más importantes de este trabajo es el de analizar la influencia de diferentes estrategias de despliegue de los gestores en la red mallada. Dichas estrategias se han definido en base a un conjunto básico de tres parámetros que determinarían el comportamiento mejor o peor de la estrategia de despliegue, siendo los que se recogen seguidamente.

- *Probabilidad de cobertura*. Hace referencia a la probabilidad de que un agente cualesquiera pueda comunicarse con, al menos, un gestor, perteneciendo, de esa manera, a la arquitectura de gestión.
- *Número de saltos.* Una de las limitaciones que en mayor medida se le achacan a las topologías multi-salto es la interferencia adicional que pueden causar las comunicaciones de varios saltos. Por tanto, sería conveniente que, en la medida de los posible, la longitud de los caminos entre los agentes y sus gestores correspondientes fueran lo más cortas posibles.

- *Distribución de los agentes.* Con este parámetro se pretende caracterizar la bondad en la distribución de los agentes entre los gestores. En un caso óptimo, cada gestor tendría que tener el mismo número de agentes gestionados, mientras que en el peor de los supuestos, un gestor tendría todos los agentes, mientras que el resto no tendrían ninguno. En base a la diferencia relativa entre estas dos situaciones, se define el parámetro β como sigue.

$$\beta = \frac{1}{2\left(A_C - \frac{A_C}{M}\right)} \sum_m \left| A_m - \frac{A_C}{M} \right| \qquad (1)$$

  En donde A_C es el número total de agentes cubiertos, M el número de gestores y A_m el número de agentes atendidos por el gestor *m-ésimo*. Se puede comprobar que el parámetro β está restringido en el intervalo $[0, 1]$, siendo 0 el caso óptimo y 1 el peor.

En concreto se analizarán las cuatro estrategias que se presentaron en [1], y que aparecen brevemente descritas a continuación.

A. Estrategia 1: despliegue aleatorio

En este caso se asume que los gestores se despliegan de manera completamente aleatoria, sin ningún tipo de planificación previa. Esto refleja un escenario no deseado, ya que, dependiendo de la topología concreta de la red, se podría dar el caso de que hubiera gestores completamente aislados, sin ningún nodo en su área de cobertura.

B. Estrategia 2: despliegue óptimo geométrico

En este supuesto se asume que los gestores se sitúan en aquellos puntos que aseguran una cobertura (*geográfica*) máxima de toda el área bajo análisis, sin tener en cuenta la topología concreta del despliegue de nodos. Por esta razón, el escenario resultante no tiene porqué garantizar una cobertura máxima, ya que dependerá fuertemente de su posición concreta en cada topología de red analizada.

C. Estrategia 3: despliegue óptimo topológico

Se parte del conocimiento total de la topología de la red, escogiendo para el rol de gestor aquellos nodos que garanticen un *coste* global menor, entendiendo como coste el número de saltos necesarios para alcanzar a un gestor. Para resolver este problema se hace uso de la *p-mediana* [13], que establece un conjunto de *M* gestores para minimizar la siguiente función.

$$\sum_{j \in N} \left\{ \min_{i \in M} d_{ij} \right\} \qquad (2)$$

en la que N es el conjunto de nodos y d_{ij} es la distancia (en número de saltos) entre los nodos i y j.

D. Estrategia 4: despliegue sub-óptimo topológico

Una de las desventajas que tradicionalmente se le atribuye al método de la *p-mediana* es que su primer objetivo es el de cubrir todos los nodos (esto es, la demanda ha de ser satisfecha). En función de las características particulares de la topología de la red, podría darse el caso de que fuera más interesante dejar algunos nodos sin ser gestionados, con el fin de potenciar una distribución más ecuánime del resto de agentes. En este sentido se propone una sencilla modificación al algoritmo tradicional de la *p-mediana*, para que no se consideren aquellos sub-grafos con un tamaño inferior a ν nodos, siendo ν un parámetro de diseño, que tendrá que tener en cuenta el beneficio adicional que supone no gestionar a un conjunto de nodos y la pérdida (en probabilidad de ser gestionado).

IV. Protocolo de descubrimiento

Como ya se ha adelantado, el protocolo de descubrimiento incorpora dos modos de funcionamiento: proactivo, en el que los gestores anuncian periódicamente su presencia y además, de forma activa, realizan el mantenimiento de las asociaciones con los agentes y reactivo, en el que los agentes inician la búsqueda de gestores (que no anuncian su presencia) y toman un papel activo en el mantenimiento. El anuncio de presencia de los gestores, en el modo proactivo, y su búsqueda, en el reactivo, se realiza mediante tráfico *bradcast* que se propaga, haciendo uso de la topología multi-salto, a través, unicamente, de los agentes y, de acuerdo a los análisis previos referentes a la cobertura en función del número de saltos, se ha limitado el número de saltos a fin de evitar situaciones de inundación de tráfico; el resto de transmisiones son *unicast*. Por otro lado y con el fin de mantener actualizada la información recibida por el tráfico de difusión, una vez asignado el papel que va a desempeñar cada nodo, ya sea agente o gestor, se mantiene una tabla con todos los nodos de los que recibe información con un papel opuesto al suyo.

En cualquier caso, en ambos modos, son los agentes los que eligen, finalmente, al gestor con el que se intentarán asociar, mientras que los gestores confirman o rechazan la asociación en cada caso, dependiendo de si el número de agentes asociados previamente sobrepasa un límite preestablecido; esta limitación trata de evitar la congestión que se produciría en torno al nodo gestor. De esta forma, cada entrada de la tabla, ya sea de agentes o de gestores mantiene el estado correspondiente al nodo, pudiendo ser: asociado, desasociado o rechazado.

La elección del gestor por parte de los agentes se lleva a cabo a través de una función de coste compuesta por tres parámetros: número de saltos ($(P)_{\text{hops}}$), agentes asociados ($(P)_{\text{assAG}}$) y estado previo de la asociación del agente con el gestor (con este parámetro se pretende favorecer que se mantengan las asociaciones ya establecidas). Los agentes, tras recopilar información acerca de los gestores alcanzables, realiza, para cada una de las entradas de su tabla, la suma ponderada del coste de cada uno de los parámetros siguiendo la ecuación (3). Cada uno de los ω_j representa el peso relativo del parámetro j dentro de la función de coste, se ha establecido, además, que la suma de todos ellos debe ser 1.0; mientras que cada $(P)^i_j$ representa el valor del parámetro en cuestión para el gestor i;

$$f_{\text{cost}_i} = \max \sum_{j=0}^{C-1} \omega_j (P)^i_j \qquad (3)$$

$$(P)^i{}_{\text{hops}} = \frac{(\text{hops})^{\max} + 1 - (\text{hops})^i}{(\text{hops})^{\max}} \qquad (4)$$

$$(P)^i{}_{\text{assAg}} = \frac{(\text{assAg})^{\max} + 1 - (\text{assAg})^i}{(\text{assAg})^{\max}} \qquad (5)$$

Para modelar los parámetros de número de saltos y de agentes asociados (Ecs. 4 y 5) se hace uso de los valores máximos establecidos en cada caso, $(\text{hops})^{\max}$ y $(\text{assAg})^{\max}$, definiendo una relación lineal que toma el valor 1 en el mejor de los casos (un único salto y ningún agente asociado, respectivamente) y decae al aumentar el número de saltos o agentes hasta alcanzar el valor 0. En cuanto al tercer parámetro, el estado previo de asociación, únicamente tomará valores 1 ó 0, dependiendo de si el agente estaba previamente asociado al gestor o no; gracias a este parámetro se evitan inestabilidades en la asociación de un agente, que podría estar cambiando de gestor de manera indefinida.

Teniendo en cuenta la variabilidad de la topología de la red, bien por la movilidad o por la aparición o desaparación de nodos, es necesario que los agentes realicen los procesos de asociación de forma periódica, haciendo uso del temporizador T_{RMT} (*Timer Refresh Manager Table*). Ante la posibilidad del inicio no simultáneo de los nodos, pudiendo provocar que la información no se propague por la red, ambos protocolos implementan un mecanismo de *back-off* sobre el T_{RMT} que se activa, en los agentes, en caso de no tener conocimiento de ningún gestor.

Por otro lado, el inicio simultáneo de todos los nodos podría provocar situaciones de congestión en torno a los gestores, lo que hace necesario que se aleatorice el inicio del descubrimiento.

La Tabla I muestra la información contenida en cada uno de los tipos de paquetes de descubrimiento, cuyo significado se detallará en la explicación de cada uno de los modos del protocolo. En todos ellos se indica la dirección origen y destino del paquete, así como el número de secuencia, que permite descartar los paquetes recibidos por más de un camino, en el caso tráfico *broadcast*; los paquetes de asociación, y de su correspondiente mantenimiento, que surjan del envío de un determinado paquete *broadcast* mantienen el número de secuencia de éste. El número de saltos únicamente está presente en aquellos paquetes que pueden ser enviados de forma *broadcast* (*Manager Announcement* y *Manager Request*), con el fin de poder llevar a cabo el control de saltos; finalmente, el número de agentes asociados se envía en los paquetes *Manager Announcement*, al ser necesario para el cálculo de la función de coste.

A. Modo Proactivo

Como se ha dicho anteriormente en este modo de operación (Figs. 1 y 2), los gestores anuncian su presencia mediante el envío de tráfico *broadcast* (*Manager Announcement*) de forma periódica utilizando el temporizador T_{MA} e incrementando el valor del número de secuencia en cada nuevo envío. Se ha decidido, además, que el primer envío de este paquete se realice de forma aleatoria en un intervalo dado por MSI (*Manager Start Interval*), evitando, así, posibles problemas de

Tabla I
ESTRUCTURA DE LOS PAQUETES DE DESCUBRIMIENTO.

Tipo	$Orig.$	$Dest.$	$Saltos$	Seq	Ag_{asoc}
MA_{nn}	√	√	√	√	√
MR_{eq}	√	√	√	√	
AR_{eq}	√	√		√	
AR_{ej}	√	√		√	
AC_{onf}	√	√		√	

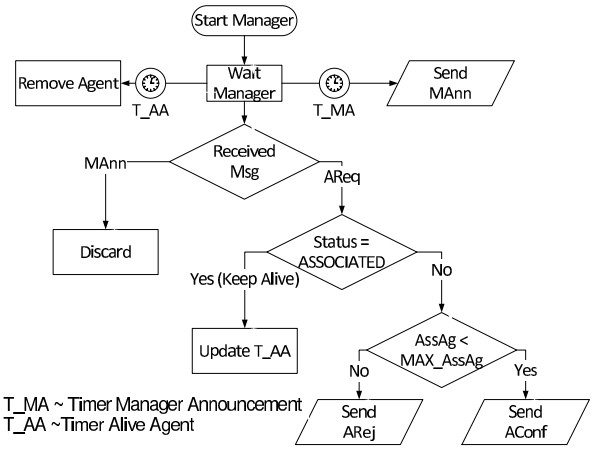


Fig. 1. Descripción del funcionamiento del papel de gestor en modo Proactivo

congestión. Los MA son propagados a través de la red multi-salto por los agentes (los gestores simplemente los descartan). Tras almacenar la información de los gestores, el nodo agente inicia el proceso de asociación, al expirar el temporizador T_{RMT}, cuyo primer valor se define como la suma del MSI y un tiempo aleatorio en el intervalo dado por ASI (*Agent Start Interval*); de este modo, el proceso de asociación consiste en seleccionar al mejor gestor, de los que mantiene en su tabla, y enviar una petición de asociación (*Association Request*) al mismo. En caso de que el gestor esté a una distancia mayor de un salto, la ruta por la que irá el paquete estará determinada por el protocolo de encaminamiento subyacente. Ante la recepción de esta solicitud el gestor seleccionado confirma o rechaza la asociación (*Association Confirm* o *Association Reject*) en función de si el número de agentes asociados ha llegado o no al máximo establecido.

Una vez completada la asociación, el mantenimiento se lleva a cabo mediante la respuesta del agente a la recepción de los sucesivos MA que envía su gestor. Al recibir este anuncio de presencia, el agente inicia un temporizador aleatorio de mantenimiento (T_{KA}, *Timer Keep Alive*) que, al expirar, provoca el envío de un AR_{eq}; conviene resaltar que, una vez establecida la conexión, los paquetes de petición de asociación tienen otro significado y son tratados por el gestor para garantizar la presencia de los agentes que supervisa. El temporizador T_{KA} evita el envío simultáneo de paquetes de mantenimiento por parte de los agentes, que se sincronizarían ante la recepción de los MA.

Es importante destacar que, en realidad, un nuevo proceso de asociación, tras la expiración del temporizador T_{RMT}, no genera necesariamente tráfico, ya que, únicamente, se comprueba si hay en la tabla de gestores uno mejor que aquel con el que se mantiene la asociación. Si no fuera así; esto es, si el mejor gestor siguiera siendo el mismo, no se llevaría a cabo ningún proceso adicional.

B. Modo Reactivo

En este caso (Figs. 3 y 4), el descubrimiento lo inician los agentes, quienes, al expirar su temporizador T_{RMT}, comienzan el proceso de búsqueda; el primer valor de este temporizador (al inicio de la simulación) toma un valor aleatorio dentro del intervalo ASI. A diferencia del modo proactivo, los

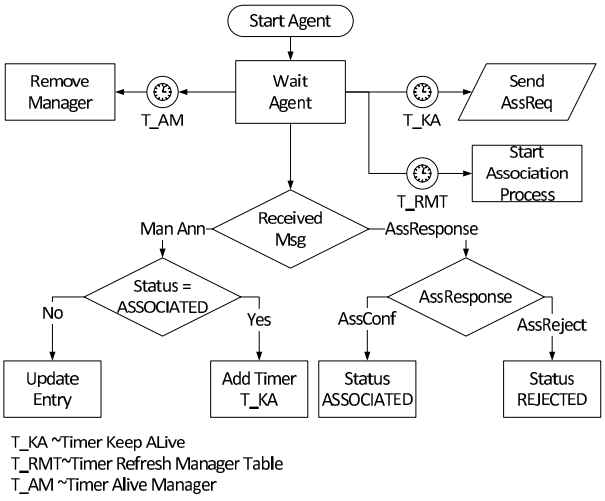


Fig. 2. Descripción del funcionamiento del papel de agente en el modo Proactivo

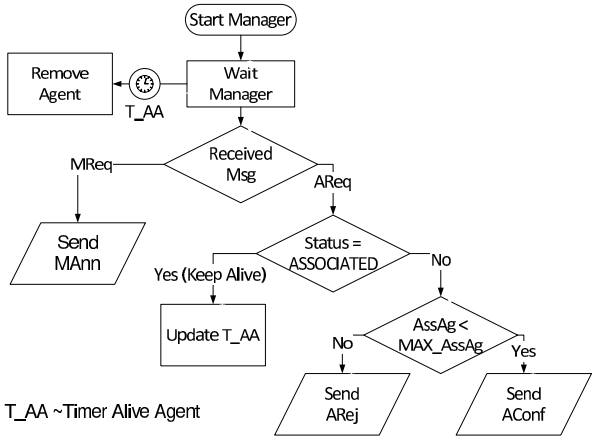


Fig. 3. Descripción del funcionamiento del papel de gestor en modo Reactivo

agentes obtienen la información de los gestores alcanzables durante este proceso de búsqueda, ya que los gestores no anuncian su presencia. Para ello, envían un paquete *broadcast* de petición de gestores (*Manager Request*), en cada nueva petición de gestor se incrementa el número de secuencia. Ante la recepción de los MR_{eq}, los gestores responden con un MA, que se envía al agente correspondiente mediante una transmisión *unicast*; de esta manera los agentes consiguen obtener la información de todos los gestores disponibles. A continuación, y tras esperar un tiempo suficiente como para recibir todos las posibles respuestas de los gestores alcanzables (T_{WMA}, *Timer Wait Manager Announcement*), el nodo agente es el encargado de seleccionar al mejor gestor (tal y como sucedía en el modo proactivo) y enviar la petición de asociación (AR_{eq}), a la que el gestor responderá de acuerdo al número de agentes que supervisa en ese momento en particular. Con la recepción de la confirmación de asociación el agente inicia un temporizador T_{KA} que utilizará para indicar el intervalo de mantenimiento de la asociación; así, al expirar, se envía un paquete AR_{eq} de mantenimiento, que es respondido por el gestor con un AC_{onf}.

A diferencia del esquema proactivo, al iniciarse cada nuevo proceso de asociación (T_{RMT}), el nodo agente no es conocedor de los gestores disponibles, por lo que debe reiniciar la búsqueda aún cuando no vaya a variar su asociación.

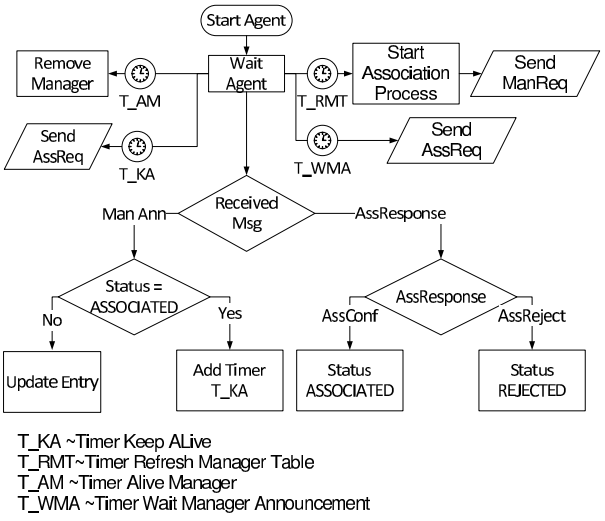


Fig. 4. Descripción del funcionamiento del papel de agente en el modo Reactivo

V. Resultados

En esta sección se presentan los principales resultados obtenidos durante la evaluación de las diferentes estrategias de asignación de los papeles de gestión, ya definidas en la Sección III (medidas estáticas), así como del comportamiento intrínseco de los protocolos de descubrimiento (medidas dinámicas).

Para la realización de las medidas estáticas se han utilizado dos procedimientos complementarios: el primero es un análisis fundamental basado en un simulador propietario y el segundo un estudio basado en el simulador de red $NS2$. Los resultados del análisis fundamental se han presentado con gran detalle en [1], lo que permite compararlos con los obtenidos mediante simulación de red, con lo que se consigue validar la implementación llevada a cabo en el simulador $NS2$ y, por lo tanto, poder extraer conclusiones de unas medidas que el simulador propietario no podría llevar a cabo. Los parámetros en que se van a centrar las medidas estáticas van a ser la probabilidad de cobertura y el parámetro β, descrito en la Sección III. Las medidas dinámicas comparan las prestaciones, en cuanto a tráfico y tiempos, de los dos modos de funcionamiento del protocolo de descubrimiento; dicho estudio es posible gracias a la utilización del simulador $NS2$, en el que se ha integrado la arquitectura de gestión propuesta.

Para la realización de las medidas se han elegido los siguientes parámetros de partida: 80 nodos desplegados de forma aleatoria sobre un escenario cuadrado de 100 metros de lado y cada nodo con un radio de cobertura, de la tecnología de comunicaciones subyacente, de 15 metros. Partiendo de la topología básica de 80 nodos se ha variado el número de gestores N, asignándoles sobre el escenario según las distintas estrategias y realizando 100 simulaciones independientes de 600 segundos por cada una de las combinaciones de gestores/agentes y estrategias, con el fin de asegurar unos resultados lo suficientemente fiables.

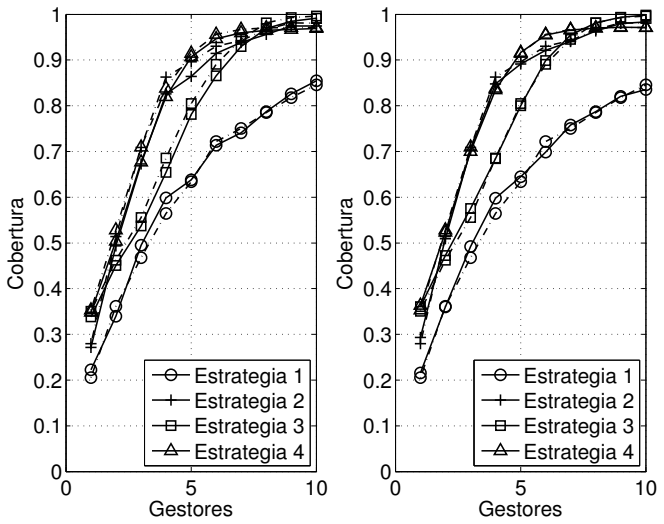


Fig. 5. Cobertura de Gestión, comparación del análisis fundamental con cada uno de los modos (izquierda Proactivo y derecha Reactivo). Los resultados del análisis fundamental se representan con trazo discontinuo (-.-) y los de los modos con línea continua (-).

Para el caso concreto de la estrategia 4 se ha establecido que los sub-grafos de tamaño igual o menor a 2 no se gestionarán, por lo que no se tendrán en cuenta a la hora de resolver la p-mediana para posicionar los gestores de forma óptima.

A. Medidas estáticas

La Fig. 5 muestra la probabilidad de cobertura de gestión, entendida como el número de agentes asociados respecto al total de agentes. En la figura se representan, además de los resultados obtenidos para los dos modos de descubrimiento, aquellos obtenidos con el análisis fundamental, lo que permite corroborar su validez. Se puede observar que, tanto el modo proactivo como el reactivo, ofrecen los mismos resultados que el análisis fundamental, y que, por tanto, no presentan diferencias entre ellos. En cuanto a la diferencia entre las cuatro estrategias, se pone de manifiesto que la número 1 presenta el peor comportamiento, que es lo esperado si se tiene en cuenta su caracter aleatorio. Por otro lado, las estrategias 2 y 4 dan lugar a valores muy parecidos, mientras que la 3 ofrece una cobertura ligeramente inferior hasta que se alcanza un número de gestores lo suficientemente alto; se puede observar que a partir de 8 gestores la estrategia 3 supera en cobertura a la 4, que se mantiene constante, por lo que se deduce que la diferencia de cobertura se corresponderá a los agentes contenidos en sub-grafos de 2 o menos nodos.

Como complemento a las medidas reflejadas por la Fig. 5 se ha estudiado la probabilidad de cobertura resultante al hacer desaparecer nodos gestores, representada en la Fig. 6 como el $\%$ de la cobertura resultante respecto a la inicial, la que habría con todos los gestores en funcionamiento; resaltar que en esta simulación se parte de 80 nodos y 10 gestores y se van desactivando uno a uno los gestores. Esta medida es un indicador de la robustez de las estrategias frente a la pérdida de nodos y de cuántos gestores sería aceptable perder antes de que fuera apropiado realizar una reasignación de papeles. Los valores se han obtenido utilizando el análisis fundamental, ya que la probabilidad de cobertura no difiere entre éste y las medidas de simulación.

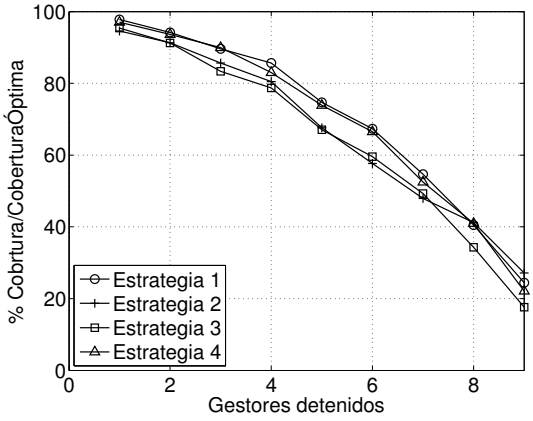


Fig. 6. Cobertura con pérdida de gestores, 80 nodos y 10 gestores iniciales. Medidas del análisis fundamental

Se observa que las que se presentan como más vulnerables son las estrategias 2 y 3, mientras que el despliegue sub-óptimo topológico (estrategia 4) tiene un comportamiento similar al de la estrategia aleatoria. Cabe destacar el comportamiento de la estrategia 2, que aún mostrando un comportamiento muy parecido a la 4 en probabilidad de cobertura (ver Fig. 5) presenta hasta casi un 10% de diferencia ante la pérdida de gestores. Notar además que, a pesar de que a la vista de la Fig. 6, podría parecer que la estrategia aleatoria presente un comportamiento adecuado es, como se vio anteriormente, la que ofrece una cobertura más pequeña, por lo que es razonable que la pérdida asociada a la desaparición de gestores no sea,en términos relativos, elevada.

Otra cualidad que debe presentar la red de gestión es una buena distribución de los nodos agentes entre los gestores; como ya se expuso, esta capacidad se analiza a través del parámetro β. De nuevo, no hay diferencias notables entre las implementaciones del protocolo y el análisis fundamental, como se puede ver en la Fig. 7, con la excepción de la la estrategia 1, debido al carácter aleatorio de la asignación de gestores.

Esta medida evidencia la clara diferencia en comportamiento de las estrategias 3 y 4, en las que, a pesar de tener en cuenta la topología de la red, se observan valores de β muy diferentes. Gracias a no cubrir los sub-grafos de menor tamaño, el despliegue sub-óptimo topológico es capaz de asegurar una carga más equilibrada entre sus gestores. En cuanto al despliegue aleatorio, aunque no presenta valores bajos de β, se comporta incluso mejor que la estrategia 3, lo que evidencia la penalización en la que incurre el algoritmo de la p-mediana por el hecho de tener que dar servicio a todos los nodos. Por último, la estrategia 2 presenta el mejor comportamiento, lo que puede deberse al hecho de que los nodos se despliegan de manera uniforme en el escenario.

B. Medidas dinámicas

Como ya se indicó al inicio de esta sección, el conjunto de medidas que se representan seguidamente se han llevado a cabo con el simulador de redes NS2 y permiten analizar características propias del comportamiento de los protocolos de descubrimiento.

En primer lugar, se ha analizado el tiempo que tardan los agentes en asociarse, haciendo variar los intervalos aleatorios

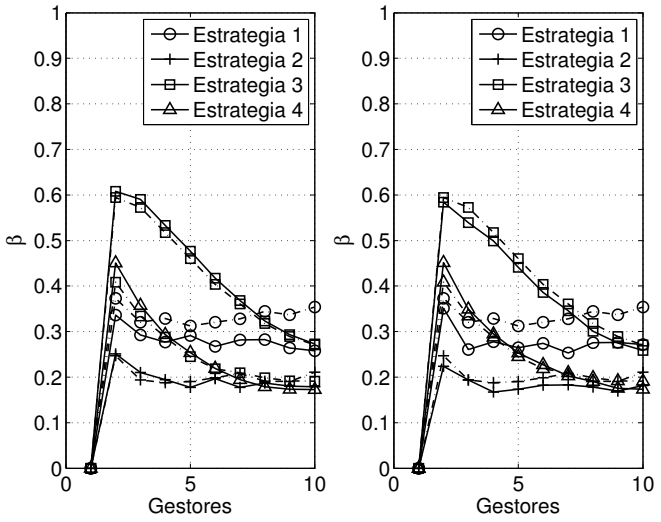


Fig. 7. Parámetro Beta, comparación del análisis fundamental con cada uno de los modos (izquierda Proactivo y derecha Reactivo). Los resultados del análisis fundamental se representan con trazo discontinuo (-.-) y los de los modos con línea continua(-).

que se usan para inicializar gestores y agentes, y que se comentó en la Sección IV (MSI y ASI). Las medidas se han realizado sobre una única topología de red de 80 nodos y 10 gestores; los resultados obtenidos permiten concluir que no existen diferencias notables entre las cuatro estrategias en este aspecto, por lo que se ha empleado una única estrategia de asignación de gestores (estrategia 4) y se ha ido variando el valor del intervalo ASI. A fin de comprobar la repercusión real del valor de dicho parámetro, en las medidas realizadas no se ha activado el mecanismo de *back-off* en el temporizador T_{RMT}, por lo que sólo se tiene en cuenta el tiempo necesario para completar la primera asociación, en caso de darse. El valor elegido para el intervalo MSI ha sido de 1 segundo, mientras que para el intervalo del temporizador T_{WMA} (sólo para el modo reactivo) se ha establecido en 3 segundos.

Tal y como se ha descrito previamente, en el modo proactivo, un agente recibirá los MA de los gestores en el intervalo MSI para, pasado este tiempo, iniciar el proceso de asociación con el "mejor" de los gestores registrados, en un tiempo dentro del intervalo ASI, al expirar el valor inicial del temporizador T_{RMT}. Teniendo esto en cuenta, el tiempo de asociación estará distribuido de manera uniforme dentro del intervalo $\left[MSI, MSI + ASI\right]$, por lo que su función de probabilidad acumulada será una recta con pendiente $\frac{1}{ASI}$ en dicho intervalo.

La Fig. 8 muestra que, en el modo proactivo, para valores elevados de ASI, el tiempo medio de asociación de los agentes sí sigue la distribución uniforme, mientras que, al reducir su valor, el comportamiento se vuelve menos predecible e incluso se puede ver que la probabilidad de no asociarse, que coincide con el valor de la función de probabilidad acumulada en $t = 0$, crece de manera considerable.

En el modo reactivo los agentes inician el envío de paquetes de petición de gestores (*Manager-Request*) dentro del intervalo ASI; posteriormente, tras esperar un tiempo indicado por T_{WMA}, proceden a enviar la solicitud de asociación. Por tanto, para cada valor de ASI el tiempo de asociación volvería a estar uniformemente distribuido en el intervalo $\left[T_{WMA}, T_{WMA} + ASI\right]$.

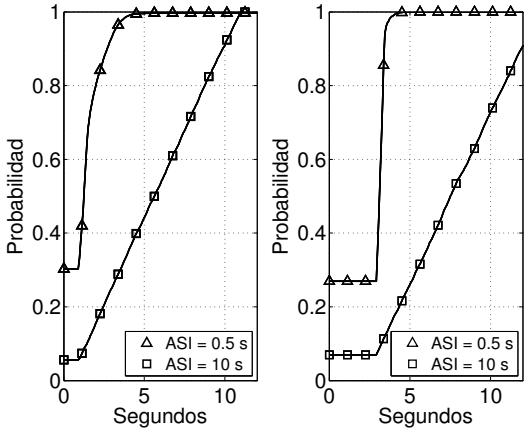


Fig. 8. Probabilidad acumulada de tiempo de asociación (izquierda Proactivo y derecha Reactivo).

En este caso (Fig. 8), se observa un mejor comportamiento que en el modo proactivo, debido a que la aleatorización de los agentes es más influyente, al ser éstos los que realizan el proceso de búsqueda de gestores. Se observa que, incluso con valores muy pequeños de ASI, el tiempo de asociación sigue el comportamiento esperado (mantiene una tendencia lineal), aunque la probabilidad de cobertura se ve nuevamente penalizada.

En la última medida se ha analizado la influencia del número de gestores en el tráfico generado por cada agente cubierto, entendiendo que un menor valor de esta medida representa una mayor eficiencia del tráfico generado. La Fig. 9 muestran los paquetes enviados por minuto y por agente cubierto; poniendo de manifiesto que, a excepción de la estrategia 1, que parece aprovechar de manera menos eficiente el tráfico generado, no hay diferencias notables entre estrategias; por el contrario, sí se puede observar una clara diferencia de comportamiento entre los modos de operación del protocolo (proactivo y reactivo).

Se observa que, mientras el tráfico por agente cubierto en el modo proactivo tiene un comportamiento constante y de valor bajo, con una ligera tendencia creciente, en el modo reactivo el comportamiento es diferente; inicialmente toma valores muy superiores al modo proactivo, pero disminuye de manera brusca con el aumento de gestores (cayendo a la mitad del valor inicial con 5 gestores), llegando a estabilizarse en valores similares al modo de operación anterior y manteniendo la tendencia decreciente. Esta diferencia radica en el hecho de que en el modo proactivo los gestores envíen periódicamente *Manager-Announcement*, que se propagan por toda la red , mientras que en el modo reactivo el tráfico *broadcast* aparece únicamente durante los procesos de búsqueda, al expirar el T_{RMT}, que, en el caso de tener un número bajo de gestores presentes en la topología de red, es frecuente debido a la técnica de *back-off* (IV) que se activa en los agentes en caso de no lograr asociarse.

VI. Conclusiones

En este trabajo se ha analizado el comportamiento de una arquitectura de gestión autónoma sobre un escenario de comunicaciones inalámbrico y multi-salto (red mallada).

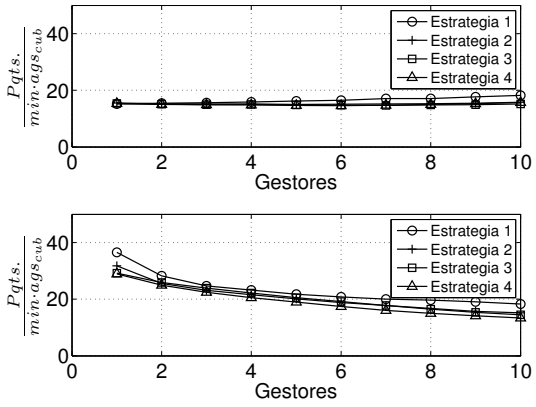


Fig. 9. Tráfico de Descubrimiento, expresado como el número de paquetes generado por minuto entre el número de agentes cubiertos. En la parte superior el modo Proactivo y en la inferior el Reactivo.

Se ha partido de un modelo de organización descentralizado/jerárquico, ya que reduce la penalización que las tareas de gestión pueden causar en la operación de la red subyacente.

Para lograr dicho objetivo se han propuesto unas estrategias de asignación de gestores en base a una serie de métricas de mérito. Las medidas presentadas (obtenidas a través de un estudio más analítico y de una implementación en el marco del simulador *ns-2*), permiten extraer una serie de conclusiones; en primer lugar, es fundamental llevar a cabo una selección apropiada de los gestores, ya que puede haber grandes diferencias según la estrategia particular de selección; además, la heurística propuesta como estrategia de asignación novedosa presenta unos resultados muy interesantes, ya que aporta un mejor comportamiento en términos de la distribución de los agentes entre los gestores seleccionados sin penalizar sustancialmente la probabilidad de cobertura.

Desde una perspectiva de aplicación real, una vez que los gestores se han seleccionado/desplegado, es necesario que los agentes los descubran y que se asocien con ellos. Para ello se han propuesto sendos mecanismos de descubrimiento, fundamentales en el modelo autónomo en el que se está trabajando. El protocolo de descubrimiento (con los modos de funcionamiento proactivo y reactivo) se ha diseñado e implementado en el marco del simulador *ns-2* y, en dicha herramienta, se ha estudiado su comportamiento respecto a los tiempos de estabilización y aprendizaje (tiempos de asociación) y respecto a la carga de tráfico que se origina en la red. A la vista de los resultados obtenidos se puede concluir que ambos modos de funcionamiento tienen un comportamiento acorde con el esperado (el que se obtuvo con el análisis fundamental), ya que no hay prácticamente diferencia alguna en el comportamiento medido en términos de probabilidad de cobertura y de la distribución de agentes entre gestores. En lo que se refiere al comportamiento de los protocolos de descubrimiento, se ha visto que, en las condiciones del escenario analizado, el protocolo reactivo tiene un comportamiento ligeramente mejor, pues pone de manifiesto una estabilidad mayor frente al intervalo de inicialización de los agentes y, lo que es más importante, estabiliza la sobrecarga asociada al protocolo de descubrimiento, que llega incluso a disminuir, a medida que se incrementa el número de gestores en el escenario.

A partir de este trabajo se abren varias líneas de investigación que se están comenzando a explorar. Por una parte sería interesante valorar la idoneidad de las estrategias teniendo en cuenta otros escenarios de aplicación, como la conexión con una red de infraestructura (Internet) a partir de los gestores (que se comportaría como *gateways*) seleccionados. También sería interesante utilizar la plataforma de la que se dispone para analizar los procedimientos de gestión en redes malladas, como pudieran ser la asignación óptima de canales, de potencias de transmisión, etc.

AGRADECIMIENTOS

Los autores querrían expresar su agradecimiento al Gobierno de España por su financiación en el proyecto "Comunicaciones Cognitivas y Cooperativas sobre Entornos Heterogéneos", C3SEM (TEC2009-14598-C02-01)

REFERENCES

[1] J. Irastorza, R. Aguero, y L. Muñoz, "Manager selection over a hierarchical/distributed management architecture for personal networks," in *2nd International ICST Conference on Mobile Networks and Management, 2010. MONAMI 2010.*, 2010.

[2] "IEEE 802.11 wireless LAN medium access control (MAC) and physical layer (PHY) specification - ESS mesh networking (IEEE 802.11s) - draft 8.0," December 2010.

[3] "IEEE 802.16 air interface for fixed and mobile broadband wireless systems. amendment 1: Multiple relay specification (IEEE 802.16j)," June 2009.

[4] R. Schoenen, R. Halfmann, y B. H. Walke, "MAC performance of a 3GPP-LTE multihop cellular network," in *Proceedings of the IEEE International Conference on Communications, ICC*. IEEE, 2008, pp. 4819–4824.

[5] R. Pabst, B. Walke, D. Schultz, P. Herhold, H. Yanikomeroglu, S. Mukherjee, H. Viswanathan, M. Lott, W. Zirwas, M. Dohler, H. Aghvami, D. Falconer, y G. Fettweis, "Relay-based deployment concepts for wireless and mobile broadband radio," *IEEE Communications Magazine*, vol. 42, no. 9, pp. 80–89, 2004.

[6] L. Ferreira, M. De Amorim, L. Iannone, L. Berlemann, y L. Correia, "Opportunistic management of spontaneous and heterogeneous wireless mesh networks," *Wireless Communications, IEEE*, vol. 17, no. 2, pp. 41 –46, april 2010.

[7] U. Ashraf, S. Abdellatif, y G. Juanole, "Gateway selection in backbone wireless mesh networks," in *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE*, 2009.

[8] T. Matsuda, H. Nakayama, S. Shen, Y. Nemoto, y N. Kato, "On gateway selection protocol for dymo-based manet," in *Networking and Communications, 2008. WIMOB '08. IEEE International Conference on Wireless and Mobile Computing,*, 2008, pp. 32 –37.

[9] F. Setiawan, S. Bouk, y I. Sasase, "An optimum multiple metrics gateway selection mechanism in manet and infrastructured networks integration," in *Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE*, 2008, pp. 2229 –2234.

[10] A. Mian, R. Baldoni, y R. Beraldi, "A survey of service discovery protocols in multihop mobile ad hoc networks," *Pervasive Computing, IEEE*, vol. 8, no. 1, pp. 66 –74, 2009.

[11] C. Ververidis y G. Polyzos, "Service discovery for mobile ad hoc networks: a survey of issues and techniques," *Communications Surveys Tutorials, IEEE*, vol. 10, no. 3, pp. 30 –45, 2008.

[12] J. Hoebeke, I. Moerman, B. Dhoedt, y P. Demeester, "Analysis of decentralized resource and service discovery mechanisms in wireless multi-hop networks," *Comput. Commun.*, vol. 29, pp. 2710–2720, August 2006.

[13] M. G. C. Resende y R. F. Werneck, "A hybrid heuristic for the p-median problem," *Journal of Heuristics*, vol. 10, pp. 59–88, January 2004.

Modelado y validación de 6LoWPAN para el simulador de redes OPNET

Jasone Astorga, Eduardo Jacob, Maider Huarte
Departamento de Electrónica y Telecomunicaciones
Universidad del País Vasco / Euskal Herriko Unibertsitatea
Escuela Superior de Ingeniería. Alameda de Urquijo s/n. 48013 - Bilbao
jasone.astorga@ehu.es, eduardo.jacob@ehu.es, maider.huarte@ehu.es

***Resumen*- Gracias al impulso de la *Internet de las Cosas,* en los últimos años se están invirtiendo numerosos esfuerzos en integrar sensores y dispositivos de capacidades muy reducidas en el mundo IP, principalmente mediante el desarrollo de mecanismos que permitan minimizar la sobrecarga introducida en las comunicaciones por dicho protocolo de nivel de red. En esta línea, destaca la propuesta del grupo de trabajo del IETF 6LoWPAN, basada principalmente en la implementación de funcionalidades de fragmentación y de compresión de cabeceras IPv6. En este trabajo se muestra cómo se han modelado y validado dichas funcionalidades en el simulador de redes OPNET, con el objetivo de evaluar la mejora de rendimiento obtenida gracias a las mismas, en la transmisión de datagramas IPv6 en redes de sensores, principalmente en términos de ahorro de energía.**

***Palabras Clave*- sensores, IPv6, 6LoWPAN**

I. INTRODUCCIÓN

La popularidad de las redes de sensores se ha incrementado de forma considerable en los últimos años. De hecho, gracias a los últimos avances en semiconductores y tecnologías electrónicas y a la disponibilidad cada vez más común de servicios de Internet, es cada día más fácil encontrar diferentes dispositivos con un pequeño microprocesador y capacidades de comunicación inalámbricas como Wi-Fi, Bluetooth, ZigBee, UWB, etc integradas. Este tipo de sistemas posibilitan el diseño y despliegue de una nueva generación de aplicaciones, caracterizadas principalmente por una cobertura de adquisición de datos sin precedentes, la invisibilidad del proceso de adquisición, la gran cantidad de datos recogidos y una amplia capacidad de interconexión. Algunos ejemplos comunes de las aplicaciones previstas son los siguientes:

- **Aplicaciones militares**: el despliegue a gran escala de diferentes tipos de redes de sensores posibilitará la monitorización de tropas y equipamiento enemigo, la evaluación del campo de batalla e incluso la detección de ataques químicos o biológicos.
- **Aplicaciones medioambientales**: pueden abarcar desde el seguimiento e identificación de animales, monitorización de factores ambientales en cosechas, detección de fuegos forestales, detección de inundaciones, sistemas de prevención de desastres o monitorización de áreas afectadas por desastres naturales.
- **Aplicaciones médicas**: una nueva era de sensores podría soportar aplicaciones tan críticas como la monitorización de variables fisiológicas en pacientes con algún tipo de dolencia e incluso aplicar remedios en algunos casos (por ejemplo, la inyección instantánea de medicamentos de emergencia a la sangre).
- **Aplicaciones de edificios/entornos inteligentes:** el desarrollo de diferentes tipos de sensores de alto rendimiento, así como cualquier otro tipo de dispositivos electrónicos, con capacidades de procesamiento y comunicaciones inalámbricas integradas, ha dado lugar a una nueva área de investigación relacionada con la automatización de los hogares y los entornos inteligentes. En este campo se están desarrollando incontables aplicaciones orientadas a ayudar a las personas en sus rutinas diarias.

Desde el punto de vista físico, los sensores se caracterizan principalmente por disponer de memorias reducidas, capacidad de cómputo muy limitado y por utilizar baterías para su funcionamiento [1]. Además, muchas veces las baterías integradas en estos dispositivos no son recargables, con lo que cuando las baterías se agotan los dispositivos dejan de ser utilizables. Por este motivo, el consumo de energía es un aspecto crítico para cualquier protocolo o aplicación que haya de funcionar en tales dispositivos.

Desde el punto de vista de las comunicaciones, dado su reducido tamaño, los sensores normalmente hacen uso de tecnologías de comunicación inalámbricas, mediante la integración de una pequeña antena. Además, habitualmente estos dispositivos se organizan en LoWPANs (Low-Power Wireless Personal Area Networks o Redes Inalámbricas de Área Personal de Baja Potencia). Estas redes se constituyen habitualmente como islas aisladas, conectadas a otras redes IP a través de routers frontera.

En cuanto a las comunicaciones dentro de la red LoWPAN, existe un cierto consenso acerca de la utilización del estándar IEEE 802.15.4 [2] en los niveles físico y de enlace. Este protocolo, estandarizado por el IEEE en 2004, proporciona un mecanismo de comunicación de corto alcance, baja tasa binaria y baja potencia. Sin embargo, en cuanto a la pila de protocolos por encima del nivel de enlace, no existe ningún consenso ni ninguna tecnología que prevalezca claramente sobre las demás. Aunque se han promovido algunas pilas de protocolos populares, como pueden ser ZigBee [3], SP100a [4] y WirelessHART [5], el mercado sigue fragmentado, sin que ninguna de ellas haya eclipsado al resto, provocando como resultado que cada fabricante implemente su propio protocolo propietario para soportar transferencias de datos sobre enlaces radio IEEE 802.15.4.

ISBN: 978-84-694-5948-5

No obstante, la utilización de protocolos propietarios presenta varias desventajas, ya que cada uno de ellos implementa el funcionamiento de la capa de red de una forma diferente y normalmente dejan sin definir los mecanismos para llevar a cabo el transporte de paquetes de datos entre la red IEEE 802.15.4 y dispositivos conectados a otras redes. La forma más común de hacer frente a esta última dificultad es mediante el despliegue de un proxy entre el dispositivo sensor y la red externa [6], [7]. Sin embargo, esta solución presenta también diversas desventajas, como el hecho de que las comunicaciones dejan de ser extremo-a-extremo y que la implementación de un proxy es normalmente específica para una tarea o protocolo concreto. Por lo tanto, una solución de este tipo requeriría el desarrollo de proxies específicos para cada aplicación existente.

Por todos los motivos ya mencionados, sería deseable poder utilizar IP, el protocolo de comunicaciones por excelencia en Internet hoy en día, como protocolo de nivel de red en entornos LoWPAN. Debido a esta necesidad surgió el grupo de trabajo 6LoWPAN [8], cuyo objetivo es definir cómo las comunicaciones basadas en IP pueden transportarse de forma eficiente sobre enlaces radio IEEE 802.15.4 [9]. A pesar del gran potencial del enfoque de 6LoWPAN, todavía existen algunas dudas sobre su viabilidad en despliegues reales. Por lo tanto, en este trabajo se describe primero cómo se han modelado las principales funcionalidades propuestas por el grupo de trabajo 6LoWPAN en el entorno de simulación OPNET [10], el cual es un potente simulador de redes basado en eventos discretos. En segundo lugar, se muestra cómo hemos utilizado el modelo desarrollado, consistente en una nueva capa de adaptación situada entre la capa IP y la capa MAC IEEE 802.15.4, para evaluar los efectos derivados de la introducción de comunicaciones basadas en IP en redes de sensores, mediante la medición de parámetros clave como el consumo de energía y la sobrecarga de las comunicaciones. Por último, los resultados obtenidos mediante la simulación de un escenario sencillo nos han servido para validar el modelo desarrollado.

II. Retos del Enfoque IP

IP es una tecnología largamente probada y desplegada a nivel mundial. Por lo tanto, la capacidad de proporcionar servicios IP en redes heterogéneas embebidas implica ventajas claras, como facilidad de integración y amplia interoperabilidad, posibilidad de conexión directa entre la red LoWPAN y otras redes externas, reutilización de infraestructuras y herramientas ya existentes para el diagnóstico, gestión, etc. Sin embargo, las redes de sensores presentan limitaciones muy severas con respecto a las capacidades físicas y de comunicación de las entidades que constituyen la red. Como consecuencia, hasta hace muy poco la posibilidad de introducir tecnología IP en dispositivos sensores se pensaba inviable, ya que se consideraba que el protocolo IP era demasiado pesado como para poder implementarse en dispositivos con recursos tan escasos.

En este sentido, uno de los mayores problemas a la hora de implementar comunicaciones basadas en IP en redes LoWPAN se debe al gran tamaño de los paquetes IP en comparación con la longitud de las tramas de nivel de enlace. Concretamente, IPv6 determina una MTU mínima de 1280 bytes, mientras que el estándar IEEE 802.15.4 especifica una longitud máxima de las tramas de la capa MAC de 127 bytes. Siendo la longitud mínima de la cabecera MAC de 5 bytes, el número máximo de bytes disponibles para los datos de niveles superiores es 102. Este hecho deriva en la necesidad de implementar tanto funcionalidades de compresión de cabeceras como de fragmentación, para adaptar el tamaño de los datagramas IPv6 al de las tramas del nivel inferior.

Otro problema que surge a la hora de transportar paquetes IP sobre redes LoWPAN está relacionado con las severas limitaciones de energía y rendimiento de dichas redes, lo que las hace más propensas a fallos en los enlaces y a interferencias. Se ha demostrado que protocolos como TCP alcanzan un rendimiento muy bajo en entornos inalámbricos principalmente por estos motivos [11]. Por lo tanto, un protocolo de nivel de red apropiado para redes de sensores debería ser sensible a estas limitaciones, siendo capaz de adaptarse a las condiciones de la red mientras mantiene un nivel de consumo de energía eficiente.

Con respecto a la configuración de direcciones IP y otros parámetros del nivel de red, en redes IP ordinarias estos parámetros se asignan a cada interfaz conectada a la red bien de forma manual o utilizando algún mecanismo dinámico como DHCP. Sin embargo, ninguno de estos esquemas de configuración es adecuado para el caso de las redes de sensores: por una parte, en redes de sensores de gran escala la configuración manual no es viable; y por otra, los métodos dinámicos son normalmente demasiado costosos desde el punto de vista de las comunicaciones. Teniendo en cuenta estas dificultades, y el hecho de que habitualmente las redes de sensores están compuestas por un gran número de dispositivos, IPv6 proporciona algunas características muy deseables, como capacidades de autoconfiguración y un espacio de direccionamiento extremadamente grande. No obstante, el disponer de un espacio de direccionamiento grande implica que las direcciones de red a utilizar han de ser relativamente largas (128 bits), lo cual supone un inconveniente desde el punto de vista de la sobrecarga de las transmisiones. Por lo tanto, es necesario implementar algún mecanismo para reducir la información acerca de las direcciones IPv6 a transmitir sobre el enlace radio.

Por ultimo, el encaminamiento en redes IP se lleva a cabo normalmente a nivel de red, independientemente de las capas inferiores. Sin embargo, las redes IEEE 802.15.4 se organizan habitualmente en topología de estrella o mallada e implementan mecanismos de reenvío a nivel 2. Por este motivo, es necesario implementar alguna capa de adaptación que soporte el encaminamiento tanto a nivel de red como a nivel de enlace. Además, cualquier mecanismo de encaminamiento diseñado para su implementación en entornos LoWPAN debería minimizar la sobrecarga introducida en la red por los paquetes del protocolo. En este sentido, el hecho de que las redes de sensores no se desplieguen habitualmente como redes de tránsito puede ser útil para simplificar los protocolos de rutado a implementar en estos entornos.

III. La Solución Propuesta por 6LoWPAN

El objetivo del grupo de trabajo 6LoWPAN es definir cómo los paquetes IPv6 pueden transportarse de forma efectiva y eficiente sobre enlaces inalámbricos IEEE 802.15.4. El enfoque seguido para esta definición se basa en

la filosofía *paga únicamente por lo que uses*. Es decir, 6LoWPAN define una cabecera básica extremadamente compacta y ofrece la posibilidad de expandir dicha cabecera cuando se requieran funcionalidades adicionales.

Teniendo en cuenta las dificultades y retos presentados en el apartado anterior, el grupo de trabajo 6LoWPAN propone la inclusión de una cabecera autocontenida independiente para hacer frente a cada uno de dichos retos, resultando en la definición de las siguientes tres cabeceras:

- Una **cabecera de direccionamiento mallado**, para soportar el reenvío de unidades de datos de nivel 2.
- Una **cabecera de fragmentación**, para hacer frente a la dificultad de acomodar datagramas IPv6 dentro de tramas IEEE 802.15.4.
- Una **cabecera de compresión**, para minimizar la sobrecarga introducida por las cabeceras IPv6 en las comunicaciones.

Cada una de estas cabeceras se introduce únicamente en caso de que sea necesaria, siguiendo el esquema de la Fig. 1.

Además, como los sensores se despliegan habitualmente a gran escala y de forma desatendida, 6LoWPAN soporta también la auto-configuración sin estado (*stateless*) de direcciones IPv6 en los nodos sensores.

IV. Diseño e Implementación de Funcionalidades 6LoWPAN en OPNET

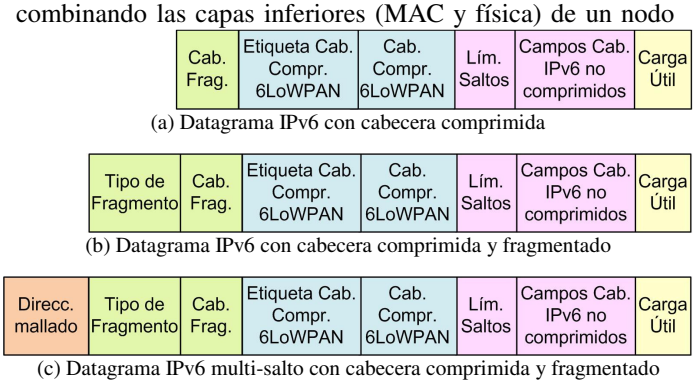
6LoWPAN propone un esquema de compresión sin estado, de diferentes campos de la cabecera IPv6. El escenario óptimo es el siguiente:

- La versión IP es IPv6.
- Las direcciones origen y destino son de tipo *link-local*.
- Los identificadores de interfaz IPv6 de las direcciones origen y destino pueden inferirse de las direcciones MAC correspondientes.
- La *Longitud del Paquete* puede obtenerse a partir de campos de otras capas, como la *Longitud de la Trama* o la *Longitud del Datagrama* (en caso de fragmentación).
- Los campos de *Clase de Tráfico* y *Etiqueta de Flujo* son cero.
- El campo de *Siguiente Cabecera* es UDP, ICMP o TCP.

Después, dependiendo de cuánto se acerque cada paquete concreto a este caso óptimo, puede que algunos campos no sean comprimibles, teniendo que ser transportados "en línea".

A. Modelo de Nodo 6LoWPAN

Para implementar las funcionalidades 6LoWPAN descritas anteriormente, hemos creado un nuevo modelo de nodo para el simulador OPNET. Este modelo se ha creado combinando las capas inferiores (MAC y física) de un nodo IEEE 802.15.4 con las capas superiores (IP) de un nodo Ethernet estándar. Además en este nuevo modelo de nodo se ha introducido una nueva capa de procesamiento, desarrollada íntegramente desde cero, entre la capa de red IPv6 y la capa de nivel de enlace IEEE 802.15.4, tal y como se muestra en la Fig. 2.



(a) Datagrama IPv6 con cabecera comprimida

(b) Datagrama IPv6 con cabecera comprimida y fragmentado

(c) Datagrama IPv6 multi-salto con cabecera comprimida y fragmentado

Fig.1. Estructura de cabeceras 6LoWPAN

Además, para permitir una configuración flexible del comportamiento de este nodo a nivel de funcionalidades 6LoWPAN, se han creado una serie de nuevos atributos de nodo, tal y como se puede observar en la Fig. 3.

- ***Compression*:** determina si la compresión de cabeceras a nivel 6LoWPAN está habilitada o no.
- ***File Enable*:** permite habilitar o deshabilitar el almacenamiento de logs del modelo de proceso 6LoWPAN.
- ***File Directory*:** cuando se almacenan los logs en un fichero, este atributo define el directorio donde se encuentra dicho fichero de log.
- ***6lowpan.GlobalAddress*:** se trata de un atributo compuesto que define las características de las direcciones globales IPv6 a utilizar:
  - ***Active*:** indica si van a usarse direcciones IPv6

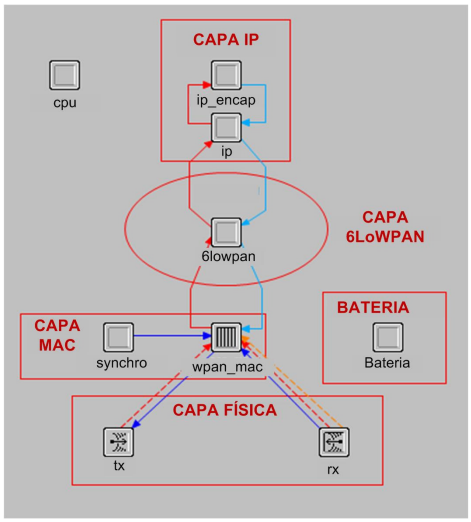


Fig. 2. Modelo de nodo con soporte 6LoWPAN en OPNET

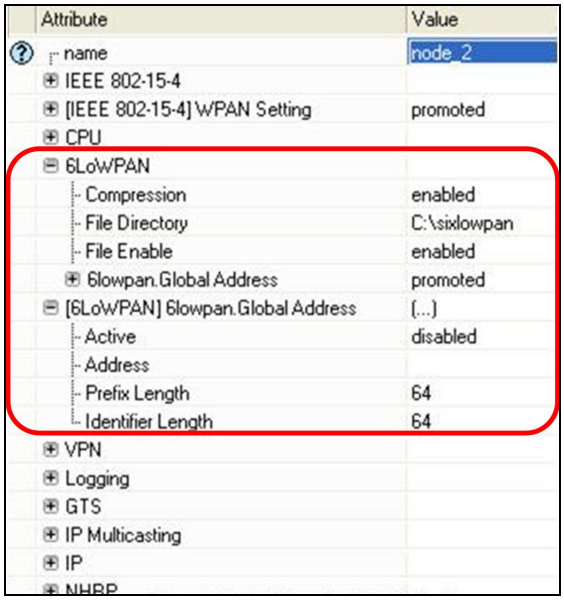


Fig. 3. Atributos 6LoWPAN en OPNET

globales. Si este atributo está deshabilitado, las direcciones IPv6 a utilizar serán de tipo link-local.

- ***Address*:** parámetro que almacena la dirección IPv6 global asignada al nodo.
- ***Prefix Length*:** indica la longitud del prefijo IPv6, lo cual es necesario para las tareas de compresión, tal y como se verá más adelante.
- ***Identifier Length*:** indica la longitud del identificador de interfaz dentro de la dirección IPv6 global. Este parámetro se utiliza junto con la longitud del prefijo para comprimir de forma eficiente las direcciones IPv6 globales.

Por otra parte, para permitir la comunicación del nuevo modelo de proceso *6lowpan* con los modelos de proceso de las capas adyacentes *ip* y *wpan_mac*, ha sido necesario definir nuevos flujos de datos bidireccionales que posibiliten el intercambio de paquetes de datos entre dichas capas.

Además, también ha sido necesario llevar a cabo ciertas modificaciones en los modelos y librerías asociados con dichas capas adyacentes. Por una parte, se han extendido algunas librerías del modelo IP estándar de OPNET, concretamente aquellas que definen las funciones encargadas de la creación de direcciones de red IPv4 e IPv6, para añadir nuevas funciones necesarias para crear direcciones IPv6 a partir de identificadores de enlace IEEE 802.15.4, tal y como determina el mecanismo de autoconfiguración sin estado de 6LoWPAN. Por otra parte, en el modelo de proceso original de la capa IEEE 802.15.4 la dirección de enlace del nodo destino de una comunicación se configuraba a nivel de atributo de nodo. Sin embargo, como el objetivo de la introducción de la capa de adaptación 6LoWPAN es implementar servicios IP sobre enlaces radio IEEE 802.15.4, en este caso las direcciones de nivel de enlace deberían derivarse de las direcciones IPv6 correspondientes a cada nodo. Por este motivo, hemos generado un mecanismo que permite obtener direcciones MAC a partir de direcciones IPv6, tal y como se explicará en detalle más adelante.

B. Modelo de Proceso 6LoWPAN

El modelo de proceso definido para llevar a cabo el procesamiento relativo a las funcionalidades 6LoWPAN tanto para los paquetes enviados como recibidos, es muy simple, consistiendo únicamente de dos estados, tal y como se puede observar en la Fig. 4.

Estado "*Init*": se trata del estado inicial ejecutado cuando se arranca el proceso. Las principales tareas llevadas a cabo por este estado son las siguientes:

- Registro del proceso en un registro de modelos global.
- Procesamiento de los atributos de configuración del nodo.
- Inicialización de variables de estado.

Estado "*Wait*": este estado consiste en un proceso en bucle que está continuamente esperando la llegada de un nuevo paquete, indiferentemente de si llega de la capa de red o de enlace. Por lo tanto, es en este estado donde se lleva a cabo realmente el procesamiento de todos los paquetes IPv6 enviados o recibidos por el nodo, de acuerdo con la especificación 6LoWPAN.

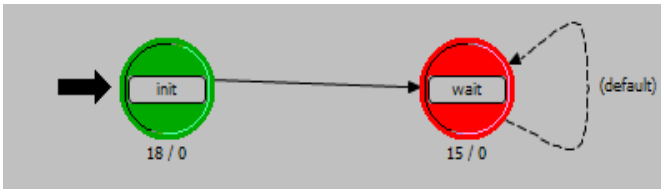


Fig. 4. Modelo de proceso 6LoWPAN

Para cada paquete entrante, el primer paso consiste en determinar si dicho paquete ha sido recibido de la capa de red o de enlace. A partir de esta información, se invoca la función apropiada para la gestión del paquete, pasándole el paquete recibido como parámetro.

En el caso de que el paquete recibido provenga de la capa IP, la capa de adaptación 6LoWPAN sigue el esquema mostrado en la Fig. 5 para adaptar dicho paquete al enlace IEEE 802.15.4 sobre el que ha de ser transmitido, basándose en las condiciones de la red y del tráfico, así como en los atributos específicos configurados en el nodo.

Tal y como se puede observar en la Fig. 5, primero se procesa la cabecera IPv6 del paquete recibido para extraer las direcciones IPv6 origen y destino. Después, se utilizan estas direcciones para calcular las direcciones MAC correspondientes y el identificador PAN (Personal Area Network) asociado a la red. En este punto cabe destacar que en las pruebas de simulación realizadas configuramos las direcciones IPv6 de forma que siempre fuera posible obtener las direcciones MAC correspondientes de forma automática, sin necesidad de utilizar ningún protocolo adicional como ARP. No obstante, consideramos que ésta es una práctica común y realista para la configuración de redes 6LoWPAN, ya que teniendo en cuenta las limitaciones de energía y computación que gobiernan el despliegue de cualquier red de sensores, consideramos que cualquier intento de desplegar una red de este tipo de forma eficiente, debería seguir invariablemente este enfoque.

Mencionar también que los identificadores de interfaz utilizados en redes IEEE 802.15.4 se ajustan a la estructura EUI-64 [12]. Sin embargo, aunque todos los dispositivos pertenecientes a una red IEEE 802.15.4 han de tener una dirección IEEE EUI-64 única, en algunos casos también pueden utilizarse direcciones cortas de 16 bits, tal y como se

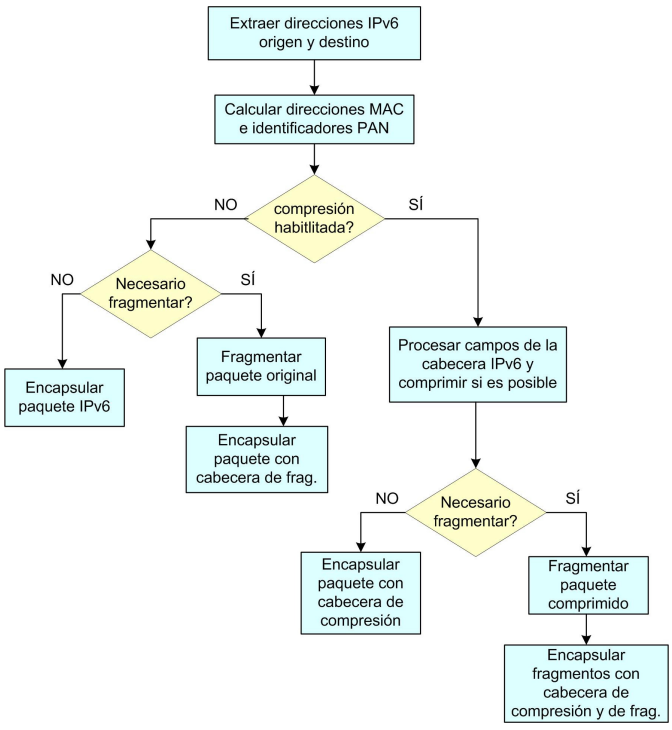


Fig. 5. Diagrama de bloques de la función encargada de gestionar los paquetes recibidos de la capa IP

Attribute Name	Type	Default Value
src_addr	integer	0
src_pan_id	integer	0
dest_addr	integer	0
dest_pan_id	integer	0

Fig. 6. Estructura del fichero ICI utilizado para transportar datos desde la capa 6LoWPAN a la capa de nivel de enlace

define en el estándar IEEE 802.15.4 [2]. Las direcciones MAC cortas de 16 bits se obtienen de forma sencilla dividiendo la dirección IPv6 en los 8 grupos de 16 bits que la componen y cogiendo únicamente el último grupo de 16 bits. Por lo tanto, los últimos 16 bits de la dirección IPv6 de un cierto nodo se corresponden directamente con la dirección MAC corta de dicho nodo. De forma similar, el identificador PAN se corresponde con el quinto grupo de 16 bits de la dirección IPv6. Por lo tanto, utilizando la notación común usada para representar direcciones IPv6, la dirección IPv6 de un cierto nodo se podría escribir de la siguiente manera: X:X:X:X:PAN_ID:X:X:MAC.

Por otra parte, como tanto las direcciones MAC origen y destino, como el identificador PAN, se calculan a partir de las correspondientes direcciones IPv6 dentro del modelo de proceso 6LoWPAN, es necesario implementar algún mecanismo para transportar está información desde la capa 6LoWPAN a la capa de nivel de enlace IEEE 802.15.4. De hecho, el protocolo de nivel de enlace necesita estos datos para rellenar los campos de direccionamiento de las tramas IEEE 802.15.4. En el simulador OPNET este tipo de comunicación entre capas que no se corresponde con el intercambio de paquetes de datos, se lleva a cabo mediante un tipo especial de archivos, denominados “archivos ICI”. La Fig. 6 muestra el contenido concreto del archivo ICI utilizado para el intercambio de datos entre los procesos 6LoWPAN e IEEE 802.15.4.

Después de calcular las direcciones MAC a partir de las direcciones IPv6, se comprueba si a nivel de nodo se ha habilitado la compresión de cabeceras o no. Si la compresión de cabeceras no está habilitada, el módulo 6LoWPAN deja de procesar la cabecera IPv6, ya que ha de ser transmitida entera. No obstante, el proceso 6LoWPAN ha de continuar evaluando el tamaño del paquete para decidir si es necesario fragmentarlo o no.

Si por el contrario, el atributo de compresión de cabecera a nivel de nodo está habilitado, el módulo 6LoWPAN continúa procesando la cabecera IPv6 para determinar qué campos pueden ser comprimidos. De hecho, simplemente por pertenecer a la misma red 6LoWPAN, diferentes dispositivos comparten un estado común que permite eliminar algunos de los campos de la cabecera IPv6 sin que esto afecte a la comunicación. La Fig. 7 muestra el formato de una cabecera IPv6 básica.

Fig. 7. Formato de cabecera IPv6 básica

Por lo tanto, 6LoWPAN lleva a cabo una compresión sin estado de las cabeceras IPv6 teniendo en cuenta las siguientes directrices:

- ***Versión*:** la versión del protocolo IP utilizada es siempre 6, por lo que no es necesario transmitir esta información.
- ***Clase de Tráfico* y *Etiqueta de Flujo*:** normalmente estos campos suelen ser 0. Si es así, no se transmiten. En cualquier otro caso se transmiten en línea los 8 bits completos de la *Clase de Tráfico* y los 20 bits de la *Etiqueta de Flujo*.
- ***Longitud de la Carga:*** se omite, ya que puede calcularse a partir de otros valores como el campo de *Longitud de Trama* de la cabecera MAC o el de *Tamaño de Datagrama* de los fragmentos de paquete, cuando se produce fragmentación.
- ***Siguiente cabecera:*** para optimizar la transmisión de este campo en la mayor parte de los casos, se han definido códigos de 2 bits para los protocolos más comunes, a saber, UDP, TCP e ICMP. Si los paquetes IP transportan información de algún otro protocolo de nivel superior, han de transmitirse los 8 bits completos de este campo.
- ***Límite de Saltos:*** éste es el único campo que no puede comprimirse nunca, ya que su valor cambia con cada salto que atraviesa el paquete, y por lo tanto, los 8 bits de este campo han de transmitirse en línea.
- ***Direcciones Origen y Destino:*** el nivel de compresión que puede aplicarse a las direcciones IPv6 difiere dependiendo de si estas direcciones son de tipo *link-local* o global. En el caso de direcciones de enlace local, se componen de un prefijo conocido y un identificador que puede inferirse de los datos de nivel de enlace, por lo que pueden obviarse totalmente. En el caso de direcciones globales, dependiendo de la dirección concreta, será necesario transmitir bien el prefijo, el identificador o ambos campos.

Gracias a estos mecanismos es posible reducir una cabecera de 40 bytes a tan solo 2 bytes, uno para la cabecera de compresión 6LoWPAN y otro para el campo *Límite de Saltos*. El formato de la cabecera de compresión 6LoWPAN es el que se muestra en la Fig. 8, donde el valor de cada uno de sus campos especifica si el campo de la cabecera IPv6 correspondiente se ha comprimido o se ha transmitido en línea a continuación.

Una vez se ha completado el proceso de compresión, el módulo 6LoWPAN pasa a evaluar si la fragmentación es necesaria, en cuyo caso se invoca la función correspondiente.

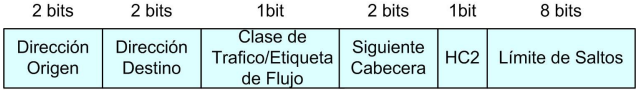
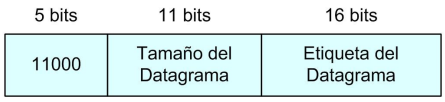
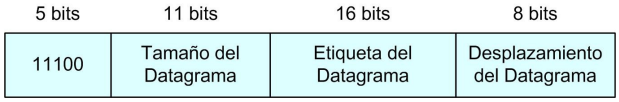


Fig. 8. Formato de la cabecera de compresión 6LoWPAN



(a) Primer fragmento de un datagrama



(b) Resto de fragmentos del datagrama

Fig. 9. Formato de la cabecera de fragmentación 6LoWPAN

La función de fragmentación se encarga de dividir el datagrama original en fragmentos que puedan ser transportados en tramas del nivel de enlace IEEE 802.15.4 y también de calcular y añadir la cabecera de fragmentación 6LoWPAN correspondiente a cada fragmento resultante. En este caso, ha de tenerse en cuenta que tal y como se muestra en la Fig. 9, la cabecera de fragmentación 6LoWPAN es diferente para el primer fragmento de un paquete y para el resto, ya que en el caso del primer fragmento de un datagrama el desplazamiento siempre es cero, y por lo tanto, este valor puede obviarse.

Por otra parte, si el paquete recibido por la capa 6LoWPAN procede de la capa MAC, el procesamiento a llevar a cabo es más sencillo, gracias al hecho de que el modelo de proceso de la capa IEEE 802.15.4 ofrece la posibilidad de pasar las tramas a la capa superior con la cabecera IEEE 802.15.4 incluida. De esta forma, las direcciones MAC origen y destino, así como los identificadores PAN correspondientes, pueden obtenerse de forma sencilla. Por su parte, el contenido del campo de carga útil de la trama se corresponde directamente con un paquete 6LoWPAN, el cual habrá de ser procesado adecuadamente para reconstruir el datagrama IPv6 correspondiente en el formato esperado por la capa IP. La Fig. 10 muestra un esquema de las tareas llevadas a cabo para crear un paquete IPv6 completo a partir de un paquete 6LoWPAN comprimido.

Tal y como se puede observar en la Fig. 10, el

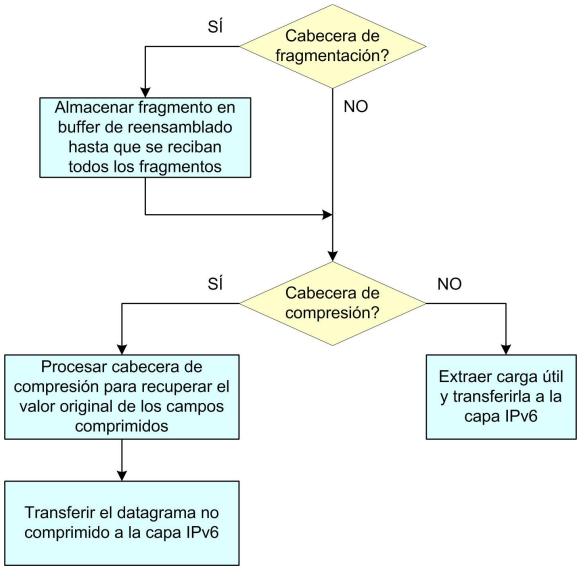


Fig. 10. Diagrama de bloques de la función encargada de gestionar los paquetes recibidos de la capa MAC

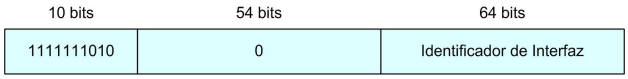


Fig. 11. Formato de dirección IPv6 generada a partir de dirección MAC

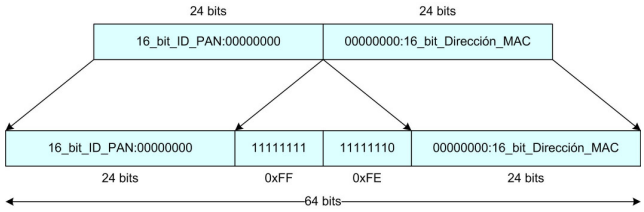


Fig. 12. Mecanismo de generación de identificador de interfaz

procesamiento del paquete 6LoWPAN comienza con la búsqueda de la cabecera de fragmentación. Si esta cabecera no existe, el proceso continúa buscando la cabecera de compresión. Si esta cabecera tampoco existe, la carga útil del paquete 6LoWPAN se corresponde directamente con el datagrama IPv6 completo, el cual puede extraerse y transmitirse a la capa de red sin necesidad de mayor procesamiento.

Si por el contrario, el paquete recibido es un fragmento de un datagrama IPv6 de mayor tamaño, el proceso se encarga de leer el valor de los campos de la cabecera de fragmentación, los cuales utiliza para almacenar el fragmento en la posición correcta del buffer de reensamblado hasta que se reciban el resto de los fragmentos del datagrama original. Una vez se ha reconstruido el paquete completo, éste es objeto del mismo procesamiento que cualquier paquete 6LoWPAN no fragmentado.

En el caso de que el paquete recibido haya sido objeto de compresión de cabeceras 6LoWPAN, se procesan los diferentes campos de la cabecera de compresión 6LoWPAN utilizando la misma lógica explicada para la operación de compresión, pero en sentido inverso. Un caso interesante es el correspondiente a la generación de direcciones IPv6 de enlace local totalmente comprimidas a partir de las direcciones MAC origen y destino correspondientes. El formato de las direcciones IPv6 generadas es el que se muestra en la Fig. 11.

Con respecto al cálculo del identificador de interfaz, los 32 bits más significativos se obtienen concatenando 16 ceros a los 16 bits del identificador PAN. Estos 32 bits se concatenan después con los 16 bits de la dirección MAC corta. Finalmente, el identificador de interfaz se genera a partir de estos 48 bits siguiendo la especificación de IPv6 sobre Ethernet [13]. De acuerdo con este estándar, el identificador de interfaz se obtiene incluyendo el valor hexadecimal FFFE en la parte central de la pseudo-dirección de 48 bits, tal y como se muestra en la Fig. 12.

Este procedimiento de cálculo del identificador de interfaz es útil también para la transmisión de direcciones de tipo global comprimidas. En este caso, para recuperar la dirección IPv6 completa, se utiliza el valor del identificador de red PAN configurado en el propio nodo donde se está realizando el cálculo.

V. ANÁLISIS DE RENDIMIENTO

A. *Escenario de Simulación*

Para evaluar la mejora del rendimiento de las comunicaciones IPv6 proporcionada por el proceso de modelo diseñado, hemos creado un escenario de simulación muy simple representando una pequeña red PAN compuesta únicamente por dos nodos. La topología de red utilizada es la representada en la Fig. 13.

A pesar de que este escenario no se ajusta a la topología habitual de las redes de sensores, en las que se despliegan grandes cantidades de dispositivos minúsculos, se adapta perfectamente a nuestro propósito de evaluar el efecto de la capa 6LoWPAN desarrollada y poder así validar su correcto funcionamiento. De hecho, un escenario de simulación tan simple permite identificar de forma sencilla y precisa los efectos causados por el procesamiento 6LoWPAN, sin interferencia de otros factores externos, como puede ser la pérdida de paquetes debida a colisiones o interferencias, bucles de rutado, etc.

Por otra parte, las pruebas se han realizando configurando las dos entidades comunicantes de la Fig. 13 tanto con direcciones IPv6 de enlace local, como con direcciones globales, permitiendo de esta forma evaluar diferentes niveles de compresión 6LoWPAN.

Por último, la demanda de tráfico configurada consiste en la transmisión de paquetes IPv6 de tamaño fijo de 40 bytes a un ritmo constante de un intercambio petición/respuesta por segundo, comenzando en el segundo 20 de la simulación y siendo la tasa de transmisión de datos de la red de 250 Kbps.

B. *Estadísticos Evaluados*

Con el objetivo de evaluar cómo afecta el procesamiento 6LoWPAN a la transmisión de tráfico IPv6 en redes de sensores, hemos creado dos nuevos estadísticos que nos permitan cuantificar dicho efecto:

- **Bits de datos transmitidos (%):** este estadístico está orientado a evaluar la eficiencia de la funcionalidad de compresión de cabeceras de 6LoWPAN, determinando qué porcentaje de la cantidad total de bits transmitidos por el aire se corresponde realmente con bits de datos del nivel superior y qué porcentaje corresponde a la sobrecarga introducida por los protocolos IPv6, 6LoWPAN e IEEE 802.15.4.
- **Energia consumida:** este estadístico pretende evaluar cómo se va reduciendo el recurso más limitado de las redes de sensores, la energía. Para ello, se ha modelado el consumo de energía de un sensor asignando diferentes valores a los estados de "dormido", "inactivo", "transmitiendo" y "recibiendo". Los valores asignados en nuestro caso son los correspondientes a una plataforma MicaZ, una de las plataformas de sensores más comunes utilizada hoy en día. Dichos valores, los cuales han sido extraídos de las hojas de especificaciones del fabricante, son los indicados en la Tabla 1.

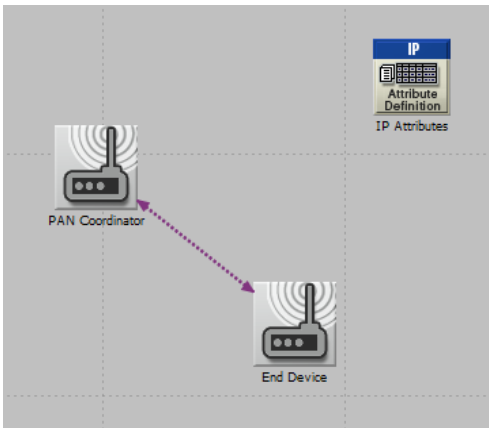


Fig. 13. Escenario de simulación

Estado	Consumo
Dormido	16 µA
Inactivo	35 µA
Recepción	27,7 mA
Transmisión (0 dBm)	17,4 mA

Tabla 1. Consumo de MicaZ en diferentes estados

C. *Resultados de Simulación y Validación del Modelo*

Primero, hemos evaluado la relación entre los bits totales transmitidos por cada trama IEEE 802.15.4 y los bits de información de niveles superiores contenidos en cada una de estas tramas, con y sin la funcionalidad de compresión de cabeceras 6LoWPAN habilitada. Los resultados obtenidos son los que se muestran en la Fig. 14, donde la línea azul representa el porcentaje de bits de datos transmitidos en cada trama con respecto a los bits totales en el caso de que no se esté utilizando la compresión de cabeceras y la línea roja representa el mismo porcentaje en el caso óptimo de la compresión de cabeceras 6LoWPAN. Mencionar que dado el reducido tamaño de los paquetes IPv6 transmitidos (40 bytes) no ha sido necesario llevar a cabo fragmentación.

Como puede observarse, en el caso de paquetes de red pequeños, la funcionalidad de compresión de cabeceras 6LoWPAN permite aumentar la capacidad de transmisión de datos de la red en más de un 50%. Teniendo en cuenta las severas limitaciones de recursos de los dispositivos que componen las redes de sensores, la mayoría de las aplicaciones a desplegar sobre este tipo de redes serán aplicaciones ligeras de tipo petición/respuesta, las cuales pueden beneficiarse de forma significativa de las mejoras obtenidas mediante la compresión de cabeceras.

Otro beneficio derivado de la funcionalidad de compresión de cabeceras es que reduce la necesidad de

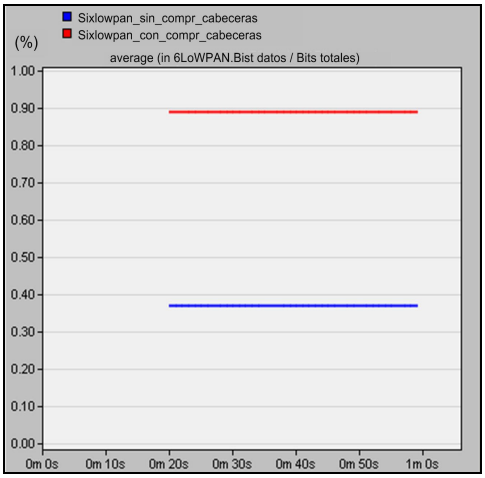


Fig. 14. Comparación de la capacidad de transmisión de datos con y sin compresión de cabeceras

fragmentación, ya que los paquetes comprimidos pueden encajar en tramas de nivel de enlace demasiado pequeñas para transportar el datagrama IPv6 original completo sin comprimir.

Otra ventaja importante derivada de la reducción de la cantidad de bits totales transmitidos por el medio radio para transportar la misma cantidad de información es la reducción de la energía consumida por los nodos involucrados en la comunicación, ya que la mayor causa de consumo en el caso de los dispositivos sensores es la transmisión de bits a través del medio radioeléctrico [14]. Cabe destacar que el ahorro de energía es una propiedad particularmente deseable en el caso de los entornos considerados en este trabajo, donde la energía es el recurso más limitado. La Fig. 15 muestra la relación entre el consumo de energía de uno de los nodos del escenario de simulación de la Fig. 13 con y sin la funcionalidad de compresión de cabeceras 6LoWPAN habilitada. Además, las funcionalidades de compresión de cabeceras se han evaluado en dos casos: utilizando direcciones link-local y utilizando direcciones globales de las cuales el identificador de interfaz puede derivarse de la dirección MAC correspondiente. Por su parte, la Fig. 16 muestra la energía restante en dicho nodo en los mismos

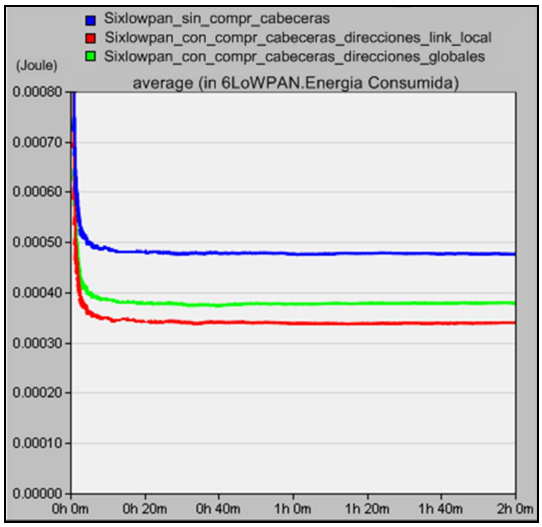


Fig. 15. Comparación de la energía consumida con y sin compresión de cabeceras

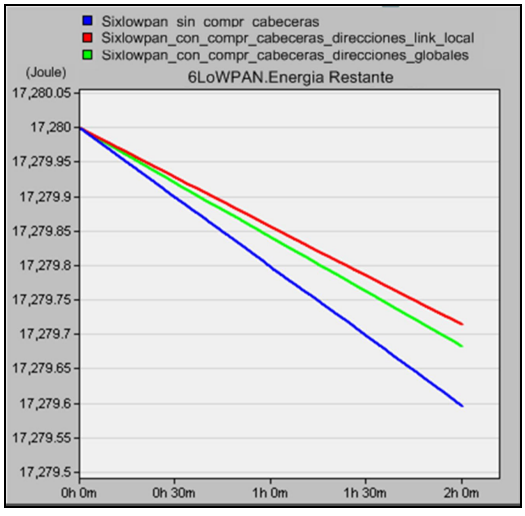


Fig. 16. Comparación de la energía restante después de dos horas de simulación con y sin compresión de cabeceras

casos. La energía inicial configurada en cada nodo es la correspondiente a 2 pilas AA (1,5v, 1600mAh) en Joules.

Tal y como puede observarse, en el caso de utilizar la compresión de cabeceras 6LoWPAN, el consumo medio de energía durante la transmisión/recepción de paquetes IPv6 se reduce en 0,1 mJ para el caso de utilizar direcciones globales y más aún si las direcciones utilizadas son de tipo enlace local. Esto deriva en que en una simulación de dos horas la cantidad de energía ahorrada gracias a la compresión de cabeceras sea de 90 mJ en el peor de los casos. Teniendo en cuenta que la energía total consumida durante estas dos horas de simulación sin compresión de cabeceras es de 600 mJ, el ahorro conseguido gracias a la compresión de las cabeceras en el peor de los casos es de un 22,5%, lo que se traduce en la prolongación de la vida útil de dichos dispositivos en el mismo porcentaje.

VI. Conclusiones

En este trabajo se ha presentado una alternativa de diseño e implementación en el simulador de redes OPNET de las funcionalidades de fragmentación y compresión de cabeceras propuestas por el grupo de trabajo 6LoWPAN. Teniendo en cuenta los resultados obtenidos mediante simulación, los cuales se corresponden con los resultados esperados, consideramos que el modelo desarrollado es válido y refleja fielmente la realidad. Por lo tanto, dicho modelo podría utilizarse en el futuro para llevar a cabo estudios más exhaustivos de la mejora de rendimiento obtenida mediante la implementación de 6LoWPAN en entornos de sensores complejos compuestos por un gran número de entidades.

Referencias

[1] J. Al-Muhtadi, D. Mickunas, and R. Campbell, “Wearable security services”, in *Proceedings of the 21st International Conference on Distributed Computing Systems*, pp. 266-271, Apr. 2001.

[2] IEEE 802.15.4 Standard, “Wireless Medium Access (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal AreaNetworks (WPANs)”, Dec. 2006

[3] Zigbee Alliance, Jun. 2009, http://www.zigbee.org/

[4] Int’l Soc. Automation (ISA), “ISA100.11a Status,” 2007, http://www.isa.org/isa100

[5] HART Communication Foundation (HCF), Jun. 2009, http://www.hartcomm.org/

[6] D. J. Abadi, W. Lindner, S. Madden and J. Schuler, “An integration framework for sensor networks and data stream management systems”, in *Proceedings of the Thirtieth international Conference on Very Large Data Bases*, vol 30, pp. 1361-1364, Toronto, 2004

[7] A. Kansal, M. Goraczko and F. Zhao, “Building a sensor network of mobile phones”, in *Proceedings of the 6th international conference on Information processing in sensor networks*, pp. 547-548, Cambridge, Massachusetts, USA, 2007

[8] IPv6 over Low power WPAN (6lowpan) Working Group, http://datatracker.ietf.org/wg/6lowpan/charter/

[9] G. Montenegro, N. Kushalnagar, J. Hui and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, Sep. 2007

[10] OPNET, http://www.opnet.com/

[11] H. Balakrishnan, S. Seshan, E. Amir, and R. Katz, “Improving TCP/IP performance over wireless network”, in Proceedings of the First International Conference on Mobile Computing and Networking (MobiCom’95), California, USA, 1995

[12] IEEE, “Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority”, http://standards.ieee.org/regauth/oui/tutorials/EUI64.html

[13] M. Crawford, "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, Dec. 1998.

[14] V. Shnayder, M. Hempstead, B-r. Chen, G. W. Allen, and M. Welsh, “Simulating the power consumption of large-scale sensor network applications”, *Proc. 2nd Int. Conf. Embedded Networked Sensor Systems, (SenSys 2004)*, Baltimore, MD, USA, 2004

Prevención de egoísmo basada en verosimilitud en redes MANET

Alberto Rodriguez-Mayol y Javier Gozalvez
Ingeniería de Comunicaciones
Universidad Miguel Hernández de Elche
Avda. de la Universidad s/n 03202 Elche
f.rodriguez@umh.es, j.gozalvez@umh.es

***Resumen*- En las redes ad-hoc móviles es necesario que los nodos que componen la red colaboren en la retransmisión de paquetes cuando los nodos origen y destino no se encuentran dentro del rango de transmisión. Sin embargo, algunos nodos pueden negarse a cooperar para ahorrar recursos de batería o comunicación. En la literatura se han propuesto mecanismos de prevención de egoísmo basados en reputación, en los cuales los nodos observan el comportamiento de sus vecinos para detectar y aislar a aquellos nodos egoístas que no retransmiten los paquetes. Por tanto, se necesitan mecanismos de detección exactos y rápidos para distinguir los nodos cooperativos y los egoístas. Este trabajo presenta un mecanismo de detección con un novedoso enfoque basado en verosimilitud que mejora las prestaciones de los mecanismos Bayesianos empleados tradicionalmente, y que es más robusto frente al comportamiento egoísta impredecible de los nodos.**

***Palabras Clave*- Redes MANET, redes ad-hoc móviles, prevención de egoísmo, watchdog, reputación.**

I. INTRODUCCIÓN

Para asegurar unos niveles de conectividad adecuados, las redes móviles ad-hoc (MANET *Mobile Ad-hoc Networks*) requieren la cooperación de los nodos individuales para la retransmisión de paquetes desde el nodo origen al nodo destino [1]. Dado que algunos nodos pueden negarse a cooperar, por ejemplo para ahorrar recursos de batería o computación, en los últimos años ha habido un trabajo intenso en el desarrollo de esquemas de prevención de egoísmo, que pueden clasificarse genéricamente en basados en reputación, basados en crédito y basados en teoría de juegos [2]. Las estrategias basadas en reputación detectan el comportamiento egoísta o cooperativo de los nodos observando sus retransmisiones, y registrando su nivel de cooperación en tablas. Estas tablas son posteriormente utilizadas por los protocolos de enrutamiento para seleccionar la ruta más segura y aislar y evitar a los nodos egoístas identificados.

Se han propuesto diferentes técnicas para observar a los nodos y detectar su comportamiento egoísta o cooperativo. Una de las técnicas más relevantes, debido a su nivel de aceptación, simplicidad y eficiencia, dado que el proceso de observación no introduce una sobrecarga de comunicación adicional, es la técnica *watchdog* [3]. Con *watchdog*, un nodo (el nodo precursor) que transmite un paquete hacia un nodo retransmisor, usa el modo promiscuo de la MAC para observar la retransmisión del paquete dentro del tiempo establecido. Si la retransmisión es observada correctamente, el nodo precursor registra una acción positiva del nodo retransmisor en su tabla de reputación; en otro caso, se registra una acción egoísta. Sin embargo, errores esporádicos en el canal o colisiones de paquetes pueden impedir la observación correcta de la retransmisión, lo cual provoca que se registre una acción egoísta incorrectamente [4]. Estos registros incorrectos pueden afectar negativamente al funcionamiento y al rendimiento del proceso de detección, que debe decidir si un nodo debe ser o no acusado de comportarse egoístamente a partir de la información registrada por la técnica de observación. Los estudios más avanzados en esta materia basan su proceso de detección en un enfoque Bayesiano [5]-[7], que generalmente requiere la utilización de un número elevado de observaciones para reducir la probabilidad de acusar incorrectamente a un nodo cooperativo o de no detectar a un nodo egoísta. Además, los mecanismos Bayesianos se caracterizan por la necesidad de establecer un compromiso entre la exactitud y la rapidez del proceso de detección, que puede afectar negativamente a su rendimiento y al funcionamiento global de la red. Si se reduce el mínimo número de observaciones necesarias para acusar a un nodo, se mejora la rapidez en la detección, pero en un alto porcentaje de ocasiones las acusaciones son incorrectas. Si un nodo cooperativo se ve incorrectamente acusado, será aislado, no podrá participar en la red en lo sucesivo ni como origen o destino ni retransmisor, reduciendo por tanto la conectividad de la red y siendo castigado injustamente. Por el contrario, si se eleva el número mínimo de observaciones, se reduce notablemente el error en las acusaciones, pero a su vez esto provoca que los nodos egoístas descarten un número elevado de paquetes antes de que sean finalmente identificados. En este contexto, este trabajo propone un novedoso enfoque alternativo basado en verosimilitud que supera a las técnicas Bayesianas tradicionales tanto en exactitud como en rapidez de decisión. No existen en la literatura antecedentes de técnicas de detección basadas en verosimilitud. Para demostrar sus ventajas, se aplica al mecanismo de detección *watchdog*, aunque podría ser aplicado a otras técnicas de observación.

II. TÉCNICAS DE DETECCIÓN DE EGOÍSMO BAYESIANAS

Considérese una red MANET en la cual ciertos nodos rechazan selectivamente la retransmisión de algunos de los paquetes que deben retransmitir para otros nodos, con

probabilidad p_s. p_s es una variable aleatoria, con un valor distinto para cada nodo, y con función densidad de probabilidad $f_{ps}(x)$. $f_{ps}(x)$ describe la distribución del parámetro p_s en una red determinada (es decir, qué proporción de nodos descartan paquetes con probabilidad p_s=x, para x variando entre 0 y 1). Sea p_e la probabilidad de error de la técnica de observación, es decir, la probabilidad de que una acción cooperativa sea tomada por una acción egoísta. En el caso de *watchdog*, p_e es equivalente a la probabilidad de error de recepción de paquete debido a errores en el canal y colisiones de paquetes. Sea D el proceso aleatorio que describe la observación de las retransmisiones, con dos posibles eventos a considerar: D=0 si la retransmisión es observada, D=1 en otro caso. Existen dos razones por las cuales el resultado de D puede ser la no observación de la retransmisión: bien el nodo no ha retransmitido el paquete (p_s), o bien la retransmisión no ha sido correctamente detectada, con probabilidad $(1-p_s)p_e$. Este proceso se repite con cada retransmisión de paquete, conformando un proceso de tipo Binomial D_n con probabilidad p_d:

$$\Pr(D=1) = p_s + (1-p_s)p_e = p_s + p_e - p_s p_e = p_d \qquad (1)$$

Tras n observaciones, el mecanismo de detección debe decidir si el nodo está ocasionalmente actuando de manera egoísta (p_s>0). Esta decisión debe ser exacta, es decir, debe minimizar el cociente de acusaciones incorrectas *IA* (*Incorrect Accusations*) y de no acusaciones incorrectas *INA* (*Incorrect No Accusations*), y al mismo tiempo debe ser rápida, para minimizar el número δ de paquetes descartados por un nodo egoísta antes de ser detectado. *IA* se define como el cociente entre el número de nodos cooperativos acusados incorrectamente de comportarse de manera egoísta y el número total de nodos cooperativos. *INA* se define como el cociente entre el número de nodos egoístas no detectados y el número total de nodos egoístas.

Los mecanismos de detección más avanzados propuestos en la literatura son variantes del enfoque Bayesiano propuesto en [5]. Este enfoque asume que las observaciones de retransmisión permiten al nodo precursor estimar la p_s real del nodo. Para ello se supone que la $\hat{p}_s$ estimada sigue una distribución de tipo Beta, Beta(α,β). Esta distribución depende de dos parámetros, α y β. En este enfoque, α y β se inicializan a 1, y se incrementan cada vez que se observa un comportamiento egoísta o cooperativo, respectivamente. Al comienzo, cuando todavía no se ha realizado ninguna observación, el nodo precursor no tiene ninguna certeza sobre la p_s real del nodo, y por ello la función Beta(1,1) es una distribución uniforme, indicando que la $\hat{p}_s$ estimada puede tomar cualquier valor entre 0 y 1 con la misma probabilidad. Cuanto mayor es el número de observaciones realizadas, más exacta es la aproximación de la función Beta(α,β) a la p_s real del nodo retransmisor. Cuando el número de observaciones es suficientemente grande, el nodo precursor estima el valor de la p_s real, usando para ello el valor esperado de la distribución Beta(α,β) en ese momento. Este valor esperado es utilizado por el nodo precursor como una métrica para decidir si el nodo retransmisor está actuando egoístamente. Si el valor de la métrica supera el umbral de acusación τ, que es un parámetro de configuración de entrada de cada técnica que expresa el nivel máximo de egoísmo aceptable, entonces el veredicto es que el nodo retransmisor está actuando egoístamente. Se han propuesto distintas técnicas Bayesianas basadas en este procedimiento que se exponen a continuación.

La primera de las técnicas, denominada en el presente trabajo BIW (*Bayesian with Infinite Window*) [5]-[6], define la métrica como

$$M_{BIW}(n,\alpha,l) = \left.\frac{\alpha(n)}{n}\right|_{n\geq l} \qquad (2)$$

donde $\alpha(n)$ representa el número de acciones negativas registradas en las últimas n observaciones, y l es el mínimo número de observaciones que aseguran la validez estadística de la métrica. Si el valor de dicha métrica supera el umbral de acusación τ, entonces el nodo es acusado de actuar egoístamente. La segunda métrica, denominada BFW (*Bayesian with Finite Window*), que fue empleada en [7], tiene en cuenta sólo las últimas l observaciones, como se refleja en:

$$M_{BFW}(n,\alpha,l) = \left.\frac{\alpha(n-l,n)}{l}\right|_{n\geq l} \qquad (3)$$

[5] propone una mejora de la métrica BIW, denominada aquí BDF (*Bayesian with Discount Factor*) por claridad, la cual introduce un factor de descuento:

$$u = 1 - \frac{1}{l} \qquad (4)$$

Sea s el resultado de la última observación D_i. Esto quiere decir que si s=0 entonces se ha observado una retransmisión y si s=1 se ha observado un comportamiento egoísta. Entonces, la actualización de α y β sería en este caso:

$$\begin{aligned} \alpha_{DF}(i) &:= u\alpha_{DF}(i-1) + s \\ \beta_{DF}(i) &:= u\beta_{DF}(i-1) + (1-s) \end{aligned} \qquad (5)$$

El factor de descuento u atenúa la importancia de las observaciones más antiguas frente a las más recientes, de manera que el nodo retransmisor debe cooperar continuamente a lo largo del tiempo, ya que un comportamiento positivo en el pasado no compensa comportamientos negativos recientes. La métrica de egoísmo BDF puede definirse entonces como:

$$\begin{aligned} M_{BDF}(n,\alpha,l) &= \left.\frac{\alpha_{DF}(n)}{\alpha_{DF}(n)+\beta_{DF}(n)}\right|_{n\geq l} = \\ &= \left.\frac{1-u^{\alpha(n)}}{2-u^{\alpha(n)}-u^{l-\alpha(n)}}\right|_{n\geq l} \end{aligned} \qquad (6)$$

Donde las siglas *DF* indican que $\alpha_{DF}(,n)$ se refiere, no al número de observaciones de acciones egoístas en el instante n, sino al valor de α en el instante n computado según la ecuación (5). Análogamente al caso del enfoque Bayesiano original, una métrica Bayesiana con factor de descuento BDF y con una ventana de tamaño l, BFDF (*Bayesian Finite window with Discount Factor*) puede definirse como:

$$\begin{aligned} M_{BFDF}(n,\alpha,l) &= \left.\frac{\alpha_{DF}(n-l,n)}{\alpha_{DF}(n-l,n)+\beta_{DF}(n-l,n)}\right|_{n\geq l} = \\ &= \left.\frac{1-u^{\alpha(n-l,n)}}{2-u^{\alpha(n-l,n)}-u^{l-\alpha(n-l,n)}}\right|_{n\geq l} \end{aligned} \qquad (7)$$

Para un funcionamiento correcto de las técnicas de detección Bayesianas es necesario seleccionar de manera óptima el valor de los parámetros de configuración l y τ. El objetivo de la optimización es maximizar la exactitud (o equivalentemente, minimizar el error *IA* e *INA*) y la rapidez (δ) de la detección de egoístas. En la selección óptima del valor de de estos parámetros se debe tener en cuenta la influencia de los parámetros $f_{ps}(x)$ y p_e. Mientras que el valor de p_e puede ser estimado en tiempo real, como en [8], o a través de mensajes de señalización como en [9], la estimación del valor de la función de distribución $f_{ps}(x)$ en una red MANET es una tarea difícil. Además, se necesitarían un gran número de observaciones l para obtener unos cocientes de error aceptables, lo cual por otro lado incrementaría el número promedio de paquetes descartados por nodos egoístas δ. Para entender mejor la importancia de la selección de los parámetros (τ,l), se realiza a continuación una estimación analítica de su influencia en los parámetros de resultado *IA* e *INA* para la técnica BFW. Si se considera un número suficientemente grande de experimentos, los cocientes *IA* e *INA* pueden aproximarse por la probabilidad de que un nodo cooperativo sea acusado y de que un nodo egoísta no sea detectado, respectivamente. Sea A_n el evento de que un nodo haya sido acusado en el instante n. Si se asume, para facilitar los cálculos, que la métrica de egoísmo BFW se calcula solamente cada l observaciones, entonces los eventos de acusación en instantes diferentes son independientes e idénticamente distribuidos, y la probabilidad de acusación tras n observaciones se puede calcular como:

$$\Pr(A_n) = 1 - \Pr(\overline{A}_n) \approx 1 - \Pr\left(\bigcap_{i=1}^{n/l} \overline{A}_{i\cdot l}\right) \tag{8}$$

Dado que el evento de ser acusado en un instante es independiente del de ser acusado en cualquier otro instante, se puede expresar como un producto, y después se aplica la métrica BFW de la ecuación (3):

$$\Pr(A_n) \approx 1 - \prod_{i=1}^{n/l} \Pr(\overline{A}_{i\cdot l}) = 1 - \prod_{i=1}^{n/l} \Pr\left(\alpha((i-1)l, i\cdot l) \Big/ l \le \tau\right) \tag{9}$$

Igualmente, se puede simplificar el producto de probabilidades a una potencia, dado que la probabilidad de que sea acusado en un instante o en otro es la misma, por ser el tamaño de ventana siempre idéntico.

$$\Pr(A_n) \approx 1 - \Pr\left(\alpha(1,l) \Big/ l \le \tau\right)^{n/l} \tag{10}$$

Finalmente, recuérdese de la Teoría de Probabilidad, la expresión de la función de distribución binomial. Dada una variable X que sigue una distribución binomial con probabilidad p, la función de distribución F describe precisamente la probabilidad de que, tras n pruebas, se hayan dado x éxitos

$$F(x;n,p) = \Pr(X \le x) = \sum_{i=0}^{\lfloor x \rfloor} \binom{n}{i} p^i (1-p)^{1-i} \tag{11}$$

En el caso que nos interesa, las n pruebas corresponden a las n observaciones del nodo, la probabilidad p corresponde a la p_d definida anteriormente. Con todo esto, a partir de las ecuaciones (10) y (11) se puede llegar a la expresión siguiente:

$$\Pr(A_n) \approx 1 - \Pr(\alpha(1,l) \le l\tau)^{n/l} = 1 - F(\lfloor l\tau \rfloor; l, p_d)^{n/l} \tag{12}$$

Usando la ecuación (1), los las probabilidades de *IA* e *INA* se pueden expresar como:

$$\Pr(IA) = 1 - F(\lfloor l\tau \rfloor; l, p_e)^{N/l} \tag{13}$$

$$\Pr(INA) = F(\lfloor l\tau \rfloor; l, p_e + p_s - p_s p_e)^{N/l} \tag{14}$$

La Fig. 1 subraya la dependencia de las probabilidades de *IA* e *INA* respecto al umbral de acusación τ para distintos valores de probabilidad de egoísmo del nodo (p_s) y un tamaño de ventana l fijo e igual a 12 (se observa una tendencia similar si se varía l manteniendo τ fijo). Un objetivo de diseño podría ser minimizar *IA* e *INA*; sin embargo, la Fig. (1) muestra que ambos parámetros siguen tendencias opuestas. Para valores de p_s mayores o iguales a 0.2, las probabilidades *IA* e *INA* son minimizadas simultáneamente en el punto τ=0.45. Por otro lado, si p_s es igual a 0.1, τ debería tomar un valor entre 0.35 y 0.4, pero en este caso las probabilidades *IA* e *INA* no podrían ser reducidas más allá de 0.1.

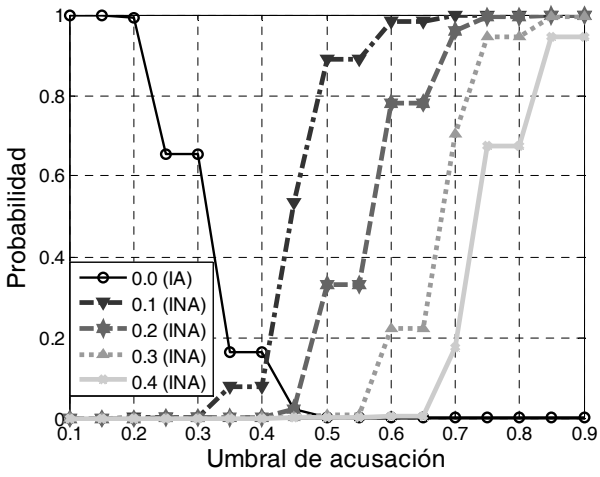


Fig. 1. Probabilidades de *IA* e *INA* en función del umbral de acusación τ. La leyenda corresponde a distnitos valores de p_s.

Con otra estrategia, la probabilidad de *IA* se puede reducir incrementando el número de observaciones l, pero sin embargo, esto conlleva a un incremento del número de paquetes descartados por nodos egoístas (δ) antes de que su comportamiento sea detectado y sean aislados. Como se ha visto, es importante subrayar que los valores óptimos de τ y l varían en función de cual sea la distribución en la red del parámetro de egoísmo p_s entre los nodos, la cual no es conocida a priori. Por ello, el rendimiento y las perspectivas de implementación de las técnicas basadas en el enfoque Bayesiano se ven considerablemente limitados.

III. Propuesta de Detección de Egoísmo Basada en Verosimilitud

Para superar las limitaciones de las técnicas de detección de egoísmo Bayesianas, este trabajo propone un nuevo enfoque basado en verosimilitud y desarrolla un nuevo mecanismo a partir de este enfoque. En el enfoque de verosimilitud se define una nueva métrica diseñada para comprobar, después de cada observación D_n, si lo más

probable es que el número de acciones egoístas observado $\alpha(n)$ corresponda exclusivamente a equivocaciones del proceso de observación, provocadas por errores en el canal radio y colisiones de paquetes, representado por la probabilidad p_e, o si por el contrario es más probable que se deba a la combinación de este efecto unido al comportamiento egoísta del nodo retransmisor, representado por la probabilidad p_s. En este contexto, se necesita una función que mida la probabilidad de que la hipótesis nula "*las acciones egoístas observadas se deben exclusivamente a la inexactitud del método de observación*" sea verdadera. Este trabajo utiliza como punto de partida para hallar dicha función de verosimilitud a la función de distribución Binomial expresada en la ecuación (11). Dada la complejidad de la expresión exacta de la función de distribución Binomial, se toma la aproximación basada en la desigualdad de Hoeffding:

$$F(x,n,p) \approx \exp\left(-2\frac{(np-x)^2}{n}\right) \tag{15}$$

A partir de dicha expresión, este trabajo propone una función de verosimilitud F_L, que tiene en cuenta el número de observaciones n, el número de observaciones de acciones egoístas $\alpha(n)$ y la probabilidad de error de observación p_e:

$$F_L(\alpha,n,p_e) \approx \exp\left(-2\frac{(\Delta_-(np_e-\alpha(n)))^2}{n}\right) \tag{16}$$

donde se define Δ_- como:

$$\Delta_-(x) = \frac{x-|x|}{2} = \begin{cases} 0 & x \geq 0 \\ x & x < 0 \end{cases} \tag{17}$$

Las métricas propuestas, denominadas métrica de verosimilitud con ventana infinita LIW (*Likelihood Infinite Window*) y métrica de verosimilitud con ventana finita LFW (*Likelihood Finite Window*), se definen como el promedio de la función de verosimilitud F_L para todas las observaciones y para las últimas l observaciones, respectivamente:

$$M_{LIW}(n,\alpha,l,p_e) = \frac{1}{n}\sum_{i=1}^{n}\exp\left(-2\frac{(\Delta_-(ip_e-\alpha(i)))^2}{i}\right)\Bigg|_{n\geq l} \tag{18}$$

$$M_{LFW}(n,\alpha,l,p_e) = \frac{1}{l}\sum_{i=n-l+1}^{n}\exp\left(-2\frac{(\Delta_-(ip_e-\alpha(i)))^2}{i}\right)\Bigg|_{n\geq l} \tag{19}$$

Al contrario de lo que sucede en las técnicas Bayesianas, la propuesta de detección basada en verosimilitud solamente acusa a un nodo de actuar egoístamente cuando el valor de la métrica de verosimilitud calculado es inferior al umbral de acusación τ. En tal caso, no se confirmaría la hipótesis nula, y opr tanto el nodo es egoísta. En este contexto, es importante señalar que τ no es una medida del valor máximo aceptable de egoísmo, sino una medida de la mínima verosimilitud de la hipótesis de que el nodo no está comportándose de manera egoísta. Por consiguiente, el parámetro τ no está aquí directamente relacionado con la distribución del parámetro p_s entre el conjunto de nodos de la red, lo cual facilita la selección de un valor adecuado del parámetro τ y mejora las perspectivas de implementación de la propuesta.

IV. Evaluación de Rendimiento

Para la evaluación del rendimiento de la técnica propuesta y su comparación con las técnicas Bayesianas tradicionales se han realizado extensos lotes de simulaciones. Cada prueba básica de simulación consiste en la valoración por parte de cada técnica de una realización concreta del proceso de detección Binomial descrito en la sección II y la determinación (usando el criterio de acusación de la técnica considerada) de si dicha realización se corresponde o no con la de un nodo egoísta. Por una realización entendemos una muestra concreta $\{\alpha(i)\}$ i=1,…,N, de observaciones de acciones egoístas, siendo N el número total de observaciones realizadas, sobre un nodo con egoísmo p_s mediante una técnica de observación con una probabilidad de error p_e. En dicha realización $\alpha(i)$ representa el número de observaciones de paquetes descartados realizadas tras i observaciones. Si tras i observaciones, con $l \leq i \leq N$, se cumple el criterio de acusación de la técnica de detección, entonces el veredicto es que el nodo está descartando paquetes. Por el contrario, si eso no ocurre para ningún valor $\alpha(i)$, con $l \leq i \leq N$, entonces el veredicto es que las observaciones de paquetes descartados en realidad corresponden a los errores de la técnica de detección de *watchdog*. El resultado de una prueba básica es por tanto, para cada técnica de detección implementada, un veredicto negativo o positivo de acusación del nodo retransmisor. La rapidez de la técnica en la detección de nodos egoístas podrá ser evaluada contando el número de observaciones k que han sido necesarias antes de la acusación, en caso de que el egoísmo del nodo en la realización concreta de la prueba fuera p_s>0. Es decir, la rapidez sería el mínimo valor de i tal que la secuencia $\{\alpha(i)\}$ i=1,…,N cumple la condición de acusación en la observación i=k. Por otro lado, la exactitud de la técnica se puede evaluar comparando el veredicto de la técnica con el egoísmo real del nodo retransmisor p_s. Una acusación es incorrecta si el egoísmo real del nodo es nulo p_s=0, y una no acusación es incorrecta si por el contrario p_s>0. A través de la realización de un gran número de pruebas en distintas condiciones, se pueden obtener promedios numéricos de los parámetros de evaluación considerados: δ, *IA* y *NIA*. A continuación se explicarán cuáles han sido las condiciones consideradas y el procedimiento para hallar los valores óptimos de los parámetros de configuración (τ,l).

Se realizaron simulaciones preliminares para seleccionar los valores óptimos de los parámetros (τ,l) para cada una de las técnicas, en escenarios con diferentes valores de p_s. Es necesario recalcar que los valores óptimos de (τ,l) no tienen porque ser iguales para todas las técnicas. En primer lugar, el parámetro τ expresa conceptos diferentes en las técnicas Bayesianas y de verosimilitud, como ya se ha comentado. Por simplicidad, se ha usado la misma notación en ambos casos, puesto que se trata de un cierto umbral que toma valores entre 0 y 1 y que es comparado con la métrica para decidir si el nodo es egoísta. Sin embargo, en las técnicas Bayesianas τ expresa un umbral de máximo egoísmo permitido, mientras que en las de verosimilitud determina la mínima certeza aceptable de que el nodo no sea egoísta. Además, el funcionamiento de cada técnica requiere unos valores específicos de (τ,l). La selección de los valores óptimos de (τ,l) variará también para cada red concreta en función de la

distribución $f_{ps}(x)$ del parámetro p_s del egoísmo de los nodos en dicha red.

Por las razones mencionadas, en el proceso de selección de los valores óptimos de (τ,l) para cada técnica, se deben tener en cuenta los distintos valores posibles del parámetro p_s de un nodo retransmisor. Sea $x_i \in \{0.0,...,1.0\}; i = 1,...,N_{ps}$ el conjunto finito y discreto de los posibles valores del parámetro p_s dentro de la distribución $f_{ps}(x)$. Entonces, los valores promedio de $IA(m,\tau,l)$, $INA(m,p_s,\tau,l)$ y $\delta(m,p_s,\tau,l)$ para cada valor de p_s, para cada técnica de detección m y para un conjunto de N nodos puede computarse usando el siguiente principio de proporcionalidad:

$$\begin{aligned} IA(m,f_{ps},\tau,l) &= f_{ps}(0.0)IA(m,\tau,l) \\ INA(m,f_{ps},\tau,l) &= \sum_{i=2}^{N_{ps}} f_{ps}(x_i)INA(m,x_i,\tau,l) \\ \delta(m,f_{ps},\tau,l) &= \sum_{i=2}^{N_{ps}} f_{ps}(x_i)\delta(m,x_i,\tau,l) \end{aligned} \quad (20)$$

Dada la infinidad de posibles distribuciones $f_{ps}(x)$ del parámetro de egoísmo p_s en una red real, el paso siguiente ha sido definir un conjunto de distribuciones $\{f_{ps}(x)_i\}$ que sean representativas de las potenciales distribuciones en una red MANET real. En este trabajo, las características más influyentes de la distribución $f_{ps}(x)$ son la proporción de nodos no egoístas $f_{ps}(0)$ y la proporción de nodos egoístas con un nivel de egoísmo p_s reducido pero no nulo. A continuación se expone un ejemplo para ilustrar la primera característica. En una red con muchos nodos no egoístas será más perjudicial que la técnica de detección empleada tenga un nivel alto de *IA* que de *INA*, es decir, dado que la mayoría son no egoístas, con un nivel elevado de *IA* habrá un número considerable de acusaciones incorrectas. En cambio no sería muy perjudicial que tuviera un valor más elevado de INA, dado que hay pocos egoístas en la red. Dado que, como se ha comentado en la sección II, existe un compromiso entre los parámetros *IA* e *INA*, la selección de la pareja de parámetros (τ,l) deberá tener en cuenta la proporción de nodos no egoístas $f_{ps}(0)$. Por ello, a la hora de determinar las distribuciones $f_{ps}(x)$ a tener en cuenta, se han considerado 8 valores para la proporción de nodos no egoístas: $f_{ps}(0) \in \{0.2,0.3,...,0.9\}$.

Respecto a la segunda característica, los nodos que tienen un bajo nivel de egoísmo p_s son más difíciles de detectar que los nodos con un p_s alto, es decir, obtienen valores elevados de *INA*, dado que su comportamiento se puede enmascarar con la probabilidad de observación errónea p_e. De ahí que un nodo con un nivel de egoísmo reducido pueda confundirse más fácilmente con un nodo no egoísta, y viceversa. Esto puede apreciarse claramente además en la Figura 1, en las curvas correspondientes a la probabilidad de *INA* para distintos valores de p_s. Por ejemplo, fijando un valor de τ=0.6, la probabilidad de que un nodo con egoísmo reducido (p_s=0.1) pase desapercibido es casi completa, mientras que para un nodo con mayor nivel de egoísmo (por encima de $p_s \geq 0.4$) es nula. Por consiguiente, se han considerado 3 tipos de función para la $f_{ps}(x)$ de los nodos egoístas: uniforme, linealmente creciente (menor proporción de nodos con egoísmo difícil de detectar) y linealmente decreciente. Combinando estos 3 tipos de función, con los 8 valores mencionados de proporción de nodos no egoístas $f_{ps}(0)$, se han obtenido un conjunto de 24 distribuciones representativas $\{f_{ps}(x)_i\}$.

En el proceso de selección de los valores óptimos de los parámetros de configuración (τ,l) debe tenerse en cuenta el compromiso anteriormente comentado entre la rapidez y la exactitud en las técnicas de detección, especialmente en las técnicas Bayesianas. Por ello, se han considerado dos criterios diferentes. El criterio de exactitud consiste en seleccionar la pareja (τ,l) que minimice la suma de los promedios de los cocientes de *IA* e *INA*. Por otro lado, el criterio de rapidez tiene en cuenta tanto la suma de los promedios de *IA* e *INA* como también el número de paquetes descartados antes de que un nodo egoísta sea detectado δ. La Tabla 1 muestra los valores de (τ,l) que en promedio se ajustan mejor a los criterios de optimización de exactitud y rapidez para cada una de las técnicas de detección. El conjunto de valores de (τ,l) evaluados fueron todas las posibles combinaciones de $\tau \in \{0.1,0.15,\ldots,0.9\}$ y $l \in \{3,6,12,24,48\}$, y los resultados han sido obtenidos considerando una p_e igual a 0.1. La Tabla 1 también muestra el porcentaje de ocasiones en que el valor seleccionado de (τ,l), que mejor se ajustaba a los criterios exactitud y rapidez en promedio, resultaba asimismo ser la configuración óptima en cada una de las veinticuatro distribuciones $\{f_{ps}(x)_i\}$ consideradas individualmente (parámetro *Opt*. en Tabla 1).

En primer lugar, debe señalarse que en todas las técnicas Bayesianas el valor de l que ha resultado seleccionado es el más alto (48) de entre todos los que han sido considerados, cuando se utilizaba el criterio de exactitud. Por otro lado, si se considera el criterio de rapidez, se requiere un valor apreciablemente menor del parámetro l, para reducir su impacto sobre el número de paquetes descartados δ. Pero esto a su vez, conlleva a un reajuste del parámetro τ seleccionado a un valor mayor y por tanto más permisivo con el egoísmo, para compensar el efecto colateral negativo de reducir el número de observaciones l sobre el cociente de *IA*. El compromiso entre la rapidez y la exactitud de la detección, especialmente en las técnicas Bayesianas, queda patente en este resultado.

Por otro lado, las técnicas de verosimilitud no necesitan un valor elevado del parámetro l, incluso cuando se considera el criterio de exactitud, según muestra la Tabla 1, lo cual es de esperar que reduzca el número de paquetes descartados por egoístas δ. Los resultados mostrados también ponen de manifiesto que la configuración óptima en promedio de (τ,l) para el mecanismo LFW resultaba ser la configuración óptima para casi todas las distribuciones consideradas en el conjunto $\{f_{ps}(x)_i\}$. En concreto, resulta ser óptima en el 100% de los casos cuando se considera el criterio de exactitud y en el 66% de los casos cuando se considera el criterio de rapidez. En este aspecto las restantes técnicas resultan ser considerablemente inferiores. La configuración de la técnica BDF resulta ser óptima en el 100% de los casos para el criterio de exactitud y el 50% para el de rapidez, pero los valores de (τ,l) de ambas configuraciones son completamente diferentes, lo cual no ocurre en el caso de la técnica LFW. Esta es una ventaja de implementación muy notable de LFW respecto al resto de técnicas, ya que la distribución del parámetro de egoísmo entre los nodos de la red es desconocida y con esta única configuración para LFW se optimiza a la vez la exactitud y la rapidez para casi todas las

distribuciones consideradas del parámetro p_s. Puede concluirse por tanto a partir de la Tabla I que la elección de unos parámetros de configuración para las técnicas Bayesianas que maximicen la exactitud, disminuyen considerablemente la rapidez en la detección de egoístas, y viceversa. Este compromiso no existe en las técnicas basadas en verosimilitud, especialmente en la técnica con ventana finita LFW, dado que una misma configuración resulta ser óptima tanto en rapidez como en exactitud.

		BIW	BFW	BDF	BFDF	LIW	LFW
τ	exact.	0.20	0.30	0.45	0.35	0.30	0.10
	rapidez	0.35	0.35	0.50	0.50	0.35	0.10
l	exact.	48	48	48	48	12	3
	rapidez	6	24	3	12	3	3
Opt	exact.	58.33	66.67	100.0	62.50	66.67	100.0
	rapidez	16.67	16.67	50.00	41.67	4.17	66.67

Tabla 1: Configuración óptima de (τ,l).

La Tabla 2 muestra el promedio de los parámetros de resultados *IA*, *INA* y δ obtenidos para el conjunto $\{f_{ps}(x)_i\}$ de distribuciones y para los distintos mecanismos de detección. Los resultados mostrados en la Tabla 2 fueron obtenidos usando la configuración óptima de (τ,l) mostrada en la Tabla 1, para cada técnica y criterio de selección. Ante todo, debe señalarse que ambas propuestas de verosimilitud, y especialmente LFW, obtienen el mejor rendimiento tanto con el criterio de exactitud como con el criterio de rapidez, lo cual permite conseguir un porcentaje muy bajo de acusaciones incorrectas de nodos cooperativos y de nodos egoístas no detectados, además de lograr el menor número de paquetes descartados por los nodos egoístas antes de ser detectados. Si bien alguna de las técnicas Bayesianas puede obtener una marca mejor de rendimiento en alguno de los parámetros analizados, esto siempre está asociado a una degradación considerable de otro parámetro de rendimiento. En particular, los valores altos de l de las configuraciones que optimizan el criterio de exactitud en las técnicas Bayesianas están correlacionados con cocientes de *IA* e *INA* bajos, pero también con una gran degradación en el número de paquetes descartados por los nodos egoístas antes de ser detectados (δ). Por otro lado, también en las técnicas Bayesianas, el criterio de rapidez reduce el parámetro δ, pero esto se consigue a expensas de incrementar el error de detección, ya sea en el cociente de *IA* o de *INA*. Las diferencias entre las técnicas Bayesianas y las técnicas de verosimilitud se pueden explicar de la siguiente manera. En el enfoque Bayesiano, τ se corresponde al umbral máximo de egoísmo aceptable. Por consiguiente, al incrementar su valor para reducir el cociente de acusaciones incorrectas *IA* (de manera que se compense el efecto de reducir el parámetro l) provoca que los nodos con un egoísmo menor que el estipulado por el umbral $p_s<\tau$ no sean correctamente detectados y por tanto el cociente *INA* aumenta. Por el contrario, en las técnicas de verosimilitud, τ mide la mínima verosimilitud requerida para que la hipótesis de que el nodo no está actuando egoístamente sea aceptada. En este caso, la reducción de τ, es decir, la exigencia de un menor nivel de verosimilitud, rebaja el cociente de acusaciones incorrectas *IA*, pero no provoca que los nodos con un bajo nivel de egoísmo p_s no sean detectados, ya que τ se refiere en este caso a la verosimilitud y no directamente al nivel de egoísmo p_s del nodo. Entre las técnicas de verosimilitud, es preferible emplear LFW ya que además de lograr el mejor rendimiento promedio con la configuración óptima seleccionada, sus prestaciones son también óptimas para la mayoría de las distribuciones del parámetro f_{ps} consideradas en el conjunto $\{f_{ps}(x)_i\}$ y para los dos criterios de exactitud y de rapidez. Por tanto puede concluirse que, con la técnica LFW de verosimilitud se consigue superar el compromiso existente en las técnicas Bayesianas entre la rapidez y exactitud de la detección que impide hallar una configuración de (τ,l) que optimice a la vez los tres parámetros de rendimiento considerados de *IA*, *INA* y δ, independientemente del número de nodos egoístas que participen en la red y de su grado de egoísmo.

		BIW	BFW	BDF	BFDF	LIW	LFW
IA [%]	exact.	1.86	3.76	0.46	3.38	0.94	0.06
	rapidez	11.32	2.18	22.70	6.44	0.62	0.06
INA [%]	exact.	3.40	0.20	0.00	0.40	0.00	0.60
	rapidez	3.20	3.40	3.00	1.00	2.80	0.60
δ	exact.	12.14	13.13	17.03	13.04	4.87	2.35
	rapidez	2.04	7.33	1.51	6.04	2.78	2.35

Tabla 2: Rendimiento promedio.

V. CONCLUSIONES

En trabajos anteriores sobre redes MANET, han sido propuestos mecanismos de prevención de egoísmo basados en reputación para detectar y aislar a posibles nodos egoístas que no colaboran en la retransmisión de paquetes para otros nodos, generando problemas de conectividad. Los mecanismos de detección propuestos hasta la fecha se basan en un enfoque Bayesiano con distintas variantes que se caracteriza por un compromiso entre la exactitud y la rapidez del proceso de detección. En este contexto, este trabajo propone un novedoso enfoque para el diseño de mecanismos de detección basado en verosimilitud, que tiene en cuenta de manera explícita la probabilidad de error del método de observación del comportamiento de los nodos. Con dicho enfoque, la técnica de detección no evalúa directamente el egoísmo del nodo, sino la verosimilitud de que las observaciones realizadas correspondan al comportamiento de un nodo no egoísta. El estudio llevado a cabo demuestra que la propuesta basada en verosimilitud supera el rendimiento de las propuestas Bayesianas, tanto en términos de exactitud como de rapidez de detección. Otro resultado importante obtenido es que con el método de verosimilitud se obtiene un rendimiento óptimo con la misma configuración de los parámetros de entrada de la técnica para la gran mayoría de las distribuciones del parámetro de egoísmo de los nodos consideradas. De esta manera, no es necesario estimar a priori la distribución real en la red de este parámetro, lo cual sería además difícilmente realizable.

AGRADECIMIENTOS

Este trabajo ha sido posible por el apoyo del Ministerio de Ciencia e Innovación del Gobierno de España y de los fondos FEDER a través de los proyectos TEC2008-06728, de la Generalitat Valenciana a través de los proyectos ACOMP/2010/111 y BFPI/2007/269.

REFERENCIAS

[1] S. Buchegger, J. Mundinger y J.-Y Le Boudec, "Reputation Systems for Self-organized Networks," *IEEE Technology and Society Magazine*, vol. 27, no. 1, pp. 41–47, Marzo 2008.

[2] Y. Yoo y D.P. Agrawal, "Why Does it Pay to Be Selfish in a MANET?," *IEEE Wireless Communications Magazine*, vol. 13, no. 6, pp. 87-97, Diciembre 2006.

[3] S. Marti, T. J. Giuli, K. Lai y M. Baker, "Mitigating Routing Misbehavior in Mobile Ad-hoc Networks," *en Libro de Actas de la ACM International Conference on Mobile Computing and Networking*, pp. 255-265, 2000.

[4] A. Rodriguez-Mayol y J. Gozalvez, "On the Implementation Feasibility of Reputation Techniques for Cooperative Mobile Ad-hoc Networks," *en Libro de Actas del 16th European Wireless*, Abril 2010.

[5] S. Buchegger y J.-Y Le Boudec, "A Robust Reputation System for P2P and Mobile Ad-hoc Networks," *en Libro de Actas del 2nd Workshop on the Economics of Peer-to-Peer Systems*, Junio 2004.

[6] D. Djneouri y N. Badache, "On Eliminating Packet Droppers in MANET: a Modular Solution," *Ad hoc Networks*, vol. 7, no. 6, pp. 1243-1258, Septiembre 2009.

[7] L. Yang, J.M. Kizza, Alma-Cemerlic y F. Liu, "Fine-Grained Reputation-based Routing in Wireless Ad Hoc Networks," *en Libro de Actas del IEEE Intelligence and Security Informatics*, pp. 75-78, Junio 2007.

[8] H. Jiang, Y. Yang, J. Xu y L. Wang, "Estimation of Packet Error Rate at Wireless Link of Vanet," *Advances in Wireless Sensors and Sensor Networks, Lecture Notes in Electrical Engineering*, vol. 64, pp. 329-359, 2010.

[9] B. Han y S. Lee, "Efficient Packet Error Rate Estimation in Wireless Networks," *en Libro de Actas de la 3rd International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities*, pp. 1-9, Mayo 2007.

Eficiencia Energética de un Mecanismo de Selección Dinámica de Interfaces en Redes Ad-hoc Inalámbricas

Luis Sánchez, Jorge Lanza, Luis Muñoz
Departamento de Ingeniería de Comunicaciones
Universidad de Cantabria
Laboratorios I+D Telecomunicación, Plaza de la Ciencia s/n, 39005 Santander, España
{lsanchez, jlanza, luis}@tlmat.unican.es

***Resumen*- Energy efficiency is critical to ensuring scalability, embedding, and portability of emerging computing and communication systems. It is of particular interest in the design of mobile computing systems because of the limitations in energy and power availability. This paper presents and compares in terms of energy efficiency two strategies for the dynamic selection of the outbound interface on multi-radio devices in wireless ad-hoc networks. Findings from the studies show that intelligent selection of communication interface in heterogeneous ad-hoc networks leads to more efficient use of the energy consumed while assuring the quality of service parameters necessary for the correct provision of applications running on top of wireless ad-hoc mobile networks.**

***Palabras Clave*- Energy efficiency, Ad-hoc network, Universal Convergence Layer**

I. INTRODUCCIÓN

Aunque las redes inalámbricas se vengan usando desde hace bastantes años, sólo recientemente se ha empezado a prestar una atención explícita a la eficiencia energética en este tipo de tecnologías. Es evidente que cuando las fuentes de energía son costosas o escasas, la eficiencia energética se torna fundamental. En algunos casos, las fuentes de energía de los nodos que componen la red son completamente no renovables, lo que impone unas restricciones que es obligatorio atender durante el diseño y operación de la red.

Sin embargo, las cosas no son siempre tan simples. Si la eficiencia energética fuera el único aspecto a tener en cuenta, la mejor política sería no transmitir nada. Las baterías durarían eternamente, logrando así una eficiencia energética óptima. Obviamente, es necesario tener en cuenta otros aspectos de rendimiento de las comunicaciones, lo que hace que la forma de incluir la eficiencia energética en la ecuación esté lejos de estar clara. Una posible solución es tratar de minimizar el consumo energético habida cuenta de que se cumplen ciertas condiciones en cuanto a rendimiento (tasa binaria, retardo, etc.). Alternativamente se puede tratar de maximizar la tasa binaria o minimizar el retardo por cada Julio consumido. Ninguna de estas formulaciones del problema conduce a una metodología fácilmente implementable.

Se han propuesto diferentes aproximaciones al problema de optimizar el consumo energético en diferentes capas de la pila de comunicaciones. A nivel de enlace, es posible evitar la transmisión de información cuando las condiciones del canal sean demasiado adversas tal y como se estudia en [1]. También se ha probado a combinar técnicas de corrección de errores (FEC) y de retransmisión (ARQ) para tratar de conservar energía, esto es, llegar a una solución de compromiso entre el número de retransmisiones y la longitud del código corrector que maximicen el rendimiento del sistema) tal y como se plantea en [2]. Otras alternativas plantean protocolos de enrutamiento energéticamente eficientes que establezcan rutas entre los nodos de la red que aseguren que todos ellos consumen sus baterías de manera uniforme tal y como se propone en [3][4] o que evitan usar rutas que pasan por nodos con un nivel de batería bajo. Soluciones más complejas como las expuestas en [5] explotan el paradigma de la optimización entre capas y controlan la topología de la red variando la potencia de transmisión de ciertos nodos de forma que se cumplan ciertas propiedades de la red.

El diseño entre capas es particularmente apropiado para abordar las restricciones energéticas, ya que no sólo es posible abordar la minimización del consumo energético en las diferentes capas de la pila de protocolos sino que a la vez se puede considerar las necesidades en cuanto a rendimiento y parámetros de calidad del sistema que es necesario cumplir. Mientras que las soluciones que son específicas de una capa pueden obviar información relevante de otras capas, las soluciones basadas en diseño entre capas explotan un conocimiento más global del sistema que puede adoptar soluciones sub-óptimas en cada capa que combinadas resultan en una solución óptima a nivel de sistema. En este artículo, se presenta de manera breve el marco sobre el cual se ha implementado un mecanismo de selección dinámica de la interfaz de transmisión. Asimismo se evalúan dos estrategias de selección distintas desde el punto de vista de su eficiencia energética.

II. CAPA DE CONVERGENCIA

El concepto de aislar las capas superiores de las tecnologías de acceso inalámbricas y con ello conseguir dar soporte de manera transparente a sistemas multimodo, en el que los dispositivos están equipados con varias interfaces de acceso a la red, es posible al introducir una capa de convergencia en la pila de protocolos. La Capa de Convergencia Universal (UCL) que se propone en este

artículo tiene básicamente dos misiones. Principalmente, se plantea como un habilitador para el uso de múltiples tecnologías de acceso de manera transparente en un mismo dispositivo. De esta manera se asegura la compatibilidad tanto hacia adelante como hacia atrás a la vez que se permite explotar la heterogeneidad, en términos de posibilidades de acceso a la red, en favor del usuario. Esto se consigue definiendo una interfaz común hacia la capa de red mientras se maneja y hace uso de las diferentes tecnologías de acceso subyacentes, independientemente de las características de sus capas de Control de Acceso al Medio (MAC) y Física (PHY). Además, la UCL también posibilita el paradigma de la optimización entre capas. Su privilegiada posición en la pila de protocolos permite que la UCL tenga acceso a información de varias capas y con ello manejar los flujos de datos con el fin de maximizar el rendimiento del sistema. Así entre otras posibilidades, el uso de información del estado del canal enriquecerá la selección de rutas de un protocolo de enrutamiento y mejorará la gestión de los parámetros de nivel MAC de una interfaz de acceso, dependiendo del estado de batería o de los requerimientos de Calidad de Servicio (QoS) exigidos por un flujo de datos.

La Figura 1 presenta los diferentes bloques funcionales en los que se divide la UCL. Cada uno de estos módulos se especializa en proveer una de las funcionalidades que ofrece la UCL. Este diseño modular permite añadir o quitar funcionalidades dependiendo de cuales sean los requerimientos y características del equipo y la red sobre la que se opere. Por claridad, en la Figura 1 se muestra una arquitectura simplificada donde únicamente se presentan los módulos involucrados en los mecanismos de selección dinámica de la interfaz de transmisión que se evalúan en este artículo.

La acción coordinada de los módulos que se describen a continuación permite implementar las estrategias de gestión de las comunicaciones que se describirán en la siguiente sección.

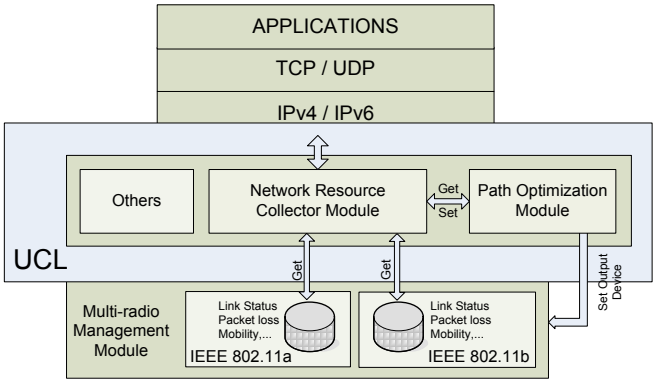


Fig. 1. Arquitectura de alto nivel simplificada de la UCL

A. Multi-radio Management Module

La UCL oculta la complejidad asociada a la posible multiplicidad de tecnologías de acceso ofreciendo una única interfaz hacia las capas superiores. De esta forma, los dispositivos multi-radio podrán mantener un único identificador de red (una única dirección IP) independientemente del número de tecnologías de acceso de que dispongan. Esto elimina los problemas asociados al multihoming a la vez que permite hacer el uso que más convenga de las múltiples posibilidades de comunicación que poseen por su naturaleza multi-radio. La posibilidad de usar diferentes enlaces para comunicarse con otro dispositivo permite que la UCL decida cual es la mejor forma de comunicarse con él en cada momento dependiendo de las necesidades de la comunicación y las condiciones de contorno.

B. Network Resource Collector

La información exportada por las tecnologías de acceso puede ser muy valiosa para otras capas de la pila de protocolos. Teniendo en cuenta que esta información puede tener diferentes significados dependiendo de donde y por quién sea utilizada, dentro de la UCL se define este módulo cuya funcionalidad es la de recoger la información de los interfaces de red y ofrecerla tanto a los otros módulos de la arquitectura como a otros componentes en las capas superiores. De esta manera se habilita la optimización entre capas.

C. Path Optimization Module

Seleccionar cual es la interfaz de acceso óptimo que se debe usar para transmitir en cada momento puede depender de las preferencias del usuario, de las restricciones de calidad de servicio que haya que garantizar y del estado de la red. En este módulo se implementan las estrategias de decisión con las que seleccionar esta interfaz en base a la información recogida por el módulo Network Resource Collector.

III. Optimización del consumo basada en selección dinámica de la interfaz de transmisión

La UCL emplea un algoritmo de Decisión basada en Múltiples Atributos (MADM, *Multiple Attribute Decision Making*) para determinar cuál de los interfaces disponibles usar en cada momento para conseguir un rendimiento óptimo del sistema. Para ello, se evalúa una función de utilidad para cada uno de los posibles enlaces que se pueden utilizar para transmitir y aquél que en cada momento arroje el mejor resultado será el que se emplee para enviar el paquete a su destino.

Para implementar la función de utilidad es posible utilizar diferentes políticas así como emplear diferentes parámetros. En este artículo proponemos dos posibilidades que serán evaluadas y comparadas. En ambas es el transmisor quien determina la interfaz por la que se transmiten los paquetes hacia su destino.

En la primera de ellas, la UCL decide qué interfaz utilizar en base a la Relación Señal a Ruido (SNR) observada en los diferentes canales inalámbricos disponibles. Para ello, se definen una serie de niveles umbral para cada tecnología de acceso de las que disponga el dispositivo. Estos niveles se basan en la SNR observada y cada uno de ellos lleva asociado un valor de utilidad diferente que trata de maximizar la tasa binaria y minimizar las perdidas de paquetes. Es importante destacar que la SNR es un parámetro que permite hacer una estimación de la calidad del enlace y el estado del canal.

En las pruebas realizadas para evaluar la solución implementada, se han usado dos tecnologías de acceso inalámbricas, IEEE 802.11b y IEEE 802.11a. Para este caso, cuando el canal 802.11a se degrada y entra el nivel

denominado malo, el canal 802.11b todavía está en el nivel bueno por lo que la función de utilidad de la interfaz 802.11b supera a la de la interfaz 802.11a.

La segunda técnica de selección dinámica de la interfaz utilizada para la transmisión se basa en evaluar el estado del canal usando el número de paquetes perdidos. Más específicamente, la UCL evalúa las ráfagas de paquetes recibidos correcta e incorrectamente para decidir cuál es el estado del canal y con ello seleccionar la interfaz a utilizar en cada momento.

Los canales inalámbricos se comportan de manera rafagueante debido a los procesos de desvanecimiento que soportan. Por ello, es posible que, incluso en casos en los que la SNR a priori indica condiciones adversas, se puedan recibir paquetes de manera correcta y viceversa. Esta estrategia trata de beneficiarse de este comportamiento y reaccionar de manera rápida ante la aparición de estos periodos (buenos o malos). Al no ser posible obtener esta información de la tecnología de acceso que no se está empleando, la UCL toma la decisión fijándose en las ráfagas de paquetes erróneos o correctos transmitidos por la interfaz en uso. De este modo, cuando se observa una ráfaga de paquetes perdidos de una longitud predefinida, la UCL decide que la transmisión pase a una interfaz más robusta frente a los errores en el canal. Obviamente, si ya se está utilizando la tecnología de acceso más robusta, no es posible hacer nada por lo que la transmisión continúa por dicha interfaz. En el caso contrario, cuando se reciben de manera consecutiva un número pre-definido de paquetes correctamente, la UCL pasará a utilizar una interfaz que ofrezca mayor rendimiento aunque sea menos robusto frente a los efectos adversos del canal radio. Para evitar un efecto ping-pong, se define un mecanismo que incrementaría el tamaño de las ráfagas que conllevan un cambio de interfaz con lo que se suavizaría la frecuencia de cambio de interfaz.

El número de paquetes perdidos en una ráfaga en el cual se establece la decisión de pasar la transmisión a una tecnología más robusta (denominado umbral de *downgrading*) se puede adaptar para llegar a una solución de compromiso entre la tasa binaria, las perdidas de aplicación y la eficiencia energética. Igualmente el número de paquetes correctamente recibidos de manera consecutiva a partir del cual se decide pasar a una tecnología más rápida y eficiente pero menos robusta (denominado umbral de *upgrading*) también es adaptable.

Es importante mencionar que dado que las decisiones se toman a nivel local en el transmisor, no es necesaria ningún tipo de señalización entre los nodos involucrados en la comunicación.

IV. Evaluación de la Eficiencia Energética de los Mecanismos de Selección Dinámica de Interfaz

En esta sección se presentan los resultados obtenidos a través de la evaluación experimental realizada para validar los beneficios de la selección inteligente de la interfaz de red más adecuada para la transmisión de datos implementada en la UCL.

El escenario escogido para llevar a cabo los experimentos permite evaluar diversas condiciones de canal. Desde un canal muy bueno (Localización 1 en la Figura 2) hasta un canal que genera una alta degradación de las comunicaciones (Localización 3 en la Figura 2). El uso de este entorno de pruebas permite valorar el comportamiento de la UCL en un escenario real, lo que posibilita extraer conclusiones directamente aplicables a la propia experiencia de usuario. Se han empleado dos equipos portátiles, cada uno de ellos equipado con una interfaz IEEE 802.11a y otra IEEE 802.11b. Esta elección está justificada por el hecho que hoy en día son las tecnologías más usadas en redes de inalámbricas de área personal (WPAN) y de área local (WLAN). Además, dado que operan en diferentes rangos de frecuencia, emplean diferentes capas físicas y capas MAC ligeramente diferentes, son un buen ejemplo del concepto de heterogeneidad. Los test realizados han consistido en desplazar un terminal entre las Localizaciones 1 y 3, en ambos sentidos, manteniendo el otro nodo fijo cerca de la Localización 1 (representado por la cruz roja en la Figura 2). El intercambio de datos se ha realizado siendo el nodo fijo el transmisor y forzando la velocidad de transmisión a los máximos posibles para cada tecnología, 11 Mbps para IEEE 802.11a y 54 Mbps para IEEE 802.11b.

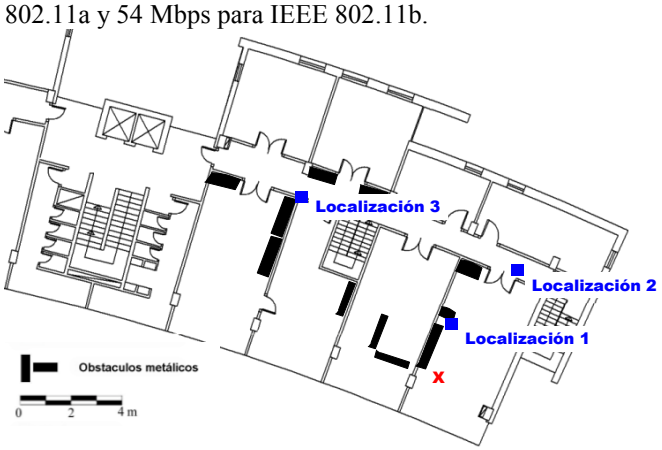


Fig. 2. Entorno de realización de la campaña de medidas

Adicionalmente, se ha simulado en Matlab® este escenario mediante una secuencia de estados que modelan la recepción de las tramas transmitidas. Se ha empleado el modelo Gilbert-Elliot donde los parámetros de partida tomados de [6] se han optimizado a partir de la evaluación experimental del escenario de la Figura 2. Este modelo se ha empleado por su simplicidad, si bien se están desarrollando estudios similares usando modelos de canal más precisos [12]. Cada simulación incluye la transmisión de 60000 tramas de nivel de enlace. En el escenario de simulación planteado, dos tercios de las tramas se transmiten en un canal muy bueno (Localización 1) mientras que las restantes se distribuyen equitativamente entre los otros dos canales (Localización 2 y 3). Es importante señalar que la capa MAC de IEEE 802.11 implementa un esquema ARQ por el cual una trama es retransmitida en caso de error. Típicamente, se retransmite hasta cuatro veces. Por tanto, la tasa de error de trama (FER) no es equivalente a la tasa de error de paquete (PER). Este hecho también ha sido tenido en cuenta en las simulaciones. Se ha seguido el método Monte Carlo, repitiendo el proceso de simulación 1000 veces.

Basándose en los valores de consumo de potencia por bit extraídos de [7], se puede comparar la eficiencia energética de las diferentes estrategias y cuantificar la mejora obtenida por la introducción de la optimización de la selección

inteligente de la interfaz inalámbrica frente a una selección a ciegas.

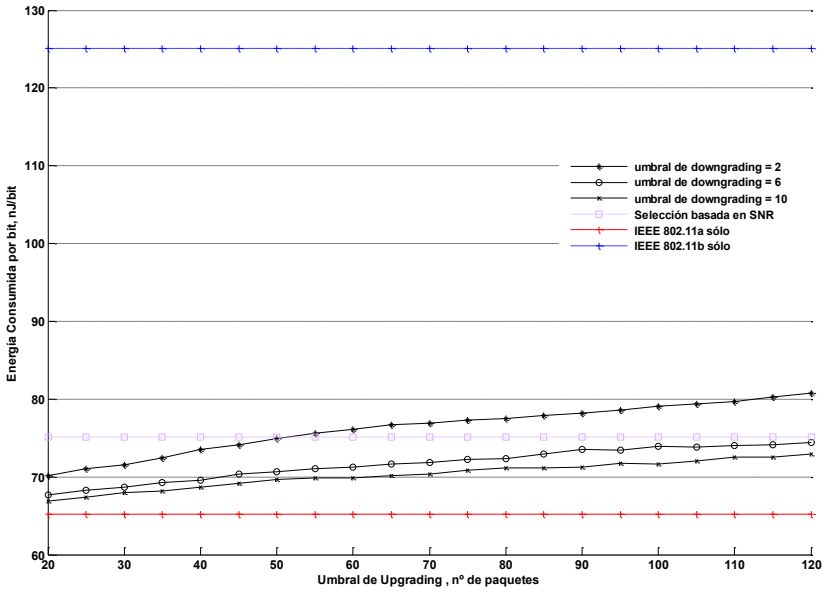


Fig. 3. Eficiencia energética de las soluciones UCL y la no-UCL

Tal como se observa en la Figura 3, la selección de la interfaz 802.11a conlleva una leve mejora de la eficiencia energética. Esto es debido a que, por definición, presenta una mejor eficiencia energética tal como se observa en la Figura 4.

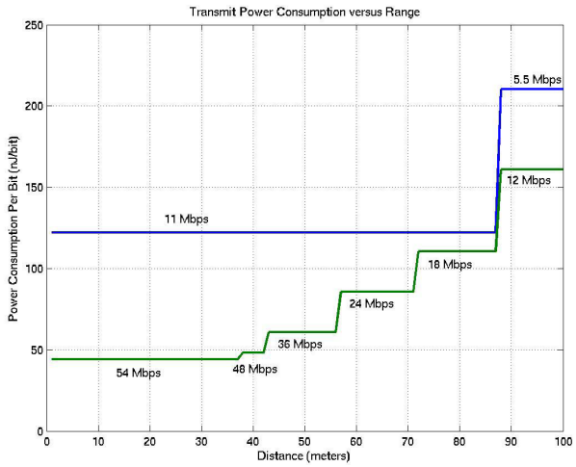


Fig. 4. Consumo por bit transmitido (802.11b vs 802.11a). Fuente: [7]

El consumo de batería por unidad de información es menor que en las otras estrategias estudiadas. Sin embargo, IEEE 802.11a es una tecnología menos robusta y presenta una FER más alta en la mismas condiciones. Es por esto que, tal como se refleja en la Figura 5, la simulación muestra una significante perdida de paquetes al emplear únicamente la interfaz IEEE 802.11a. Los valores representados en la Figura 5 corresponden a los paquetes perdidos a nivel de aplicación, es decir, aunque cada trama errónea se retransmita hasta 4 veces a nivel de enlace, ninguna de esas retrasmisiones ha sido recibida sin error. A pesar de esta pérdida de paquetes y las correspondientes retransmisiones, lo cual degrada notablemente la comunicación, la inherente eficiencia energética en transmisión de las interfaces 802.11a hace que los resultados obtenidos, al analizarlos desde un enfoque puramente energético, sean ligeramente mejores.

Esta situación es causada en parte por el porcentaje de uso del canal bueno y del canal malo (i.e. 2/3 vs 1/3). Una diferente distribución habría reportado resultados diferentes en cuanto a eficiencia energética en el caso de solo emplear IEEE 802.11a.

Por el contrario, al usar solo 802.11b, la eficiencia es mucho menor aunque la tasa de error de trama obtenida es menor en cualquiera de los canales atravesados en el escenario móvil.

Dicho esto, para realizar una correcta comparativa no es suficiente con tener en cuenta una única métrica, sino que es necesario evaluar la ganancia de las soluciones propuesta tanto desde el punto de vista de eficiencia energético como desde la mejora de la calidad de servicio.

Esta afirmación se refleja mejor en la Figura 5, donde el eje de ordenadas izquierdo representa el consumo de energía por bit y el eje de ordenadas derecho la pedida de paquetes a nivel de aplicación para cada uno de los casos. Una elección adecuada de los parámetros para la estrategia de optimización de pérdida de paquetes, hace que la UCL consiga un relativo menor consumo de potencia por bit unido a una gran reducción de la perdida de paquetes a nivel de aplicación. Por ejemplo, definiendo el umbral de *downgrading* en 6 paquetes y el de *upgrading* en 65, se obtiene tres veces menos paquetes perdidos al tiempo que se sitúa el consumo global de potencia por bit sólo un 10% por encima.

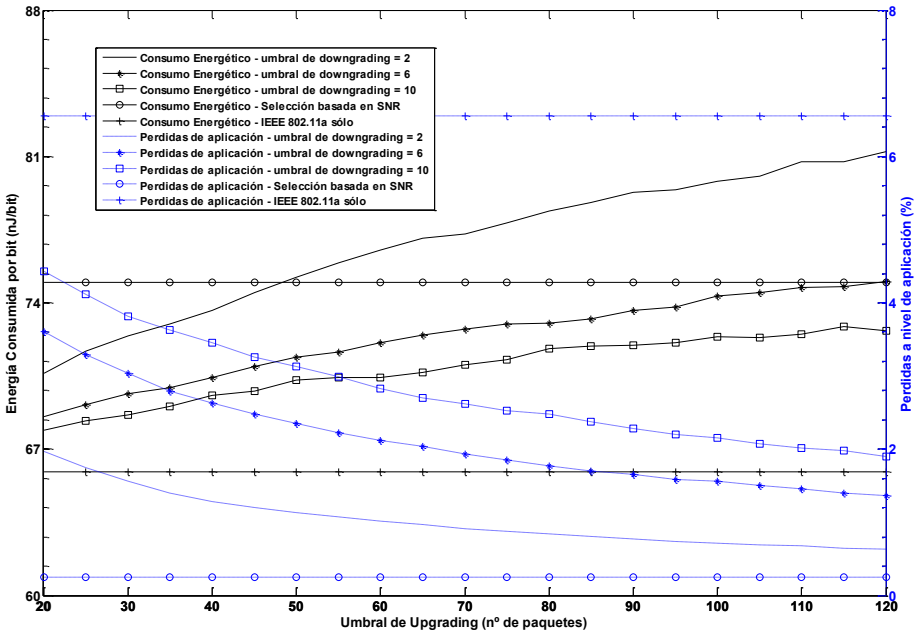


Fig. 5. Eficiencia energética y perdida de paquetes usando la UCL ó sólo el interfaz 802.11a

Empleando la SNR como métrica, la tasa de perdida se reduce drásticamente (alrededor de 20 veces menos), manteniendo el consumo de potencia sólo un 15% por encima del caso en que se usa 802.11a todo el tiempo.

Realizando estudios similares tomando el ancho de banda como referencia, el uso de la UCL reporta mejoras de al menos 2.5 veces más ancho de banda que empleando únicamente 802.11b pero reduciendo el consumo global por bit en un 40%. En la Figura 6 se representa esta situación habiéndose fijado los umbrales en 6 paquetes para el de *downgrading* y 64 para el de *upgrading*. Esto significa que se está reduciendo a la mitad la potencia consumida durante la simulación. Para la aproximación basada en SNR, también se duplica el ancho de banda obtenido manteniendo una reducción del consumo de potencia de un 40% frente al uso único de 802.11b.

Se puede concluir por tanto, que el uso inteligente de las interfaces disponibles conlleva un consumo de potencia no óptimo pero que por el contrario no compromete el rendimiento global del sistema, como sería el caso del uso exclusivo de 802.11b (disminución del ancho de banda) o de 802.11a (elevada pérdida de paquetes).

Los ensayos se han realizado empleando exclusivamente tráfico UDP, pues los mecanismos de control de congestión de TCP habrían detenido la transmisión cuando las condiciones del canal fueran malas, haciendo inviable la comparación de la eficiencia energética de las diferentes alternativas mostradas. En ese caso, la UCL mantendría el rendimiento en unos niveles razonables haciendo uso de la tecnología inalámbrica más robusta en el canal con malas condiciones [8]. Por el contrario usando solo IEEE 802.11a, la transmisión se interrumpirá tan pronto se aprecie una degradación del canal [9]. Así pues, en términos de eficiencia energética esta última opción reportaría mejores resultados, pero disminuyendo notablemente el ancho de banda y la calidad del servicio ofrecido.

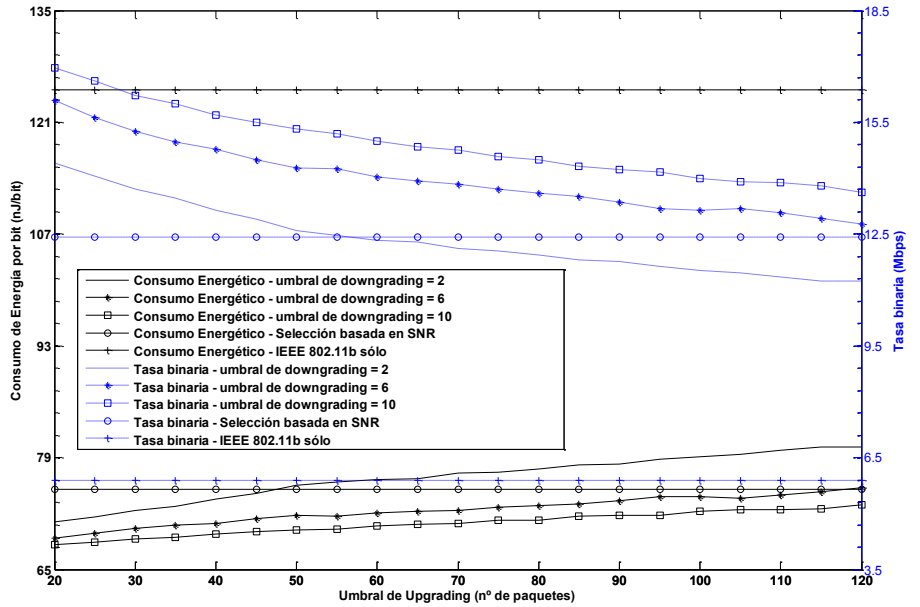


Fig. 6. Tasa binaria y eficiencia energética usando la UCL ó sólo 802.11b

V. CONCLUSIONES

En este artículo se ha demostrado que la eficiencia energética de las redes inalámbricas ad-hoc puede mejorarse considerando las posibilidades multi-radio de los dispositivos. Al contrario que otras metodologías, como los mecanismos de adaptación de la velocidad binaria [10] [11], que se centran en una sola tecnología, la solución analizada en este artículo permite la provisión de servicios en dispositivos móviles multi-radio de forma transparente. De esta forma, permitimos trabajar en circunstancias de heterogeneidad, permitiendo una mayor versatilidad en las comunicaciones, pues, en función de la aplicación y los requerimientos del usuario, se pueden explotar las ventajas específicas de todas las tecnologías WPAN y WLAN disponibles. A pesar de que para emplear la interfaz más adecuada se deba forzar un traspaso vertical, la sesión no se verá afectada. Por ello, es importante destacar que si bien la validación se ha hecho utilizando dos variantes del estándar IEEE 802.11, la flexibilidad de la UCL permitiría integrar en el proceso de decisión otras tecnologías más diversas.

La solución propuesta ha demostrado ser capaz de ofrecer un alto grado de adaptación reportando buenos resultados. En este sentido no se ha buscado la identificación de los parámetros de configuración óptimos, sino evaluar las mejoras que las soluciones de optimización propuestas pueden ofrecer. Este aspecto es relevante ya que permite adaptar el comportamiento según el servicio, las necesidades y requerimientos del usuario y las condiciones del canal.

Resulta especialmente destacable que la visión del sistema que se obtiene desde la capa de convergencia, teniendo acceso a información proveniente múltiples capas y diferentes tecnologías de acceso, hacen posible inferir mejoras globales a éste. Así, considerando los requerimientos de usuario, la UCL puede tomar las decisiones más apropiadas buscando adaptarse a las necesidades del usuario final y optimizando su experiencia de uso, no solo gracias a un mayor ancho de banda sino también mejorando la eficiencia energética y/o disminuyendo la perdida de paquetes debida a las deficiencias del canal de comunicación. En este sentido, las estrategias de selección implementadas en la UCL se basarán en sistemas de decisión multi-paramétricos que incluirán desde el nivel de batería, hasta el tipo de aplicación o las preferencias de usuario.

Los resultados experimentales, realizados tomando la SNR como métrica de decisión, sustentan las conclusiones de este trabajo.

AGRADECIMIENTOS

Los autores desean expresar su agradecimiento a todos los socios del proyecto MAGNET Beyond (IST-027396), y especialmente a los del WP2 por su colaboración.

Asimismo, los autores desean expresar su agradecimiento al Ministerio de Ciencia e Innovación, por su financiación en el proyecto "Comunicaciones Cognitivas, Cooperativas y Gestión Autónoma de Servicios", C3SEM (TEC2009-14598-C02-01)

REFERENCIAS

[1] Zorzi, M., Rao, R. R., "Energy constrained error control for wireless channels", IEEE Personal Communications Magazine, vol. 4, nº 6, pp. 27–33, Diciembre 1997.

[2] Lettieri, P., Fragouli, C., Srivastava, M.B., "Low power error control for wireless links", Proceedings from the 3rd annual ACM/IEEE international conference on Mobile computing and networking, pp.139–150, Septiembre 1997.

[3] Woo, M., Singh, S., Raghavendra, C.S., "Power Aware routing in mobile ad hoc networks", Proceedings from the 4th annual ACM/IEEE international conference on Mobile computing and networking, pp.181–190, Octubre 1998.

[4] Chang, J-H., Tassiulas, L., "Energy conserving routing in wireless ad-hoc networks", Proceedings from the 19th Annual Joint Conference of the IEEE Computer and Communications Societies, pp. 22–31, Marzo 2000.

[5] Ramanathan, R., Rosales-Hain, R., "Topology Control of Multihop Wireless Networks using Transmit Power Adjustment", Proceedings from the 19th Annual Joint Conference of the IEEE Computer and Communications Societies, pp. 404–413, Marzo 2000.

[6] Arauz, J. and Krishnamurthy, P.: "Markov modeling of 802.11 channels", Proceedings from the 58th IEEE Vehicular Technology Conference, Octubre 2003.

[7] Texas Instruments White Paper, "Low Power Advantage of 802.11a/g vs. 802.11b", Diciembre 2003.

[8] Sanchez, L., Lanza, J. and Muñoz, L.: "Experimental Assessment of a Cross-Layer Solution for TCP/IP Traffic Optimization on Heterogeneous Personal Networking Environments", Lecture Notes in Computer Science (Vol 4217), pp. 284-296, Septiembre 2006.

[9] Garcia, M., Agüero, R. and Muñoz, L.: "On the unsuitability of TCP RTO estimation over bursty error channels", Lecture Notes in Computer Science (Vol. 3260), pp. 343–348, Septiembre 2004.

[10] Lacage, M., Manshej, M. H. and Turletti, T.: "IEEE 802.11 Rate Adaptation: A Practical Approach", In Proc. MSWiM 2004, pp. 126–134, Venice, Junio 2004.

[11] Ji-Hoon Yun, "Throughput analysis of IEEE 802.11 WLANs with Automatic Rate Fallback in a lossy channel", IEEE Transactions on Wireless Communications 8 (2), pp. 689-693, Febrero 2009.

[12] L. Muñoz, M. García, R. Agüero, "Bear: a bursty error auto-regressive model for indoor and mobile radio communications", 18th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Atenas (Grecia). Septiembre 2007.

Sesión 5.A
Aplicaciones distribuidas y P2P

Estudio exploratorio de la capacidad de discriminación de tráfico P2P usando reglas de similitud entre flujos

José Camacho, Pablo Padilla, F. Javier Salcedo-Campos, Pedro García-Teodoro, Jesús Díaz-Verdejo
Teoría de la Señal, Telemática y Comunicaciones,
Universidad de Granada,
C/ Periodista Daniel Saucedo Aranda s/n 18071 GRANADA (Spain).
josecamacho@ugr.es, pablopadilla@ugr.es, fjsalc@ugr.es, pgteodor@ugr.es, jedv@ugr.es

***Resumen*—Existe un claro interés en la clasificación de tráfico en red sin acceder a la información contenida en el *payload* de los paquetes. En particular, resulta especialmente relevante la identificación del tráfico *peer-to-peer* (*P2P*) circulante en una red. El presente artículo evalúa la aplicabilidad de reglas de similitud entre flujos de datos para la clasificación de tráfico, con especial énfasis en la distinción entre el tráfico *P2P* del que no lo es. En concreto, el trabajo se centra en evaluar los parámetros que permiten crear parejas de flujos asociados a un mismo protocolo. Este trabajo es un paso previo necesario para identificar relaciones entre flujos de cara a la clasificación de tráfico.**

***Palabras Clave*—Clasificación de tráfico, peer-to-peer, K-Nearest Neighbors**

I. Introducción

La creciente popularidad y expansión de las redes y aplicaciones *peer-to-peer* (P2P) han dado paso paralelamente a la aparición de cuestiones relacionadas con la ingeniería del tráfico y la seguridad. Por un lado, los proveedores de Internet se ven perjudicados por el uso intensivo de los recursos de red que implican las actividades P2P, debiendo manejar este gran volumen de tráfico con el mínimo impacto posible para el resto de los servicios de red. Por otra parte, la capacidad de comunicación e intercambio de cualquier tipo de información entre los llamados pares (*peers* en inglés), la mayoría de ellos anónimos, representa un riesgo de seguridad. Este riesgo afecta en primer lugar a los usuarios, puesto que la información que se intercambia podría contener virus, gusanos y *malware* en general. Existe, además, un riesgo desde el punto de vista de la infraestructura de las redes, pues las aplicaciones P2P pueden ser utilizadas coordinadamente para apoyar otras actividades perjudiciales como ataques DoS, *botnets*, etc.

En este contexto, queda patente la necesidad de diferenciar el tráfico P2P de cualquier otro tipo de tráfico. Este problema de identificación del tráfico P2P forma parte de uno más general relativo a la identificación del tráfico de red [1].

Existen tres problemas principales que se plantean en la identificación del tráfico en una red:

1) Parametrización del tráfico: Son numerosas las características y agrupaciones de éstas que se han propuesto en la literatura para representar y clasificar el tráfico de red. De este modo, la información utilizada comprende desde datos estadísticos de las conexiones a partir de informes de *routers* SNMP [2] (baja granularidad) hasta datos extraídos de cabeceras TCP, incluyendo los bits de señalización y los primeros *bytes* del *payload* (alta granularidad) [3].
2) Nivel de identificación: Una vez que el tráfico ha sido parametrizado, en la literatura se consideran tres niveles para llevar a cabo la identificación [4], [1]: identificación basada en nodo, basada en flujo y basada en paquetes. En el primer caso, el objetivo es detectar los nodos que generan un determinado tipo de tráfico [5]. En el caso basado en flujo, el objetivo es clasificar a cada flujo por el protocolo de nivel de aplicación que lo produce. Por último, en el basado en paquetes el objetivo es clasificar cada paquete individualmente.
3) Proceso de identificación: Por último, los sistemas utilizados para llevar a cabo la identificación en sí cubren una amplia variedad de técnicas. Desde heurísticas o mediante firmas [6], [1], [7] hasta minería de datos o algoritmos de reconocimiento de patrones [8], [4].

Concretando en la identificación de flujos P2P, de entre las numerosas técnicas que se han utilizado en esa tarea, destaca la clasificación mediante la distancia con los vecinos más próximos o KNN (del inglés K-Nearest Neighbors) gracias a su simplicidad y la alta tasa de reconocimiento que consigue. En este sentido destacan los trabajos de Jun [9] y Lim [10]. En el primer caso se realiza una comparación entre diferentes técnicas, como Naïve Bayes, árboles de decisión y otros métodos, incluyendo los KNN, para clasificar 12 tipos de protocolos de aplicación diferentes, entre los que se encuentran protocolos P2P (Bittorrent y Gnutella) y no P2P (HTTP, DNS, POP3, etc.). Los resultados muestran que los KNN son la mejor técnica en términos de tasa de precisión. En el segundo trabajo [10] se propone una discretización de los parámetros estándar que se extraen de los flujos (puertos, tamaños de los paquetes, número de paquetes, duración del flujo, etc.), y se evalúan 4 técnicas de clasificación: *support vector machines* (SVM), KNN, Naïve Bayes y árboles de decisión. Los resultados indican que KNN obtiene resultados similares a la mejor técnica cuando se aplica el método de discretización propuesto, con sólo alrededor de un 2% de precisión inferior a SVM, que consigue una tasa de precisión del 98%.

El presente artículo realiza, en una primera parte, un estudio exploratorio para confirmar el buen rendimiento de KNN en la

identificación de flujos P2P e investigar las causas que llevan a dicho resultado. Los resultados obtenidos sugieren la propuesta de un nuevo nivel de identificación en la clasificación de tráfico que no ha sido considerado previamente en la literatura: la clasificación basada en relaciones entre flujos. La segunda parte del trabajo se centra en el estudio de la viabilidad de la identificación de flujos pertenecientes a un mismo protocolo, siendo éste un paso previo necesario a la propuesta de un clasificador concreto.

El artículo se organiza de la siguiente manera. En la Sección II se introducen los conjuntos de datos utilizados. En la Sección III, se motiva la propuesta utilizada en el análisis subsiguiente a partir de la exploración de los datos. En la Sección IV se propone un esquema paramétrico de detección de vecinos para su implementación en línea y se identifican sus parámetros. La Sección V realiza la validación del modelo con datos independientes. Finalmente, en la Sección VI se presentan las conclusiones del trabajo.

II. Conjuntos de datos de experimentación

A. Adquisición de las bases de datos

La evaluación de métodos de identificación de tráfico requieren la disponibilidad de una base de datos con ejemplos correctamente clasificados. Esta base de datos que se utiliza como referencia para determinar la exactitud de los resultados obtenidos se denomina ”ground truth”, y debe contener suficientes datos para que sea representativa. Sin embargo, la obtención de una base de datos lo suficientemente grande con tráfico real, y además correctamente etiquetado, no es una tarea fácil, pues realizar el etiquetado manualmente no es asumible. Por lo tanto, para evaluar el sistema propuesto se ha desarrollado un dispositivo experimental construido a partir de dos componentes principales: una base de datos de tráfico real capturado en una red académica, y una herramienta para clasificar automáticamente los paquetes y flujos en función de sus cargas útiles mediante la inspección de los paquetes (Deep Packet Inspection o DPI). De esta manera, la base de datos de referencia o ”ground truth” se construye mediante el análisis y la identificación de cada flujo y cada paquete suponiendo que la herramienta escogida, en nuestro caso openDPI [11], es la mejor actualmente disponible para este propósito, y que el número de errores de clasificación en los que incurre es insignificante. Sin embargo, el número de paquetes y flujos que openDPI no es capaz de clasificar es su principal limitación.

OpenDPI es una versión de domino público derivada de un producto comercial llamado PACE de Ipoque. El núcleo de openDPI es una librería software diseñada para clasificar tráfico de Internet en función de los protocolos de aplicación. En [12] los autores explican que la clasificación de protocolos de aplicación basada en DPI se consigue mediante la combinación de una serie de técnicas diferentes:

- Búsqueda de patrones, mediante el análisis de cadenas y patrones de *bytes* en cualquier parte del paquete, incluyendo el *payload*. De esta manera, openDPI busca firmas de protocolos conocidos.
- Análisis de comportamiento, mediante la búsqueda de patrones de comportamiento conocidos de una aplicación en el tráfico observado. Los datos utilizados incluyen el tamaño absoluto y relativo de los paquetes por flujo, la tasa de paquetes, y el número de flujos y la tasa de nuevos flujos por aplicación.
- El análisis estadístico, calculando algunos indicadores que pueden utilizarse para identificar los tipos de transmisión, como la media, la mediana y la variación de los valores utilizados en el análisis del comportamiento y la entropía de un flujo.

Por lo tanto, openDPI no es un producto DPI puro, pues no sólo está basado en la detección de firmas de protocolos, sino que también incorpora información de otras fuentes. De esta manera, la precisión de la clasificación es mayor, aunque no es capaz de identificar algunos paquetes y flujos. Esto, junto, con la disponibilidad y la calidad de las firmas, hizo que openDPI fuera seleccionado para la generación del ”ground truth” en este trabajo. A partir de las librerías de openDPI, se ha diseñado una herramienta capaz de identificar los protocolos de aplicación y de seguir y diferenciar los paquetes de cada flujo.

B. Descripción de las bases de datos

La base de datos de tráfico obtenida contiene datos capturados durante 3 días hábiles en una red universitaria. La adquisición de datos se llevó a cabo en el *router* de acceso con el fin de poder controlar todo el tráfico de entrada y de salida de todos los nodos. Por tanto, los flujos son capturados completamente en ambos sentidos de la comunicación.

Para el presente trabajo, se han considerado dos subconjuntos de datos denominados F14 y F51, que contienen tráfico de diferentes nodos, para probar y validar el método, respectivamente. Las Tablas I y II muestran información sobre ambos conjuntos de datos. Los resultados proporcionados por la herramienta de openDPI para la base de datos considerada detectadon un total de 31 protocolos diferentes, de los cuales sólo 19 aparecen en el subconjunto F14 y 27 en el F51. Los resultados muestran que HTTP es el protocolo con mayor número de flujos, mientras que la proporción relativa de los protocolos P2P es del 12.5% de los flujos. Si bien este porcentaje en puede parecer reducido, el volumen de tráfico asociado es alto debido al tamaño de cada flujo P2P. Un análisis más detallado de los datos muestra que sólo un escaso número de nodos generan tráfico P2P, siendo importantes los protocolos de *videostreaming*, que contribuyen al tráfico HTTP (por ejemplo, el tráfico de Youtube). El resto de los flujos no P2P incluye sobre todo protocolos habituales, tales como DNS, SSL y protocolos de correo. La mayor parte de los flujos P2P pertenecen a BitTorrent, mientras que Gnutella y otros están presentes en menor proporción. Esta proporción de flujos puede ser consecuencia de las características específicas de los protocolos. La relación entre el tráfico P2P y *no-P2P* es similar entre ambos grupos (ver Tabla I). El conjunto F14 se utilizará para el análisis exploratorio y el ajuste del sistema, y el F51 para validar los resultados.

C. Extracción de parámetros de los flujos

La salida proporcionada por la herramienta desarrollada consiste en 3 listas: una con los flujos encontrados y su clasificación, otra con los paquetes y su clasificación y una tercera que relaciona los flujos y los paquetes de cada flujo. A partir de esta información, se realiza el proceso de parametrización,

Tabla I
ESTADÍSTICA BÁSICA DEL TRÁFICO EMPLEADO EN LOS EXPERIMENTOS.

	Flujos				
Conjunto	Total	Etiquetado	Flujos P2P	Flujos no P2P	Desconocidos
F14	135202	67015	16005	51010	68187
F51	193409	45167	9878	35289	148242
Total	328611	112182	25883	86299	216429

Tabla II
DISTRIBUCIÓN DE FLUJOS EN PROTOCOLOS PARA EL CONJUNTO DE DATOS DE CALIBRACIÓN (F14) Y EL CONJUNTO DE DATOS DE EVALUACIÓN (F51). CADA FLUJO PERTENECE ÚNICAMENTE A UN PROTOCOLO DE ACUERDO A LA CLASIFICACIÓN REALIZADA POR OPENDPI.

protocolo	Nº de flujos en F14	Nº de flujos en F51
DNS	27178	401
ICMP	980	2520
BitTorrent	15723	9362
HTTP	20057	29613
FTP	61	6
SSL	816	1960
HTTP+Flash	162	409
Mail_POP	148	3
Mail_SMTP	1512	0
MSN	35	112
SSL+MSN	3	6
MySQL	19	0
SSH	5	0
NETBIOS	6	39
NTP	1	7
HTTP+RealMedia	2	0
HTTP+MPEG	23	29
Gnutella	282	475
MPEG	2	16
DirectDownloadLink	0	32
Yahoo	0	11
SIP	0	8
iMESH	0	9
Flash	0	35
HTTP+Quicktime	0	38
STUN	0	25
WindowsMedia	0	6
WindowsMedia+MPEG	0	2
IRC	0	9
SMB	0	29
Oscar	0	5
TOTAL	67015	45167

con el que se obtiene un vector de características con 62 componentes para cada flujo, como se muestra en la Tabla III. Los vectores contienen toda la la información necesaria para su tratamiento posterior, incluyendo una etiqueta de identificación del flujo (FLOW_ID), el protocolo que ha detectado openDPI e información básica sobre el flujo (tupla de flujo). Las direcciones IP de cada flujo se han ordenado considerándolas como enteros (en representación de red) y, por tanto, los dos sentidos que pueden tener los paquetes se tienen en cuenta en la parametrización: UP (ascendente) indica que los paquetes van de la IP baja hacia la IP alta, y DOWN (descendente) para el sentido opuesto. Esta ordenación responde a motivos de eficiencia en el procesamiento posterior.

Los valores que se han considerado en cada vector de parámetros son medidas estadísticas básicas relacionadas con las propiedades del flujo, la mayoría de ellas separadas en medidas totales, ascendentes (_UP) y descendientes (_DOWN). Los parámetros empleados son los que habitualmente se incluyen en la literatura: tamaño medio de los paquetes, medidas de tiempo y duración de los flujos, número de paquetes, etc. No obstante, se ha incluido una descripción más detallada a nivel temporal y de señalización (por ejemplo, los tiempos entre llegadas y el número de paquetes URG).

En el resto del artículo se utilizará el término ”observación” para referirse al vector de características asociado a un flujo y el término ”etiqueta” para referirse a la clase a la que pertenece. Se considerarán dos tipos de clasificación. En primer lugar, se evaluará la clasificación de los flujos atendiendo al protocolo involucrado. En segundo lugar, se agruparán los distintos flujos en dos clases: tráfico P2P y el resto de tráfico de red. El conjunto de datos F14 se utilizará para el análisis exploratorio en la Sección III y la calibración del modelo paramétrico en la Sección IV. El conjunto de datos F51 se utilizará para la validación de ese modelo, en la Sección V.

III. MOTIVACIÓN

En esta sección se evaluará el rendimiento de la técnica KNN en la clasificación de los flujos en el conjunto de datos F14, tomando como referencia los resultados de [9] y [10]. En segundo lugar, se realizará un análisis exploratorio para interpretar los resultados observados con KNN.

A. Clasificación de tráfico con los K vecinos más cercanos

La técnica KNN, aplicada a la clasificación, usualmente identifica la etiqueta de una observación a partir de la moda en las etiquetas de los K vecinos más cercanos. Para establecer qué observaciones se corresponden con los vecinos más cercanos a una observación dada, es necesario definir una función de cercanía entre observaciones basada típicamente en la noción de distancia. Tomando como referencia experiencias previas, se utilizará la distancia *Manhattan* o *cityblock*, que mide la distancia entre dos puntos como la suma de las diferencias absolutas entre sus coordenadas. Así, la distancia entre los puntos $\mathbf{x}$ e $\mathbf{y}$ se corresponde con la norma $\mathbf{L}_1$ de su diferencia:

$$d_1(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_1 \quad (1)$$

Para obtener una estimación del rendimiento de KNN en la clasificación, se utilizará una técnica de remuestreo basada en la repetición del siguiente esquema:

Tabla III
COMPONENTES DE LOS VECTORES DE PARÁMETROS DE CADA FLUJO.

Valor	Descripción
Identificación de flujos	
N_PROT	Numero de protocolos detectado
IP_LOW	Dirección IP menor el la tupla de la sesión
IP_UPPER	Dirección IP mayor el la tupla de la sesión
PORT1	Puerto asociado a la menor IP (IP_LOW)
PORT2	Puerto asociado a la mayor IP (IP_UPPER)
PROT_UDP	Protocolo de transporte UDP
PROT_TCP	Protocolo de transporte TCP
PROT_UNK	ICMP
DIR	Dirección del primer paquete observado
FIRST_TIME	Marca de tiempo del primer paquete (μs)
LAST_TIME	Marca de tiempo del último paquete (μs)
Relacionados con la transferencia	
NPACKETS	Número de paquetes en el flujo
NPACKETS_UP	Idem dirección hacia arriba
NPACKETS_DOWN	Idem dirección hacia abajo
PACKETS_SIZE	Tamaño total de los paquetes intercambiados
PACKETS_SIZE_UP	Idem dirección hacia arriba
PACKETS_SIZE_DOWN	Idem dirección hacia abajo
PAYLOAD_SIZE	Tamaño total de los payloads
PAYLOAD_SIZE_UP	Idem dirección hacia arriba
PAYLOAD_SIZE_DOWN	Idem dirección hacia abajo
MEAN_PACK_SIZE	Tamaño medio de los paquetes
MEAN_PACK_SIZE_UP	Idem dirección hacia arriba
MEAN_PACK_SIZE_DOWN	Idem dirección hacia abajo
SHORT_PACKETS	Número de paquetes cortos
SHORT_PACKETS_UP	Idem dirección hacia arriba
SHORT_PACKETS_DOWN	Idem dirección hacia abajo
LONG_PACKETS	Número de paquetes largos
LONG_PACKETS_UP	Idem dirección hacia arriba
LONG_PACKETS_DOWN	Idem dirección hacia abajo
MAXLEN	Tamaño máximo de los paquetes
MAXLEN_UP	Idem dirección hacia arriba
MAXLEN_DOWN	Idem dirección hacia abajo
MINLEN	Tamaño mínimo de los paquetes
MINLEN_UP	Idem dirección hacia arriba
MINLEN_DOWN	Idem dirección hacia abajo
Relacionados con el tiempo	
DURATION	Duración del flujo (μs)
MEAN_INTERAR	Tiempo medio entre paquetes consecutivos
MEAN_INTERAR_UP	Idem sólo para paquetes hacia arriba
MEAN_INTERAR_DOWN	Idem sólo para paquetes hacia abajo
MAX_INTERAR	Tiempo máximo entre paquetes consecutivos
MAX_INTERAR_UP	Idem sólo para paquetes hacia arriba
MAX_INTERAR_DOWN	Idem sólo para paquetes hacia abajo
MIN_INTERAR	Tiempo mínimo entre paquetes consecutivos
MIN_INTERAR_UP	Idem sólo para paquetes hacia arriba
MIN_INTERAR_DOWN	Idem sólo para paquetes hacia abajo
Señalización	
N_SIGNALING	Número de paquetes con flags
N_SIGNALING_UP	Idem dirección hacia arriba
N_SIGNALING_DOWN	Idem dirección hacia abajo
NACKS	Número de paquetes con ACK activo
NFIN	Idem FIN
NSYN	Idem SYN
NRST	Idem RST
NPUSH	Idem PSH
NURG	Idem URG
NECE	Idem ECE
NCWD	Idem CWD
NACK_UP	Número de paquetes hacia arriba con ACK activo
NACK_DOWN	Idem dirección hacia abajo
NFIN_UP	Idem FIN & UP
NFIN_DOWN	Idem FIN & DOWN
NRST_UP	Idem RST & UP
NRST_DOWN	Idem RST & DOWN

i. Dividir aleatoriamente el conjunto de datos en observaciones de test y de calibración.
ii. Utilizar el algoritmo KNN con la distancia *cityblock* para estimar el protocolo asociado a las observaciones de test.
iii. Calcular el porcentaje de aciertos.

La repetición de este esquema de remuestreo permite estimar con cierta precisión la media y la desviación típica asociadas al número de aciertos en la clasificación basada en KNN. Los parámetros específicos del análisis aparecen detallados en la Tabla IV. El procedimiento de remuestreo anterior ha sido repetido para la clasificación de los flujos de acuerdo a su protocolo (considerando por tanto 19 clases, ver Tabla II), así como para la discriminación entre tráfico P2P y el resto (considerando únicamente dos clases). Tras el análisis realizado, se obtiene una media de porcentaje de aciertos en la clasificación de protocolos igual al 83.7% y una desviación típica del 3.5%. En cuanto a la discriminación de tráfico P2P, se obtiene una media del 90.8% y una desviación típica del 2.7%. Nótese que el presente análisis únicamente tiene por objeto evaluar el rendimiento de KNN en la clasificación del conjunto de datos considerado, y no identificar los parámetros que permiten una clasificación óptima.

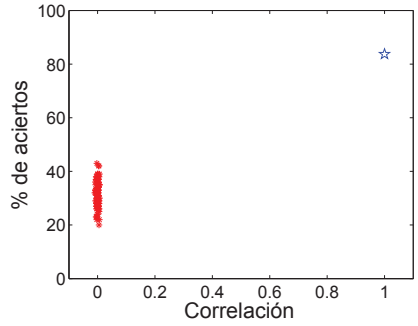
Tabla IV
PARÁMETROS DEL ANÁLISIS DEL RENDIMIENTO DE KNN EN LA CLASIFICACIÓN ENTRE TRÁFICO P2P Y EL RESTO DE TRÁFICO.

Parámetro	Valor
Número de repeticiones	100
Tamaño del conjunto de test	100
Distancia	*cityblock*
Número de vecinos	1
Características utilizadas	Todas

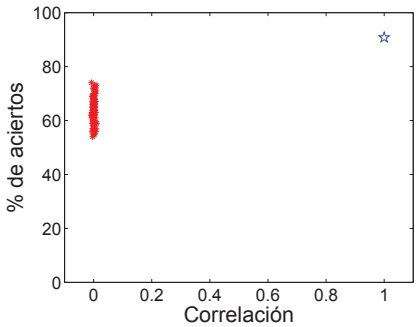
Aunque el porcentaje de aciertos obtenido con KNN es elevado, es coveniente comprobar que no se debe a elementos casuales y específicos del conjunto de datos considerado. Nótese que el porcentaje de aciertos puede ser muy elevado cuando el número de observaciones de cada clase no está balanceado, sin necesidad de que exista una capacidad clasificatoria real. A modo de ejemplo, en el caso límite, cuando sólo hay observaciones de una clase, cualquier clasificador tipo KNN ofrece un resultado del 100%. Para la comprobación, se pueden utilizar de nuevo técnicas de remuestreo, en este caso con cambio de etiquetas. Estas técnicas son conocidas en la literatura como tests de permutación (*permutation tests*) o de aleatorización (*randomization tests*) [13], [14]. El procedimiento seguido es repetir el análisis con el mismo conjunto de datos pero donde las etiquetas han sido reordenadas aleatoriamente. De esta manera, si se produce un elevado número de aciertos, éste puede considerarse consecuencia de elementos casuales, como el porcentaje de observaciones en cada clase. El resultado de KNN con los datos permutados ofrece una muestra de referencia para un test de hipótesis. El test se realiza con la hipótesis nula de que el resultado de KNN sobre los datos originales pertenece a la misma población que la muestra de referencia y, por tanto, se debe a elementos casuales. La hipótesis alternativa es que el resultado es debido a una capacidad real de clasificación del clasificador KNN.

En la Figura 1 se muestra el diagrama de dispersión asociado a los tests de permutación, al estilo propuesto en [13]. En la Figura 1(a) se presenta el gráfico correspondiente a la clasificación de protocolos y en la Figura 1(b) el correspondiente a la discriminación de tráfico P2P. En las figuras, cada uno de los puntos de la izquierda se corresponde con el resultado obtenido en una permutación. Para ello, una vez que las etiquetas han sido reordenadas aleatoriamente, se repite el proceso de tres pasos especificado anteriormente para computar el número de aciertos. En las abcisas de las figuras se muestra la correlación entre el vector de etiquetas

original y el aleatorizado. Esta correlación permite evaluar hasta qué punto la aleatorización de las etiquetas ha producido una ordenación de las etiquetas muy similar a la original. Teniendo en cuenta el elevado número de observaciones, algo mayor de 67.000, es razonable obtener valores de correlación muy bajos, lo que implica que las asignaciones aleatorias de etiquetas no se corresponden con la realidad en la mayoría de los casos. Esto es ideal desde el punto de vista de obtener una muestra de referencia que dependa únicamente de elementos casuales. En ordenadas se presenta el porcentaje de aciertos obtenidos. Los resultados conseguidos en las permutaciones se comparan con el original, situado en el punto de correlación 1. Ambas gráficas muestran que el resultado obtenido por KNN es estadísticamente significativo (p-valor $< 10^{-8}$, asumiendo una distribución normal de las permutaciones).



(a)



(b)

Fig. 1. Diagrama de dispersión asociado a los test de permutación: (a) clasificación de protocolos y (b) discriminación de tráfico P2P.

B. Exploración de los datos

Una vez comprobado el buen rendimiento de KNN en ambas clasificaciones, es conveniente establecer las causas del mismo. Dado el elevado número de características asociadas a los flujos, una técnica apropiada de exploración es la denominada mínimos cuadrados parciales (*Partial Least Squares* o *PLS*) en su versión discriminante (*PLS-Discriminant Analysis* o *PLS-DA*) [15], [16], [17]. PLS permite identificar unas pocas características, denominadas *variables latentes*, que maximizan la correlación entre las características originales y un número de variables respuesta. En PLS-DA, las variables respuestas son variables indicadoras (*dummy*), tantas como el número de clases, que toman valor igual a 1 para una de las clases y -1 para el resto. En el caso de este estudio, las características originales son las 62 características registradas para los flujos (Tabla III) y las variables respuesta son variables indicadoras definidas según la pertenencia de los flujos a las distintas clases. Si entendemos las observaciones como puntos en un espacio M-dimensional, donde M es el número de características originales, PLS-DA permite identificar el subespacio de mayor correlación con la discriminación entre clases. Las variables latentes se identifican por orden de correlación, de forma que la primera es la que presenta la mayor correlación y así sucesivamente.

En la Figura 2 se muestra el gráfico de dispersión de las observaciones, denominado gráfico de valores (*scores plot*), en el subespacio correspondiente a las variables latentes 3 (LV3) y 4 (LV4). Por simplicidad, se considera únicamente la discriminación entre tráfico P2P y *no-P2P*. Las primeras dos variables latentes no se muestran debido a que su mayor nivel de correlación viene más determinado por la variabilidad asociada a las características originales que por la asociada a la variable respuesta. No son, por tanto, variables latentes que permitan discriminar entre las clases. Las variables latentes 3 y 4 son las de mayor capacidad discriminativa, siendo ésta baja en cualquier caso. Como se puede observar en la figura, las observaciones correspondientes a tráfico P2P aparecen en zonas donde se sitúan también observaciones *no-P2P*. Esto indica que no es posible determinar reglas (lineales) que permitan, a partir del valor de las características, distinguir tráfico P2P. Este resultado lleva a la hipótesis de que la buena clasificación obtenida con la asociación entre vecinos no esté determinada por elementos absolutos (ej. determinados valores específicos de las características), sino relativos entre vecinos (similitud en características). Esta hipótesis no es trivial, dado que un escenario donde la clasificación fuera motivada por elementos absolutos, por ejemplo donde las observaciones correspondientes a cada clase aparecieran agrupadas, también llevaría a un buen rendimiento de KNN.

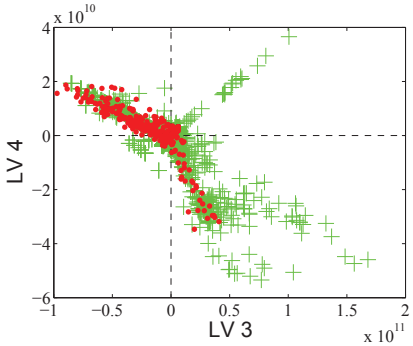


Fig. 2. Gráfico de dispersión de las observaciones en el subespacio PLS-DA correspondiente a las variables latentes 2 y 3. Discriminación entre tráfico P2P (puntos) y *no-P2P* (cruces).

Por otro lado, teniendo en cuenta el uso de la distancia *cityblock* en la técnica KNN, es de interés revisar la escala de las distintas variables involucradas en el análisis. Esto es así debido a que la distancia *cityblock* se ve muy afectada por las diferencias en variabilidad entre las características. La Figura 3 presenta la desviación típica muestral del vector de características. Podemos observar que unas pocas características tienen una desviación típica muy superior al resto y, por tanto, determinan prácticamente el valor de la distancia *cityblock*. Éstas son variables asociadas a la diferencia

temporal entre el comienzo y el final de los flujos vecinos, su duración y el tiempo entre paquetes. Cabe resaltar que la clasificación con KNN únicamente basada en esas variables obtiene un porcentaje de aciertos similar al del conjunto completo de variables. Este resultado parece indicar que los flujos pertenecientes a una misma clase aparecen próximos entre sí en el tiempo y presentan características dinámicas similares. Sin embargo, y en la línea del análisis con PLS-DA, se constata la inexistencia de discriminación lineal entre las clases en ninguno de los subespacios asociados a las parejas de características de alta variabilidad. Por tanto, no podremos discriminar las clases a partir de, a modo de ejemplo, un determinado valor de duración o de tiempo entre paquetes.

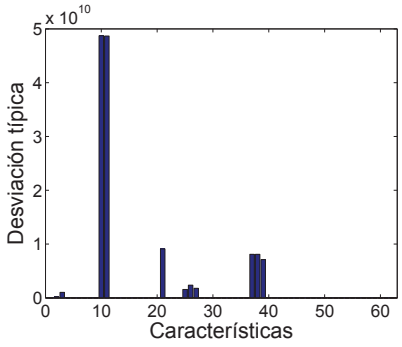


Fig. 3. Desviación típica muestral de las características.

Otro análisis interesante para comprender el buen rendimiento de KNN en la clasificación es la observación directa de algunas de las características de los flujos y sus vecinos asociados. A modo ilustrativo, en la Tabla V se muestran algunos parámetros correspondientes a flujos emparejados como vecinos de acuerdo a la distancia *cityblock*. A partir de esta muestra, podemos introducir un número de hipótesis que serán de interés en el análisis subsecuente. En primer lugar, en varios casos coinciden las direcciones IPs en los flujos vecinos, así como el puerto destino. Esto podría ser una característica generalizable en algunos protocolos. Por ejemplo, en el caso de la navegación web, un cliente realiza numerosas conexiones TCP al mismo servidor para la apertura de distintas páginas alojadas en éste. Así, el cliente elige puertos dinámicos cercanos, si no consecutivos, al hacer esta navegación, como se muestra en varias de las parejas de flujos HTTP. También asociado a la navegación web podemos ver un comportamiento parecido en el protocolo DNS. Durante la navegación, es común acceder a páginas en distintos servidores web. Para acceder a estos servidores será necesario realizar la pertinente conversión DNS. En cuanto a los flujos P2P, parece lógico esperar la coincidencia únicamente de la dirección IP origen, que inicia la conexión con distintos pares prácticamente en paralelo. Cabe resaltar que en aquellas parejas de flujos muestreadas donde la clasificación basada en el vecino más cercano falla, no hay coincidencia de IPs o puertos.

Tabla V
PARÁMETROS CORRESPONDIENTES A UN SUBCONJUNTO DE FLUJOS EMPAREJADOS COMO VECINOS DE ACUERDO A LA DISTANCIA *cityblock*. EN CASO DE COINCIDENCIA EN EL VALOR DE PUERTOS O SERVICIO ENTRE AMBOS FLUJOS EMPAREJADOS, EL VALOR SE MUESTRA UNA ÚNICA VEZ.

IPs coin.	Pto origen	Pto destino	Servicio	Dist. en tpo (sg)	Coincidencia en clases
OyD	33116 /16486	53	DNS	0.0	Sí
OyD	60945 /60946	80	HTTP	28	Sí
-	1210 /51666	80 /16578	HTTP /P2P	21	No
OyD	47275 /47279	80	HTTP	4	Sí
Origen	1254 /1261	61836 /44763	P2P	56	Sí
OyD	3081 /3083	80	HTTP	1	Sí
Origen	4361 /4367	80	HTTP	66	Sí
OyD	12160 /26695	53	DNS	0.3	Sí
Origen	3352 /3438	28100 /64139	P2P	23	Sí
Origen	1950 /44009	64905 /53	P2P /DNS	68	No

IV. FORMALIZACIÓN E IDENTIFICACIÓN DEL MODELO PARAMÉTRICO

Considerando los resultados discutidos en la sección anterior, podemos concluir que existe un buen rendimiento de clasificación de tráfico usando KNN, y que éste se debe principalmente a elementos relativos entre flujos. Por tanto, parece interesante investigar la posibilidad de definir un clasificador basado en las relaciones (dinámicas) entre flujos, en lugar de un clasificador que se limite a considerar características de un único flujo. Para ello, en el resto de este trabajo se persigue la identificación de parejas de flujos en base a reglas de similitud. Siendo éste un trabajo preliminar para evaluar la viabilidad de dicho clasificador, consideraremos que el porcentaje de aciertos del clasificador será igual al porcentaje de parejas de flujos en las que ambos flujos pertenecen a la misma clase.

De acuerdo a las hipótesis sugeridas por la muestra recogida y analizada en la Tabla V, parece conveniente buscar parejas de flujos cercanos en el tiempo y que presenten ciertas similitudes a nivel de direcciones IP y puertos. Para ello, se propone la definición de una función objetivo asociada a una pareja de flujos. Así, dado un determinado flujo, se considerará que su vecino más cercano es aquel que maximice la siguiente función objetivo:

$$F = \alpha \cdot (|N_{IP} - 1| + \frac{1}{d_{pto1} + k_1} + \frac{1}{d_{pto2} + k_1} + \frac{1}{d_t + k_2}) \quad (2)$$

donde N_{IP} es el número de IPs coincidentes entre flujos, d_{pto1} y d_{pto2} son las distancias (diferencias cuadráticas) entre los puertos asociados a las IPs, d_t es la distancia en tiempo entre el comienzo de los flujos, en segundos, k_1 y k_2 son los parámetros del modelo y:

$$\alpha = \begin{cases} 1, & para \quad N_{IP} \geq 1 \\ \infty, & para \quad N_{IP} = 0 \end{cases} . \quad (3)$$

Como se observa en las ecuaciones (2) y (3), la selección de vecinos se restringe a aquellos flujos que comparten al menos una dirección IP. El objetivo es sólo emparejar aquellos flujos originados por un mismo cliente, o bien destinados a un mismo servidor. El hecho de que ambas direcciones IP coincidan es normalizado al valor unidad en la función objetivo. Los parámetros k_1 y k_2 (referidos a la contribución de los puertos y del tiempo entre flujos, respectivamente) nos permiten ajustar el peso de las distancias en el cómputo global de la función objetivo. Inicialmente no se fija una restricción en el tiempo máximo entre flujos admisible para permitir una relación de vecindad.

Los parámetros k_1 y k_2 son ajustados a partir del porcentaje de aciertos en el conjunto de datos F14, con la distribución de flujos en protocolos previamente ofrecida en la Tabla II. La Figura 4 muestra el resultado del ajuste. De dicha gráfica se desprende el excelente ajuste del clasificador basado en la relación de vecindad. De acuerdo a la misma, se fijan los parámetros a los valores $k_1 = 1$ y $k_2 = 2$. Cabe destacar que, si bien estos valores producen el ajuste óptimo, pequeñas variaciones de los mismos no suponen una gran reducción en el rendimiento (del 99.5% al 97.5%).

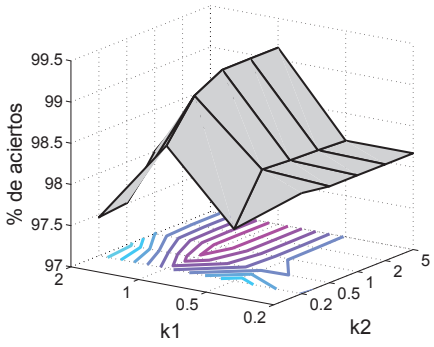
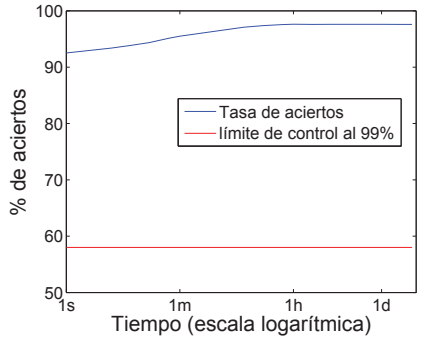


Fig. 4. Estudio paramétrico de posibles valores de k_1 y k_2 para el establecimiento de la relación de vecindad entre flujos.

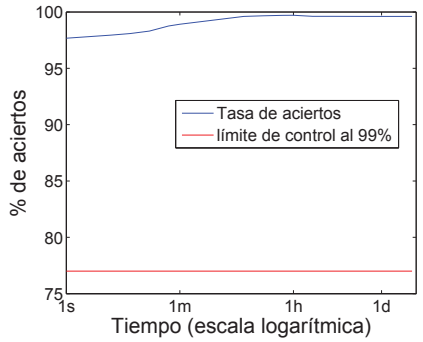
V. Validación del modelo paramétrico

El objetivo de esta sección es validar el modelo paramétrico definido y ajustado en la sección previa, para lo cual utilizaremos el conjunto de datos F51, descrito en la segunda columna en la Tabla II. Adicionalmente, considerando la aplicabilidad real de la propuesta en un nodo de red con memoria limitada, resulta adecuado estudiar el tiempo máximo permitido entre flujos y el rendimiento del modelo para dicha restricción temporal.

En la Figura 5 se presenta la tasa de éxito del modelo ($k_1 = 1$ y $k_2 = 2$) aplicado al conjunto de datos F51 considerando distintos tiempos máximos permitidos entre flujos (abscisas), tanto en la clasificación de protocolos (Fig. 5(a)), como en la discriminación de tráfico P2P (Fig. 5(b)). Los resultados se comparan con el límite de control al 99% de confianza (p-valor = 0.01) correspondiente a los test de permutación en el conjunto de datos F51 (gráficos no mostrados). Como se observa, una mayor restricción del tiempo entre flujos a la hora de establecer la vecindad provoca un descenso en la tasa de éxito a la hora de formar duplas de flujos con la misma etiqueta. No obstante, los gráficos nos indican que restricciones de tiempos muy fuertes (intervalos muy reducidos) siguen permitiendo un agrupamiento con resultados satisfactorios, muy por encima del nivel establecido por el límite de control.



(a)



(b)

Fig. 5. Detalle de la evolución de la tasa de acierto en función de la restricción de tiempo máxima existente entre flujos de cada dupla. (a) Para la lista de protocolos detallada, (b) para la lista de protocolos agrupados en P2P o no P2P.

Un análisis de los resultados detallado para cada protocolo puede ser interesante para mejorar la interpretación de los mismos. En las Figuras 6 y 7 se muestra la mediana de los tiempos entre vecinos, seleccionados de acuerdo al modelo paramétrico, para los conjuntos de calibración y test, respectivamente. La mediana ha sido escogida en lugar del valor medio por su robustez ante valores anormalmente altos. En las gráficas se observa que algunos protocolos presentan tiempos elevados entre flujos (ej. SSH), responsables de la bajada del rendimiento del modelo al restringir el tiempo máximo entre flujos. Téngase en cuenta, no obstante, que el número de flujos asociado a un mismo protocolo es bastante reducido en algunos casos (ver Tabla II), con lo que la mediana puede no ser representativa del comportamiento del protocolo. En cualquier caso, la mayoría de los protocolos establecen una vecindad óptima para intervalos de tiempo menores al minuto, según el valor de su mediana. Adicionalmente, si se fuerza a restringir el tiempo, el rendimiento no es penalizado en un alto grado, como se refleja en las Figuras 5(a) y 5(b).

Cabe destacar la capacidad generalizadora de la propuesta, ya que en el conjunto de test considerado existen protocolos que no aparecen en el conjunto de calibración. Por otro lado, se pueden observar diferencias significativas en las medianas obtenidas para ciertos protocolos en el conjunto de datos F14 y el F51 (ej. Gnutella o DNS). Si bien la tasa de coincidencia de los protocolos en flujos emparejados no se ve afectada

por ese hecho, lo que es otra manifestación de la robustez de la propuesta, estas diferencias en mediana pueden ser una limitación de cara a la futura clasificación.

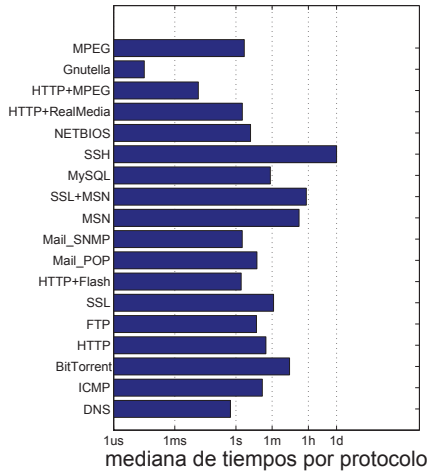


Fig. 6. Mediana de tiempos entre flujos de las distintas duplas, detallado por protocolos, para el conjunto de calibración.

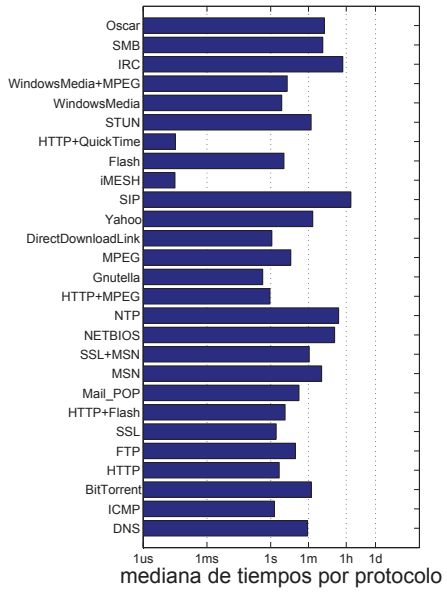


Fig. 7. Mediana de tiempos entre flujos de las distintas duplas, detallado por protocolos, para el conjunto de evaluación.

VI. Conclusiones

El presente trabajo estudia la definición de reglas de similitud entre flujos de datos en red para su aplicación en clasificación de tráfico, con especial énfasis en la discriminación de tráfico *peer-to-peer* (P2P). El objetivo perseguido es la definición de un modelo paramétrico que permita identificar parejas de flujos con un mismo protocolo asociado. Para ello, se han utilizado dos conjuntos de datos con 62 características calculadas sobre 67.015 y 45.167 flujos, respectivamente. Las características no utilizan ninguna información del *payload* en los paquetes de tráfico. El primer conjunto de datos se utilizó para el diseño y la calibración del modelo paramétrico y el segundo para su validación. Los resultados indican que es posible identificar parejas de flujos en los que coincida el protocolo de aplicación en un porcentaje superior al 90%, y en los que ambos flujos sean o no tráfico P2P en un porcentaje superior al 97%, ambos resultados en validación y para ventanas temporales de menos de 1 segundo. Si se permiten ventanas temporales mayores, el porcentaje de coincidencia es aún mayor. Este resultado valida la viabilidad de definir un clasificador de tráfico, ubicado en un nodo de red, que tome decisiones a partir de las relaciones entre flujos en una ventana temporal.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Ciencia e Innovación (MICINN) del gobierno de España con el proyecto TEC2008-06663-C03-02.

Referencias

[1] A. Callado, C. Kamienski, G. Szabo, B.P. Gero, J. Kelner, "A Survey on Internet Traffic Identification," *IEEE Communications Surveys & Tutorials*, vol. 11, n. 3, pp. 37-52, 2009.

[2] S. Sen and J. Wang, "Analyzing Peer-to-Peer TrafficAcross Large Networks," *IEEE/ACM Transactions on Networking*, vol. 12, n. 2, pp. 219-232, 2004.

[3] Madhukar, A., Williamson, C., "A Longitudinal Study of P2P Traffic Classification", *Proc. of Int. Symposium on Modeling, Analysis and Simulation*, pp. 179-188, 2006.

[4] R. Keralapura, A. Nucci, and C. Chuah, "A Novel Self-Learning Architecture for P2P Traffic Classification in High Speed Networks," *Computer Networks*, vol. 54, pp. 1055-1068, 2010.

[5] Xuan-min, L., Jiang, P., Ya-jian, Z., *"A New P2P Traffic Identification Model Based on Node Status"*, In Int. Conference on Mangement and Service Science, pp. 1-4, 2010.

[6] X. Li and Y. Liu, "A P2P Network Traffic Identification Model Based on Heuristic Rules,". *Int. Conference on Computer Application and System Modeling*, vol. 5, pp. 177-179, 2010.

[7] W. JinSong, Z. Yan, W. Qing, and W. Gong, "Connection Pattern-based P2P Application Identification Characteristic," *Proc. of Int. Conference on Network and Parallel Computing Workshops*, pp. 437-441, 2007.

[8] M. Soysal and E.G. Schmidt, "Machine Learning Algorithms for Accurate Flow-Based Network Traffic Classification: Evaluation and Comparison," *Performance Evaluation*, vol. 67, n. 6, pp. 451-467, 2010.

[9] L. Jun, Z. Shunyi, L. Yanqing, and Z. Zailong, "Internet traffic classification using machine learning," *Second International Conference on Communications and Networking in China (CHINACOM'07)*, pp 239-243, 2007.

[10] Y. Lim, H. Kim, J. Jeong, C. Kim, T.T. Kwon, and Y. Choi, "Internet traffic classification demystified: on the sources of the discriminative power," *Proceedings of the 6th International Conference On Emerging Networking Experiments And Technologies (CoNEXT'10)*, 2010.

[11] OpenDPI, 2011. Available at http://www.opendpi.org

[12] Mochalski, K., Schulze, H., "Deep Packet Inspection. Technology, applications & net neutrality", White Paper, 2009. Available at http://www.ipoque.com/resources/white-papers

[13] F. Lindgren, B. Hansen, W. Karcher, M. S. ostr om, and L. Eriksson, "Model validation by permutation tests: Applications to variable selection," *Journal of Chemometrics*, vol. 10, pp. 521–532, 1996.

[14] S. Wiklund, D. Nilsson, L. Eriksson, M. S. ostr om, S. Wold, and K. Faber, "A randomization test for pls component selection," *Journal of Chemometrics*, vol. 21, pp. 427–439, 2007.

[15] H. Wold and E. Lyttkens, "Nonlinear iterative partial least squares (nipals) estimation procedures," in *Bull. Intern. Statist. Inst. Proc., 37th session, London*, 1969, pp. 1–15.

[16] P. Geladi and B. Kowalski, "Partial least-squares regression: a tutorial," *Analytica Chimica Acta*, vol. 185, pp. 1–17, 1986.

[17] M. Barker and W. Rayens, "Partial least squares for discrimination," *Journal of Chemometrics*, vol. 17, pp. 166–173, 2003.

Aplicación adaptativa multi-fuente para streaming de vídeo en redes P2P inalámbricas

Juan Caubet, Carlos Gañán, Sergi Reñé, Juanjo Alins y Jorge Mata-Díaz
Departament d′Enginyeria Telemàtica
Universitat Politècnica de Catalunya (UPC)
{juan.caubet, carlos.ganan, sergi.rene, juanjo, jorge.mata}@entel.upc.es

***Resumen*—El *streaming* consiste en la distribución de contenido multimedia a un elevado número de clientes a través de una red. Proporcionar un servicio de *streaming* para nodos móviles inalámbricos presenta muchos desafíos. Una solución *peer-to-peer* (P2P) presenta la gran ventaja de escalar fácilmente para tamaños de población arbitrarios, ya que cada nodo que recibe contenido multimedia puede ofrecer al mismo tiempo su propio ancho de banda para distribuir dicho contenido al resto de nodos. En este trabajo se presenta el diseño e implementación de NeuroCast: una aplicación P2P sin estructura para la transmisión de vídeo. NeuroCast implementa un algoritmo de planificación robusto que reduce el retraso de la distribución de vídeo. Por otra parte, teniendo en cuenta los contenidos heterogéneos, retrasos y anchos de banda de la red, NeuroCast optimiza el problema de la asignación de subcanales. Por lo tanto, éste es adecuado para escenarios inalámbricos dada su capacidad para adaptarse a las condiciones cambiantes de la red.**

I. Introducción

Durante la última década, el crecimiento de los sitios web más populares que sirven contenidos multimedia ha llevado al incremento de aplicaciones de *streaming* de vídeo. Sin embargo, el *streaming* de vídeo a través de una red inalámbrica tiene que hacer frente a múltiples desafíos: 1) no se garantiza nada acerca del ancho de banda, el retardo, y la tasa de pérdida de paquetes; 2) es difícil predecir el ancho de banda, el retardo, y la tasa de pérdida de paquetes, ya que son desconocidos y variables con el tiempo; 3) la heterogeneidad de las prestaciones de los receptores es un problema importante cuando las secuencias de vídeo están distribuidas por toda la red de multidifusión; y 4) se debe emplear un mecanismo de control de congestión para prevenir los períodos de congestión en la red inalámbrica.

Se han propuesto varias formas de abordar estos desafíos. El modelo tradicional *cliente-servidor* es adecuado para la transmisión de contenidos, pero presenta problemas de escalabilidad y una pérdida de eficiencia en la explotación de los recursos. De hecho, un servidor tiene un ancho de banda limitado y no puede servir a más de un número limitado de clientes de forma simultánea. La mejor manera de distribuir un contenido de un origen a un grupo de *hosts* al mismo tiempo es utilizar *IP multicast*. Sin embargo, la implementación de *IP multicast* está limitada debido a varias razones. En primer lugar, requiere cambios en los dispositivos de red, incrementando la complejidad y la sobrecarga en los *routers*. Pero también presenta problemas comerciales, ya que muchos ISPs tienen desactivada esta opción de envío. Otra posibilidad es utilizar redes *peer-to-peer* para la transmisión de contenidos. En este tipo de redes, los receptores actúan como clientes y servidores al mismo tiempo (*Servents*) replicando los paquetes que reciben. El objetivo de esta técnica es distribuir contenidos a un gran número de nodos de una manera escalable, robusta y eficiente. En este sentido, presentamos NeuroCast, una solución P2P de código abierto para el *streaming* de vídeo sobre redes inalámbricas.

Los sistemas P2P se esfuerzan por optimizar tres factores importantes: i) la demora de la puesta en marcha, ii) el retardo extremo a extremo, y iii) el índice de continuidad de la reproducción de vídeo. La mayoría de estos sistemas se pueden clasificar según el tipo de grafo de distribución que aplican: árbol o malla, aunque también existen una gran cantidad de soluciones híbridas. Los esquemas basados en árbol aplican un grafo de distribución en el que la raíz es el origen del contenido y éste se distribuye al resto por las ramas del árbol [1], [2], [3], [4]. En principio, cada nodo recibe datos de un nodo primario, que puede ser la fuente o no. Si los vecinos de los nodos no varían con demasiada frecuencia, este sistema requiere de pocos costes indirectos. De hecho, los paquetes pueden ser enviados de un nodo a otro sin necesidad de mensajes adicionales. Sin embargo, en ambientes muy cambiantes (es decir, rotación rápida de los vecinos) el árbol debe ser continuamente reconstruido, un proceso que requiere una sobrecarga considerable de mensajes de control. Como efecto secundario, los nodos deben almacenar los datos por lo menos durante el tiempo necesario para reparar el árbol, a fin de evitar la pérdida de los paquetes. Por contra, los sistemas basados en malla [5], [6], [7], como su propio nombre indica, implementan un grafo de distribución en malla, donde cada nodo contacta con un subconjunto de nodos para obtener un número de fragmentos suficiente para reconstruir el fichero multimedia deseado. En estos esquemas, cada nodo tiene que saber que fragmentos poseen los otros nodos y explícitamente cuales necesita. Este tipo de sistemas implican costes adicionales, debido por una parte al cambio de rutas entre nodos (los nodos han de anunciar el conjunto de ficheros que poseen) y por otra al proceso de recepción (cada nodo envía una solicitud para recibir los trozos). Pero gracias al hecho de que cada nodo recibe de varios vecinos para recuperar el contenido, los esquemas basados en mallas ofrecen buena resistencia a la caída de nodos. Como desventaja, éstos requieren de grandes búferes para soportar el suministro de

ISBN: 978-84-694-5948-5

fragmentos del vídeo, éstos son necesarios para aumentar las posibilidades de encontrar los fragmentos que faltan en la secuencia de reproducción. En este sentido, NeuroCast evoluciona un sistema basado en árbol como es PeerCast [2] para convertirlo en un sistema mallado idóneo para la distribución de vídeo en redes inalámbricas.

PeerCast es una herramienta de *streaming multicast*, de código abierto, que se utiliza generalmente para la transmisión de audio, y hace uso de un enfoque de distribución de ancho de banda donde los usuarios pueden elegir el nodo desde donde descargar la información. NeuroCast extiende PeerCast con el fin de mejorar sus capacidades para permitir ver vídeos, incluso en escenarios inalámbricos. Por otra parte, NeuroCast modifica el diseño de la red para permitir a cualquier usuario convertirse en un organismo de difusión. Entre las mejoras realizadas en NeuroCast, destaca la capacidad para transmitir y recibir desde múltiples fuentes con distintas capacidades. En contraste con PeerCast, NeuroCast resuelve la asimetría típica de la capacidad de los enlaces mediante el uso de múltiples fuentes. De esta manera, permite transmitir vídeo a través de una red inalámbrica.

En este trabajo se aborda, en primer lugar, el problema de los medios de transmisión desde un punto de vista práctico. Seguidamente, se presenta el diseño de NeuroCast, una aplicación P2P para redes inalámbricas, que: 1) es compatible con entornos altamente cambiantes, 2) soporta distribuciones heterogéneas de la capacidad de subida de los nodos, 3) tiene la capacidad de utilizar múltiples fuentes para aprovechar eficientemente la capacidad de subida disponible, y 4) emplea una adaptación rápida para recuperarse ante los cambios en las condiciones de la red. Debido a esta rápida adaptación, NeuroCast puede operar en un escenario inalámbrico donde los nodos son propensos a sufrir desconexiones y los retardos son mayores.

El resto del documento está estructurado de la siguiente manera. La sección II introduce el sistema PeerCast: conceptos básicos e implementación. La sección III describe el sistema NeuroCast. La sección IV presenta un análisis de NeuroCast, llevado a cabo mediante la emulación de un escenario inalámbrico. Por último, la sección V concluye este trabajo.

II. PeerCast

PeerCast [2] es una aplicación diseñada en sus orígenes para la difusión radiofónica en Internet. Por lo que respecta a su implementación, PeerCast es multi-hilo, de modo que los diferentes procesos se ejecutan al mismo tiempo. El número de procesos en PeerCast es variable y depende de la cantidad de conexiones que se establezcan. El hilo principal además de crear los procesos hijos, se encarga de esperar de forma pasiva la petición de recursos. En general, para cada solicitud que recibe se crea un subproceso que va a atender dicha solicitud hasta su muerte.

II-A. PCP: Packet Chain Protocol

PCP es el protocolo usado en PeerCast para permitir la comunicación entre los diferentes clientes. El objetivo de este protocolo es la reutilización de los mismos flujos de datos que se utilizan para enviar la información multimedia para enviar la información de control. El encadenamiento PCP permite a los usuarios descargar ficheros a través de otros usuarios.

En PCP, el primer paquete del grupo indica el tipo y el número de paquetes que se envían a continuación. Este tipo de paquetes que indican el tipo del grupo se denominan *paquetes padre*. El gran potencial de este protocolo radica en la forma en que los paquetes están encadenados, ya que cada uno de los paquetes dentro de un grupo puede ser al mismo tiempo *paquete padre* de otro tipo de paquetes.

II-B. Entidades

PeerCast utiliza la programación orientada a objetos, es decir, implementa clases que representan las diferentes entidades del sistema. En este sentido, PeerCast está formado por las siguientes clases: *Servent*, *Channel*, *Buffer* y *Root*.

Servent. Esta clase es esencial para el rendimiento de la aplicación PeerCast. Como cualquier aplicación P2P en tiempo real, la aplicación depende de las condiciones de la red, que pueden causar que los paquetes lleguen desordenados. Por lo tanto, es fundamental tener un buen sistema para ordenar y almacenar los paquetes de manera eficiente. La clase *Servent* gestiona el canal de envío y escucha las solicitudes procedentes de la red o del mismo *host*. Las instancias de *Servent* ejercen de servidor y cliente al mismo tiempo. Éstas se clasifican según el tipo de transmisión. Así, las más utilizadas son:

- `Server`: Es el servidor del hilo principal que se ocupa de cualquier solicitud. La presente subclase creará otras clases de servidores para gestionar las distintas peticiones.
- `Relay`: Esta clase gestiona la retransmisión de un canal entre las diferentes instancias de PeerCast. En este modo, los datos se envían usando el protocolo PCP.
- `Direct`: Esta clase se utiliza para retransmitir el archivo multimedia directamente sin encapsular los datos. Esta clase se utiliza para controlar el envío de paquetes a un reproductor de ficheros multimedia.

Channel. La clase *Channel* implementa el canal. Un canal se origina cuando un usuario comienza a retransmitir los datos multimedia sobre PeerCast. Este usuario, es el primero en subir el canal a la red, por ello se denomina *emisor*, y se convierte en el único contacto entre la fuente original y la red PeerCast. En este sentido, el canal se comparte entre los diferentes nodos de la red.

Buffer. Es evidente la necesidad de un búfer, ya que cada nodo puede funcionar como cliente y como servidor, y por lo tanto debe almacenar en la memoria los datos con el fin de ser capaz de retransmitirlos.

Root. A pesar de que PeerCast fue diseñada para funcionar de una manera descentralizada, su despliegue la convirtió en centralizada. Inicialmente PeerCast sólo tenía clientes. Si un grupo de usuarios querían compartir un fichero de audio, sólo necesitaban saber la dirección IP de algunos de ellos y el identificador de canal. De esta forma creaban un árbol de descarga. Por otro lado, si un usuario quería dicho audio

pero no conocía a ningún usuario del grupo de distribución no podía escucharlo. PeerCast se ocupa de esta cuestión con el concepto de los nodos raíz. Para cada red de difusión, un nodo actuará como raíz, siendo la principal fuente del flujo de datos, mientras que los otros recibirán estos datos y, posiblemente, los retransmitirán. Además, los nodos raíz recopilan información sobre otros clientes con el fin de crear un directorio de información.

III. NeuroCast

Las redes P2P de distribución de ficheros han evolucionado de las típicas topologías árbol hacia topologías de malla. Esta evolución permite a NeuroCast utilizar el ancho de banda de N usuarios, que no podrían transmitir un vídeo por sí solos, para ser capaz de lograr la difusión del vídeo. NeuroCast es una aplicación multi-hilo, de manera que cada usuario de la red es capaz de recibir un flujo de datos de diferentes nodos a la vez, y retransmitirlo a cualquier otro usuario.

Los usuarios interactúan con NeuroCast a través de una interfaz web donde pueden gestionar la aplicación, así como obtener información acerca de las transmisiones que se están produciendo o acerca de los nodos de los que se están descargando un vídeo.

Además de los nodos que actúan como fuentes de datos, hay otro tipo de clientes en NeuroCast: los *trackers*. Estos clientes se encargan de la recopilación de información sobre los nodos que comparten un canal en particular. Así, cuando un usuario inicia una sesión, el *tracker* le proporcionará a ese usuario una lista con los nodos que están ofreciendo el canal de vídeo solicitado. Estos nodos que están compartiendo el canal se denominan *hits*. En este sentido, es esencial que los *trackers* dispongan siempre de una lista de *hits* actualizada. NeuroCast se ha implementado de tal manera que todos los usuarios actúan como *trackers*. Por lo tanto, cualquier nodo puede compartir con otro su lista de *hits*.

Por otra parte, utilizando un enfoque de malla surge un nuevo concepto: el *substreaming*. El *substreaming* aparece como consecuencia de la necesidad de recibir un flujo de varias fuentes al mismo tiempo. De esta manera, el *substreaming* consiste en dividir el flujo original en N partes, que se distribuyen desde las diferentes fuentes que están disponibles. Por otra parte, podría ser interesante recibir más paquetes de una fuente que de otra. Por lo tanto, existen nuevos problemas que tratar, como el orden de llegada de los paquetes o la distribución de los paquetes de las fuentes disponibles. Los paquetes que conforman el subflujo se denominan *chunks*.

Los acciones que un nodo tiene que realizar cuando se une a la red NeuroCast son las siguientes:

- *Obtención de la lista de hits.* Durante el inicio, la aplicación NeuroCast se conecta al *tracker*, quien envía de vuelta una lista de *hits* del canal solicitado.
- *Selección de nodos.* Una vez que un nodo tiene la lista de *hits* solicitada tiene que seleccionar el mejor conjunto de nodos de esa lista para sus fines.
- *Asignación de paquetes.* NeuroCast envía los paquetes agrupados en *chunks* numerados, que se asignan a la vez que se transmite el flujo.
- *Estado de la red y adaptación de carga.* NeuroCast supervisa el estado de la red de forma permanente durante la descarga del canal.

III-A. Entidades

El número de procesos en NeuroCast es variable y depende de la cantidad de conexiones establecidas. Al igual que en PeerCast, NeuroCast tiene un hilo principal que se ocupa de las peticiones entrantes y crea los procesos hijo que se encargan de estas solicitudes. En esta sección se describen brevemente las dos entidades principales de NeuroCast.

Servent. La entidad *Servent* maneja las solicitudes del sistema y envía el flujo de datos, ya sea a otra instancia NeuroCast o a un nodo; actúa como servidor y cliente. Al igual que en PeerCast, existen tres tipos de diferentes de *Servents*: *Server, Relay y Direct*.

Channel. Los canales son los principales elementos en NeuroCast. Se utilizan para tratar los flujos multimedia que se están compartiendo en la red. Cada flujo utiliza un canal diferente. NeuroCast sólo distribuye partes de la secuencia, no todo el flujo a la vez. Cada canal tiene su fuente original. En general, esta fuente es un nodo (emisor) que crea el canal y se convierte en el único vínculo físico entre los clientes NeuroCast y el archivo multimedia original. El canal tiene dos elementos principales:

- *El búfer*: Gestiona y almacena los paquetes entrantes temporalmente para permitir una posterior retransmisión.
- *El flujo del canal*: Se encarga del protocolo de enlace y la recepción de canales. La clase *Channel* permite controlar el flujo de los paquetes entrantes.

NeuroCast también introduce la posibilidad de utilizar varios nodos al mismo tiempo, dividiendo el flujo entre ellos. Los subcanales son los elementos de la aplicación que permiten identificar los fragmentos del vídeo. Así, cada una de las fuentes se asocia a un subcanal.

III-B. Balanceo de carga

Como se menciona en la Introducción, NeuroCast permite equilibrar la carga de acuerdo con las condiciones de la red. De esta manera, NeuroCast implementa diferentes algoritmos para optimizar la selección de nodos entre los *hits* del canal solicitado y la distribución de la carga entre los elegidos. A continuación se muestra como NeuroCast es capaz de adaptarse a las condiciones de la red y redistribuir la carga de acuerdo con estas condiciones.

Selección de nodos.

Una de las partes más críticas en cualquier reproductor basado en una red P2P es la selección de nodos. A diferencia de las aplicaciones P2P para la distribución de archivos, el hecho de trabajar con vídeo crea una dependencia fuerte y directa entre los nodos que transmiten el flujo.

Una vez que la lista de *hits* se ha recibido, es necesario calcular cuál es el subconjunto de ellos, con las mejores condiciones relativas a la red, que permiten descargar el flujo. NeuroCast siempre trata de dar prioridad a nodos con un

alto ancho de banda disponible, con una tasa de pérdida de paquetes baja, y con una alta disponibilidad. NeuroCast utiliza un algoritmo de selección de nodos que se basa en el algoritmo usado en CollectCast [5]. Los puntos fuertes de este algoritmo son la selección de fuentes, la monitorización del estado de la red y la redistribución de fuentes periódicamente con el fin de adaptarse a las condiciones variantes de la red. El proceso de selección se puede dividir en tres fases:

1. Obtener la lista de *hits* asociados al canal solicitado.
2. Enumerar los conjuntos de *hits* que satisfagan las restricciones impuestas por:

$$R_l \leq \sum_{x \in \bar{P}_{act}} R_x \leq R_u, \quad (1)$$

   donde $R_x = R_p$ que es la tasa máxima de envío con la que un nodo puede contribuir, R_l es el límite inferior de la tasa total de envío de un conjunto de nodos, R_u es el límite superior de la tasa total de envío de un conjunto de nodos, y P_{act} es el conjunto de nodos activos.
3. Seleccionar el mejor conjunto de nodos de entre los obtenidos en el paso anterior.

Al solicitar un nuevo canal, NeuroCast obtiene un conjunto de parámetros de cada fuente. Así, el solicitante obtiene el parámetro R_p de cada fuente, que es la máxima tasa de envío con la que un nodo puede contribuir. Entonces, NeuroCast calcula el rendimiento y las pérdidas con cada uno de los *hits*. Para hacer estas mediciones, NeuroCast se aprovecha de una versión modificada de la aplicación Iperf [8].

Asignación de paquetes.

Los nodos activos ($\bar{P}_{act}$) colaboran colectivamente para enviar el vídeo segmento por segmento: todos cooperan en el envío del primer segmento, luego el segundo, y así sucesivamente. El archivo está dividido en segmentos de datos de igual longitud. Al nodo p se le asigna un número de paquetes D_p para enviar en proporción a su tasa real de transmisión:

$$D_p = \left\lceil \Delta \cdot \frac{R_p}{\sum_{x \in \bar{P}_{act}} R_x} \right\rceil. \quad (2)$$

donde Δ es el número de *chunks* en los que se divide el flujo para poder funcionar con grupos de subcanales. El valor D_p se utiliza para distribuir los *chunks* entre los diferentes nodos del conjunto.

Adaptación de red.

Para garantizar la carga máxima de la red, NeuroCast añade una nueva funcionalidad para adaptar en todo momento el número de *chunks* que un nodo se está descargando. La variable utilizada para determinar cuando redistribuir el número de *chunks* es el tiempo entre llegadas de paquetes. Por lo tanto, NeuroCast establece un umbral que permite, una vez sobrepasado, redistribuir el número de *chunks* que cada nodo retransmite. Este umbral se calcula mediante la siguiente ecuación:

$$th = \frac{\frac{b_{size}}{rate} \cdot \Delta_{packets} \cdot \Delta}{n_s}, \quad (3)$$

donde b_{size} es el tamaño del búfer, $rate$ es la tasa de bits del vídeo reproducido, $\Delta_{packets}$ es el retraso máximo permitido entre las llegadas de paquetes y n_s es el número de *chunks* en los que se divide el subcanal que se retransmite.

Para alertar al resto de *hits* de la redistribución de la carga, el nodo solicitante les envía el mensaje `CHANGE_PEER_CHUNKS`. A continuación, el nodo receptor lee la petición y habilita el indicador asociado con la redistribución para que el servidor de retransmisión que difunde el canal se ponga a la espera de recibir. Una vez que el subcanal está listo para la redistribución de la carga, sigue estos pasos:

1. Notifica a los otros subcanales acerca de la redistribución.
2. Disminuye en uno el número de *chunks* que se están descargando de su *hit*.
3. Busca un subcanal con un tiempo entre llegadas menor con el fin de enviar el *chunk* que le falta.
4. Espera hasta que el subcanal de menor tiempo entre llegadas haya redistribuido los *chunks*.

El resto de los subcanales, una vez que han sido informados de que una redistribución de carga va a tener lugar, también entran en el proceso de redistribución. Aquí hay que distinguir dos situaciones diferentes: por un lado los subcanales de menor tiempo entre llegadas y, por otro, el resto de los subcanales. Estos últimos simplemente esperan a que la redistribución se realice para continuar con la descarga de los nuevos *chunks* asignados. El subcanal con el menor tiempo entre llegadas sigue estos pasos:

1. Comprueba que el canal tiene más de un subcanal en ese momento. En caso de que sólo exista un subcanal activo empieza a descargar el flujo completo.
2. Si hay más subcanales, aumenta en uno el número de *chunks* del subcanal.
3. Por último, redistribuye la carga entre todos los *hits* de la lista de acuerdo a su número de *chunks*.

Un caso diferente al presentado anteriormente, pero que también requiere de la redistribución de la carga, se produce cuando un nodo no recibe paquetes de un subcanal determinado. En estas situaciones la redistribución de carga se fuerza inmediatamente.

Otra de las novedades que se han introducido en NeuroCast es la liberación de los *hits* que están enviando un solo *chunk*, y a pesar de eso, siguen enviando paquetes con retraso. En estos casos, se les quita de la lista de *hits* y se descarga el último *chunk* de otro nodo, dejando a los subcanales en estado de alerta hasta que el canal no esté disponible o se decida desconectar.

IV. Evaluación

En esta sección se presentan los resultados de la emulación mediante la cual se ha evaluado el comportamiento de NeuroCast. Se han configurado dos entornos diferentes con el fin de comprobar el rendimiento de NeuroCast en condiciones de red heterogéneas. El escenario de referencia para nuestras emulaciones se ha creado utilizando VNUML [9], como se muestra en la Figura 1.

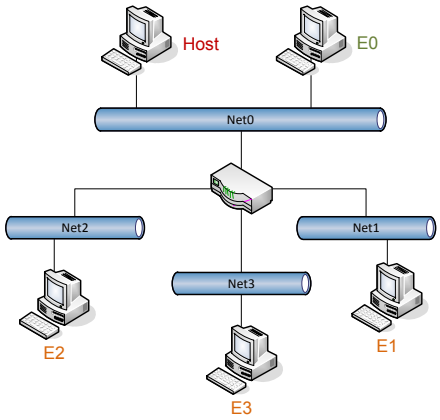


Figura 1. Topología de la red.

Las principales características de la red virtual y de las máquinas virtuales se muestran en la Tabla I. Utilizando esta configuración base se modifican las condiciones de la red donde estos nodos están operando para recrear dos entornos distintos. Estos entornos pretenden emular las condiciones de operación en una red inalámbrica.

	Host	**E0**	**E1**	**E2**	**E3**
maxRelays	1	4	4	4	4
Rp	2 Mbps	1 Mbps	1 Mbps	700 Kbps	1 Mbps
minRp	100 Kbps	100 Kbps	100 Kbps	100 Kbps	100 Kbps
Δ	0	10	10	10	15
factor_sep_pack	1	1.3	1.3	1.3	1.3
delay_margin	0	3	3	3	3
$\Delta_{packets}$	0	30	30	30	16
Rl	100 Kbps	100 Kbps	100 Kbps	100 Kbps	100 Kbps
Ru	2 Mbps	2 Mbps	3 Mbps	3 Mbps	3 Mbps

Tabla I
PARÁMETROS INICIALES DE LA RED VIRTUAL.

IV-A. Impacto de los cambios de topología

En un escenario móvil, es común tener topologías de red muy cambiantes. En esta sección, se evalúa cómo NeuroCast logra adaptarse a estas condiciones cambiantes sin alterar la calidad del servicio de *streaming* de vídeo. Así, al descargar un vídeo de más de una fuente, si una de estas fuentes deja la red, entonces el nodo solicita la redistribución de la carga y descarga los subcanales de otras fuentes. El proceso que se lleva a cabo consiste en:

1. Comprobar si hay un nodo libre en la lista de *hits*. Es decir, un *hit* del que no se está descargando ningún *chunk* ni al que se le esté enviando algún *chunk*.
2. Solicitar una nueva lista de *hits* al *tracker* y volver a comprobar de nuevo si hay algún *hit* libre.
3. Descargar los *chunks* "perdidos" de uno de los *hits* que ya se esté utilizando.

En este último caso se requiere una reasignación de los subcanales. Dependiendo de la situación, NeuroCast lleva a cabo una serie de cambios:

- Si el cliente está descargando más de un subcanal desde el mismo *hit*, entonces pasamos a agrupar los *chunks* en el mismo subcanal.
- En caso de eliminar un subcanal que no está al final de la lista, se ocupa la posición que se pierde con el último subcanal, de forma que no queden espacios vacíos.
- Si un subcanal descarga todos los *chunks* en que dividimos el flujo, ponemos el valor de número de *chunks* a 1 y ya no dividimos el flujo en diferentes partes.

A continuación, se muestra un ejemplo donde el nodo $E0$ sale de la red. Inicialmente se tiene en el nodo $E3$ la configuración de las Tablas II y IV-A:

Lista de *hits*	
Hit	**Dirección IP**
0 (tracker)	10.0.0.1
1	10.0.0.2
2	10.0.1.2
3	10.0.2.2

Tabla II
LISTA INICIAL DE HITS.

Lista de Subcanales			
	Dirección IP	**Número de *Chunks***	***Chunks***
Subcanal 0	10.0.0.2 ($E0$)	6	0 2 4 6 8 10
Subcanal 1	10.0.1.2 ($E1$)	3	1 3 5
Subcanal 2	10.0.2.2 ($E2$)	3	7 9 11

Tabla III
DISTRIBUCIÓN INICIAL DE LOS CHUNKS ENTRE LOS TRES SUBCANALES.

Después de algún tiempo, el nodo $E0$ sale de la red. Los nodos solicitantes comprueban que no hay ningún *hit* libre en su lista de *hits*, por lo que solicita una lista nueva de *hits* al *tracker*. (véase la Tabla IV).

Lista de *hits*	
Hit	**Dirección IP**
0 (tracker)	10.0.0.1
1	10.0.1.2
2	10.0.2.2
3	10.0.3.2

Tabla IV
LISTA FINAL DE HITS.

Lista de subcanales			
	Dirección IP	**Número de *chunks***	***Chunks***
Subcanal 0	10.0.2.2 ($E2$)	3	7 9 11
Subcanal 1	10.0.1.2 ($E1$)	9	0 1 2 3 4 5 6 8 10

Tabla V
DISTRIBUCIÓN DEFINITIVA DE LOS CHUNKS ENTRE LOS 2 SUBCANALES.

Como la única nueva incorporación a la lista de *hits* es la propia IP del nodo solicitante, sólo le queda la posibilidad de reaprovechar uno de los *hits* que ya le está transmitiendo algún *chunk*. En este caso hemos escogido el nodo $E1$, como se muestra en la Tabla V. En resumen, NeuroCast es capaz

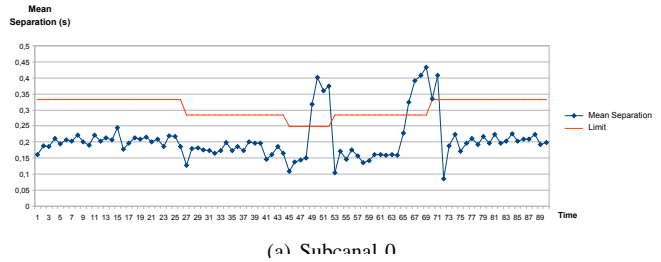
de adaptarse a cualquier cambio en la topología de la red sin importar si el *hit* ha abandonado la red porque perdió la conectividad o porque ha decidido detener la transmisión.

IV-B. Impacto del tráfico interferente

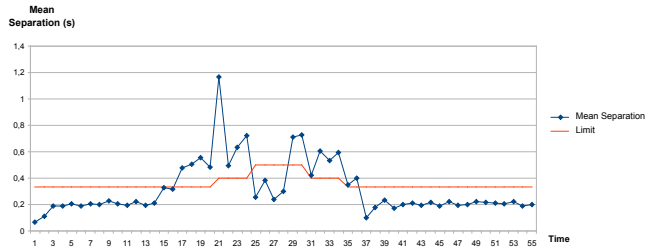
Otra situación que tiende a ocurrir en un escenario inalámbrico es sufrir interferencias de otro nodo de la red. Para mostrar como funciona NeuroCast en estas condiciones, siguiendo con la topología de red de la Figura 1, partimos de la situación en la que tenemos 4 máquinas funcionando con NeuroCast y reproduciendo un vídeo: $Host$, $E0$, $E1$ y $E2$. En esta situación la máquina $E0$ se descarga el flujo del $Host$, la máquina $E1$ se lo descarga de $E0$ y finalmente $E2$ se descarga el vídeo desde $E0$ y $E1$ al mismo tiempo.

En esta sección, nos centramos en estudiar el rendimiento de $E2$ mientras sufre interferencias de tráfico. Para lograr este entorno interferente, se introduce tráfico en el enlace $Net2$ utilizando el *Distributed Internet Traffic Generator* (D-ITG) [10] desde el nodo $E3$.

Antes de inyectar el tráfico, $E2$ se descarga 6 chunks de cada fuente dado que las condiciones de $Net1$ y $Net2$ son prácticamente idénticas. Sin embargo, cuando se introduce un tráfico de 800 kbytes/s en el enlace $Net2$, este equilibrio se rompe. Por lo tanto, los *chunks* procedentes de $E1$ se ven afectados y, en consecuencia, el tiempo entre llegadas de paquetes se incrementa. Precisamente esto se refleja en los gráficos de las Figuras 2(a) y 2(b).



(a) Subcanal 0



(b) Subcanal 1.

Figura 2. Evolución del tiempo medio de llegada de los paquetes.

La Figura 2 muestra, además del tiempo medio entre llegadas de paquetes, el umbral teórico que NeuroCast utiliza para determinar cuándo es necesario una redistribución de los *chunks*. Es importante señalar que la redistribución se produce cuando se supera 3 veces este umbral, dado que este es el valor del parámetro `delay_margin` definido en el fichero de configuración de NeuroCast.

Como hemos visto anteriormente, algunos parámetros en el fichero de configuración afectan directamente al comportamiento de la aplicación durante las variaciones de las condiciones de la red. Los siguientes experimentos se han llevado a cabo variando dos parámetros tales como Δ, el número de *chunks* en que se divide un vídeo, y $\Delta_{packets}$, el número de paquetes observados en el búfer para calcular el tiempo medio entre llegadas de paquetes, y hemos observado en cada caso el tiempo necesario para redistribuir los *chunks*.

Para forzar la redistribución cambiamos la capacidad del enlace de uno de los *hits* que retransmite el vídeo utilizando *Network Emulator* (NetEm) [11]. Así, han pasado de los 1000 Kbps disponibles al comienzo del período de sesiones, a 112 kbps.

Analizando los resultados de la Tabla VI vemos que el parámetro Δ no afecta significativamente al tiempo obtenido, mientras que el parámetro $\Delta_{packets}$ es decisivo.

	Test1	**Test2**	**Test3**	**Test4**	**Test5**
Δ	20	5	20	20	5
$\Delta_{packets}$	20	20	50	5	5
Tiempo de redistribución	90s	86s	173s	31s	31s

Tabla VI
RENDIMIENTO CON UNA INTERFERENCIA DEL TRÁFICO DEL 90 % DE LA CAPACIDAD DEL ENLACE.

En consecuencia, como puede deducirse de estos resultados, cualquier nodo usando NeuroCast, incluso sufriendo interferencias, es capaz de funcionar sin sufrir una degradación mayor de la calidad del servicio.

IV-C. Evolución del búfer

La capacidad del búfer de NeuroCast está limitada por defecto a 64 paquetes. Esto significa que con un flujo de datos de 1 Mbps y paquetes de 8192 bytes el búfer se renueva cada 4 segundos. A través de dos parámetros como son el `lastPos` (la posición en el búfer del último paquete consecutivo disponible) y el `nextSendPos` (la posición en el búfer del siguiente paquete que tenemos que enviar al subcanal i), se puede evaluar la utilización del búfer a lo largo de la sesión. Según el valor que tome la diferencia entre los dos parámetros distinguiremos 3 situaciones:

- Diferencia cercana a 0: Funcionamiento óptimo. Significa que el *servent* que envía paquetes utiliza los que acaban de llegar al búfer.
- Diferencia incrementándose hasta 63: Problemas con el envío. Significa que se tienen pendientes de enviar tantos paquetes como indica la diferencia, y por lo tanto, se está a punto de "saturar" el búfer.
- Diferencia disminuyendo hasta -63: Problemas con la recepción. Significa que están pendientes para escribirse en el búfer tantos paquetes como indica la diferencia, ya que el *servent* pide más paquetes nuevos.

Para observar el comportamiento del búfer se simula un escenario donde el *host* envía el flujo a un único nodo (véase la Figura 3). En esta situación los paquetes llegan al *host*

de forma ordenada y sin retrasos, ya que estamos conectados directamente al servidor.

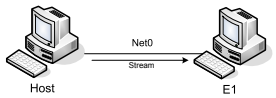


Figura 3. Escenario: 1 Host y 1 Nodo.

La estructura del búfer del $Host$ presenta la estructura mostrada en la Figura 4. Los 3 elementos básicos que actúan en el *host* son: el búfer, el *servent* que envía paquetes a la máquina $E1$ y flujo del canal que lee los paquetes del servidor de *streaming* para escribirlos en el búfer. En la Figura 4 se muestra la evolución de los dos parámetros: el `nextSendPos` y el `lastPos`. En el instante actual suponemos que se ha terminado de escribir en la posición 16 del búfer el paquete 80, y el *servent* acaba de enviar el paquete 78 situado en la posición 14. Por lo tanto, el próximo que se enviará (instante $t + 1$) será el 79 (posición 15), y el `lastPos` apuntará al paquete 80 (posición 16).

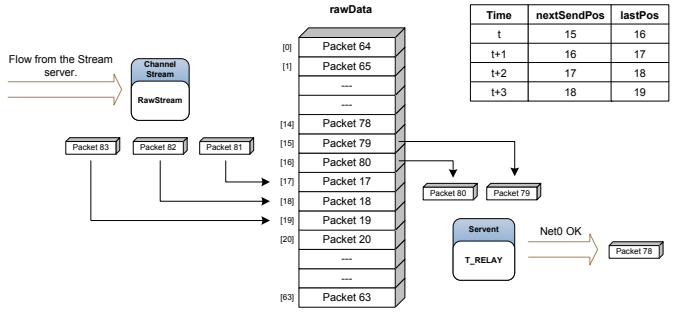


Figura 4. Búfer durante la recepción de paquestes.

A continuación, para ver cómo evoluciona NeuroCast cuando varían las condiciones en la red, saturamos el enlace $Net0$ de manera que la máquina $E1$ no pueda recibir los paquetes del *host*. En cuanto a la llegada de paquetes al sistema, la situación se mantiene igual, de manera que el índice `lastPos` evoluciona como en el caso anterior. El último paquete escrito en el sistema es el 80 en la posición 16 y en los siguientes instantes se incorporarán los paquetes 81, 82, 83, etc. El problema surge con la salida, ya que una vez el *servent* deja de enviar paquetes a la red, el índice `nextSendPos` se mantiene constante, como en el caso del ejemplo que se mantiene en la posición 15. En esta situación el número de paquetes pendientes de ser enviados al búfer se va incrementando progresivamente. El gráfico de la Figura 5 muestra la evolución del número de elementos en cola para ser enviados.

Para saturar el enlace se envía un tráfico intermitente a $E1$ a través de $Net0$, de manera que ocupe todo el ancho de banda durante 30 segundos. De esta manera el $Host$ sólo puede enviar algún paquete a $E1$ de forma esporádica, ya que se encuentra el canal ocupado con alta probabilidad. En el gráfico de la Figura 5 podemos observar que, tanto al inicio

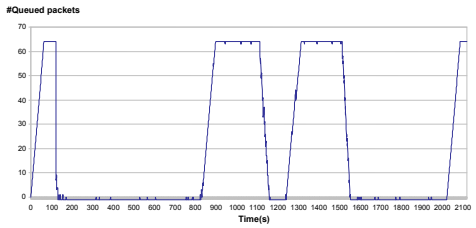


Figura 5. Evolución del tamaño del búfer con problemas de transmisión.

de la sesión como con el tráfico intrusivo, la cola del búfer se incrementa rápidamente hasta los 64 paquetes, que es el límite del búfer. El crecimiento en estos casos es constante en las 6 "ráfagas" que se representan, debido a que la cola se incrementa paquete a paquete a medida que el flujo del canal introduce nuevos paquetes.

A continuación estudiamos la evolución del búfer en un entorno P2P como el de la Figura 1. En este caso se analiza el comportamiento del búfer en el $Host$ y en $E2$. De esta manera podemos estudiar un caso donde tenemos una máquina que recibe el flujo de dos fuentes diferentes ($E0$ y $E1$), y a la vez envía parte del vídeo a otra máquina ($E3$). La secuencia que se ha seguido ha sido la siguiente: se han arrancado (en este orden) las máquinas $Host$, $E0$, $E1$, $E2$ y $E3$. A continuación se ha introducido un tráfico interferente en la máquina $E1$ durante aproximadamente unos 10 segundos, y a continuación la máquina $E0$ ha finalizado su sesión. Unos instantes después lo hacen las máquinas $E3$, $E2$, $E1$ y $Host$. La Figura 6 muestra la evolución del búfer en el $Host$ respecto al parámetro `nextSendPos`.

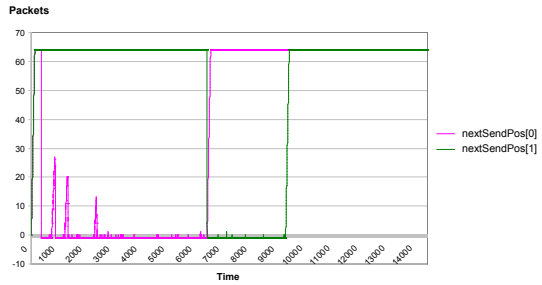


Figura 6. Evolución del búfer en el $Host$.

Analizando el resultado obtenido observamos como al principio el búfer se llena rápidamente hasta llegar a los 64 paquetes, y una vez $E0$ inicia su sesión le envía el flujo. A continuación se observan 3 picos, que en ningún caso superan los 30 paquetes, y corresponden a la entrada en el sistema de las máquinas $E1$, $E2$ y $E3$. Estos son debidos al cálculo del ancho de banda disponible en la máquina $E0$, que provoca un ligero retraso en la recepción de los paquetes. Aproximadamente hasta el instante 6,500 la utilización del búfer se mide respecto al `nextSendPos[0]`, asociado al *servent* que envía paquetes en *hit* $E0$. Pero cuando se pasa a enviar el flujo al *hit* $E1$ se utiliza un nuevo *servent* y el parámetro de referencia en este caso es el $nextSendPos[1]$.

Por otra parte la Figura 7 muestra la evolución del búfer de la máquina $E2$.

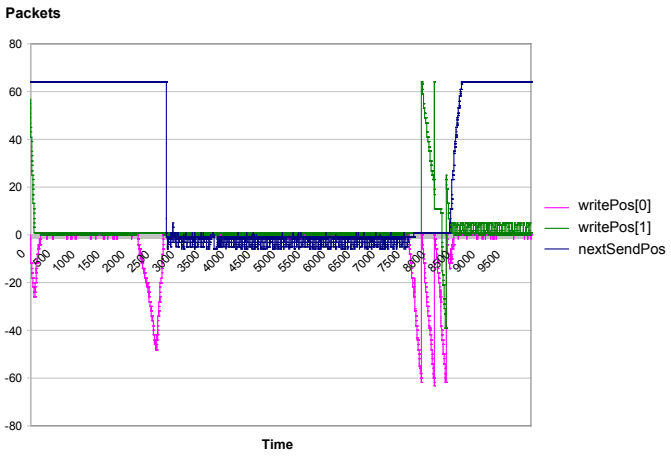


Figura 7. Evolución del búfer en $E2$.

En este caso el análisis es más complejo, ya que con el búfer interactúan tres elementos diferentes:

- `writePos[0]`: relacionado con el *ChannelStream* que escribe paquetes en el búfer del *hit* $E0$.
- `writePos[1]`: relacionado con el *ChannelStream* que escribe paquetes procedentes del *hit* $E1$.
- `nextSendPos`: relacionado con el *servent* que envía *chunks* a la máquina $E3$.

Podemos observar como al inicio de la sesión las dos fuentes empiezan a enviar paquetes. En un principio el subcanal 1 va un poco por delante del 0, pero enseguida se estabilizan las diferencias con `lastPos` entorno a los 0 paquetes. El primer pico negativo que encontramos en el instante 2,500 se debe a pequeños retrasos producidos por la entrada en el sistema del *hit* $E3$, y unos momentos después ya observamos como el *servent* empieza a actuar y el `nextSendPos` se actualiza con el valor de `lastPos`. A partir de aquí los tres elementos relacionados con la lectura y escritura de paquetes en el búfer continúan evolucionando al mismo ritmo. Como hemos comentado anteriormente, se ha introducido un tráfico interferente entre el *hit* $E1$ y el $E0$. Es por ello que durante unos segundos se dejan de recibir paquetes de esta máquina, de forma que el `lastPos` se mantiene constante y como consecuencia la función `writePos[0]` decrece hasta los -63 paquetes. En este punto se empiezan a reescribir paquetes del búfer y el `lastPos` se actualiza de acuerdo al último paquete escrito por el *hit* $E0$, por lo que se invierten los papeles, y ahora es la función de `writePos[1]` la que toma el valor de 63 debido a los paquetes que tiene pendientes de enviar, mientras que `writePos[0]` vuelve a valer 0. Aproximadamente en el instante 8,500 desaparece el tráfico interferente y la situación se vuelve a estabilizar en cuanto a los dos subcanales de entrada. También observamos como el `nextSendPos` relacionado con el *hit* $E0$ vuelve a los 63 paquetes una vez éste ha abandonado la sesión y, por lo tanto, dejamos de enviarle el flujo.

V. Conclusiones

La tecnología P2P permite nuevas oportunidades para definir una eficiente aplicación de *streaming* de video, pero al mismo tiempo trae una serie de desafíos técnicos debidos a los problemas que acarrea su carácter dinámico y heterogéneo. En este trabajo hemos presentado la aplicación NeuroCast, un sistema P2P no estructurado para *streaming* de vídeo que es capaz de adaptarse a las condiciones de una red inalámbrica. El diseño NeuroCast se basa en una arquitectura de malla no estructurada. De esta forma NeuroCast optimiza la asignación de nodos de acuerdo a su ancho de banda y combina estrategias de selección dinámica de nodos basándose en la información implícita de la recepción de datos.

El rendimiento de NeuroCast es evaluado en entornos emulados de redes inalámbricas. Los experimentos indican que NeuroCast es capaz de operar en un entorno hostil, teniendo en cuenta las condiciones de red. Nuestros resultados también confirman la capacidad de los sistemas no estructurados mallados para soportar altos niveles de desconexiones que pueden resultar del dinamismo de los usuarios y de la red (la rotación, los fracasos, la congestión, etc.)

Referencias

[1] H. Deshpande, M. Bawa, and H. Garcia-Molina. Streaming live media over a peer-to-peer network. Technical Report 2001-30, Stanford InfoLab, 2001.

[2] Peercast: P2p broadcasting for everyone. [online] http://www.peercast.org, accessed on January 2011.

[3] M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. Splitstream: High-bandwidth multicast in cooperative environments. *9th ACM Symposium on Operating Systems Principles*, 2003.

[4] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for largescale peer-to-peer systems. *IFIP/ACM Intl. Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany*, 2001.

[5] M. Hefeeda, A. Habib, D. Xu, B. Bhargava, and B. Botev. Collectcast: A peer-to-peer service for media streaming. *ACM Multimedia*, 11:68–81, 2003.

[6] Gnutella. [online] http://rfc-gnutella.sourceforge.net, accessed on January 2011.

[7] X. Zhang, J. Liu, B. Li, and T. Yum. Coolstreaming/donet: A data-driven overlay network for efficient peer-to-peer live media streaming. *Proceedings of IEEE Infocom*, 2005.

[8] A. Tirumala, M. Gates, F. Qin, J. Dugan, and J. Ferguson. Iperf: The (tcp/udp) bandwidth measurement tool, [online] http://iperf.sourceforge.net/, accessed on January 2011.

[9] Virtual network user-mode-linux (vnuml). [online] http://www.dit.upm.es/vnuml, accessed on January 2011.

[10] S. Avallone, S. Guadagno, D. Emma, A. Pescapè, and G. Ventre. D-itg distributed internet traffic generator. *International Conference on Quantitative Evaluation of Systems*, pages 316–317, 2004.

[11] S. Hemminger. Network emulation with netem. In *Linux Conf Au*, [online] http://linux-net.osdl.org/index.php/Netem, accessed on January 2011.

Evaluación de la configuración de clasificadores KNN para la detección de flujos P2P

F. Javier Salcedo-Campos, Jesús E. Díaz-Verdejo y Pedro García-Teodoro
Centro de Investigación TIC, Departamento de Teoría de la Señal, Telemática y Comunicaciones
Universidad de Granada
Dirección Postal 18071
fjsalc@ugr.es, jedv@ugr.es, pgteodor@ugr.es

***Resumen*—En los últimos años se ha producido un aumento de la popularidad de las redes y aplicaciones *peer-to-peer* (*P2P*), lo que se traduce en nuevos riesgos de seguridad para los usuarios y los nodos, así como nuevos escenarios en la gestión del tráfico de las redes. En este sentido existe un claro interés en la detección del tráfico *P2P* de la red sin acceder a la información contenida en el *payload* de los paquetes. Los clasificadores KNN se han mostrado muy efectivos para este fin, aunque no han sido estudiados en profundidad. El presente trabajo se centra en evaluar clasificadores KNN con diferentes configuraciones de distancia, número de vecinos más próximos y reglas de decisión para determinar si un flujo corresponde a un protocolo *P2P* o no, obteniéndose resultados superiores al 93% en la detección de flujos *P2P* y cercano al 97% en la precisión.**

***Palabras Clave*—Clasificación de tráfico, K-Nearest Neighbors, peer-to-peer**

I. Introducción

La identificación del tráfico de red trata de clasificar paquetes (identificación basada en paquetes), los flujos (identificación basada en flujos), o los nodos (identificación basada en nodos) en una red dada de acuerdo a los protocolos de aplicación asociados. Tradicionalmente esta tarea era sencilla, puesto que cada aplicación tenía asignados uno o varios puertos para realizar la comunicación. De este modo, para determinar el protocolo de aplicación, bastaba con realizar una simple inspección de la cabecera de la capa de transporte en busca de los puertos origen y destino de la comunicación. Sin embargo, esta situación ha cambiado debido a que muchas aplicaciones de Internet, incluyendo los protocolos *P2P* (acrónimo del término inglés *peer-to-peer*), están empleando técnicas de ofuscación de puertos, y otras como la encriptación y el tunelado [1], que hacen mucho más difícil su identificación. Por otra parte, la creciente popularidad y expansión de las redes y aplicaciones *P2P* plantean nuevos problemas de seguridad y de control del tráfico. En primer lugar, la capacidad de comunicación e intercambio de cualquier tipo de información entre los nodos que ejecutan aplicaciones *P2P*, en la mayoría de los casos de manena anónima, representa un riesgo para la seguridad de los usuarios y de la red. La información que se intercambia podría contener virus, gusanos y *malware* en general, que afecten a los nodos *P2P*. Por otra parte, las aplicaciones *P2P* son susceptibles de ser utilizadas coordinadamente para apoyar otras actividades perjudiciales como ataques DoS [3], *botnets* [4], etc.; lo que supone también un riesgo para la infraestructura de las redes. Además de los problemas relacionados con la seguridad, el uso cada vez más extendido de las aplicaciones *P2P* provoca la acaparación de las infraestructuras de red de los proveedores de Internet por el tráfico *P2P*. Por lo que necesitan desarrollar métodos que minimizen el impacto de este tipo de tráfico en el resto de los servicios de red.

Por ello, la identificación del tráfico de red en general, y del tráfico generado por aplicaciones *P2P* en particular, se está convirtiendo en un objetivo prioritario en el área de ingeniería de redes [2]. Los principales problemas que se plantean en la identificación del tráfico en una red son los siguientes:

1) La parametrización del tráfico: Es necesario extraer datos característicos del tráfico de red para poder representarlo y clasificarlo posteriormente. La información que se utiliza puede provenir de fuentes variadas, y ser de muy diverso tipo. Por ejemplo, puede comprender desde datos estadísticos de las conexiones a partir de informes de *routers* SNMP [5] (baja granularidad) hasta datos extraídos de cabeceras TCP, incluyendo los bits de señalización y los primeros *bytes* del *payload* (alta granularidad) [6].
2) El nivel de identificación: A partir del tráfico ya parametrizado, existe la posibilidad de realizar su identificación en tres niveles[7], [2]. La primera opción, basada en nodo, el objetivo es detectar los nodos que generan un determinado tipo de tráfico [8]. La segunda posibilidad es hacerla basada en flujo, cuyo objetivo sería clasificar a cada flujo por el protocolo de nivel de aplicación que lo produce. La tercera y última opción sería realizar la clasificación de cada paquete individualmente, que es la que se denomina basada en paquetes.
3) El proceso de identificación: Existen múltiples técnicas para llevar a cabo la identificación. Por citar algunos de los tipos de técnicas más importantes se podrían destacar las que emplean minería de datos o algoritmos de reconocimiento de patrones [11], [7] y las heurísticas o las basadas en detección de firmas [9], [2], [10].

Uno de los métodos de identificación de tráfico disponibles que ofrece mejores resultados, inspecciona la información de la capa de aplicación (*payload* en inglés) para encontrar patrones o firmas características de protocolos. Este método es el denominado DPI (del inglés *Deep Packet Inspection* o Inspección Profunda de Paquetes). No obstante, existen dos factores: rendimiento y privacidad que se consideran como requisitos importantes para las herramientas de red actuales y que DPI no satisface. Además DPI no es capaz de inspeccionar *payloads* cifrados, lo que está forzando a los desarrolladores a buscar soluciones alternativas en la identificación del tráfico *P2P*.

ISBN: 978-84-694-5948-5

Una de las técnicas que cumple a priori con los requisitos de rendimiento y privacidad en la identificación del tráfico es el clasificador KNN (del inglés *K-Nearest Neighbors*). Centrándonos en el nivel de identificación basado en flujo (la que se ha investigado en el presente trabajo), en lo que respecta al rendimiento, dado que los KNN se basan en determinar la similitud entre varios ejemplos de flujos conocidos y el flujo a reconocer, el proceso de clasificación es bastante eficiente, pues no se necesita realizar ningún tipo de entrenamiento. Por otra parte, es posible emplearlos sin invadir la privacidad de los usuarios, parametrizando los flujos únicamente con datos estadísticos y con información extraída de las cabeceras de los paquetes. Además, en la literatura se ha puesto de manifiesto que los clasificadores KNN ofrecen buenos resultados en la identificación de protocolos respetando la privacidad.

Jun demuestra en su trabajo que los KNN son la mejor técnica en términos de tasa de precisión [12]. Para ello realiza un estudio comparativo entre diferentes técnicas, como Naïve Bayes, árboles de decisión y otros métodos, incluyendo los KNN, para clasificar 12 tipos de protocolos de aplicación diferentes, entre los que se encuentran protocolos *P2P* (Bittorrent y Gnutella) y no *P2P* (HTTP, DNS, POP3, etc.). Otra muestra de la bondad de los clasificadores KNN queda patente en la contribución de Lim[13], donde se propone una discretización de los parámetros estándar que se extraen de los flujos (puertos, tamaños de los paquetes, número de paquetes, duración del flujo, etc.), y se evalúan 4 técnicas de clasificación: *support vector machines* (SVM), KNN, Naïve Bayes y árboles de decisión. Los resultados indican que KNN obtiene resultados similares a la mejor técnica cuando se aplica el método de discretización propuesto, con sólo alrededor de un 2% de precisión inferior a SVM, que consigue una tasa de precisión del 98%.

Por todos los motivos expuestos anteriormente, el presente trabajo ha consistido en realizar un estudio de la capacidad de clasificación de los KNN en la identificación de flujos *P2P* empleando diferentes configuraciones. Se han explorado 4 distancias distintas: cityblock, coseno, correlación y euclídea, además de un número varible de los vecinos más próximos; e incluso 3 reglas de distintas de decisión: mayoría, aleatoria y consenso para determinar si los flujos son *P2P* o no.

El artículo se organiza de la siguiente manera. En la Sección II se introducen los conjuntos de datos utilizados. En la Sección III, se presentan los fundamentos teóricos relacionados con los clasificadores KNN, y se introducen las medidas para determinar la efectividad de un clasificador. La Sección IV recoge los resultados experimentales que se han obtenido y se discuten. Finalmente, en la Sección V se presentan las conclusiones del trabajo.

II. Conjuntos de datos de experimentación

Se necesita una base de datos con ejemplos correctamente etiquetados para poder realizar una evaluación de las técnicas de clasificación de tráfico. Además, esta base de datos debe contener suficientes datos obtenidos de tráfico real para que sea representativa y pueda emplearse como referencia para determinar la exactitud de los resultados obtenidos. A esta base de datos de referencia se le denomina "ground truth". Por lo tanto, para evaluar el sistema propuesto se ha desarrollado un dispositivo experimental construido a partir de dos componentes principales: una base de datos de tráfico real capturado en una red académica, y una herramienta que proporciona los mejores resultados en la clasificación automática de los paquetes y flujos. Esta sección se organiza del siguiente modo: en el primer apartado se describe la herramienta de clasificación basada en openDPI. En el segundo se explica cómo se ha adquirido la base de datos y se muestran sus características, y finalmente en el tercero se detalla la parametrización que se ha aplicado a los flujos de la base de datos.

A. La herramienta openDPI

De cara a la realización del presente trabajo, se ha construido una herramienta basada en la biblioteca openDPI [14] que es capaz no sólo de identificar los protocolos de aplicación, sino que también realiza la diferenciación y el seguimiento de los paquetes en cada flujo. De esta manera, se proporcionan dos clasificaciones: basada en flujo y basada en paquetes. La herramienta funciona en modo de proceso por lotes, de modo que cuando se conoce el protocolo de un flujo, todos los paquetes desconocidos pertenecientes a dicho flujo se etiquetan con el protocolo con el que se ha identificado el flujo.

OpenDPI es una versión de domino público derivada de un producto comercial llamado PACE de Ipoque. El núcleo de openDPI es una librería software diseñada para clasificar tráfico de Internet en función de los protocolos de aplicación. En [15] los autores explican que la clasificación de protocolos de aplicación basada en DPI se consigue mediante la combinación de una serie de técnicas diferentes:

- Búsqueda de patrones, mediante el análisis de cadenas y patrones de *bytes* en cualquier parte del paquete, incluyendo el *payload*. De esta manera, openDPI busca firmas de protocolos conocidos.
- Análisis de comportamiento, mediante la búsqueda de patrones de comportamiento conocidos de una aplicación en el tráfico observado. Los datos utilizados incluyen el tamaño absoluto y relativo de los paquetes por flujo, la tasa de paquetes, y el número de flujos y la tasa de nuevos flujos por aplicación.
- El análisis estadístico, calculando algunos indicadores que pueden utilizarse para identificar los tipos de transmisión, como la media, la mediana y la variación de los valores utilizados en el análisis del comportamiento y la entropía de un flujo.

Por lo tanto, se puede afirmar que openDPI no es un producto DPI puro, pues además de estar basado en la detección de firmas de protocolos, también incorpora información de otras fuentes. De esta manera, la precisión de la clasificación es mayor, aunque deja algunos paquetes y flujos sin identificar, precisamente para evitar producir clasificaciones erróneas. La precisión de la clasificación, junto, con la disponibilidad y la calidad de las firmas, hizo que openDPI fuera seleccionada para realizar el etiquetado automático de la base de datos empleada en este trabajo.

Es necesario aclarar en este punto que para ser capaz de manejar paquetes UDP, se ha generalizado el concepto de flujo a través del uso de sesiones. Las sesiones se consideran definidas por el intercambio de información asociada a una

tupla (direcciones IP, puertos y el protocolo de transporte). De este modo, si el tráfico es TCP, una sesión puede ser identificada como un flujo TCP asumiendo que el número de los puertos efímeros utilizados por una determinada entidad no es mayor que 65535 en el período observado. No obstante, en este trabajo, se va a utilizar el término flujo para referirse a una sesión.

El número de paquetes y flujos que openDPI no es capaz de clasificar es su principal limitación. Además, los problemas de privacidad que su uso acarrea (inspecciona los payloads de los paquetes) y el alto coste computacional que implica la inspección profunda de paquetes; hacen que no sea una herramienta adecuada en la clasificación de tráfico en línea. Aunque, como ya se ha apuntado, su precisión en la clasificación sí lo hacen adecuado actualmente para la creación de bases de datos de referencia o "ground truth", puesto que en su versión actual es capaz de reconocer hasta 101 protocolos distintos, incluyendo los más comunes (HTTP, DNS, etc.) y varios tipos de protocolos *P2P*.

B. Descripción de las bases de datos

La base de datos de referencia o "ground truth" se ha construido mediante el análisis y la identificación de cada flujo y cada paquete con la herramienta realizada con las librerías de openDPI. Para obtener una base de datos lo suficientemente grande con tráfico real, se ha capturado tráfico durante 3 días hábiles en varios nodos de una red universitaria. Con el fin de poder controlar todo el tráfico de entrada y de salida de todos los nodos, la adquisición de datos se llevó a cabo en el *router* de acceso. De este modo, los flujos se han capturado completamente en ambos sentidos de la comunicación.

Para el presente trabajo, se han considerado dos subconjuntos de datos denominados S1 y S2, que contienen tráfico de diferentes nodos, para probar y validar el método, respectivamente. La Tabla I y las gráficas 1 y 2 muestran información sobre ambos conjuntos de datos. Los resultados proporcionados por la herramienta openDPI para la base de datos considerada indican que existe un gran número de flujos que no es capaz de etiquetar, el más del 50% en el conjunto S1 y casi el 80% en S2. Respecto a los flujos etiquetados destacan los pertenecientes al protocolo HTTP, que es el más utilizado en ambos conjuntos de datos, mientras que la proporción relativa de los protocolos *P2P* está por encima del 20% de los flujos. La mayor parte de los flujos *P2P* pertenecen a BitTorrent, mientras que Gnutella y otros están presentes en menor proporción. El resto de los flujos no *P2P* incluye sobre todo protocolos habituales, tales como DNS, SSL y protocolos de correo. No obstante la relación entre el tráfico *P2P* y *no-P2P* es similar entre ambos conjunto (ver Gráfica 3). En el presente trabajo, el conjunto S1 se utiliza para realizar el estudio de los KNN con diferentes distancias, vecinos más próximos y criterios de decisión, mientras que S2 se emplea para la validación de los resultados.

C. Extracción de parámetros de los flujos

La salida proporcionada por la herramienta desarrollada consiste en 3 listas: una con los flujos encontrados y su clasificación, otra con los paquetes y su clasificación y una tercera que relaciona los flujos y los paquetes de cada flujo. A partir de esta información, se realiza el proceso de parametrización,

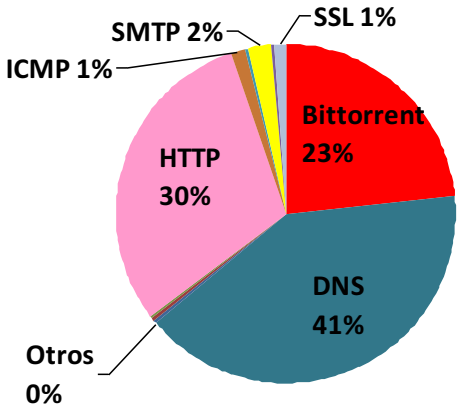


Fig. 1. Gráfico de distribución de los principales protocolos en el conjunto S1.

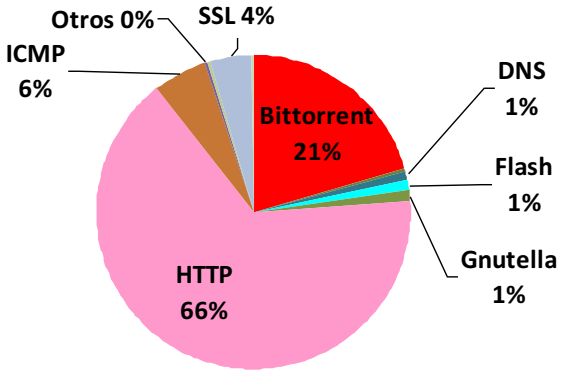


Fig. 2. Gráfico de distribución de los principales protocolos en los conjuntos de flujos S1 y S2.

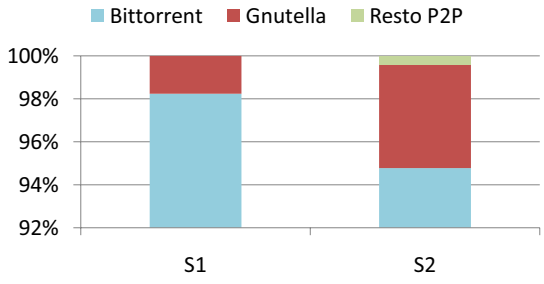


Fig. 3. Proporción comparativa de los protocolos P2P en los conjuntos S1 y S2.

Tabla I
ESTADÍSTICA BÁSICA DEL TRÁFICO EMPLEADO EN LOS EXPERIMENTOS.

	Flujos				
Conjunto	Total	Etiquetado	Flujos *P2P*	Flujos no *P2P*	Desconocidos
S1	70797	33524	8091	25433	37273
S2	107860	22645	4695	17950	85215
Total	178657	56169	12786	43383	122488

con el que se obtiene un vector de características con 62 componentes para cada flujo, como se muestra en la Tabla II. Los vectores contienen toda la la información necesaria para su tratamiento posterior, incluyendo una etiqueta de identificación del flujo (FLOW_ID), el protocolo que ha detectado openDPI e información básica sobre el flujo (tupla de flujo). Las direcciones IP de cada flujo se han ordenado considerándolas como enteros (en representación de red) y, por tanto, los dos sentidos que pueden tener los paquetes se tienen en cuenta en la parametrización: UP (ascendente) indica que los paquetes van de la IP baja hacia la IP alta, y DOWN (descendente) para el sentido opuesto.

Los valores que se han considerado en cada vector de parámetros son medidas estadísticas básicas relacionadas con las propiedades del flujo, la mayoría de ellas separadas en medidas totales, ascendentes (_UP) y descendientes (_DOWN). Los parámetros empleados son los que habitualmente se incluyen en la literatura: tamaño medio de los paquetes, medidas de tiempo y duración de los flujos, número de paquetes, etc. No obstante, se ha incluido una descripción más detallada a nivel temporal y de señalización (por ejemplo, los tiempos entre llegadas y el número de paquetes URG).

Los valores de los parámetros han sido obtenidos de la lista de paquetes de un flujo analizando sólo su tamaño, marcas de tiempo, señalización TCP y la dirección de los paquetes. De este modo sólo se han analizado las cabeceras TCP/UDP, dejando sin inspeccionar el *payload*. Por tanto, se preserva la privacidad de los usuarios en la capa de aplicación. La complejidad de la evaluación es baja, puesto que sólo se calculan o extraen valores mínimos, máximos, medias y conteos según sea el parámetro considerado. En el presente trabajo se utilizan todos los parámetros relacionados con la señalización, el tiempo y la transferencia; y a éstos sólo se añaden los puertos asociados a cada una de las direcciones IP entre las que se establece el flujo. De este modo, un flujo quedará representado por un vector de 53 parámetros.

III. FUNDAMENTOS TEÓRICOS

En el presente trabajo, la clasificación de los flujos se realiza mediante clasificadores KNN con diferentes configuraciones de vecinos, distancias y reglas para determinar la clase de los flujos test. Por lo tanto, es necesario describir algunos conceptos básicos sobre los KNN junto con las medidas más habituales que se usan para medir la efectividad de un clasificador.

A. Clasificación a partir de los K vecinos más cercanos o KNN

El algoritmo de clasificación por los K vecinos más cercanos, o KNN, es un método de clasificación de objetos basado en la cercanía a los ejemplos más cercanos en un espacio de características [17]. Es uno de los algoritmos más simples de

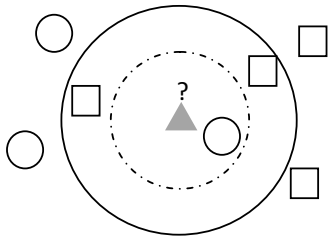


Fig. 4. Ejemplo de clasificación con KNN: $K = 1$, (círculo grande con línea discontinua), $K = 3$ (círculo mayor con línea continua).

clasificación: un objeto es clasificado en función de los votos de sus vecinos: esto es, la clase que se le asigna al objeto es la clase más común de entre sus K vecinos más cercanos. Supongamos que queremos clasificar el triángulo gris como un círculo o un cuadrado en el espacio mostrado en la Figura 4. Si $K = 1$, será clasificado como un círculo, porque su objeto más cercano es un círculo. Sin embargo, será clasificado como un cuadrado si $K = 3$, ya que dos de los tres objetos más cercanos son cuadrados.

El valor óptimo de K depende de los datos que se tengan. Valores más altos de K en general suelen reducir el efecto del ruido en la clasificación, pero, sin embargo, deja menos claros los límites entre clases distintas. Un valor adecuado de K puede ser seleccionado mediante diversas técnicas heurísticas, por ejemplo, la validación cruzada. Los ejemplos de entrenamiento son vectores en un espacio de características multidimensional etiquetados correctamente con su clase. Por lo tanto, la fase de entrenamiento del algoritmo consiste sólo en almacenar los vectores de características y las etiquetas de las muestras de entrenamiento, que en nuestro caso será ser P2P o no. En la fase de clasificación, K es una constante definida por el usuario. Un vector observado se clasifica mediante la asignación de la clase más frecuente, de entre las K muestras más cercanas a ese vector. Existen algunas variaciones en la toma de decisión de la clase del vector test [18]. Las 3 reglas evaluadas evaluadas en el presente trabajo han sido las siguientes:

1) Mayoría: La clase se elige igual a la de la mayoría de los vecinos más próximos según la distancia escogida. Si el valor de K es par, en caso de empate, se elige como clase la del vecino más cercano de entre las dos clases para romper el empate.
2) Aleatoria: Es igual que la anterior regla, sólo que en caso de empate se elige aleatoriamente la clase de entre los dos grupos de vecinos más cercanos.
3) Consenso: Es la más restrictiva, pues sólo asigna una clase a aquellos puntos cuyos K vecinos más próximos pertenecen a la misma clase, dejando el resto sin clasificar. De este modo se trata de asegurar que la clase

Tabla II
COMPONENTES DE LOS VECTORES DE PARÁMETROS DE CADA FLUJO.

Valor	Descripción
Identificación de flujos	
N_PROT	Numero de protocolos detectado
IP_LOW	Dirección IP menor el la tupla de la sesión
IP_UPPER	Dirección IP mayor el la tupla de la sesión
PORT1	Puerto asociado a la menor IP (IP_LOW)
PORT2	Puerto asociado a la mayor IP (IP_UPPER)
PROT_UDP	Protocolo de transporte UDP
PROT_TCP	Protocolo de transporte TCP
PROT_UNK	ICMP
DIR	Dirección del primer paquete observado
FIRST_TIME	Marca de tiempo del primer paquete (μs)
LAST_TIME	Marca de tiempo del último paquete (μs)
Relacionados con la transferencia	
NPACKETS	Número de paquetes en el flujo
NPACKETS_UP	Idem dirección hacia arriba
NPACKETS_DOWN	Idem dirección hacia abajo
PACKETS_SIZE	Tamaño total de los paquetes intercambiados
PACKETS_SIZE_UP	Idem dirección hacia arriba
PACKETS_SIZE_DOWN	Idem dirección hacia abajo
PAYLOAD_SIZE	Tamaño total de los payloads
PAYLOAD_SIZE_UP	Idem dirección hacia arriba
PAYLOAD_SIZE_DOWN	Idem dirección hacia abajo
MEAN_PACK_SIZE	Tamaño medio de los paquetes
MEAN_PACK_SIZE_UP	Idem dirección hacia arriba
MEAN_PACK_SIZE_DOWN	Idem dirección hacia abajo
SHORT_PACKETS	Número de paquetes cortos
SHORT_PACKETS_UP	Idem dirección hacia arriba
SHORT_PACKETS_DOWN	Idem dirección hacia abajo
LONG_PACKETS	Número de paquetes largos
LONG_PACKETS_UP	Idem dirección hacia arriba
LONG_PACKETS_DOWN	Idem dirección hacia abajo
MAXLEN	Tamaño máximo de los paquetes
MAXLEN_UP	Idem dirección hacia arriba
MAXLEN_DOWN	Idem dirección hacia abajo
MINLEN	Tamaño mínimo de los paquetes
MINLEN_UP	Idem dirección hacia arriba
MINLEN_DOWN	Idem dirección hacia abajo
Relacionados con el tiempo	
DURATION	Duración del flujo (μs)
MEAN_INTERAR	Tiempo medio entre paquetes consecutivos
MEAN_INTERAR_UP	Idem sólo para paquetes hacia arriba
MEAN_INTERAR_DOWN	Idem sólo para paquetes hacia abajo
MAX_INTERAR	Tiempo máximo entre paquetes consecutivos
MAX_INTERAR_UP	Idem sólo para paquetes hacia arriba
MAX_INTERAR_DOWN	Idem sólo para paquetes hacia abajo
MIN_INTERAR	Tiempo mínimo entre paquetes consecutivos
MIN_INTERAR_UP	Idem sólo para paquetes hacia arriba
MIN_INTERAR_DOWN	Idem sólo para paquetes hacia abajo
Señalización	
N_SIGNALING	Número de paquetes con flags
N_SIGNALING_UP	Idem dirección hacia arriba
N_SIGNALING_DOWN	Idem dirección hacia abajo
NACKS	Número de paquetes con ACK activo
NFIN	Idem FIN
NSYN	Idem SYN
NRST	Idem RST
NPUSH	Idem PSH
NURG	Idem URG
NECE	Idem ECE
NCWD	Idem CWD
NACK_UP	Número de paquetes hacia arriba con ACK activo
NACK_DOWN	Idem dirección hacia abajo
NFIN_UP	Idem FIN & UP
NFIN_DOWN	Idem FIN & DOWN
NRST_UP	Idem RST & UP
NRST_DOWN	Idem RST & DOWN

elegida para el objeto observado sea la correcta.

Por lo general la distancia euclídea es la que se utiliza como métrica para determinar la proximidad de dos objetos en el espacio de características. Sin embargo, pueden aplicarse otras medidas de distancia diferentes. En nuestro caso, se han empleado cuatro: euclídea, cityblock, coseno y correlación [18]. Por lo tanto, entre dos vectores de dimensión M $X = (x_1, x_2, ..., x_M)$ e $Y = (y_1, y_2, ..., y_M)$, se pueden calculan las cuatro distancias del siguiente modo:

- Euclídea:

$$d_{euc}(X,Y) = \sqrt{\sum_{i=1}^{M}(x_i^2 - y_i^2)}$$

- Cityblock:

$$d_{cit}(X,Y) = \sum_{i=1}^{M}|x_i - y_i|$$

- Coseno:

$$d_{\cos}(X,Y) = 1 - \frac{\sum_{i=1}^{M} x_i y_i}{\sqrt{\sum_{i=1}^{M} x_i^2}\sqrt{\sum_{i=1}^{M} y_i^2}}$$

- Correlación:

$$d_{cor}(X,Y) = 1 - \frac{1}{M}\sum_{i=1}^{M}\left(\frac{x_i - \bar{x}}{\sigma_x}\right)\left(\frac{y_i - \bar{y}}{\sigma_y}\right)$$

donde $\bar{x}$ e $\bar{y}$ son la media de los componentes de X e Y, y σ_x e σ_y sus desviaciones estándar que se calculan a través de las siguientes expresiones:

$$\sigma_x = \sqrt{\frac{1}{M-1}\sum_{i=1}^{M}(x_i - \bar{x})^2}$$

$$\sigma_y = \sqrt{\frac{1}{M-1}\sum_{i=1}^{M}(y_i - \bar{y})^2}$$

B. Medidas de eficiencia en clasificadores

Se han empleado tres conocidas medidas en el ámbito de los clasificadores con el objetivo de comparar los resultados obtenidos [16]: el porcentaje de verdaderos positivos (VP), porcentaje de verdaderos negativos (VN) y la precisión de la clasificación (PC).

Denominando N_{P2P} y N_{otro} al número total de flujos P2P y no P2P, respectivamente. Considerando que $n_{X\to Y}$ es el número de flujos de la categoría X –no-P2P, $otro$, o P2P, $P2P$– clasifiacdos como pertenecientes a la categoría Y ($otro$ o $P2P$). Las medidas de eficiencia mencionadas anteriormente pueden ser definidas en nuestro caso como se indica a continuación:

1) **VP, Verdaderos positivos**: el porcetaje de flujos P2P correctamente clasificados como P2P respecto al número total de flujos P2P.

$$VP = \frac{n_{P2P\to P2P}}{N_{P2P}} \cdot 100 \quad (1)$$

2) **VN, Verdaderos negativos**: el porcentaje de flujos no P2P clasificados correctamente respecto al número total de flujos no P2P.

$$VN = \frac{n_{otro\to otro}}{N_{otro}} \cdot 100 \quad (2)$$

3) **PC, Precisión de la clasificación**: el porcentaje de flujos correctamente clasificados en relación al número total de flujos.

$$PC = \frac{n_{P2P\to P2P} + n_{otro\to otro}}{N_{P2P} + N_{otro}} \cdot 100 \quad (3)$$

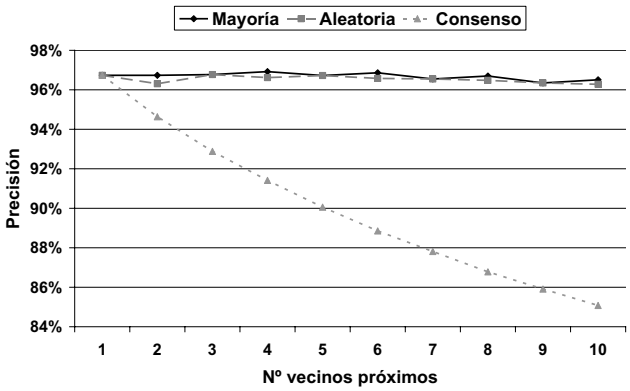


Fig. 5. Tasas de precisión obtenidas por los KNN en función del número de vecinos más próximos elegidos y del criterio de decisión con la distancia Cityblock.

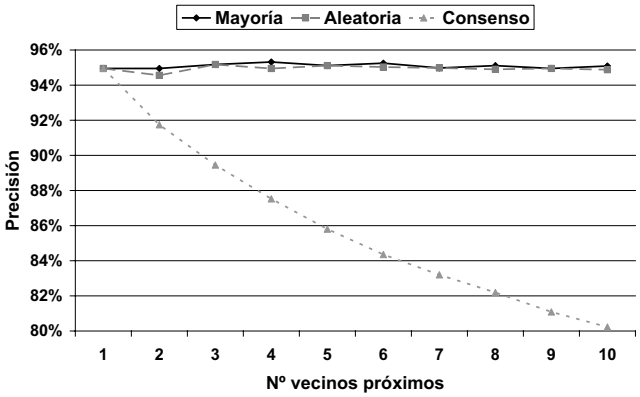


Fig. 6. Tasas de precisión obtenidas por los KNN en función del número de vecinos más próximos elegidos y del criterio de decisión con la distancia Coseno.

El sistema ideal debería proporcionar en estas tres medidas: VP, VN y PC un valor del 100%.

IV. Resultados experimentales

Los experimentos se han realizado a partir de los flujos parametrizados con 53 datos estadísticos explicados en el Apartado 2.3 de los conjuntos S1 y S2 de la base de datos.

De cara a mejorar la confianza de los resultados que se obtienen en la fase de experimentación, se ha llevado a cabo una validación cruzada (en inglés *leave-one-out*), de modo que los flujos etiquetados de S1 se han particionado en 10 subconjuntos o particiones aleatorias con el mismo número de flujos y la misma proporción entre flujos P2P y no P2P [19]. Después se toman 9 particiones con sus flujos etiquetados para reconocer la partición restante. De este modo, el procedimiento se repite 10 veces, hasta que todas las particiones son usadas en el reconocimiento. Los resultados se promedian sobre el conjunto completo de experimentos. Para incrementar aún más la confianza de los resultados, la asignación de los flujos a las particiones se ha realizado de manera aleatoria.

El presente apartado recoge en primer lugar los resultados obtenidos en la clasificación de flujos con diferentes configuraciones de vecinos, distancias y reglas del clasificador KNN sobre el conjunto S1; para posteriormente exponer los resultados de la validación sobre S2 en el segundo subapartado.

A. Exploración inicial de los parámetros de configuración

Se han realizado cuatro experimentos para determinar cómo cambia la efectividad del clasificador KNN cuando se modifica alguno de sus parámetros de configuración como la distancia, el número de vecinos más próximo y el criterio de asignación de la clase. En cada uno de estos 4 experimentos se ha fijado la distancia empleada, mientras que se ha variado el número de vecinos más próximos (de 1 a 10) y el criterio de selección de la clase: mayoría, aleatorio o consenso. De este modo se han obtenido las Figuras 5, 6, 7 y 8.

En lo que respecta a las reglas de decisión, en las gráficas se puede observar que, a pesar de lo estricta que es la regla de consenso, se consiguen resultados superiores al 80% en la precisión de la clasificación. Estas condiciones son aún más restrictivas cuando se aumenta el número de vecinos más

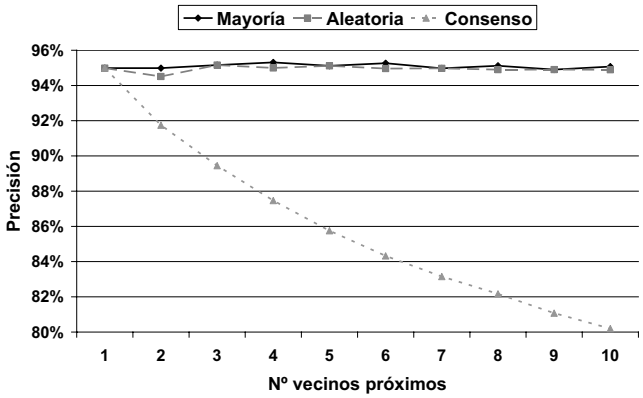


Fig. 7. Tasas de precisión obtenidas por los KNN en función del número de vecinos más próximos elegidos y del criterio de decisión con la distancia Correlación.

cercanos para asignar la clase, por ello el mínimo se obtiene siempre para $K = 10$. Por otra parte, las reglas aleatoria y por mayoría ofrecen parecidos resultados en cada una de las 4 distancias evaluadas, porque no hay diferencia en la decisión cuando K es impar. Sólo se observan pequeñas diferencias a favor de la regla por mayoría para valores de K pares, indicando que si existe empate entre las clases de los vecinos más cercanos, es más eficiente mirar la clase del siguiente

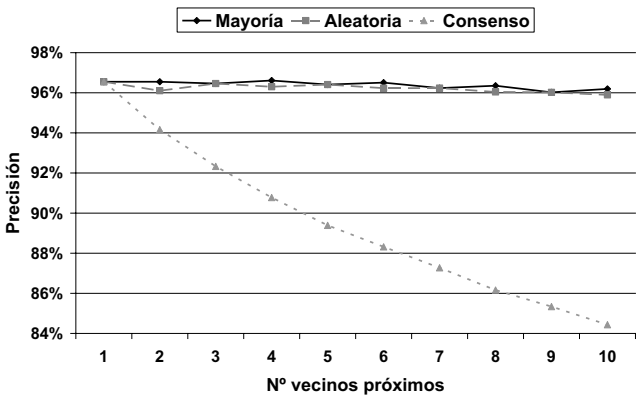


Fig. 8. Tasas de precisión obtenidas por los KNN en función del número de vecinos más próximos elegidos y del criterio de decisión con la distancia Euclídea.

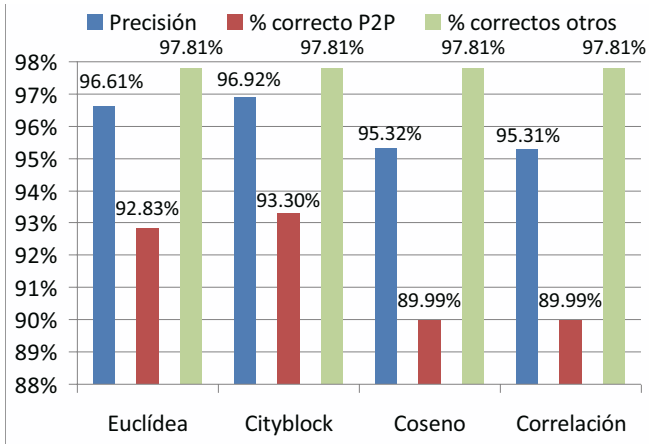


Fig. 9. Precisión y tasa de reconocimiento correcto de cada clase (VP y VN) para cada una de las distancias consideradas usando la regla por mayoría y los 4 vecinos más próximos.

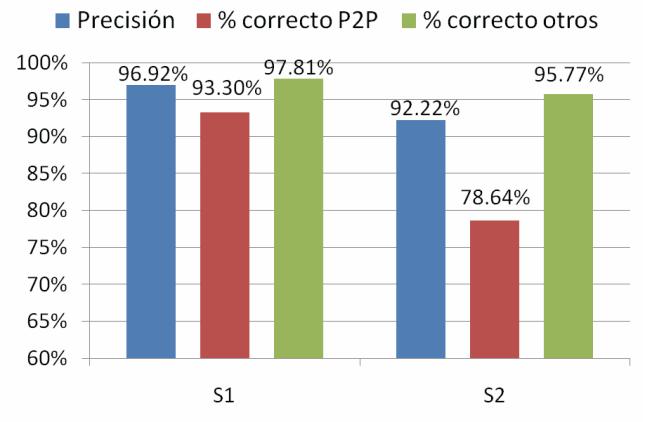


Fig. 10. Precisión y tasa de reconocimiento correcto de cada clase (VP y VN) en los conjuntos de flujos S1 y S2 usando la distancia cityblock y la regla de decisión por mayoría de los 4 vecinos más próximos.

más próximo para tomar la decisión que hacerlo de manera aleatoria.

Atendiendo al número de vecinos más próximos, de los experimentos se ha obtenido que el número óptimo de vecinos más próximos con los que determinar la mejor clase de los flujos es 4, y que es independiente de la distancia que se utilice. No obstante, como puede apreciarse en las gráficas, no existe una diferencia significativa entre distintos valores de K, y ya se obtienen resultados muy similares al máximo con sólo emplear el vecino más próximo.

Las gráficas indican también que las mejores distancias, en cuanto a precisión, son la euclídea y la cityblock. Para observar mejor las diferencias entre las distancias utilizadas, se ha realizado la Figura 9, en la que se representan los resultados de reconocimiento (PC, VP y VN) emplando la mejor regla (por mayoría) y el mejor número de vecinos más proximos (cuatro).

A la vista de la Figura 9, la mejor distancia es la Cityblock, por la detección individual sobre P2P, porque la detección del tráfico no P2P es la misma para las 4 distancias. El diferente rendimiento de los clasificadores KNN con distintas distancias se basa exclusivamente en la capacidad de reconocimiento de los flujos P2P.

B. Validación de los resultados

Para validar los resultados se ha empleado la mejor configuración del clasificador KNN obtenida sobre el conjunto S1, distancia cityblock con la decisión de la clase por mayoría de los 4 vecinos más próximos, sobre el conjunto de datos S2 y se comparan los resultados. La Figura 10 muestra los resultados que se han obtenido del experimento. En ella se aprecia que hay un descenso en la tasa de precisión de un 4,7% motivado sobre todo por la menor capacidad de detección de flujos P2P, que descende casi un 15%. Hay dos razones que podrían explicar este comportamiento. La primera puede ser la dependencia de la clasificación que realizan los KNN con los nodos. Recordemos que los conjuntos S1 y S2 contienen tráfico de muchos nodos distintos y que la clasificación que hacen los KNN se basan en la similitud del flujo respecto a otros ejemplos conocidos. La manera de realizar la clasificación también explica que se produzcan diferencias al clasificar S1 y S2 puesto que la proporción de protocolos en los flujos P2P y no P2P no es igual en ambos conjuntos. Centrándonos en los protocolos P2P, en S2, el 1% del tráfico total (4,8% del P2P) es Gnutella y el resto casi todo es Bittorrent (21% del total y 94,8% del tráfico P2P), mientras que en S1 el tráfico P2P es prácticamente todo Bittorrent (98,2% del P2P) y menos de 1,8% es Gnutella. Estas diferencias son aún mayores si atendemos a los principales protocolos no P2P como HTTP y DNS (Ver figuras 1 y 2).

V. Conclusiones

En el presente trabajo se ha realizado un estudio sobre la mejor configuración de los clasificadores KNN para detectar flujos P2P. Se ha determinado que parametrizando los flujos con 53 valores característicos relacionados con la señalización, el tiempo y la transferencia, incluyendo los puertos; lo más efectivo es usar la distancia cityblock y la regla de decisión por mayoría de los 4 vecinos más próximos para detectar flujos P2P con un clasificador KNN. Los resultados que se han obtenido son prometedores, puesto que en ningún caso la tasa de precisión ha resultado ser inferior al 80%, siendo prácticamente el 97% en la configuración óptima. A pesar de ello, la configuración óptima de los KNN parece sensible a la distribución de protocolos y a los nodos que generan el tráfico. Por este motivo, parece necesario trabajar en dos direcciones: primero en la mejora de la selección y el tratamiento de las variables con las que se parametrizan los flujos. Y en segundo lugar en la mejora de la eficacia de los KNN, explorando distancias más adecuadas al contexto y a los parámetros que se extraen de los flujos. En este sentido también podrían explorarse otros modos de aplicar los KNN sobre los vectores de parámetros, como por ejemplo, realizar el reconocimiento en varias fases. Las variables podrían agruparse en conjuntos mediante algún criterio y aplicar los KNN sobre cada conjunto de manera independiente. De este modo se obtendrían varios resultados o estimaciones que podrían emplearse posteriormente en un segundo nivel de decisión para clasificar los flujos de muestra.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el MICINN a través del proyecto TEC2008-06663-C03-02.

REFERENCIAS

[1] T.T.T. Nguyen, G. Armitage, "A survey of techniques for internet traffic classification using machine learning" *IEEE Communications Surveys & Tutorials*, vol. 10, num. 4, pp. 56-76, 2008.

[2] A. Callado, C. Kamienski, G. Szabo, B.P. Gero, J. Kelner, "A Survey on Internet Traffic Identification", *IEEE Communications Surveys & Tutorials*, vol. 11, n. 3, pp. 37-52, 2009.

[3] M. Qi, "P2P Network-Targeted DDoS Attacks", *Second International Conference on Applications of Digital Information and Web Technologies, ICADIWT '09*, pp. 843-845, 4-6 Aug. 2009.

[4] S. Chang, L. Zhang, Y. Guan, T. E. Daniels; " A Framework for P2P Botnets ", 2009 International Conference on Communications and Mobile Computing", 2009 International Conference on Communications and Mobile Computing

[5] S. Sen and J. Wang, "Analyzing Peer-to-Peer TrafficAcross Large Networks," *IEEE/ACM Transactions on Networking*, vol. 12, n. 2, pp. 219-232, 2004.

[6] Madhukar, A., Williamson, C., "A Longitudinal Study of P2P Traffic Classification", *Proc. of Int. Symposium on Modeling, Analysis and Simulation*, pp. 179-188, 2006.

[7] R. Keralapura, A. Nucci, and C. Chuah, "A Novel Self-Learning Architecture for *P2P* Traffic Classification in High Speed Networks," *Computer Networks*, vol. 54, pp. 1055-1068, 2010.

[8] Xuan-min, L., Jiang, P., Ya-jian, Z., *"A New P2P Traffic Identification Model Based on Node Status"*, In Int. Conference on Mangement and Service Science, pp. 1-4, 2010.

[9] X. Li and Y. Liu, "A *P2P* Network Traffic Identification Model Based on Heuristic Rules,". *Int. Conference on Computer Application and System Modeling*, vol. 5, pp. 177-179, 2010.

[10] W. JinSong, Z. Yan, W. Qing, and W. Gong, "Connection Pattern-based *P2P* Application Identification Characteristic," *Proc. of Int. Conference on Network and Parallel Computing Workshops*, pp. 437-441, 2007.

[11] M. Soysal and E.G. Schmidt, "Machine Learning Algorithms for Accurate Flow-Based Network Traffic Classification: Evaluation and Comparison," *Performance Evaluation*, vol. 67, n. 6, pp. 451-467, 2010.

[12] L. Jun, Z. Shunyi, L. Yanqing, and Z. Zailong, "Internet traffic classification using machine learning," *Second International Conference on Communications and Networking in China (CHINACOM'07)*, pp 239-243, 2007.

[13] Y. Lim, H. Kim, J. Jeong, C. Kim, T.T. Kwon, and Y. Choi, "Internet traffic classification demystified: on the sources of the discriminative power", *Proceedings of the 6th International Conference On Emerging Networking Experiments And Technologies (CoNEXT'10)*, 2010.

[14] OpenDPI, 2011. Available at http://www.opendpi.org

[15] Mochalski, K., Schulze, H., "Deep Packet Inspection. Technology, applications & net neutrality", White Paper, 2009. Available at http://www.ipoque.com/resources/white-papers

[16] Gomez, J.M., Puertas, E., Maña, M.J., "Evaluating cost-sensitive unsolicited bulk email categorization", *Proc. of the ACM symposium and applied computing*, ACM Press, pp. 615-620, 2002.

[17] Duda, R.O., Hart, P.E., Stork, D.G., "Pattern Classfication", Ed. John Wiley & Sons, 2001.

[18] T. Mitchell, "Machine Learning", Ed. McGraw-Hill, 1997.

[19] Kohavi, R., "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection ", *Proc. of the 14th International Joint conference on Artifcial Intelligence*, Montreal, Canada, 1995.

Nuevas heurísticas para la detección de nodos y flujos eDonkey

Rafael A. Rodríguez-Gómez, Gabriel Maciá-Fernández, Pedro García-Teodoro
Departamento de Teoría de la Señal, Telemática y Comunicaciones,
CITIC - Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación, Universidad de Granada
C/Periodista Daniel Saucedo Aranda s/n E-18071
rodgom@ugr.es, gmacia@ugr.es, pgteodor@ugr.es

***Resumen*—El uso de aplicaciones basadas en redes peer-to-peer (P2P) ha experimentado un incremento exponencial, lo que ha provocado que el volumen de tráfico generado por éstas llegue a suponer alrededor de un 80% de todo el ancho de banda de la red. Por este motivo, el interés de los proveedores de servicio de Internet (ISPs) por clasificar este tráfico ha aumentado también de forma considerable.**

En este contexto, en el presente artículo se describen dos algoritmos de detección del protocolo eDonkey. El primero de ellos tiene como objeto de detección los flujos del protocolo y está basado en la hipótesis de que son flujos del protocolo eDonkey aquellos en los que el cliente que inicia la conexión envía sustancialmente más información que la que recibe. El segundo algoritmo ha sido desarrollado para detectar nodos generadores de tráfico eDonkey, basándose en la hipótesis de que son nodos generadores de tráfico eDonkey aquellos cuya tasa de subida es constante conectándose a múltiples IPs.

Ambos algoritmos de detección han sido probados en tres conjuntos de trazas. Como resultado, se ha comprobado que las hipótesis utilizadas en los algoritmos de detección son ciertas en el protocolo eDonkey. Adicionalemente, los experimentos muestran que los algoritmos propuestos tienen una elevada tasa de reconocimiento y una baja tasa de falsos positivos.

***Palabras Clave*—Clasificación de tráfico, detección de flujos, detección de nodos, P2P, eDonkey**

I. Introducción

El desarrollo y el uso de aplicaciones que utilizan redes P2P para comunicarse han experimentado un crecimiento exponencial en los últimos años. Encontramos en la actualidad múltiples ejemplos de éstas como pueden ser: eMule o uTorrent, como aplicaciones de compartición de ficheros, Skype, como aplicación de voz sobre IP, y Spotify, para la compartición de flujos de audio.

El tráfico generado por las aplicaciones P2P supone un enorme consumo del ancho de banda de la red; de hecho, Callado et al. [1] aseguran que el volumen de tráfico generado por aplicaciones P2P supone un 80% de todo el ancho de banda de red. Este consumo de ancho de banda conlleva la aparición de retardos en las comunicaciones y, en suma, una disminución en la calidad de los servicios proporcionados.

La capacidad de clasificar el tráfico P2P es una tarea de altísimo valor para los proveedores de servicio de Internet (IPSs), que ven perdido el control de su tráfico y que se ven forzados a incrementar las operaciones de mantenimiento debido a este crecimiento en el uso de las redes P2P [2]. La disponibilidad de mecanismos eficientes y eficaces de clasificación de tráfico permitirá a los ISPs definir políticas de gestión de tráfico, ofreciendo diferentes clases de servicio y aplicando a cada una de ellas una conformación del tráfico que dependa, por ejemplo, de las aplicaciones utilizadas por el usuario.

Los métodos de clasificación de tráfico pueden dividirse actualmente en tres grupos: *(i)* basados en el puerto, *(ii)* basados en el contenido de los paquetes, y *(iii)* basados en las características estadísticas de los flujos. Las aplicaciones P2P pueden utilizar cualquier puerto para comunicarse y encriptar el contenido de sus mensajes, lo que dificulta enormemente su clasificación mediante los dos primeros tipos de técnicas.

El presente trabajo se centrará, pues, en la detección basada en las características de los flujos. El protocolo a detectar es eDonkey, protocolo de comunicación para redes P2P que sigue representando, hoy por hoy, un porcentaje considerable del volumen de tráfico de Internet, y que es utilizado principalmente en aplicaciones de compartición de ficheros como eMule o aMule.

Se proponen aquí dos heurísticas para la detección de tráfico eDonkey, que son el resultado de un estudio detallado del comportamiento de este protocolo: detección de nodos basada en la tasa de subida y detección de flujos basada en la inversión del sentido de la descarga.

La detección de nodos basada en la tasa de subida se fundamenta principalmente en dos asunciones: (*i*) los usuarios limitan la tasa de subida de las aplicaciones que utilizan eDonkey, y (*ii*) la tasa de subida tiene un comportamiento constante en ciertos periodos temporales alrededor de este límite establecido por el usuario. De esta forma, si la tasa de subida de un nodo tiene un comportamiento constante en diferentes periodos temporales alrededor de un mismo valor, este nodo será detectado como nodo generador de tráfico eDonkey.

La detección de flujos propuesta se basa en la particularidad de que, en las aplicaciones de compartición de archivos que utilizan el protocolo eDonkey para comunicarse, el nodo que inicia la conexión es el nodo que envía el grueso de la información. Esto es radicalmente opuesto a lo que sucede en el paradigma de cliente-servidor, donde es el cliente quien inicia la conexión y el servidor quien envía la información requerida. Por tanto, si el sentido del envío de la información en un flujo es el contrario al común, es decir, la información viaja del nodo que inicia la conexión al que la recibe, este flujo será detectado como perteneciente al protocolo eDonkey.

El resto del artículo se divide como sigue: En la Sección II se presentan los trabajos relacionados con la clasificación de tráfico P2P, indicando la aportación aquí pretendida frente a ellos. En la Sección III se presentan conceptos generales de las redes P2P, junto a una descripción del funcionamiento

ISBN: 978-84-694-5948-5

general del protocolo eDonkey. Las heurísticas propuestas para la detección del protocolo eDonkey se detallan en la Sección IV, mostrándose en la Sección V el entorno de experimentación considerado. Por su parte, en la Sección VI se presentan y discuten los resultados experimentales obtenidos con las propuestas realizadas. Finalmente, en la Sección VII se exponen las principales conclusiones que se pueden extraer de este trabajo.

II. Trabajos relacionados

Los métodos de clasificación de tráfico existentes en la literatura pueden dividirse en tres: basados en el puerto, basados en el contenido de los paquetes y basados en las características de los flujos. Como aseguran los autores de [3], los métodos de clasificación de tráfico basados en puertos conocidos no resultan válidos actualmente en la detección de tráfico P2P y en cuanto a los basados en el contenido de los paquetes, su uso incurre en problemas legales relativos a la privacidad, lo que reduce su ámbito de aplicación enormemente.

Existe una gran cantidad de trabajos en los que se propone una clasificación basada en las características de los flujos. Por ejemplo, el trabajo de Moore y Zuev [4] utiliza análisis bayesiano para este fin. Las características utilizadas para esta clasificación son: puertos TCP, duración de los flujos, estadísticas del tiempo entre paquetes, estadísticas del tamaño de los paquetes y la transformada de Fourier del tiempo entre paquetes. Una particularidad de este trabajo es que solamente se utilizan flujos TCP completos, es decir, que contienen el establecimiento y cierre de conexión propios de este protocolo. En esta línea se puede destacar también BLINC [5], una herramienta de clasificación que asocia un equipo final con la aplicación que genera la mayor parte de su tráfico. Es decir, asocia los equipos finales con los servicios que ofrecen o utilizan, en lugar de analizar cada flujo individualmente. De forma similar a BLINC, una de las dos heurísticas presentadas en este trabajo se basa en la detección de nodos generadores de tráfico P2P.

Otros trabajos, en lugar de utilizar una única metodología para clasificar todos los protocolos existentes, clasifican únicamente un subconjunto de protocolos. Esta es la aproximación más frecuente para clasificar flujos de los protocolos utilizados en redes P2P. En esta línea se encuentra [3], el primer trabajo que intenta clasificar tráfico proveniente de aplicaciones P2P en puertos arbitrarios sin inspeccionar el cuerpo de los paquetes. Para esto se utilizan dos heurísticas: *(i)* En la primera se seleccionan las parejas IP origen y destino que se comunican utilizando tanto TCP como UDP, y *(ii)* la segunda se basa en los patrones de conexión de las parejas IP-puerto de las aplicaciones P2P. Básicamente, cuando un nodo P2P inicia una conexión con un nodo A, el puerto de destino será el puerto definido por el usuario del nodo A para atender las peticiones de la aplicación P2P. Las dos heurísticas de detección propuestas en nuestro trabajo también son capaces de clasificar el tráfico P2P encriptado en puertos arbitrarios.

Xu et al. [6] proponen un método para identificar tráfico P2P basándose en el comportamiento de la transferencia de datos de las aplicaciones P2P. Los autores aseguran que los datos descargados por un nodo de la red en las aplicaciones P2P serán subidos a otro nodo de la red con posterioridad. Así, dividen los datos descargados y subidos por los nodos en bloques de datos e intentan detectar aquellos flujos que comparten bloques de datos; éstos serán identificados como flujos P2P. De esta misma forma, las heurísticas propuestas en el presente artículo también se basan en el comportamiento de la transferencia de datos de las aplicaciones P2P, aunque, como se verá en la Sección IV, presentan grandes diferencias con respecto al trabajo de Xu et al.

Por último, concretando aún más, existen aportaciones destinadas a la clasificación de un único protocolo. Como ejemplo de detección de tráfico Skype se puede citar [7], en el que los autores proponen un marco basado en dos técnicas complementarias para detectar en tiempo real el tráfico perteneciente a dicho protocolo. La primera aproximación se basa en el test Chi-Cuadrado de Pearson y en las características del tráfico de VoIP para detectar, en la estructura de los paquetes, huellas fundamentales del tráfico de Skype, valiéndose de la aleatoriedad introducida a nivel de bit en el proceso de encriptación. La segunda se basa en la caracterización estadística del tráfico de Skype en términos de tasa de llegada y tamaño de paquetes. El presente trabajo se basa también en la detección de un protocolo concreto, en este caso, en la detección del protocolo eDonkey, cuyas características principales se describen en la siguiente sección.

III. Conceptos generales del protocolo eDonkey

Las redes P2P se pueden dividir en tres tipos, según su arquitectura: centralizadas, distribuidas e híbridas. En las redes centralizadas existe un servidor encargado de indexar los recursos de la red, asociando recursos y clientes que los comparten. En el caso de las redes distribuidas no se requiere ninguna gestión centralizada. Los nodos son los encargados de almacenar la distribución de los contenidos en la red. Por último, las redes híbridas son una combinación de las dos anteriores. En éstas los clientes pueden ser clientes y supernodos (o servidores), formando los clientes una red distribuida en torno a los supernodos, los cuales realizan las tareas de los servidores en las redes centralizadas.

En este contexto de redes P2P, el protocolo eDonkey resulta de muy amplio uso. Por ello es el objetivo central de este trabajo. Con el fin de comprender el alcance del estudio aquí realizado, el presente apartado lleva a cabo una breve discusión acerca de las principales características y operativas de eDonkey.

El protocolo eDonkey se diseñó para la comunicación de nodos en una red P2P híbrida formada por servidores y clientes. Por un lado, los servidores dan acceso a la red, se encargan de manejar la distribución de la información en estructuras similares a diccionarios que almacenan la relación entre los recursos y los nodos que los comparten. Por otro lado, los clientes son los únicos nodos que comparten datos, y son los encargados de almacenar los recursos de la red.

A continuación se presenta una breve descripción de las comunicaciones del protocolo eDonkey, de especial interés para el presente trabajo, divididas en comunicaciones cliente-servidor y comunicaciones cliente-cliente.

A. Comunicaciones cliente-servidor

Para poder acceder a los recursos de la red, los clientes deben conectarse a un servidor eDonkey. Esta conexión puede

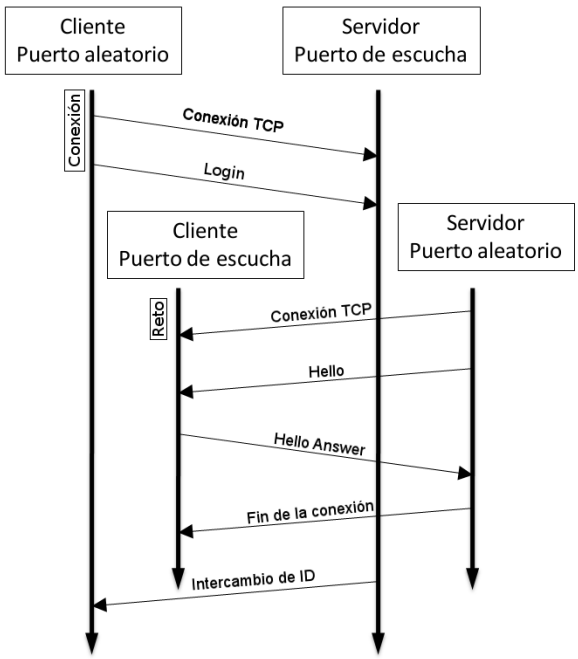


Fig. 1. Proceso de conexión cliente-servidor del protocolo eDonkey.

dividirse en dos fases (Figura 1): *(i)* Solicitud de conexión por parte del cliente, y *(ii)* reto por parte del servidor. En la solicitud de conexión, el cliente inicia una conexión TCP con el servidor y, posteriormente, envía el mensaje de registro del protocolo eDonkey, denominado `login`. El servidor responde al cliente con la segunda parte de la conexión, el reto. Para esto, se realiza una segunda conexión TCP, en este caso iniciada por el servidor y cuyo destino es el puerto definido por el cliente para la recepción de las conexiones de otros clientes de la red eDonkey. Este reto determina si el cliente es capaz o no de recibir conexiones de otros clientes. El proceso de conexión finaliza con el envío, por parte del servidor, de un identificador único. En caso de que el reto haya sido superado con éxito el cliente es identificado con *ID alta*, en otro caso con *ID baja*. En conclusión, una ID baja implica que el cliente no es capaz de aceptar conexiones de otros clientes de la red, y una ID alta que sí lo es.

Después de una conexión exitosa, tanto servidor como cliente intercambian información acerca de su estado actual. El cliente comienza enviando la lista de recursos que comparte, y el servidor envía, entre otra información, su versión y la lista de servidores que conoce.

Una vez que el cliente tiene acceso a un servidor eDonkey puede realizar búsquedas mediante palabras clave, a las que el servidor responderá con una lista de recursos relacionados. Posteriormente, el cliente decide descargar uno o más archivos de la lista obtenida enviando un mensaje al servidor en el que pregunta por el recurso concreto que desee, y al que el servidor responde con una lista de clientes que poseen dicho recurso.

B. Comunicaciones cliente-cliente

Para descargar el recurso o recursos por los que el cliente ha preguntado al servidor, es necesario realizar conexiones cliente-cliente. Estas conexiones se realizan mediante una

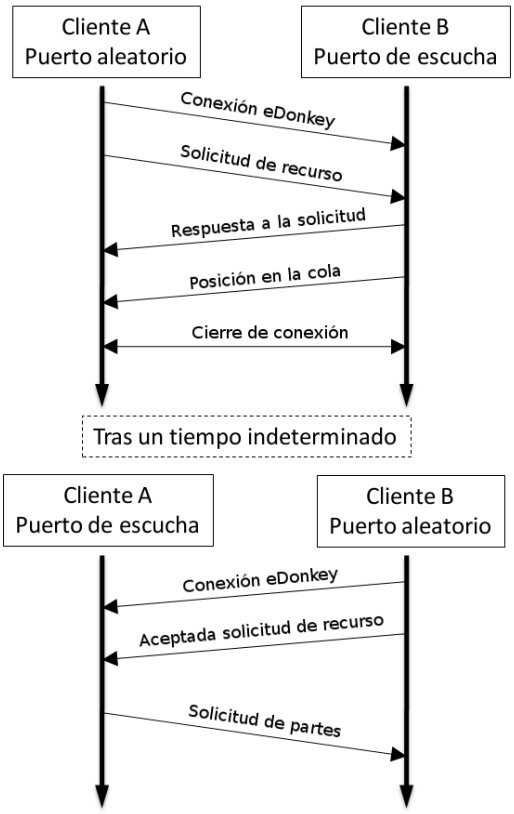


Fig. 2. Entrada en la cola de peticiones de un cliente e inicio de la descarga de un recurso.

conexión TCP inicial, seguida por un intercambio de mensajes del protocolo eDonkey `hello` y `hello answer`. Estos mensajes portan información del nodo que lo envía, principalmente el identificador de usuario y el puerto TCP en el que se reciben conexiones eDonkey.

Existe un caso especial en las conexiones cliente-cliente: aquel en el que el cliente al que se intenta conectar tiene un identificador de ID baja. Por definición, un cliente con ID baja no es capaz de aceptar conexiones, por lo que esta conexión resulta imposible mediante el mecanismo convencional. Para solventar este problema, el protocolo eDonkey dispone de un mensaje llamado `callback`. Aprovechando que el cliente con baja ID tendrá una conexión con el servidor, el cliente origen envía un mensaje `callback` al servidor indicando su intención de conectarse con el cliente destino. El servidor, como respuesta, envía otro mensaje `callback` al cliente destino, a través de la conexión establecida previamente con él (que fue generada por el cliente). Por último, el cliente destino, al recibir el mensaje del servidor, inicia una conexión con el cliente origen. Este mecanismo, sin embargo, no solventa el caso en el que cliente origen y destino posean ID baja, ya que ninguno de los dos es capaz de aceptar conexiones externas. En este caso, la conexión entre ellos no está soportada en el protocolo eDonkey.

Una vez establecida la conexión entre clientes de la red, es posible iniciar el proceso de descarga de un recurso. La situación más común en este proceso se muestra en la Figura 2. El cliente A (cliente origen) envía un mensaje solicitando el recurso buscado y el cliente B (cliente destino) responde a esta solicitud indicando que lo posee. Posteriormente, el

cliente B indica la posición que ocupa la solicitud del cliente A en su cola de peticiones a servir y cierra la conexión. Cuando la petición del cliente A alcanza una posición en la cola que le permite ser servida, el cliente B inicia una conexión con el cliente A y le envía un mensaje indicando que acepta su solicitud. Finalmente, el cliente A solicita las partes concretas del recurso buscado para que el cliente B inicie su envío. Tal y como se detalla en la Sección IV, este comportamiento se utilizará como heurística de detección de patrones de generacón de tráfico eDonkey.

Otra posibilidad, aunque bastante menos frecuente, es que la solicitud del cliente A sea aceptada sin necesidad de cerrar la conexión iniciada por éste. Esto se puede deber a que la cola de servicio del cliente B no esté completamente llena, o a que, por algún motivo, B decida servir las peticiones del cliente A con mayor prioridad.

IV. Heurísticas de detección

Planteada como se ha hecho con anterioridad nuestra intención de desarrollar mecanismos para la detección de comunicaciones P2P, concretamente referidas al protocolo eDonkey, en lo que sigue se presentan a nivel teórico las heurísticas propuestas con tal fin.

Dos son las heurísticas de detección ideadas: detección de nodos basada en la tasa de subida y detección de flujos basada en la inversión del sentido de la descarga. A través de la detección de nodos P2P se persigue determinar qué dispositivos generan en un momento determinado tráfico de este tipo, lo cual puede resultar de interés para los ISPs. Sustentada en esta primera detección, la determinación de qué flujos concretos son de tipo P2P puede resultar de interés de cara, por ejemplo, a la caracterización de los mismos orientada, por ejemplo, a un filtrado más específico.

A. Detección de flujos

La primera heurística tiene como objetivo la detección de flujos del protocolo eDonkey basándose en la hipótesis de que el cliente que inicia la conexión es el que envía los datos. Esto se debe a que el proceso más común de descarga de un archivo mediante el protocolo eDonkey (Figura 2) está formado por dos conexiones TCP. En la primera, el cliente A indica su interés por un recurso del cliente B entrando en su cola de peticiones a servir, y en la segunda el cliente B inicia la conexión con el cliente A para indicarle que su petición puede ser servida y comenzar a enviarle datos, de modo que el cliente que inicia la conexión (cliente B) es el que envía los datos.

En la mayoría de las aplicaciones cliente-servidor es el servidor el encargado de enviar los datos tras una conexión iniciada por el cliente. Este comportamiento es inverso por tanto al que se ha descrito en el protocolo eDonkey y, por este motivo, los autores proponen la siguiente hipótesis para la detección de flujos eDonkey:

Hipótesis 1. *Son flujos del protocolo eDonkey aquellos en los que el cliente que inicia la conexión envía sustancialmente más información que la que recibe.*

Esta heurística se cumple únicamente en flujos del protocolo eDonkey utilizados para transferir archivos y no en flujos de señalización. Aún así, por ejemplo para la clasificación de tráfico por parte de los ISPs, los flujos de transferencia de archivos son los especialmente relevantes, al ser los que ocupan el mayor porcentaje del ancho de banda de la red.

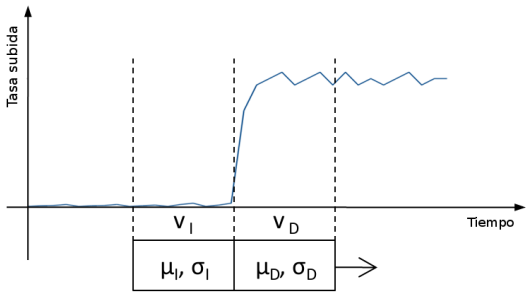


Fig. 3. Avance a lo largo del tiempo del cálculo de la divergencia de Kullback-Leibler.

Basado en la hipótesis de inversión del sentido de la descarga, se ha desarrollado el algoritmo de detección de flujos de transferencia de archivos del protocolo eDonkey mostrado más abajo como Algoritmo 1. Este algoritmo de detección ha sido desarrollado para ejecutarse en capturas de tráfico analizadas no en tiempo real. Se recorren uno a uno los flujos que conforman la captura y se extraen aquellos cuyo número de *bytes* enviados desde el cliente que inicia la conexión al que la recibe es mayor que la información recibida en sentido contrario más un umbral, Umb_B. Este umbral se debe determinar experimentalmente en base a un estudio de la distribución de tamaños de los flujos de transferencia de archivos del protocolo eDonkey. Finalmente, los flujos extraídos son los que el algoritmo detecta como flujos de compartición de archivos del protocolo eDonkey ($flujos\_eD$).

Algoritmo 1 Detección de flujos

1: **para** $i = 0$ mientras $i < num\_flujos$ **hacer**
2: **si** $bytes\_env[i] > bytes\_rec[i] + Umb_B$ **entonces**
3: $flujos\_eD \Leftarrow flujos[i]$
4: **fin si**
5: $i \Leftarrow i + 1$
6: **fin para**
7: **devolver** $flujos\_eD$

La elección del umbral utilizado en este algoritmo de detección, como se verá en la Sección VI, no resulta crítica, ya que la diferencia de tamaños en los flujos de compartición de archivos del protocolo eDonkey es muy elevada.

B. Detección de nodos

Una alternativa distinta, aunque complementaria como se ha apuntado anteriormente, a la detección de los flujos perteneciente al protocolo eDonkey es (como sucede en BLINC [5]) la detección de los nodos generadores de tráfico eDonkey. Conociendo los nodos que generan tráfico de tipo eDonkey, se podrían aplicar políticas de conformación del tráfico proveniente de los nodos que sean detectados.

Esta heurística de detección se basa en la asunción de que los usuarios limitan la tasa de subida de las aplicaciones P2P que utilizan el protocolo eDonkey como eMule o aMule, entre otras. Esto es así porque sin dicha limitación estas

aplicaciones saturarían la capacidad de subida de las conexiones a Internet de los usuarios, provocando una disminución considerable en la velocidad de la navegación Web.

La otra hipótesis en la que se basa la presente heurística es que la limitación de la tasa de subida implica, la mayor parte del tiempo, una tasa de subida constante igual al límite establecido por el usuario. Esta tasa constante representa un comportamiento extremadamente característico a nivel de nodo: un nodo se conecta con multitud de IPs diferentes a lo largo del tiempo, manteniendo un nivel aproximadamente constante de su tasa de subida. En conclusión, la hipótesis de detección queda como:

Hipótesis 2. *Son nodos generadores de tráfico eDonkey aquéllos cuya tasa de subida es constante conectándose a múltiples IPs.*

La cuestión a abordar es qué se debe considerar como un nivel constante de tasa de subida. La respuesta a esta cuestión no es trivial y para abordarla se toma como referencia el trabajo [8], en el que los autores utilizan la divergencia de Kullback-Leibler (KL) para detectar la actividad de voz en señales de audio. Esta detección consiste en determinar el instante temporal en el que la señal de audio evaluada pasa de ser únicamente ruido a contener también voz. Este cambio viene caracterizado por un cambio en la media y la varianza de la señal de audio, que se detecta gracias a la divergencia de KL. La detección de la presente heurística representa el caso contrario: se debe observar ausencia de cambios significativos en la tasa de subida (tasa constante) para determinar que un nodo está generando tráfico eDonkey.

Se puede definir la divergencia de KL como un indicador de la similitud entre dos funciones de distribución, y en el caso de dos distribuciones gaussianas p_I y p_D se representa así:

$$H(p_I||p_D) = \frac{1}{2}[log(\frac{\sigma_D^2}{\sigma_I^2}) - 1 + \frac{\sigma_I^2}{\sigma_D^2} + \frac{(\mu_I - \mu_D)^2}{\sigma_D^2}] \quad (1)$$

donde σ_D y σ_I representan las desviaciones típicas de p_I y p_D, y μ_I y μ_D las medias.

La divergencia de KL no es simétrica, lo que quiere decir que $H(p_I||p_D)$ puede ser diferente de $H(p_D||p_I)$. En la Expresión 2 se muestra la divergencia de KL de dos distribuciones gaussianas p_I y p_D en su forma simétrica:

$$\rho_{I,D} = \frac{1}{2}\left[\frac{\sigma_I^2}{\sigma_D^2} + \frac{\sigma_D^2}{\sigma_I^2} - 2 + (\mu_I - \mu_D)^2(\frac{1}{\sigma_I^2} + \frac{1}{\sigma_D^2})\right] \quad (2)$$

El algoritmo de detección derivado de la heurística propuesta puede ser descrito como sigue (véase Algoritmo 2 más adelante). Primero, los valores de la tasa de subida de un nodo son calculados en intervalos de t segundos. Los valores resultantes son filtrados mediante un filtro de mediana [9] de tamaño N. Este filtro toma N valores de tasa de subida (una ventana de tamaño N) y los ordena de menor a mayor, quedándose con el valor que ocupa el centro del intervalo.

El resultado del filtrado anterior es recorrido por dos ventanas consecutivas (v_I y v_D), cada una de tamaño N (Figura 3). En cada ventana se halla de forma independiente la media y la varianza de los valores en ellas comprendidos, asumiendo que pueden ser modelados por distribuciones gaussianas, y se calcula la divergencia de KL simétrica (Expresión 2) de estas distribuciones.

En el caso de [8] se utilizan los máximos de la divergencia de KL para encontrar los instantes en los que la señal de audio pasa de ser únicamente ruido a contener voz, o viceversa. Esto se debe a que estos valores representan los instantes en los que las distribuciones gaussianas que modelan los valores en v_I y v_D difieren en mayor medida.

La principal aportación del mecanismo de detección propuesto con respecto a [8] radica en que en la presente detección se busca que las distribuciones gaussianas que modelan los valores comprendidos en las ventanas sean lo más similares posible, lo que quiere decir que la divergencia de KL es cercana a cero. En el algoritmo propuesto esto implica que la divergencia de KL no debe superar un umbral determinado experimentalemente (Umb_{KL}) como se verá en la Sección VI.

Algoritmo 2 Detección de nodos

para $nodo = 0$ mientras $nodo < num\_nodos$ **hacer**
  para $i = 0$ mientras $i < len(tasa\_sub_{nodo})$ **hacer**
    $tasa\_filt_{nodo} \Leftarrow filt\_mediana(tasa\_sub_{nodo}, N)$
    $v_I \Leftarrow tasa\_filt_{nodo}[i : i + N]$
    $v_D \Leftarrow tasa\_filt_{nodo}[i + N + 1 : i + 2N + 1]$
    $\rho_{I,D}[i] \Leftarrow \frac{1}{2}\left[\frac{\sigma_I^2}{\sigma_D^2} + \frac{\sigma_D^2}{\sigma_I^2} - 2 + (\mu_I - \mu_D)^2(\frac{1}{\sigma_I^2} + \frac{1}{\sigma_D^2})\right]$
    si $\rho_{I,D}[i] < Umb_{KL}$ AND μ_D distinto a 0 **entonces**
      devolver Sí eDonkey
    si no
      devolver No eDonkey
    fin si
    $i \Leftarrow i + 1$
  fin para
  $nodo \Leftarrow nodo + 1$
fin para

V. Descripción del entorno experimental

Se han utilizado tres conjuntos de capturas de tráfico para realizar la experimentación relativa a las heurísticas presentadas en la Sección IV. A continuación se describen las características de estas trazas.

- *Trazas en entorno controlado (EC).* Un total de 5 usuarios aceptaron voluntariamente someterse a la monitorización de su tráfico de red durante 72 horas ininterrumpidas. En este período compartieron, mediante el programa aMule en su versión 2.2.6, una carpeta de archivos con idéntico contenido. Todos ellos se conectaron al servidor de eDonkey *se-Master Server 1* y limitaron la tasa de subida de aMule a 30kB/s. Todos los usuarios utilizaron sus PCs y su conexión a Internet sin ninguna restricción. En media cada usuario generó alrededor de 19000 conexiones del protocolo eDonkey y más de 7000 de otros protocolos, entre los que cabe destacar DNS, HTTP, SSH y SMTP.
- *Trazas de un servidor HTTP (SH).* Este conjunto de trazas contiene el tráfico generado por un servidor HTTP de una universidad europea durante 7 días. Es un servidor

Tabla I
TASA DE DETECCIÓN DEL ALGORITMO DE DETECCIÓN DE FLUJOS EN EL CONJUNTO DE TRAZAS TU.

	Tasa detección	Flujos detectados	Flujos totales
BitTorrent	0,0256	854	33304
HTTP	0,01691	50795	3003161
FTP	0,01423	35	2460
SSL	0,01244	2808	225685
IRC	0,00213	7	3281
Oscar	0,00079	2	2528
DNS	0,00001	8	1508413
Mail_POP	0,00000	0	5208
Todos	0,01139	54509	4784040

Apache versión 2.2.0 que recibe una media de 8971 conexiones por día.

- *Trazas de un troncal de universidad (TU).* Se ha almacenado todo el tráfico saliente del troncal de una universidad de Oriente Medio durante 48 horas. En este conjunto de trazas se encuentran alrededor de 73000 IPs, se han transmitido cerca de 300 millones de paquetes y los principales protocolos utilizados son: Bittorrent, HTTP, DNS, SSL y FTP entre otros varios. Tras el análisis de toda la base de datos mediante una inspección de *payload* con `OpenDPI` [10], no se ha detectado ningún paquete eDonkey. Esto se debe a que las aplicaciones P2P de compartición de archivos utilizadas en esta universidad de Medio Oriente se basan en Bittorrent en lugar de en eDonkey.

VI. RESULTADOS EXPERIMENTALES

La experimentación realizada para ambos algoritmos presentados en la Sección IV ha ido enfocada a dos objetivos principales: *(i)* Estudiar la validez de las hipótesis presentadas en las heurísticas, en base a la tasa de aciertos y de falsos negativos en las trazas de entorno controlado y, *(ii)* analizar si estas hipótesis se cumplen en otros protocolos, extrayendo para ello, en las trazas SH y TU el porcentaje de falsos positivos para protocolos diferentes a eDonkey.

A continuación se analizan separadamente los resultados obtenidos para los algoritmos de detección de flujos y de nodos.

A. Detección de flujos

En la experimentación realizada se ha supuesto que, por la propia filosofía de las redes P2P, un nodo de la red eDonkey utiliza para su funcionamiento normal más de un flujo del protocolo de forma simultánea. De modo que sólo se consideran flujos eDonkey si, aparte de ser detectados mediante la heurística propuesta, su actividad ha coincidido temporalmente con al menos otro flujo detectado en el mismo nodo.

Para aplicar la heurística de detección de flujos es necesario determinar el umbral de diferencia de *bytes* enviados frente a recibidos, Umb_B. Para esto se ha realizado un estudio del porcentaje de flujos detectados en los tres conjuntos de trazas en función del valor de Umb_B. Los resultados de este análisis indican que existe un amplio rango para la elección de este umbral dentro del cual la eficacia de este algoritmo de detección es considerable. Concretamente el valor utilizado para la experimentación de detección de flujos es de 10kB.

Las trazas EC contienen 37089 flujos de compartición de archivos eDonkey. De éstos, 28016 han sido detectados correctamente, lo que supone una tasa de acierto del 77,53%. En este conjunto de trazas no se ha producido ningún falso positivo. El porcentaje de los flujos de compartición de archivos de eDonkey que no han sido detectados es de un 22,47%. Esto se debe principalmente a dos motivos:

1) *ID baja en alguno de los extremos.* Los nodos con ID baja no pueden aceptar conexiones de la red eDonkey y, por este motivo, siempre inician la conexión, independientemente de si han de enviar los datos ellos o no. Esta situación queda, por tanto, fuera de la hipótesis del mecanismo propuesto de detección de flujos.
2) *Servicio sin cierre intermedio de conexión.* Aunque es menos frecuente, es posible que una solicitud de un recurso llegue a ser servida sin necesidad de cerrar la conexión inicial. Esto se produce cuando la petición inicial empieza a ser servida sin entrar en la cola de espera (vease Sección III). Si esto es así, la Hipótesis 1 de detección no es válida.

También se ha ejecutado el algoritmo de detección de flujos eDonkey en las trazas SH para explorar los falsos positivos que se producen en el protocolo HTTP. Los resultados de la detección indican que ninguno de los 62798 flujos totales ha sido detectado como eDonkey.

Para obtener el porcentaje de falsos positivos en las trazas de navegación real sin tráfico eDonkey (trazas SH y TU), se han etiquetado, en primer lugar, los protocolos a los que corresponden los diferentes flujos. Para ello, se ha modificado la aplicación `OpenDPI` para transformarla en una aplicación de clasificación de flujos en lugar de clasificación de paquetes.

Gracias a la modificación de `OpenDPI` se han podido extraer los resultados de detección que se muestran en la Tabla I. BitTorrent es el protocolo que mayor tasa de falsos positivos presenta y esto se debe a que al ser también un protocolo P2P para la compartición de archivos puede ser coherente con la hipótesis presentada. A diferencia de eDonkey, los flujos BitTorrent son bidireccionales, es decir, a través del mismo flujo pueden enviar partes de un recurso tanto un cliente como el otro. Por este motivo, el algoritmo no detecta un porcentaje mayor de flujos BitTorrent. Es lógico que los flujos FTP sean detectados porque es muy frecuente que el cliente envíe por FTP más que el servidor (caso de subida de ficheros al servidor). Un caso particular es el protocolo HTTP, el cual, aunque no fue detectado ningún flujo en las trazas SH, en éstas presenta un 1,69% de falsos positivos. Estudiados en detalle estos falsos positivos se descubrió que son provocados principalmente por tres motivos: (i) Campos de Cookie y URL muy extensos en el mensaje HTTP GET, (ii) respuestas del servidor con código 304 (Recurso no modificado), que son muy reducidas en tamaño y, por tanto, menores que la petición y, (iii) peticiones HTTP POST, en las que el envío de datos de formulario al servidor hacen que la petición pueda ser más grande que la respuesta.

B. Detección de nodos

Se ha verificado que la Hipótesis 2, para la detección de nodos, es coherente con la tasa de subida de cada uno de los nodos del experimento controlado, como se muestra en la

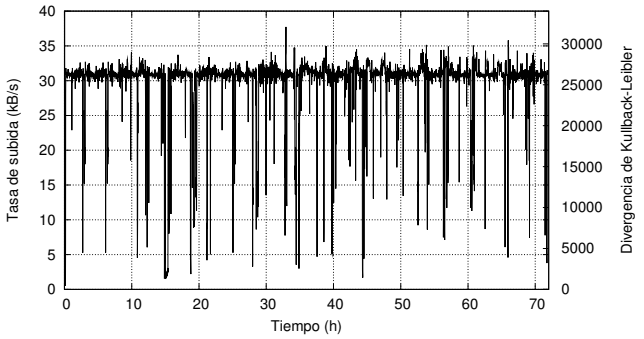


Fig. 4. Tasa de subida de uno de los nodos monitorizados en la traza EC.

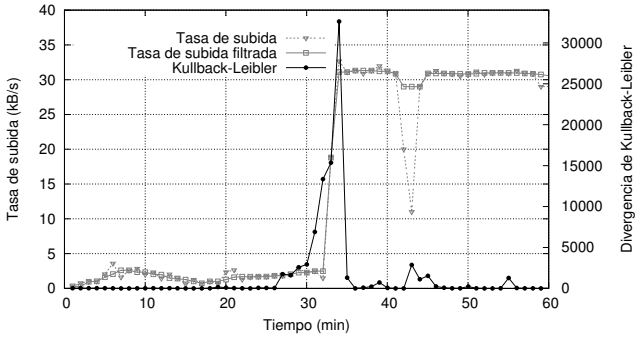


Fig. 5. Evolución de la divergencia de Kullback-Leibler en el análisis de una hora de tráfico (traza EC).

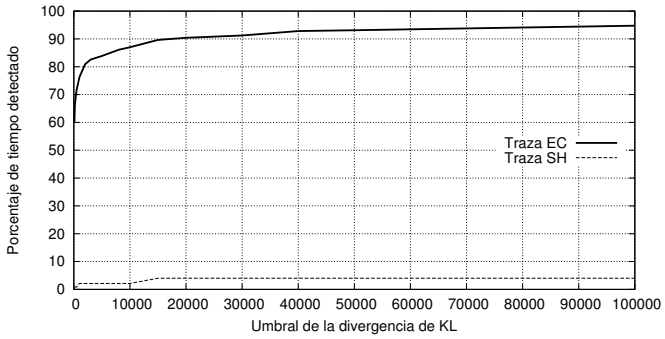


Fig. 6. Tasas de detección de la heurística de detección de nodos en las trazas EC y SH variando el valor de Umb_{KL}.

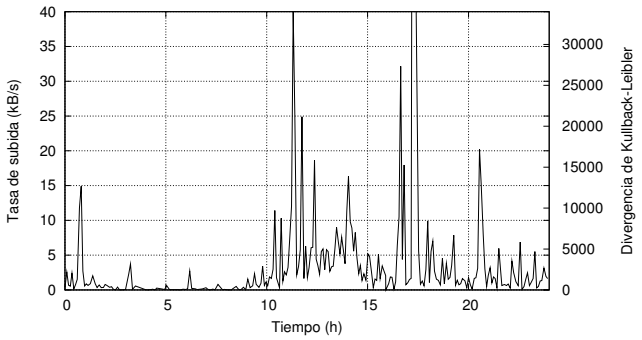


Fig. 7. Tasa de subida del servidor HTTP (traza SH) durante 24 horas.

Figura 4, en la que se representa dicha tasa para uno de los nodos. Durante las 72 horas de duración de la traza se puede apreciar claramente un comportamiento constante alrededor de 30kB/s, que es el límite superior impuesto por el usuario en el experimento.

También se observan en esta figura decaimientos de la tasa de subida. Éstos son de duración reducida y corresponden a instantes en los que se deja de enviar datos a un nodo para iniciar la transferencia con otro. El *churn* de las redes P2P (tasa de entrada y salida de nodos a la red) es elevado y es la principal razón por la que se producen decaimientos en la tasa de subida.

Los instantes en los que se supera el límite establecido por el usuario son otra característica a resaltar de la Figura 4. Éstos corresponden a actividad de red adicional al tráfico eDonkey, como puede ser: navegación HTTP, subida de algún archivo mediante SSH, envío de correo, etc. En estas trazas, un 38,7% de los flujos pertenecen a protocolos diferentes a eDonkey, entre los que destacan DNS, HTTP, SSH y SMTP.

En la Figura 5 se muestra la tasa de subida de uno de los usuarios de las trazas EC durante la primera hora del experimento. Durante los primeros 30 minutos apenas hay tasa de subida porque el servidor aún no ha dado a conocer a suficientes usuarios la existencia de este nuevo nodo en la red. A partir de esa primera media hora se aprecia que el mencionado comportamiento es constante alrededor de la tasa de 30kB/s. También se puede apreciar la reducción del efecto de los valores de tasa de subida que se desvían excesivamente de los esperados gracias a la aplicación del filtro de mediana. Por último, en esa misma representación se superpone la divergencia de KL de los valores filtrados de la tasa de subida. La media del primer tramo de la divergencia de KL es cercana a cero y la varianza reducida. Esto se debe a que la tasa de subida en este intervalo es muy similar. En el segundo tramo se puede observar un incremento en la divergencia de KL, debido al abrupto cambio de media y varianza de la tasa de subida. Por último, la media y varianza de la divergencia vuelven a ser reducidas en el tercer tramo de la representación. Este último tramo se detecta como tráfico generado por un nodo de la red eDonkey.

Adicionalemente, se ha realizado un estudio de los tres conjuntos de trazas para escoger el valor del umbral, Umb_{KL}, y el tamaño de la ventana, N, del filtro de mediana y del cálculo de la divergencia de KL. Se ha buscado un valor de Umb_{KL} que maximice la detección de los nodos generadores eDonkey y minimice la detección en los conjuntos de trazas TU y SH. Para esto se ha ejecutado el algoritmo de detección sobre estos tres conjuntos de trazas con valores del umbral en el rango de 10^2 a 10^5. Los resultados de detección se muestran en la Figura 6, en la que se puede observar que existe un amplio rango de Umb_{KL} sin grandes variaciones en las tasas de detección y de falsos positivos. Este resultado permite a los autores asumir que la elección de Umb_{KL} dentro de un rango lógico $[10^3, 10^4]$ no es crítica. Para los resultados expuestos a continuación se ha seleccionado un valor de Umb_{KL} igual a 8000.

Respecto a la elección del tamaño de ventana, N, del filtrado de mediana y del cálculo de la divergencia de KL, se ha fijado a 5 minutos por ser ésta la mínima ráfaga de tráfico constante que se pretende detectar. Un tamaño de ventana mayor dificultaría enormemente la detección de ráfagas de este tamaño. Las ráfagas de un tamaño menor no son objeto de este trabajo ya que se asume que un nodo generador de tráfico eDonkey interesante por su elevado consumo de ancho

de banda permanece conectado a la red un tiempo prolongado.

El porcentaje de tiempo durante el cual, según el algoritmo propuesto, los nodos han estado generando tráfico eDonkey ha sido de un 86,10%. Los intervalos de la tasa de subida que no se han detectado como constantes se relacionan con los decaimientos provocados por el cambio de usuario con el que se comparte.

Por último, también se ha comprobado, como puede verse en la Figura 7, que la tasa de subida del servidor HTTP no parece tener un comportamiento constante con media superior a cero. Este mismo comportamiento, aparentemente aleatorio, se observa en los nodos monitorizados en la traza TU. Solamente un 1,678% de las 168 horas en las que se monitorizó el servidor HTTP (traza SH) se clasificó a este nodo como generador eDonkey (falsos positivos).

VII. Conclusiones

En el presente artículo se aborda el problema de detectar el tráfico perteneciente al protocolo eDonkey sin inspeccionar el *payload*. Para este fin, se proponen dos algoritmos de detección. El primero para la detección de flujos y el segundo para la detección de nodos. La heurística de detección de flujos se basa en la afirmación de que los flujos del protocolo eDonkey cumplen que el número de *bytes* enviados desde el cliente que inicia la conexión al que la recibe es sustancialmente mayor que en el sentido inverso. La segunda heurística afirma que son nodos generadores de tráfico eDonkey aquellos cuya tasa de subida es constante conectándose a múltiples IPs. En base a la experimentación realizada con los tres conjuntos de trazas se puede concluir que las hipótesis de detección propuestas se cumplen para el protocolo eDonkey, presentando una buena tasa de detección y un reducido número de falsos positivos. La experimentación también ha permitido especificar que la heurística de detección de flujos es válida únicamente en flujos de compartición de archivos para nodos eDonkey con ID alta.

Actualmente se está trabajando para ampliar la experimentación a trazas reales, fuera de un entorno controlado, que contengan de forma simultánea tráfico eDonkey y de otros protocolos. Adicionalmente, se pretenden abordar en un futuro próximo dos líneas de trabajo principales:

- La combinación de la información de ambas heurísticas para la mejora de la tasa de detección y la reducción de los falsos positivos.
- Explorar la posibilidad de detectar otros protocolos P2P de compartición de archivos mediante la ejecución de la segunda heurística. Es asumible que esta detección pueda ampliarse porque los protocolos de compartición de archivos saturan la tasa de subida del usuario y esto implica la necesidad de una limitación en la misma. Esta limitación es aprovechada en la detección propuesta y por ello se cree que puede ser ampliada a otros protocolos como Kademlia o BitTorrent.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el MICINN mediante el proyecto TEC2008-06663-C03-02.

Referencias

[1] A. Callado, C. Kamienski, G. Szabo, B. Gero, J. Kelner, S. Fernandes, and D. Sadok, "A Survey on Internet Traffic Identification," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 3, pp. 37–52, Aug. 2009.

[2] A. Feldmann, "A possibility for isp and p2p collaboration," in *Broadband Communications, Networks and Systems, 2008. BROADNETS 2008. 5th International Conference on*, 2008, p. 239.

[3] T. Karagiannis, A. Broido, M. Faloutsos, and K. Claffy, "Transport layer identification of P2P traffic," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, ser. IMC '04. New York, NY, USA: ACM, 2004, pp. 121–134.

[4] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," in *Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, ser. SIGMETRICS '05, vol. 33, no. 1. New York, NY, USA: ACM, Jun. 2005, pp. 50–60.

[5] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: multilevel traffic classification in the dark," in *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '05, vol. 35, no. 4. New York, NY, USA: ACM, 2005, pp. 229–240.

[6] K. Xu, M. Zhang, M. Ye, D. M. Chiu, and J. Wu, "Identify P2P traffic by inspecting data transfer behavior," *Computer Communications*, vol. 33, no. 10, pp. 1141–1150, Jun. 2010.

[7] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli, "Revealing skype traffic: when randomness plays with you," in *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '07. New York, NY, USA: ACM, 2007, pp. 37–48.

[8] J. Ramirez, J. Segura, C. Benitez, A. de la Torre, and A. Rubio, "A new kullback-leibler vad for speech recognition in noise," *Signal Processing Letters, IEEE*, vol. 11, no. 2, pp. 266–269, 2004.

[9] J. Astola, P. Haavisto, and Y. Neuvo, "Vector median filters," *Proceedings of the IEEE*, vol. 78, no. 4, pp. 678–689, Apr. 1990.

[10] Opendpi. [Online]. Available: http://www.opendpi.org/

Propuesta de sincronización inter-destinatario adaptativa para aplicaciones multimedia distribuidas

Fernando Boronat, Mario Montagud
Departamento de Comunicaciones (DCOM) – Instituto IGIC
Universitat Politècnica de València (UPV) – Escuela Politécnica Superior de Gandia (EPSG)
C/ Paraninfo, 1, C.P. 46730, Grao de Gandia (Valencia)
fboronat@dcom.upv.es, mamontor@posgrado.upv.es

***Resumen*- La sincronización inter-destinatario es esencial en una gran variedad de aplicaciones multimedia emergentes, tales como la Televisión Social Interactiva por Internet o los juegos en red multi-jugador. En este artículo se presenta la combinación de una versión evolucionada de una propuesta de sincronización inter-destinatario, incluyendo varías políticas dinámicas para la selección de una referencia de sincronización maestra, y una novedosa técnica de ajuste suavizado de la tasa de reproducción, que posibilita la consecución de un estado de sincronización global entre usuarios distribuidos en una sesión multimedia. Las pruebas de simulación revelan la capacidad de dichas propuestas de mantener la asincronía dentro de unos límites permisibles, al mismo tiempo que se minimiza la ocurrencia de discontinuidades en los procesos de reproducción (saltos/pausas), que resultan bastante más molestas para los usuarios que variaciones suavizadas de los tiempos de reproducción.**

***Palabras Clave*- RTP/RTCP, Simulación, Sincronización Inter-Destinatario, Sistemas Multimedia,**

I. INTRODUCCIÓN

Las sistemas multimedia se caracterizan frecuentemente por la existencia de varios flujos multimedia, tanto continuos (video, audio,…) como discretos (texto, gráficos,…), transmitidos por una o varias fuentes hacia uno o varios receptores, los cuales pueden reproducir un único flujo o varios de ellos. Debido a las relaciones temporales, espaciales o semánticas entre las unidades de datos multimedia o *Media Units* (MUs) en un mismo flujo, así como entre distintos flujos (p. ej., tramas de video o muestras de voz), se requerirá algún mecanismo preciso de coordinación e integración con tal de preservar y garantizar una presentación temporalmente ordenada para cada uno de los flujos recibidos. Dicho proceso es conocido como sincronización multimedia. Podemos distinguir tres tipos de sincronización multimedia temporal: intra-flujo, inter-flujo e inter-destinatario [1]. La Fig. 1 muestra un ejemplo para cada uno de ellos. En ella podemos discernir un grupo de receptores distribuidos sobre una red IP, reproduciendo flujos de video, audio y datos. En primer lugar, la *sincronización intra-flujo* se refiere al mantenimiento de las relaciones temporales entre las MUs pertenecientes a un determinado flujo multimedia. Como ejemplo, si en una secuencia de vídeo se capturan 25 tramas por segundo, en el receptor se deberá reproducir cada una de ellas durante 40 milisegundos (Fig. 1). La *sincronización inter-flujo* se ocupa de mantener las dependencias temporales entre las MUs de distintos flujos multimedia. Por ejemplo, en un discurso retransmitido a distancia, en el receptor, deberá existir una sincronización entre las palabras que se escuchan y el movimiento de los labios del locutor. Los anteriores tipos de sincronización multimedia han sido considerados e implementados en la mayoría de aplicaciones multimedia distribuidas. Sin embargo, un tipo adicional de sincronización es esencial en multitud de aplicaciones multimedia emergentes. Se trata de la Sincronización Multimedia Inter-Destinatario o *Inter-Destination Multimedia Synchronization* (IDMS), también conocida como sincronización multipunto o de grupo, y se refiere a la sincronización simultánea entre los estados de reproducción de uno o varios flujos multimedia en varios receptores, independientemente de su localización.

En la actualidad, podemos encontrar una gran variedad de aplicaciones multimedia, como son la Televisión Social Interactiva por Internet, los juegos en red multi-jugador o la educación a distancia en tiempo real, en las que IDMS resulta de especial relevancia puesto que su inexistencia puede degradar la satisfacción de los usuarios (QoE) en diferentes aspectos [2]. Entre las anteriores aplicaciones, destaca por su creciente popularidad la primera de ellas, que posibilita la interacción y compartición de servicios entre usuarios distribuidos, en el contexto de consumo simultáneo de información multimedia, por medio de servicios de comunicación directa en tiempo real, como mensajería instantánea o audio/video conferencia [3].

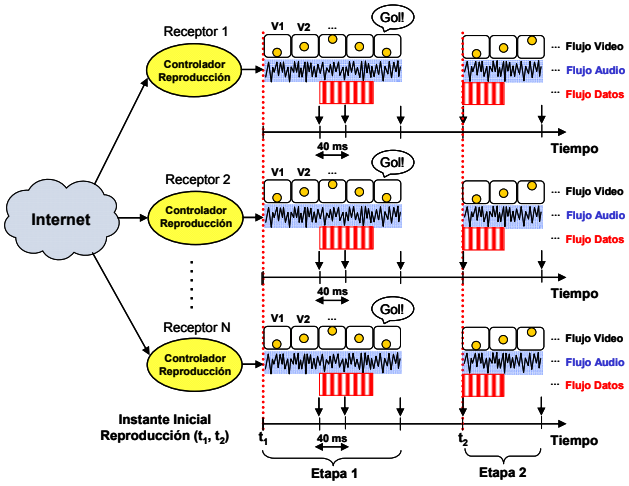


Fig. 1. Sincronización Multimedia

Un ejemplo muy claro, sería el caso en el que un grupo de amigos se pusiesen de acuerdo para ver un partido de futbol por Internet, cada uno desde su hogar. En tal caso, debería garantizarse sincronización inter-flujo entre la secuencia de video que los usuarios están visualizando en conjunto (*watching apart together*) y los flujos multimedia adicionales de comunicación directa entre los amigos (p.ej. mensajes de chat, audio/video llamadas,…). Además, eventos significativos, p.ej. un gol (Fig. 1), deberían ser percibidos por todos los usuarios simultáneamente, incluso en cada uno de los flujos multimedia involucrados, con tal de no frustrar la experiencia de los usuarios en tal interacción, puesto que resultaría muy irritante para un usuario percibir un gol a través de los mensajes o celebraciones de sus amigos/as.

El paradigma cambiante desde el uso de aplicaciones multimedia por parte de usuarios únicos hacia la compartición de servicios y experiencias en aplicaciones sociales multimedia supone una serie de retos y desafíos a afrontar, tales como: modelaje de experiencias compartidas, gestión de usuarios, sincronización y calidad de servicio (QoS). Asimismo, deben contemplarse desafíos técnicos adicionales que incluyen aspectos de escalabilidad, privacidad, directrices de diseño, reducción de ruido o integración de comunicaciones sociales en red.

Uno de los mayores desafíos a tratar, en el que se centra este artículo, es la sincronización simultánea de contenidos multimedia en diferentes sistemas finales (IDMS). A pesar de que los umbrales de asincronía (discrepancia temporal entre estados de reproducción) permisibles en aplicaciones multimedia específicas deberían ser obtenidos por medio de análisis subjetivos, en [2] se concluye que los requisitos temporales para la IDMS pueden variar entre 15 y 500 ms, en función del tipo de servicio multimedia ofrecido. Sin embargo, dichos valores se superan comúnmente en redes de distribución de contenidos convencionales (p.ej. canales de difusión IPTV) debido a retardos variantes de codificación, paquetización, red, procesado, de-paquetización, en colas, de-codificación, entrega y presentación, cuando se transmite un tipo concreto de contenido multimedia (con mecanismos específicos de codificación) a través de diferentes arquitecturas o conexiones de red (p.ej. accesos fijos o celulares, redes de diferentes operadores, …) [2]. Es por ello que la variabilidad de retardos entre diferentes sistemas finales, así como la posible variabilidad temporal de los retardos en cada uno de ellos (debido a condiciones de red cambiantes) se convierten en un serio problema a solucionar cuando se requiere interacción entre los usuarios y el contenido multimedia o entre los mismos usuarios bajo el contexto de un contenido multimedia compartido, puesto que va a dificultar o imposibilitar la provisión de servicios interactivos en dichos escenarios.

Nuestra principal motivación es la mejora de una solución de IDMS previamente propuesta ([4], [5], y [6]). En concreto, se presentan dos contribuciones principales. Por un lado, se han incluido varias políticas dinámicas para la selección de una referencia de sincronización maestra, a partir de los estados de reproducción particulares de cada uno de los participantes activos en una sesión multimedia. Por otro lado, se presenta un algoritmo adaptativo de ajuste de la tasa de reproducción (*Adaptive Media Playout* o AMP), que persigue corregir los estados de reproducción de los receptores, a través de ligeras variaciones de los tiempos de reproducción (respetando unos límites tolerables), cada vez que se detecta una discrepancia temporal entre los estados de reproducción de los receptores superando un umbral preestablecido. De esta manera, se posibilita la consecución de un estado de sincronización global, minimizando la aparición de discontinuidades o interrupciones, tales como saltos y/o pausas, que suelen provocar un efecto bastante más molesto para los usuarios que ajustes suavizados en los tasas de reproducción [7]. Nótese que aunque la idea operacional de nuestra propuesta de IDMS es válida tanto para contenidos almacenados (p.ej. Video bajo Demanda o VoD), como para flujos en directo (p.ej. un canal de difusión IPTV), aún quedan muchos desafíos técnicos y operacionales a abordar en este último caso.

El resto del artículo sigue la siguiente estructura. En la siguiente sección se presentan trabajos relacionados con las técnicas y soluciones propuestas. En la Sección III, se resume la política de *buffering* adoptada para garantizar sincronización intra-flujo. En la Sección IV, se modelan las posibles desviaciones en las tasas de reproducción y los efectos que éstas pueden conllevar sobre la sincronización local y global en una sesión multimedia. A continuación, en la Sección V se describe la solución evolucionada de IDMS, detallando las novedades y mejoras añadidas. La Sección VI presenta la evaluación de la solución propuesta con tal de mostrar sus prestaciones y capacidades. Finalmente, la Sección VII incluye las conclusiones y posibles líneas de trabajo futuras.

II. TRABAJOS RELACIONADOS

En [1], nuestro grupo presentó el más exhaustivo estudio comparativo a nivel cualitativo de las diferentes técnicas de sincronización inter-flujo e IDMS publicado hasta la fecha. Se encontraron muy pocas soluciones que proporcionen IMDS, a pesar de la relevancia que está adquiriendo en muchas aplicaciones emergentes. De hecho, recientemente la ETSI ha considerado la inclusión de un apartado relativo a este tipo de sincronización en una norma que especifica aspectos funcionales para servicios IPTV [8]. Las soluciones existentes de IDMS básicamente siguen tres esquemas en cuanto al control y operación de los algoritmos de sincronización: el maestro/esclavo (*Master/Slave* o *M/S*), el maestro o centralizado (*Synchronization Maestro Scheme o SMS*) y el distribuido (*Distributed Control Scheme o DCS*). En el esquema M/E ([9]), los receptores se clasifican en receptor maestro y receptores esclavos. El receptor maestro enviará (multicast) de forma periódica su estado de reproducción. Así, los receptores esclavos simplemente deberán ajustar sus procesos de reproducción en función del anterior, sin necesidad de enviar su propia temporización de reproducción. En el esquema DCS ([10]), todos los receptores pueden intercambiar (multicast) sus estados de reproducción con el resto de receptores y cada uno de ellos puede decidir la referencia a sincronizarse de entre todos los estados de reproducción de todos los receptores (incluyendo el suyo propio). El esquema SMS ([11]) está basado en la existencia de un maestro de la sincronización (fuente o receptor) que obtiene información de todos los receptores y corrige sus estados de reproducción distribuyendo paquetes de control específicos. En SMS, los receptores se clasifican

siguiendo un esquema M/S en el cual se selecciona un receptor como la referencia de sincronización para ajustar los estados de reproducción de los demás receptores (esclavos). En [4], nuestro grupo presentó una versión preliminar de una propuesta de IDMS empleando un esquema SMS, en el cual la fuente actuaba como el maestro y consideraba un receptor fijo como la referencia de sincronización durante toda la sesión multimedia. En [12], se presenta un esquema SMS que combina la selección de dos puntos de reproducción (el más avanzado/rezagado) como referencia de sincronización, en un escenario de juegos en red multi-jugador. En este tipo de aplicaciones, múltiples jugadores pueden colaborar en equipo o combatir contra otros múltiples jugadores. Si un jugador presenta una temporización de reproducción distinta a la de los demás participantes, no pueden garantizarse condiciones de juego equitativas. Es por ello que se necesita algún mecanismo que aporte IDMS. De este modo, este estudio examinó la influencia de los métodos de selección de referencias de sincronización maestras en base a la igualdad de condiciones (*fairness*) y al rendimiento del trabajo en equipo. Se concluyó que la IDMS mejora la eficiencia del trabajo en equipo en juegos colaborativos si los tiempos de reproducción de los jugadores se ajustan al estado de reproducción más avanzado de entre todos los jugadores bajo control. Esto ocurre porque un jugador con un tiempo de reproducción más avanzado puede ayudar a los demás jugadores más rezagados. Por otro lado, también se concluyó que la eficacia de la IDMS, en base a la equidad entre los jugadores, puede mejorarse si el punto de reproducción más atrasado se toma como referencia de sincronización global. En este trabajo se planteó como estudio futuro el diseño y comparación de políticas adicionales para la selección de la referencia maestra de sincronización (p.ej. el punto de reproducción medio de todos los receptores activos). Sin embargo, aún no existe ningún trabajo publicado que aborde este asunto. Consecuentemente, una contribución de este artículo es la mejora y extensión de nuestro esquema SMS previamente diseñado [4], de modo que el maestro de la sincronización puede gestionar varias políticas dinámicas de selección del punto de reproducción que se tomará como referencia para conseguir IDMS. Obviamente, la selección de una política específica afectará cuantitativamente a la eficacia de la sincronización, así como a la percepción/satisfacción de los usuarios. Sin embargo, no existe una política óptima para todos los escenarios porque su idoneidad vendrá fijada por las condiciones del entorno (carga de red, pérdidas, parámetros de los receptores, …) así como por los requisitos específicos de las aplicaciones multimedia bajo control. Dichas políticas serán descritas y analizadas en este artículo.

En la referencia [13] se evaluó, tanto objetivamente como subjetivamente, la calidad de sincronización labial (*lip-sync*) mediante el uso combinado de varias políticas de control reactivas basadas en saltos, pausas, extensiones y compresiones de los tiempos de reproducción de las MUs (tramas de video y muestras de voz). Además, se investigó la relación entre los resultados del análisis objetivo (se examinó el coeficiente de variación del intervalo de reproducción, la tasa de reproducción media y el error cuadrático medio de la sincronización inter-flujo) y los resultados del análisis subjetivo (*Mean Opinion Score o MOS*) por medio de análisis regresivo, obteniendo una ecuación cerrada que relaciona los anteriores parámetros. Como resultado, se concluyó que: i) el esquema de control reactivo basado en compresiones y extensiones de los periodos de reproducción ofreció la mejor calidad de sincronización, tanto para audio como para video; ii) el esquema de control reactivo basado en saltos y pausas produjo efectos molestos para los usuarios en cada uno de los flujos, especialmente en el de voz; iii) la tasa de reproducción media y el error cuadrático medio de sincronización estaban estrechamente ligados al valor de MOS, tanto para flujos en directo como para flujos almacenados; y iv) el coeficiente de variación del periodo de reproducción, que denota la calidad de la sincronización intra-flujo, también influenciaba en gran medida al valor de MOS estimado.

En este artículo no se va a evaluar la sincronización inter-flujo ya que existen una gran variedad de soluciones que la proporcionan. Por ejemplo, nuestro grupo evaluó una solución para dicho propósito con resultados satisfactorios ([4] y [5]). Así pues, nos centraremos en mejorar nuestra solución de IDMS. En este caso, tal y como sucede para la sincronización inter-flujo, los receptores pueden verse forzados a ajustar sus procesos de reproducción para corregir posibles situaciones de asincronía, causando posibles discontinuidades o interrupciones que pueden llegar a ser muy molestas para los usuarios finales. Por tanto, los resultados y conclusiones del estudio realizado en [13] pueden extrapolarse para el caso de IDMS bajo interés. Es por ello que decidimos diseñar una técnica de AMP con tal de mitigar la ocurrencia de estos efectos indeseados. Los trabajos previos relacionados con técnicas AMP se han centrado en el diseño de soluciones que mejoren la sincronización intra-flujo, tanto en aplicaciones de *streaming* de audio como de video (p.ej. [7], y [14]-[16]) y, ocasionalmente, para la mejora de la sincronización inter-flujo [13]. Sin embargo, en este trabajo se propone adaptar el uso de AMP para su aplicación en IDMS, ya que esta técnica puede habilitar a los receptores distribuidos a ajustar suavemente sus procesos de reproducción, en función de indicaciones enviadas por la fuente, cada vez que se detecta la superación de un límite de asincronía entre sus estados de reproducción.

III. SINCRONIZACIÓN INTRA-FLUJO

Las redes tradicionales de conmutación de paquetes suelen introducir retardos variables, pueden ocasionalmente alterar el orden de algunos paquetes de datos, así como perder alguno de ellos, debido principalmente a algunos factores imprevisibles (p.ej. políticas de enrutamiento dinámicas, carga de red, ancho de banda disponible, políticas de colas no eficientes, …), que pueden causar interrupciones y degradar la QoS de los servicios multimedia en tiempo real. Este fenómeno queda reflejado en la fila intermedia de la Fig. 2, que esquematiza los procesos de transmisión, *buffering* y reproducción de un flujo multimedia típico. Una técnica muy común para aliviar la aparición de dichos efectos indeseados (pero a costa de un retardo adicional) es almacenar en buffer los paquetes recibidos antes de entregarlos a los agentes de reproducción, con el objetivo de minimizar el efecto del *jitter* y restablecer las dependencias temporales de las MUs recibidas (sincronización intra-flujo), de modo que se reproduzcan siguiendo el mismo patrón temporal con el que fueron generadas (uniformemente espaciadas, si se asume

tráfico de tasa constante o *Constant Bit Rate* - CBR -), tal y como se esquematiza en la primera y tercera fila de la Fig. 2.

Sean t_n, $r_{n,i}$ y $p_{n,i}$ los instantes de tiempo en los cuales la *n-ésima* MU se transmite, recibe y reproduce en el receptor *i-ésimo*, respectivamente. El retardo de red para dicha MU se puede estimar como $l_{n,i}=r_{n,i}-t_n$. Asimismo, su retardo de reproducción viene determinado por $d_{n,i}=p_{n,i}-t_n$. Éste último debe ser mantenido uniformemente entre cada *k-ésima* y *n-ésima* MUs, es decir $(p_{n,i}-p_{k,i})\approx(t_n-t_k)$, con tal de garantizar un proceso de reproducción continuo y suavizado. El retardo de reproducción para la primera MU, d_{ini}, debe ser configurado adecuadamente y su valor se calcula en la fase inicial de nuestra propuesta de IDMS (véase Sección V). Así, denominamos Instante Inicial de Reproducción ($p_{ini,i}$) al tiempo de reproducción de la primera MU. A continuación, el controlador de reproducción se encargará de gestionar los tiempos de reproducción de las sucesivas MUs en los instantes $p_{n+1,i}=p_{n,i}+s_{n,i}$, donde $s_{n,i}$ hace referencia al tiempo de servicio (p.ej. visualización) para la *n-ésima* MU. La configuración del retraso de reproducción debe satisfacer una relación de compromiso entre el porcentaje de pérdidas de paquetes por posibles retrasos de llegada y retardo adicional permisible en el servicio multimedia.

Sin pérdida de generalidad, asumimos $r_{n,i} = \infty$ para cada MU que se pierde debido a las condiciones de la red (paquetes *2* y *n* en la Fig. 2). Además, la política de *buffering* debe descartar todos los paquetes que lleguen después de su instante de reproducción planificado, es decir, todos aquellos paquetes para los que se cumple $l_{n,i}>d_{n,i}$, como ocurre con el paquete *m* en la Fig. 2. Varias estrategias se han diseñado para reducir el efecto de tales pérdidas [17]. Por último, la política de *buffering* debe restablecer el orden original de los paquetes recibidos (paquetes *k* y *k+1* en la Fig. 2).

IV. Desviaciones De Las Tasas De Reproducción

El mantenimiento de las dependencias temporales entre las MUs en un mismo flujo o diferentes flujos multimedia puede verse afectado por los siguientes factores: retardos, jitter de red o en los sistemas finales, sobrecarga de CPU, desviaciones/fluctuaciones de los relojes o temporizadores de reproducción, etc. En esta sección vamos a considerar el efecto de la inexactitud de los mecanismos de temporización de la reproducción en los receptores sobre la sincronización local y global. Una tasa de reproducción ideal servirá las MUs entrantes con la misma frecuencia nominal con la que fueron generadas (θ MU/s), véase μ_1 en la Fig. 3. Sin embargo, los mecanismos de temporización locales pueden no ser completamente precisos. Éstos pueden presentar una tendencia de desviación o *skew* modelada por $\pm\gamma$ (μ_2 –tasa lenta– and μ_3 –tasa rápida– en la Fig. 3), expresada comúnmente en partes por millón (*ppm*). Asimismo, las tasas de reproducción pueden presentar una tasa de fluctuación o *drift*, dada por $\omega(t)$, que puede oscilar aleatoriamente con respecto al tiempo entre valores acotados por $\pm\varepsilon$ *ppm* (véase μ_4 en la Fig. 3). Estas desviaciones están estrechamente relacionadas con la resolución del reloj, su antigüedad, la estabilidad del oscilador, cambios de voltaje, la temperatura ambiental y otras variables del entorno, como el ruido [18]. Como resultado, la tasa de reproducción instantánea (en MU/s) para el receptor *i-ésimo* se puede modelar como sigue:

$$\mu_i(t)=\theta\cdot(1+\gamma_i+\omega_i(t)). \quad (1)$$

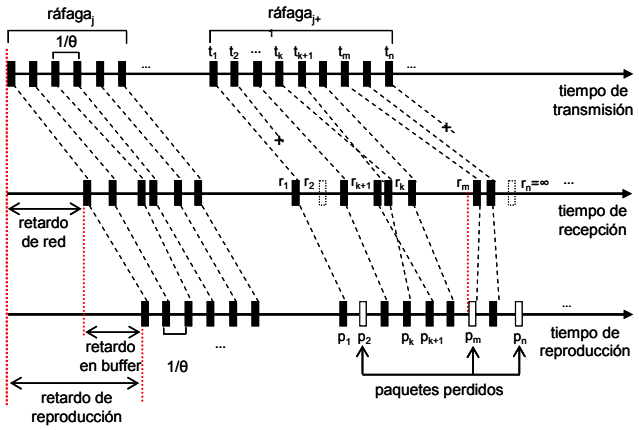


Fig. 2. Sincronización Intra-Flujo

Vamos a asumir que la fuente empieza la trasmisión del flujo multimedia en el instante t_{ini}, y que los receptores empiezan el proceso de reproducción tras un retardo de *buffering* fijo (b_{ini}). En tal caso, podemos encontrar una asincronía inicial entre los procesos de reproducción de los receptores debido a los retardos de red desiguales para cada uno de ellos: el peor caso podría ocurrir cuando el receptor más cercano ($t_{min}=t_{ini}+l_{min}+b_{ini}$) reproduce las MUs con una tasa de reproducción máxima, dada por $\theta\cdot(1+\gamma)$, mientras que el receptor más lejano ($t_{max}= t_{ini}+l_{max}+b_{ini}$) lo hace con una tasa mínima, dada por $\theta\cdot(1-\gamma)$. Además, las diferencias entre dichas tasas de reproducción causarán una asincronía creciente, A, en MUs, entre los estados de reproducción de los receptores a nivel que avanza la sesión multimedia (Fig. 3). Esta asincronía puede modelarse según la ecuación (2), en la cual el primer término refleja la contribución de la variabilidad de los retardos de red, mientras que el segundo representa el efecto de las desviaciones en las tasas de reproducción y de la evolución temporal de la sesión multimedia:

$$A(t,\gamma)=\left[\left((t_{max})-(t_{min})\right)\left(\theta\cdot(1+\gamma)\right)\right]+\left[\left(\theta\cdot(1+\gamma)\right)t-\left(\theta\cdot(1-\gamma)\right)t\right] \quad (2)$$

Esto es inaceptable para aplicaciones multimedia reales, así que, por tanto, se requerirán técnicas y mecanismos que corrijan estos efectos y aporten IDMS. Además, estas soluciones deberán ser adaptativas con respecto a cambios impredecibles de las condiciones de red y posibles congestiones y/o bloqueos en los sistemas finales.

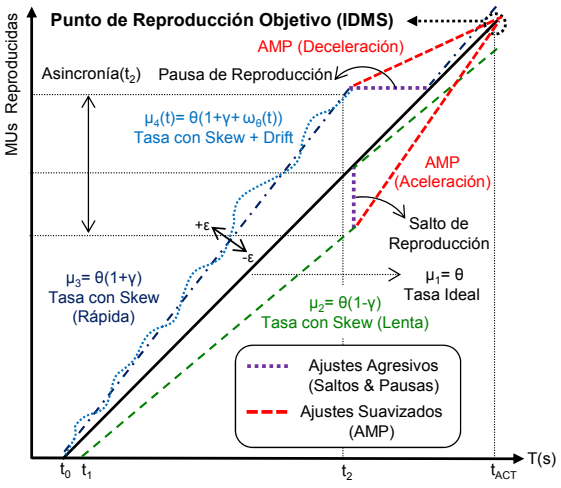


Fig. 3. Tasas y Ajustes de Reproducción

V. PROPUESTA DE IDMS

A pesar de la especial relevancia que está adquiriendo la IDMS, no existen muchas soluciones que aborden esta problemática. La gran mayoría de soluciones recopiladas en [1] definen nuevos protocolos propietarios, con sus mensajes de control específicos adicionales al tráfico de datos multimedia. Actualmente, los protocolos RTP/RTCP [17] son ampliamente utilizados en las aplicaciones multimedia. Por un lado, los números de secuencia y marcas temporales proporcionados en cada paquete de datos RTP son muy útiles para preservar las relaciones temporales (o espaciales) en el lado receptor. Por otro lado, los mecanismos de distribución de informes RTCP son muy ventajosos para informar a las fuentes multimedia sobre la QoS percibida por cada participante, así como pueden ser utilizados por los operadores de distribución de contenidos, por ejemplo IPTV, para la gestión, aislamiento y resolución de problemas [18]. Además, dichos protocolos fueron definidos de modo que se pudiesen extender sus funcionalidades para perfiles de aplicaciones específicos, y las directrices con tal de abordarlo se especifican en [19]. La IDMS implica la distribución de informes sobre la temporización de la reproducción desde/hacia los receptores de una sesión multimedia. Como dicha información puede considerarse como una métrica de QoS, decidimos diseñar una propuesta de IDMS basada en el uso y extensión de dichos protocolos, incluyendo información sobre la temporización de reproducción en informes extendidos y distribuyendo instrucciones para su corrección en nuevos informes específicos para dicho propósito. Así pues, se facilita su futura implementación e implantación en aplicaciones multimedia típicas.

Nuestra propuesta de IDMS se basa en la existencia de una referencia de tiempos global (p.ej. proporcionada mediante NTP, PTP, GPS u otras soluciones). Una ventaja importante de nuestra propuesta es que no se necesita calcular el periodo óptimo para la distribución de informes de realimentación, como así se requiere en las soluciones recopiladas en [1], ya que dichos informes se intercambian de manera más o menos periódica entre los participantes y el intervalo de envío se ajusta dinámicamente en función del número de participantes y el ancho de banda de la sesión, tal y como se especifica en [17]. Nuestra propuesta no está completamente basada en el receptor, como la mayoría de soluciones en [1], sino que sigue un esquema SMS en el cual los ajustes de sincronización son realizados por los receptores pero en base a instrucciones enviadas por la fuente. Además, se utiliza un esquema M/S con respecto a la clasificación de los receptores: un receptor (real o ficticio) es considerado como el receptor maestro y su punto de reproducción será seleccionado como la referencia para determinar el estado (avanzado o rezagado) de los estados de reproducción de los demás receptores (esclavos).

Una versión preliminar de nuestra propuesta de IDMS ya fue evaluada de manera satisfactoria en un escenario WAN real entre los Campus de nuestra Universidad en [4] y [5]. Sin embargo, decidimos optimizar dicha propuesta mediante su implementación en una plataforma de simulación ([6] y [20]), de manera que se posibilitase su evaluación en multitud de escenarios heterogéneos, bajo diversas condiciones de red y con distinto número de receptores, así como facilitar su posible comparación con otras propuestas existentes [1].

Pueden diferenciarse dos fases principales en nuestra propuesta de IDMS: la primera fase trata de garantizar que todos los receptores empiecen el proceso de reproducción del flujo multimedia simultáneamente (Fig. 1); mientras que la segunda fase se ocupa de mantener los procesos de reproducción de manera sincronizada durante la duración de la sesión multimedia. Se pueden consultar la referencia [4] para obtener una información más detallada.

A. Intercambio de Informes sobre Tiempos de Reproducción

Durante la sesión, los receptores envían informes RTCP RR [17] regularmente para informar a la fuente sobre la QoS percibida. Tal y como se posibilita en su especificación, y siguiendo las directrices marcadas en [18], se extendieron dichos informes, llamándolos RTCP RR EXT, con tal de incluir el punto de reproducción local de cada receptor *i-ésimo*, mediante los siguientes campos: i) 16 bits que identifican al número de secuencia de la MU que el receptor está reproduciendo (MU_i); ii) 64 bits relativos a su tiempo de reproducción (p_i); y iii) 8 bits que identifican el grupo específico al que pertenece el receptor. Esto conlleva una corta extensión de 3 palabras de 32 bits (baja sobrecarga de red). Nótese que dichos informes también serían enviados si nuestra propuesta de IDMS no fuese aplicada. Cuando la fuente haya reunido la información sobre el estado de reproducción de todos los receptores activos en un grupo específico, ejecutará un simple algoritmo con tal de seleccionar uno de ellos como la referencia de sincronización (maestro) y calculará la máxima asincronía entre cada uno de ellos. Si ésta excede un umbral pre-configurado (τ_{max}), la fuente enviará un nuevo mensaje de control con tal de forzar ajustes de reproducción en los receptores. Este mensaje es un nuevo paquete RTCP, llamado RTCP APP ACT, que incluye un punto de reproducción al que deberán ajustarse cada uno de los receptores, con: i) 16 bits correspondientes al número de secuencia de una MU concreta (MU_{ACT}); ii) 64 bits relativos al instante de tiempo (p_{ACT}) en el que deberá reproducirse dicha MU_{ACT}; y iii) 8 bits que identificarán al grupo específico de receptores al que va destinado dicho mensaje. La longitud total de este paquete (incluida la parte obligatoria, [17]) es de 6 palabras de 32 bits.

B. Políticas de Selección de la Referencia Maestra

Cuando la fuente debe enviar un paquete RTCP APP ACT a los receptores, ésta debe rellenar sus campos con los parámetros temporales de una referencia de sincronización específica. La fuente, a partir de la información recibida en cada informe RTCP RR EXT enviado por cada receptor *i-ésimo*, puede calcular la tendencia o desviación de su estado de reproducción de la siguiente manera:

$$\gamma_i = [(MU_i - MU_{ini})/(p_i - p_{ini})]/\theta - 1 \qquad (3)$$

De este modo, la fuente puede calcular el instante de reproducción (p_{ACT}) en el que debería reproducirse una MU específica (MU_{ACT}) si se seleccionase como maestro el receptor *i-ésimo* (i.e. $\gamma_{master}=\gamma_i$):

$$p_{ACT} = p_{ini} + (MU_{ACT} - MU_{ini})/(\theta \cdot (1 + \gamma_{master})) \qquad (4)$$

En nuestra anterior propuesta de IDMS, la fuente seleccionaba un receptor fijo como la referencia de sincronización a la que debían ajustarse el resto de receptores. En el presente trabajo se ha mejorado el anterior

esquema, de modo que la fuente sea capaz de gestionar nuevas políticas para la selección de la referencia maestra para la sincronización: i) sincronización al receptor más rápido ($\gamma_{master}=\gamma_{fastest}$); ii) sincronización al receptor más lento ($\gamma_{master}=\gamma_{fslowest}$); iii) sincronización al punto de reproducción medio ($\gamma_{master}=\gamma_{mean}$); y iv) sincronización a la tasa nominal de la fuente ($\gamma_{master}=0$). La idoneidad de cada una de estas políticas se analizará en la Sección VI.

C. Ajustes de Reproducción Agresivos (Saltos y Pausas).

Considérese que el receptor *i-ésimo* está reproduciendo una MU específica -MU_i- en el instante p_i (punto local de reproducción). Este receptor consumirá las sucesivas MUs con una tasa (posiblemente desviada) de μ_i MU/s. Así, la MU_{ACT} será reproducida en el instante p'_{ACT}, que posiblemente no incida con el instante p_{ACT} (objetivo global de sincronización). Sea $\Delta_{n,i}$ la asincronía (en segundos), para la *n-ésima* MU, entre la evolución del punto de reproducción local (p'_{ACT}) y el punto de reproducción objetivo (p_{ACT}):

$$p_{ACT} = p'_{ACT} + \Delta_{n,i} = \left[p_i + \frac{1}{\mu_i}(MU_{ACT} - MU_i) \right] + \Delta_{n,i} \qquad (5)$$

Si $\Delta_{n,i}>0$, el estado de reproducción del receptor *i-ésimo* estará avanzado con respecto al punto de reproducción objetivo. Es por ello que deberá pausar su proceso de reproducción durante $\Delta_{n,i}$ segundos para sincronizarse, originando un posible efecto de *"congelación"* (Fig. 3). Como resultado, el retardo de reproducción para la siguiente MU, d_{n+1}, será incrementado (con lo que p_{n+1} se retrasará). En caso contrario, si $\Delta_{n,i}<0$, el estado de reproducción del receptor estará rezagado. En tal caso, el controlador de reproducción deberá saltar un determinado número de MUs con tal de minimizar la asincronía detectada. De esta manera, el retardo de reproducción para la siguiente MU, d_{n+1}, será reducido (con lo que p_{n+1} se avanzará). Como se mostrará en la Sección VI, dichos ajustes resultarán en un estado de sincronización global (dentro de unos límites admisibles).

D. Ajustes de Reproducción Suavizados (AMP).

El esquema AMP propuesto trata de evitar las anteriores discontinuidades de reproducción que pueden degradar significativamente la percepción de los usuarios sobre el servicio multimedia recibido. El diagrama de flujo de dicha propuesta se esquematiza en la Fig. 4. Inicialmente, el controlador de reproducción planifica el tiempo de servicio de las MUs almacenadas en buffer con una tasa nominal de $\mu_{n,i}=1/(s_{n,i})$. Cada receptor incluye su punto de reproducción local en cada informe RTCP RR EXT que envía con tal de facilitar a la fuente el conocimiento global de los estados de reproducción. Cuando se recibe un nuevo paquete RTCP APP ACT, se registra el punto de reproducción objetivo incluido en este mensaje y se lanza el proceso AMP.

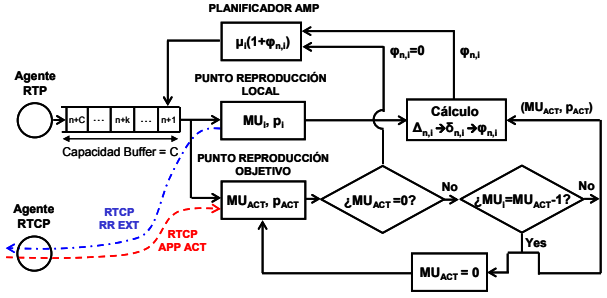


Fig. 4. Diagrama de Flujo del Esquema AMP

En este punto, dicho proceso se encargará de acelerar o ralentizar el proceso de reproducción con tal de distribuir la asincronía detectada ($\Delta_{n,i}$) entre cada una de las MUs restantes hasta alcanzar el punto de reproducción objetivo, como se ilustra en la Fig. 2. Esto se consigue incrementando/acortando el tiempo de servicio de cada *n-ésima* MU un valor de $\delta_{n,i}=(\Delta_{n,i})/(MU_{ACT}-MU_i)$ segundos (i.e. $s_{n,i}+\delta_{n,i}$). Sin embargo, deben considerarse límites acotados en la variación de la tasa de reproducción. Análisis subjetivos han mostrado que variaciones hasta el 25 % son frecuentemente imperceptibles y, dependiendo del contenido y de la frecuencia de las mismas, variaciones hasta el 50 % son algunas veces tolerables ([7] y [15]). Por tanto, se define un factor de ajuste ($\varphi_{n,i}$), acotado entre 0 y 25 % de la frecuencia nominal de la tasa de reproducción, que especificará la tasa de variación para cada *n-ésima* MU, cuyo valor será calculado, combinando (5) y (6), como sigue:

$$p_{ACT} = p_i + \frac{1}{\mu_i(1+\varphi_{n,i})}(MU_{ACT} - MU_i) \quad (6); \quad \varphi_{n,i} = \frac{1}{1+(\delta_{n,i}/s_{n,i})} \quad (7)$$

VI. Evaluación

La implementación y evaluación de la propuesta de IDMS han sido realizadas mediante NS-2 [21]. Se ha evaluado dicha propuesta en un escenario multicast con 4 LANs distribuidas. La fuente multimedia se ubicó en una de las LANs y cada uno de los 3 receptores se ubicó en cada una de las restantes LANs (Tabla I). La fuente envió un flujo CBR con una tasa de 200 kb/s (θ=25 MU/s). Adicionalmente, se configuró tráfico de *background* con tal de causar variabilidades de retardo (*jitter*) para el flujo multimedia. Se configuraron desviaciones de las tasas de reproducción de los receptores con unos valores más elevados que los típicos en osciladores comunes, que pueden oscilar entre 10-100 *ppm* [22], con tal de forzar mayores valores de asincronía entre los receptores. Además, con el objetivo de comprobar las capacidades dinámicas de intercambio de roles M/S, se cambiaron intencionadamente dichos valores en el punto intermedio de la simulación (en el quinto minuto), como se refleja en la Tabla I. La duración de la simulación fue de 10 minutos y el máximo umbral admisible de asincronía (τ_{max}) se estableció a 80 ms, con tal de forzar ajustes de sincronización antes de alcanzar un valor de 100 ms, que ya resulta perceptible y molesto en muchos escenarios [2].

A. Análisis y Resultados.

1) Instante Inicial de Reproducción. A pesar de los valores desiguales de *Round Trip Time* (RTT) estimados para cada receptor a partir de los informes RTCP RR enviados por éstos durante la sesión multimedia (Tabla I), puede apreciarse en todas las gráficas incluidas en la Fig. 5 y Fig. 6 como éstos estuvieron perfectamente sincronizados en el instante p_{ini}, calculado por la fuente de modo que el retardo de reproducción fuese común en todos ellos y de valor 500 ms (se ha enfatizado en la gráfica superior de la Fig. 5).

2) Sincronización al Receptor más Rápido. La gráfica superior de la Fig. 5 muestra la evolución de los retardos de reproducción en cada uno de los receptores durante la sesión multimedia cuando se seleccionó el receptor más rápido como la referencia maestra de sincronización, utilizando ajustes agresivos. Puede apreciarse como R1 fue el receptor más rápido ya que su tasa de reproducción presentó el valor de

skew más elevado (Tabla I). De este modo, cada vez que se detectó una asincronía superando el valor de τ_{max}, la fuente envió un nuevo paquete RTCP APP ACT, tomando como referencia el punto de reproducción de R1 ($\gamma_{master}=\gamma_{max}=\gamma_1$). Como resultado, los receptores más lentos tuvieron que saltar 0, 1 ó 2 MUs para sincronizarse ($\tau_{max}=2\cdot s_{n,i}=80$ *ms*). Sin embargo, no hubieron *"pausas"* en los procesos de reproducción de los receptores para esta política. El resumen de los ajustes de sincronización, para cada una de las políticas de selección de maestro adoptadas, se refleja en la Tabla II. También puede apreciarse en esta gráfica, así como en los valores de la cuarta columna de la Tabla II, una reducción progresiva de los retardos de reproducción en todos los receptores. Por tanto, aunque esta política de selección pueda ser adecuada en juegos de red colaborativos [12], si el receptor maestro reproduce las MUs con una tasa más rápida con la que fueron generadas, su aplicación puede suponer un vaciado gradual de los buffers de reproducción. La gráfica intermedia de la Fig. 5 ilustra el mismo proceso cuando se emplearon ajustes suavizados (AMP). En este caso, se puede apreciar como los receptores alcanzaron estados de sincronización más ajustados al punto de reproducción objetivo, puesto que distribuyeron el total de la asincronía detectada entre cada una de las MUs almacenadas en buffer hasta alcanzar el punto de reproducción objetivo. Así, la fuente tuvo que enviar menor número de mensajes de corrección (RTCP APP ACT) a los receptores.

Generalmente, cuando se utilizó AMP, en cada una de las políticas de selección, se evitó la ocurrencia de saltos/pausas, aunque el número total de MUs para las que se ajustó su tiempo de servicio fue mayor. Sin embargo, en ninguna caso, la cantidad de MUs ajustadas superó el 0,4 % (quinta columna en la Tabla II), un porcentaje muy reducido que muy posiblemente sea imperceptible por los usuarios.

3) Sincronización al Receptor más Lento. La gráfica inferior de la Fig. 5 ilustra la evolución de los procesos de reproducción de los receptores cuando se sincronizaron al receptor más lento ($\gamma_{master}=\gamma_{min}$), empleando AMP. En este caso, podemos observar como los procesos de reproducción de los receptores más rápidos se ralentizaron cada vez que se detectó una asincronía mayor que τ_{max}. Además, puede comprobarse en la tercera columna de la Tabla II, como cuando se utilizaron ajustes agresivos en esta política, los receptores tuvieron que *"pausar"* significativamente sus procesos de reproducción ($\Delta_{max}=82.2$ *ms* para R1). Dicha gráfica ilustra claramente el efecto de intercambio de roles M/S: inicialmente R3 fue el receptor más lento, pero se cambiaron intencionadamente los valores de desviación de las tasas de reproducción (Tabla I) con tal de convertir al receptor R2 en el nuevo maestro. Esta política de selección puede ser adecuada para garantizar igualdad de condiciones en aplicaciones de carácter competitivo [12], aunque en el caso que el receptor maestro (más rezagado) sea considerablemente más lento que la fuente, los buffers de los receptores podrían llenarse progresivamente, con el consiguiente incremento del retardo de reproducción y degradación de la percepción de tiempo real o pérdida de continuidad en el servicio multimedia.

4) Sincronización al Punto de Reproducción Medio. La gráfica superior de la Fig. 6 ilustra el mismo caso cuando los receptores se sincronizaron al punto intermedio de reproducción, calculado como la media de los puntos de reproducción locales de cada uno de los receptores. Esta política de selección minimiza el número de ajustes de reproducción y el valor de los mismos. Sin embargo, su aplicación no garantiza que se eviten situaciones de vaciado o desbordamiento de los buffers, puesto que la existencia de un receptor con un punto de reproducción extremadamente adelantado/rezagado tendrá un impacto significativo en el cálculo del punto de reproducción medio (objetivo). Es por ello que en todas las políticas de selección discutidas, la fuente no debe considerar puntos de reproducción excesivamente desviados, ya sea de modo accidental o malicioso, en el cálculo del punto de reproducción objetivo.

5) Sincronización a la Frecuencia Nominal de la Fuente. Esta política de selección de la referencia maestra soluciona las situaciones indeseadas que ocurren en los anteriores casos. En la gráfica inferior de la Fig. 6 se puede apreciar como utilizando dicha política y ajustes suavizados (AMP), los procesos de reproducción de los receptores siguieron una evolución uniforme durante la sesión multimedia, evitando, por tanto, situaciones de vaciado y desbordamiento de los buffers (si las condiciones de red se mantienen de manera estable). En este caso, la fuente actuó como un *receptor virtual* con un estado de reproducción *ideal* (sin desviaciones), puesto que conoce el valor de p_{ini}, de modo que cada vez que detectó un valor límite de asincronía (τ_{max}), tomó su propio estado de temporización como la referencia global de sincronización ($\gamma_{master}=0$). Utilizando este método, los receptores precisos no tienen que realizar ajustes significativos para adquirir IDMS. En la Tabla II se puede apreciar como utilizando ajustes agresivos en dicha política, el receptor R1 (más rápido) tuvo que realizar pequeñas pausas (Δ_{max}=23.7 ms), posiblemente imperceptibles, y los receptores más lentos tuvieron que saltar (descartar) un reducido número de MUs para conseguir el objetivo de sincronización global (IDMS). Por último, la Fig. 7 refleja la variación de la tasa de reproducción para cada uno de los receptores cuando se empleó dicha política combinada con el uso de AMP. Puede observarse como la tasa de reproducción fue manipulada en unos límites tolerables ($|\varphi_{max}|\leq 0.25$) cada vez que se requirieron ajustes de sincronización. Por último, la sexta columna de la Tabla II corrobora que el factor de ajuste máximo en cada uno de los casos fue siempre inferior al 25 % de la tasa nominal de envío/reproducción.

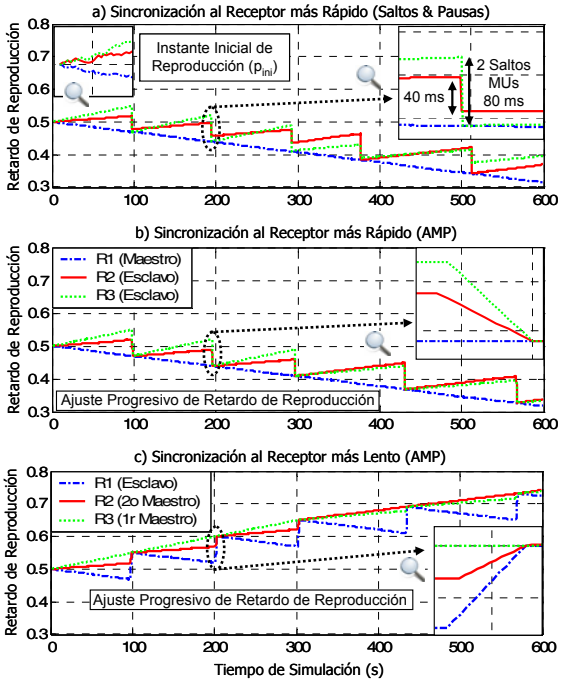


Fig. 5. Evolución del Retardo de Reproducción (Fastest/Slowest)

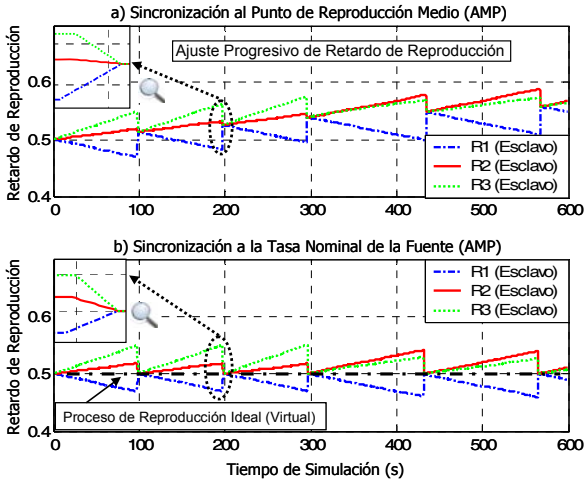


Fig. 6. Evolución del Retardo de Reproducción (Mean/Source).

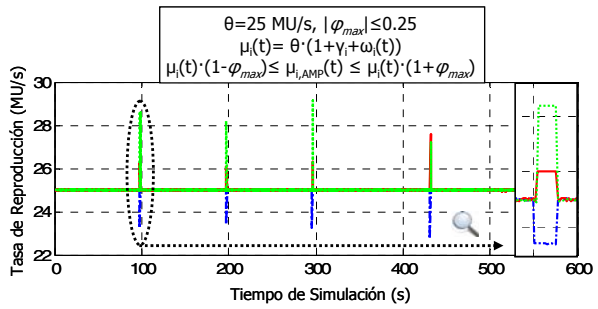


Fig. 7. Variación de la Tasa de Reproducción (Source).

TABLA I. PARÁMETROS DE LOS RECEPTORES

Receptor	*RTT Medio*	*Skew (γ)*	*Drift (ε)*
R1 (LAN 1)	44 ms	0.03 %	0.03 %
R2 (LAN 2)	125 ms	-0.02 % → -0.03 %	0.03 %
R3 (LAN 3)	208 ms	-0.05 % → -0.02 %	0.03%

TABLA II. RESUMEN DE AJUSTES DE REPRODUCCIÓN

Receptor		**Ajustes Agresivos**		**AMP**	
	Política Selección Maestro	- Saltos (%) / + Pausas (Δ_{max}) [MUs]	Variación Ocupación Buffer	Número Ajustes (%) [MUs]	φ_{max}
R1	+ Rápido	0 / 0	-184.2 ms	-	-
	+ Lento	0 / + 4 (82.2)	+215.8 ms	61 (0.4)	-0.16
	Medio	0 / + 6 (54.7)	+77.9 ms	56 (0.37)	-0.09
	Fuente	0 / + 5 (23.7)	≤τ_{max} ms	43 (0.3)	-0.08
R2	+ Rápido	- 7 (0.05) / 0	-129.8 ms	57 (0.38)	+0.23
	+ Lento	0 / + 3 (21.9)	+223.8 ms	64 (0.4)	-0.08
	Medio	0 / + 1 (8.9)	+158.2 ms	55 (0.37)	+0.06
	Fuente	- 3 (0.02) / 0	≤τ_{max} ms	48 (0.32)	+0.11
R3	+ Rápido	- 8 (0.05) / 0	-104.9 ms	52 (0.35)	+0.24
	+ Lento	0 / + 2 (14.5)	+235.4 ms	62 (0.4)	-0.05
	Medio	- 2 (0.01) / 0	+127.8 ms	53 (0.35)	+0.1
	Fuente	- 4 (0.025) / 0	≤τ_{max} ms	49 (0.33)	+0.12

a. Un total de 14967 MUs fueron enviadas durante la sesión multimedia

VII. CONCLUSIONES

Se ha presentado la combinación de una versión mejorada de una propuesta de IDMS y un novedoso esquema AMP que posibilita el mantenimiento de un estado de sincronización global (dentro de unos límites permisibles) en una sesión multimedia, minimizando la ocurrencia de discontinuidades (saltos y/o pausas) en los procesos de reproducción de los receptores. Posibles líneas de investigación futura incluyen: i) diseño de un esquema AMP que controle tanto el nivel de ocupación de buffer como la asincronía de reproducción; y ii) adaptar nuestra propuesta de IDMS para servicios interactivos (bidireccionales).

AGRADECIMIENTOS

Este trabajo ha sido financiado, parcialmente, por la Generalitat Valenciana, bajo su Programa de Apoyo a la Investigación y Desarrollo (PAID) en el Proyecto 2010/009 y por la Universitat Politècnica de València (UPV), bajo su PAID en el Proyecto PAID-01-10.

REFERENCIAS

[1] Boronat F., Lloret J., and García M., "Multimedia group and inter-stream synchronization techniques: A comparative study", Inf. Syst, 34, 1, 108-131, March 2009.
[2] V.Deventer M.O., Stokking H., Niamut O.A., Walraven F.A., Klos V.B., "Advanced Interactive Television Service Require Synchronization", *IWSSIP 2008*, Bratislava, June 2008.
[3] Vaishnavi I., Cesar P., Bulterman D., and Friedrich O., "From IPTV services to shared experiences: challenges in architecture design", IEEE Conference on Multimedia & Expo, *ICME 2010*, Singapore, July 2010.
[4] Boronat F., Guerri J.C., and Lloret J., "An RTP/RTCP based approach for multimedia group and inter-stream synchronization", Multimedia Tools and Applications Journal, Vol. 40 (2), 285-319, June 2008.
[5] Boronat, F., Montagud M., and Guerri J.C., "Multimedia group synchronization approach for one-way cluster-to-cluster applications", IEEE 34th Conference on Local Computer Networks, *LCN 2009*, Zürich, October 2009.
[6] Montagud M., Boronat F., "On the use of Adaptive Media Playout for Inter-Destination Synchronization", IEEE Communications Letters (aceptado para su publicación en 2011).
[7] Su Y., Yang Y., Lu M., Chen H., "Smooth Control of Adaptive Media Playout for Video Streaming", IEEE Transactions on Multimedia, Vol.1, No. 7, November 2009.
[8] ETSI TISPAN, "IMS-based IPTV stage 3 specification", TS 183 063 v3.4.6 (2010-12).
[9] Ishibashi Y., Tsuji A., and Tasaka S., "A Group Synchronization Mechanism for Stored Media in Multicast Communications", 6th Annual Joint Conference of the IEEE Computer and Communications Societies (*INFOCOM*), Kobe (Japan), April 1997.
[10] Ishibashi Y. and Tasaka S., "A distributed control scheme for causality and media synchronization in networked multimedia games", International Conference on Computer Communications and Networks, Miami (USA) , October 2002.
[11] Ishibashi Y. and Tasaka S.,, "A group synchronization mechanism for live media in multicast communications", *IEEE GLOBECOM' 97*, November 1997.
[12] Hashimoto T., Ishibashi Y., "Group Synchronization Control over Haptic Media in a Networked Real-Time Game with Collaborative Work", *Netgames'06*, Singapore, October 2006.
[13] Ishibashi Y., Tasaka S., Ogawa H., Media Synchronization Quality of Reactive Control Schemes, IEICE Transactions on Communications, Vol.E86-B, No.10, October 2003.
[14] Kalman M., Steinbach E., Girod B., "Adaptive Media Playout for Low-Delay Video Streaming over Error-Prone Channels", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 14 (6), 285-319, June 2004.
[15] Chuang H., Huang C., and Chiang T., "Content-Aware Adaptive Media Playout Controls for Wireless Video Streaming", IEEE Transactions on Multimedia, Vol.9, No. 6, October 2007.
[16] Park S., Kim J., "An adaptive media playout for intra-media synchronization of networked-video applications" Journal of Visual Communications & Image Representation, Vol. 19, 106-120, 2008.
[17] Schulzrinne H., Casner S., Frederick R. and Jacobson V., "RTP: A Transport Protocol for Real-Time Applications", RFC-3550, July 2003.
[18] Begen A., Perkins C. and Ött, J., "On the use of RTP for Monitoring and Fault Isolation in IPTV", IEEE Network, Vol. 24 (2), April 2010.
[19] Ott J., Perkins C., "Guidelines on Extending the RTP Control Protocol (RTCP)", RFC 5968, September 2010.
[20] Boronat F., Montagud M., and Vidal V., "A More Realistic RTP/RTCP-Based Simulation Platform for Video Streaming QoS Evaluation", Journal of Mobile Multimedia, Vol.7 No.1&2, pp 66-88, April 2011.
[21] NS-2 Simulator Home Page, http://www.isi.edu/nsnam/ns
[22] Ferrari F., Meier A., Thiele L., "Accurate Clock Models for Simulating Wireless Sensor Networks", *SIMUTools 2010*, Torremolinos (Spain), March 2010.

Web Oculta del Lado Cliente: Escala de Crawling

Manuel Álvarez, Fidel Cacheda, Rafael López-García, Víctor M. Prieto
Departamento de Tecnologías de la Información y las Comunicaciones,
Universidade da Coruña
Campus de Elviña s/n, A Coruña, España
mad@udc.es, fidel@udc.es, rafael.lopez@udc.es, victor.prieto@udc.es

***Resumen*—El objetivo de este estudio consiste en la definición de una escala para la clasificación de los sistemas de crawling en base a su efectividad accediendo a la Web Oculta del "lado cliente". Para ello se realiza un análisis exhaustivo de las diferentes tecnologías de lado cliente usadas en las páginas Web 2.0 para crear una escala con distintos niveles de dificultad. Para realizar la clasificación de los diferentes sistemas de crawling en base a la escala definida, se ha creado un sitio web contra el que comprobar su efectividad. También se proponen diferentes métodos de evaluación de la efectividad de los crawlers en base a la escala. Para la realización del estudio se han considerado tanto los crawlers de los principales buscadores web como otros sistemas de crawling OpenSource y comerciales.**

***Palabras Clave*—Web Search, crawler, Web Oculta, Web Spam, JavaScript, Redirection Spam**

I. Introducción

La WWW constituye actualmente el mayor repositorio de información jamás construido. Pero tan importante como almacenar gran cantidad de información es la gestión de la misma para permitir localizar, acceder y recopilar la que satisface las necesidades de un usuario. Los sistemas que permiten esta tarea son los crawlers, programas capaces de procesar y analizar la Web. Se puede decir que un crawler recorre los diferentes URL descubiertos, en un cierto orden, analiza el contenido descubierto y lo procesa para obtener nuevos URL que serán tratados. Existen diferentes tipos de sistemas de crawling en función de su ámbito: globales, orientados a recuperar toda o gran parte de la información de la Web, o dirigidos, orientados a una parte concreta, más reducida, de la Web.

Desde sus orígenes los sistemas de crawling han tenido que enfrentarse a sitios web orientados a usuarios humanos: navegaciones a través de menús emergentes, diferentes capas de datos que se ocultan o hacen visibles dependiendo de las acciones del usuario, sistemas de redirecciones o mecanismos de mantenimiento de sesión. El conjunto de tecnologías que posibilitan la inclusión de los aspectos comentados en los sistemas de crawling representa un gran desafío, debido a que obligan a implementar técnicas para tratarlas de forma adecuada. Los sitios web que utilizan estas tecnologías forman la que se conoce como Web Oculta [1], por contener información que no es alcanzable por la mayor parte de los sistemas de crawling. A su vez, ésta se puede dividir en Web Oculta del lado cliente o del lado servidor. En este artículo se analiza cómo tratan las tecnologías del lado cliente (Web Oculta del lado cliente) los crawlers, para intentar determinar si el uso de dichas tecnologías afecta a la visibilidad de aquellas páginas o sitios Web que las usen. Se ha realizado un análisis de las tecnologías del lado cliente más utilizadas en la creación de páginas web, como son JavaScript [2], AJAX (Asynchronous, JavaScript, And XML) [3], VbScript [4] y Flash [5]. También se han analizado problemáticas como Redirection Spam [6] y Cloacking [7] para intentar detectar la utilización de estas tecnologías para fines ilícitos.

Una vez analizadas las diferentes tecnologías, se ha realizado una enumeración de las dificultades que se le pueden presentar a los crawlers durante su recorrido, generando una escala con diferentes niveles. Para clasificar los sistemas de crawling en base a dicha escala, se ha construido un sitio Web que genera de forma dinámica enlaces, según las dificultades propuestas. Se han obtenido resultados tanto para los crawlers de los principales buscadores, como para los crawlers OpenSource [8] [9] y aquellos con licencia comercial.

La estructura del artículo es la siguiente. En la sección II se comentan los trabajos relacionados, los sistemas de crawling de la Web oculta del lado cliente y la problemática del Web Spam. La sección III introduce las tecnologías de lado cliente y su uso para la construcción de sitios web. La sección IV comenta las ocurrencias más habituales de las tecnologías explicadas para la generación de sitios web. A partir de dichas ocurrencias se crean una serie de niveles de dificultad que deberían de ser tratados por un crawler que pretenda obtener toda la información de la Web Oculta del lado cliente. A continuación se define la escala propuesta para la clasificación de los crawlers, junto con cuatro métodos que permiten evaluarlos respecto a dicha escala. La sección V describe el sitio web creado para la realización de experimentos. En la sección VI se discuten los resultados obtenidos para los diferentes crawlers y por último en las secciones VII y VIII se comentan las conclusiones obtenidas y posibles trabajos futuros.

II. Trabajos relacionados

Son muchos los estudios relacionados con el tamaño de la Web y con la caracterización de su contenido. Sin embargo, menos han sido los que se han ocupado de clasificarla en base a la dificultad que presenta hacia los crawlers. Según los datos presentados en [10] y [11] actualmente el 90% de las páginas web usan JavaScript. En 2006, M. Weideman y F. Schwenke [12] publicaron un estudio que analizaba la importancia del uso de JavaScript en la visibilidad de un sitio Web, concluyendo que la mayor parte de los crawlers no lo tratan.

Desde el punto de vista de los sistemas de crawling, son numerosos los trabajos orientados a crear sistemas que sean capaces de tratar la Web Oculta. Los crawlers de la Web Oculta del lado servidor se ocupan de la amplia cantidad de sitios Web en los cuales se accede al contenido mediante

ISBN: 978-84-694-5948-5

formularios. Este tipo de contenido es de gran cantidad y calidad. Existen investigaciones que abordan los retos marcados por la Web Oculta del lado servidor, destacando HiWE [13] por ser uno de los sistemas pioneros. También Google [14] presentó las técnicas que utiliza para el acceso a información a través de formularios. Respecto a la Web Oculta de lado cliente [15], son menos los estudios y básicamente se resumen en las siguientes dos aproximaciones: acceder al contenido y enlaces mediante intérpretes que permitan ejecutar los scripts [16] [17], o bien utilizar mini-navegadores como en el sistema propuesto por M. Álvarez et al. en [18] y [19].

Por otra parte, debido a que el uso de tecnologías de lado cliente puede utilizarse para "engañar" a los sistemas de crawling en su tarea, han aparecido varios trabajos relacionados con detectar lo que se conoce como Web Spam. Dentro del Web Spam existen diversas técnicas tales como el Cloacking [7] [20] [21] [22] o Redirection Spam [6] [22]. La primera de estas técnicas pretende detectar cuándo es un usuario normal y cuándo es un crawler el que realiza la petición de la página. Si el que realiza la petición es un motor de búsqueda el sitio Web mostrará un contenido diferente al que le mostraría si fuera un navegador de un usuario. Las técnicas de Redirection Spam pretenden ocultar las redirecciones para ser ejecutadas únicamente en un navegador. En ambos casos se consigue "mentir" al buscador de forma que indexe unos contenidos diferentes a los que realmente son.

Sin embargo, no se conocen escalas que permitan clasificar la efectividad de los sistemas de crawling respecto a su nivel de tratamiento de las tecnologías de la Web Oculta del lado cliente.

III. Tecnologías de lado cliente usadas en la construcción de sitios web

A continuación se enumeran las tecnologías de lado cliente más habituales en la creación de sitios web, normalmente usadas para mejorar la experiencia de usuario, generando contenido y enlaces de forma dinámica, en función de las acciones del usuario.

- JavaScript, dialecto del estándar ECMAScript, es un lenguaje imperativo y orientado a objetos. Permite la generación dinámica de la interfaz y su modificación en base a los eventos generados por el usuario, a través de una implementación del DOM [23].
- Applet [24], componente Java de una aplicación que se ejecuta en el cliente web. Permite tener acceso casi completo a la máquina, con velocidades similares a la de lenguajes compilados. Permite crear soluciones más escalables al número de usuarios.
- AJAX, conocida técnica de desarrollo que permite crear aplicaciones web interactivas. Utiliza JavaScript para el envío y la recepción de la petición/respuesta asíncrona del servidor, y normalmente JSON [25] como lenguaje para encapsular la información recibida.
- VbScript, lenguaje interpretado creado por Microsoft como variante del lenguaje Visual Basic. Se ha utilizado como parte esencial de aplicaciones ASP [26] y su funcionalidad es similar a la que aporta JavaScript.
- Flash, aplicación que permite crear interfaces vectoriales. La programación de dichas interfaces se hace mediante ActionScript, lenguaje de características parecidas a JavaScript y VbScript.

IV. Definición de la escala

A partir de las diferentes tecnologías descritas en el apartado anterior y del análisis de su uso por parte de los diseñadores de sitios web, se han identificado los siguientes tipos de ocurrencias:

- Enlaces de texto, que constituyen el nivel más básico de la escala.

```
<a href="a_11000100100000000000000000_test...html">Luis Ramirez Lucena</a>
```

- Navegaciones simples generadas con JavaScript, VbScript o ActionScript. Incluye enlaces generados mediante "document.write()" o funciones similares en otros lenguajes, que permiten añadir nuevos enlaces al HTML de forma dinámica.

```
<a href="JavaScript: ">Paolo Boi </a>
```

- Navegaciones generadas mediante un Applet, dentro de las cuales existen dos tipos a su vez, aquellas generadas a partir de un URL que se le pasa como argumento al Applet y aquellas otras cuyo URL está definido como una cadena en su código compilado.
- Navegaciones generadas mediante AJAX.
- Menús desplegables, generados mediante la ejecución de código script asociado a algún evento.
- Navegaciones generadas desde Flash. Existen dos tipos: aquellas que reciben el URL como argumento desde el código HTML y aquellas que lo tienen definido dentro del propio código ActionScript.
- Enlaces definidos como cadenas en ficheros .java, .class, u otro tipo de ficheros.
- Navegaciones generadas a partir de funciones que pueden estar definidas en algún lenguaje de script cuyo código puede estar embebido en el HTML o en un fichero externo.
- Navegaciones generadas mediante diferentes tipos de redirecciones:
  - Redirección en la etiqueta meta.
  - Redirección creada en el evento onLoad de la etiqueta body.
  - Redirección JavaScript, que se ejecutará en el momento en que se cargue la página.
  - Redirección creada sobre un Applet, al cargar la página con el Applet.
  - Redirección Flash, al procesar la página con su correspondiente fichero SWF.

De forma adicional, las navegaciones generadas con cualquiera de los métodos identificados pueden crear direcciones URL de tipo absoluto o relativo. Para las direcciones generadas a partir de cualquier lenguaje de script, se pueden distinguir los siguientes métodos de construcción:

- Una cadena estática dentro del Script.

```
menu_static_embedded_relative(){document.location="a_1001...html";}
```

- Una concatenación de cadenas.

```
function menu_concatenated_embedded_relative(){
var out="";out="a_1001001010000000000000000_test_menu"+
"_concatenated_embedded_relative.html"; document.location=out;}
```

- Ejecución de una función que construye en varios pasos el URL.

```
function menu_special_function_embedded_relative(){
var a1="win",a2="dow",a3=".location.",a4="replace",a5;
a5="('a_1001000110000000000000000_test_menu_special_function";
var a6="_embedded_relative.html')";var i,url="";
for(i=1;i<=6;i++) {url+=eval("a"+i);} eval(url);}
```

Por otra parte, los diferentes métodos enumerados pueden aparecer combinados. Por ejemplo, algunos sitios web construyen menús desplegables de forma dinámica, mediante la utilización de funciones "document.write()". El número de posibilidades es inabordable. Por este motivo, para este estudio se ha considerado un subconjunto reducido, pero suficientemente significativo, que se muestra en la Fig. 1. Consta de 70 niveles que representan los tipos básicos a partir de los cuales se podrían obtener el resto de casos, mediante combinaciones. El número de niveles considerados podría ser mayor, pero no aportaría más información ni sobre el uso en la Web de las tecnologías del lado cliente, ni tampoco sobre los métodos que usan los crawlers para descubrir enlaces. Siguiendo con el ejemplo de combinación comentado, no es necesario considerarlo en el estudio ya que se ha incluido de forma independiente el tratamiento de los menús y de navegaciones generadas con "document.write()". A partir de la información obtenida para los dos casos base, se podría concluir la capacidad del crawler para tratar el caso comentado.

Tipo de ruta	Tecnología	Ubicación	Tipo de cadena		Número
Relativa / Absoluta	Texto	Embebido	Estática		1 – 2
	JavaScript	Embebido / Externo	Estática / Concatenación / Función Especial		3 – 38
	Document.Write()				
	Menu JavaScript				
	Enlace en .java	Externo	Estática		39 – 40
	Enlace en .class	Externo	Estática		41 – 42
	Applet-Enlace HTML	Embebido	Estática		43 – 44
	Applet-Enlace Class	Externo	Estática		45 – 46
	Flash-Enlace HTML	Embebido	Estática		47 – 48
	Flash-Enlace SWF	Externo	Estática		49 – 50
	AJAX	Embebido	Estática		61 – 62
	JavaScript con marcador #	Embebido	Estática / Función Especial		63
					66
	VbScript	Embebido	Estática / Función Especial		67
					70
Relativa / Absoluta	Redirección	Externo	Estática	Etiqueta meta	51 – 52
				Etiqueta body	53 – 54
				JavaScript	55 – 56
				Applet	57 – 58
				Flash	59 – 60

Fig. 1. Combinaciones de los tipos de enlaces

A partir de los 70 niveles se propuso una primera agrupación en base a las tecnologías, los métodos de construcción de cadenas, la ubicación del código y los tipos de rutas del enlace. Se creó un sitio web (ver sección V) sobre el que se realizaron las pruebas mostradas en la sección VI. En base a los resultados obtenidos, se realizó un reagrupamiento de niveles teniendo en cuenta el número de crawlers que los trataron, consiguiendo de esta forma agrupar los niveles que presentaban igual dificultad para los crawlers, además de ordenarlos por complejidad.

La Fig. 2 muestra la escala propuesta. Está formada por 8 niveles, que representan de menor a mayor, la capacidad para tratar la Web Oculta del lado cliente. En los niveles más bajos se encuentran, entre otros, los enlaces de texto, redirecciones sencillas y enlaces de JavaScript generados con "document.write()" o menús con cadenas estáticas. En los niveles de dificultad alta aparecen, principalmente, enlaces en Applets, Flash, AJAX, VbScript o redirecciones complejas.

Nivel	Descripción	Escenarios probados
1	Enlace de texto	1,2
2	JavaScript/Document.Write/Menu – Cadena Estática – Embebida	3, 4, 15, 16, 27, 28
	JavaScript – Cadena concatenada – Embebida	6
3	Redireccion HTML/onBody/JavaScript	51, 52, 53, 54, 55, 56
	JavaScript con # – Cadena Estática – Embebida	63,64
	VbScript – Cadena Estática – Embebida	67,68
4	VbScript - Función Especial - Embebida	70
5	JavaScript/Document.Write – Cadena Estática – Externa/Embebida	9, 10, 18
	Document.Write/Menu – Cadena Estática – External	21, 22, 33, 34
	Menu – Cadena concatenada – Embebida	30
6	JavaScript/Document.Write/Menu – Cadena concatenada – External	12, 24, 36
	Applet – Cadena Estática en HTML	43,44
7	JavaScript – Cadena concatenada – Embebida – Relativa	5
	JavaScript – Función Especial – Embebida	7,8
	JavaScript – Cadena concatenada – Externa – Relativa	11
	JavaScript – Función Especial – Externa – Relativa	13,14
	Document.Write – Cadena concatenada – Embebida/Externa – Relativa	17,23
	Document.Write – Función Especial – Embebida/Externa	19, 20, 25, 26
	Menu – Cadena concatenada – Embebida/Externa – Relativa	29,35
	Menu – Función Especial – Embebida/Externa	31, 32, 37, 38
	Enlace en .class	41,42
	Enlace Ajax – Absoluta	62
8	Enlace .java	39,4
	Applet – Cadena Estática en .class	45,46
	Flash – Cadena Estática en HTML/SWF	47, 48, 49, 50
	Redirección Applet/Flash	57, 58, 59, 60
	Enlace Ajax – Relativa	61
	JavaScript con # – Función Especial – Embebida	65,66
	VbScript – Función Especial – Embebida	69

Fig. 2. Clasificación de los enlaces por dificultad

Una vez definida la escala, para poder clasificar los diferentes sistemas de crawling según el nivel de complejidad de los "enlaces" que traten, se proponen los siguientes métodos de evaluación:

- Media simple: trata todos los escenarios definidos, sin considerar su dificultad. Da una idea de qué crawlers tratan mayor número de escenarios y por tanto prestan mayor atención a la Web Oculta.
- Máximo nivel accedido: este modelo ordenará los crawlers según el nivel más alto de dificultad que son capaces de tratar. Se entiende que un crawler que obtiene un nivel máximo i tiene capacidad para procesar los enlaces de ese nivel e inferiores, aunque debido a motivos tales como bajo PageRank de la página web, pueda no llegar a analizarlos.
- Media ponderada: se asigna a cada escenario tratado un valor entre 0 y 1, que depende del número de crawlers que han sido capaces de procesarlo (0 cuando todos los crawlers han sido capaces de tratarlo). Este método indica qué crawlers serán los que puedan obtener mayor número de recursos de elevada dificultad en la Web Oculta de lado cliente, o recursos que la mayoría no alcanza.
- Ocho niveles: este modelo asigna 1 punto a cada nivel tratado. Si un crawler consigue procesar todos los escenarios de un nivel obtendrá 1 punto, es decir por cada escenario procesado con éxito consigue $1/n$, donde n es el número total de escenarios definidos para ese nivel.

Para los métodos de máximo nivel y ocho niveles, se obtienen valores entre 0 y 8, siendo 8 el nivel más alto. Para el resto de casos, se aplica una normalización para facilitar la comparación de resultados.

Aunque los métodos propuestos, por separado, no proporcionan una información concluyente respecto a la capacidad de los crawlers para el tratamiento de la Web Oculta del lado cliente, sí lo hacen si se consideran de forma conjunta, como puede verse en la sección VI de comparación de crawlers.

V. Sitio Web

Para poder comprobar cómo tratan los diferentes niveles los sistemas de crawling existentes, se ha creado un sitio web contra el que realizar los experimentos. En el sitio web "jstestingsite"[1], se han creado 70 enlaces, representando los 70 niveles definidos por la escala, usando y combinando las tecnologías explicadas. Con la finalidad de incentivar la indexación del sitio Web se han tomado diversas medidas, como incluirlo desde la página web del Departamento Tecnologías de la Información y las Comunicaciones, elaborar el contenido en inglés y añadir a cada nivel, la biografía de un ajedrecista. Esto último se ha hecho para evitar que los crawlers cataloguen la pagina como Web Spam. En la Fig. 3 se muestra la página principal del prototipo de sitio web. Se identifican las siguientes partes:

- En la parte superior, de izquierda a derecha, aparecen los enlaces en Menú JavaScript, los enlaces generados en el Applet y finalmente los enlaces en Flash.
- En el centro de la página aparece una tabla dividida en 4 columnas, primero el número que identificará a cada test, a continuación una pequeña descripción del test y finalmente el enlace relativo y absoluto.
- En la parte inferior, tras la tabla, aparece el contenido.

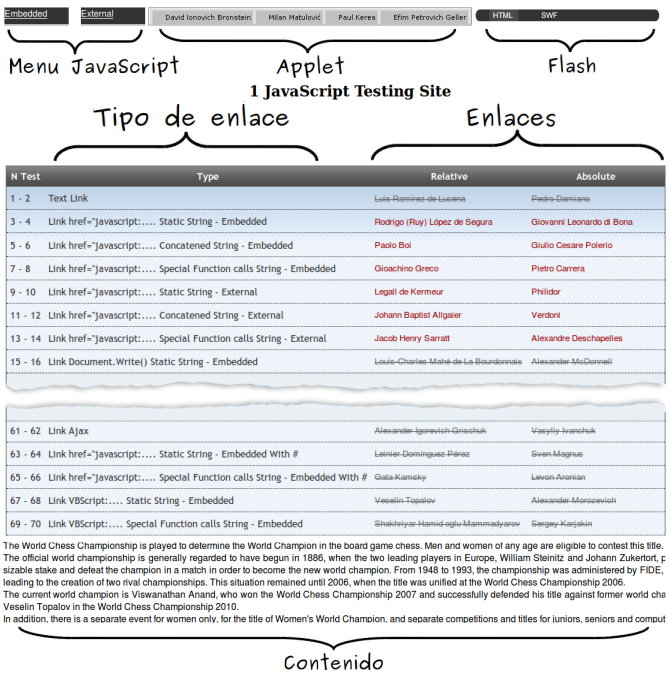


Fig. 3. Página principal del estudio

[1]http://www.tic.udc.es/~mad/resources/projects/jstestingsite/

Por otro lado se ha creado una página de resultado para cada una de las pruebas, de tal modo que si el crawler es capaz de acceder a dicho contenido significa que ha sido capaz de procesar el enlace. En la Fig. 4 se muestra una página de resultado como ejemplo, formada por los siguientes elementos:

- En la parte superior, en el centro, se muestra el número y nombre del test.
- En la parte superior izquierda aparece el código del test, una máscara binaria que representa numéricamente las características que tiene o no tiene la prueba.
- En el lateral izquierdo se muestra una tabla que enumera las características de la prueba.
- En la parte central se incluye la vida de un maestro ajedrecista.

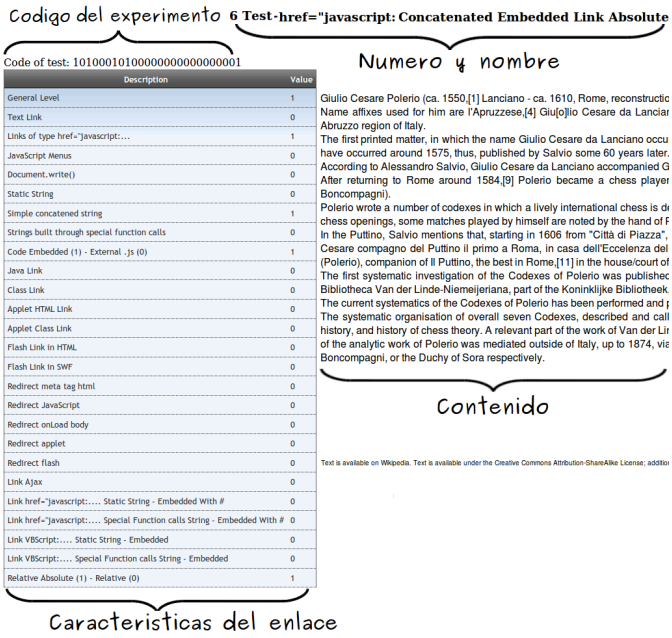


Fig. 4. Página de resultado de un test

VI. Resultados Experimentales

Esta sección incluye los experimentos realizados y los resultados obtenidos. Se han realizado pruebas contra el sitio web definido con crawlers OpenSource y comerciales, y con crawlers usados por los principales buscadores web. Para cada crawler se han obtenido dos tipos de resultados; información de indexación de contenido recopilado a través de los repositorios, en los que cada sistema almacena los documentos para permitir realizar búsquedas, e información de acceso recopilada a partir de los ficheros de log del servidor web del sitio crawleado. Tras esto se realiza una comparativa de los resultados de ambos tipos, atendiendo tanto al total de enlaces procesados como a los tipos de enlaces que han logrado tratar. Por último se presentan las puntuaciones obtenidas por cada crawler según los 4 métodos de evaluación definidos sobre la escala propuesta.

Para la ejecución de los crawlers OpenSource y comerciales se utilizaron las configuraciones que permitían maximizar el nivel de exploración de tecnologías del lado cliente. En el caso de los crawlers de buscadores web, en primer lugar se

analizaron los logs del servidor web que proporciona acceso al prototipo, para identificar los crawlers que estaban accediendo a alguna página del servidor. En base al listado publicado en la página oficial de robots [27], al User-Agent y dirección IP se determinó a qué buscadores pertenecen, y se eliminaron aquellos que no eran independientes, es decir que dependen del crawler de un tercero, y aquellos que forman parte de compañías de marketing. Para acelerar el proceso de visita e indexación se ha dado de alta manualmente el sitio web en los siguientes crawlers: Google[2], Bing[3], Yahoo![4], PicSearch[5] y Gigablast[6].
Para la generación automática de resultados de los logs del servidor, se implementó una herramienta que parte del listado de robots creado, añadiéndole las IPs y User-Agents de los crawlers OpenSource y comerciales, para automáticamente determinar qué crawlers han sido capaces de solicitar cada uno de los recursos expuestos por el servidor web.

A. Crawlers OpenSource y comerciales

Se realizó un análisis de 24 crawlers, que se muestran en la Fig. 5.

Crawler	Licencia
Advanced Site Crawler, Essential Scanner, GsiteCrawler, Heritrix, Htdig, ItSucks, Jcrawler, Jspider, Larbin, MnogoSearch, Nutch, Open Web Spider CS, Oss, Pavuk, Php Crawler, WebHTTrack	Free
JOC Web Spider, Mnogosearch, Visual Web Spider, Web Data Extractor, Web2Disk , WebCopier Pro	Shareware

Fig. 5. Crawlers OpenSource/Comerciales

Entre las características que comparten destacan: opciones de uso de proxy, autenticación (en ocasiones mediante formulario), limitaciones en número de documentos, diferentes protocolos, exclusión/inclusión de extensión de ficheros, opciones de dominio/directorio, cookies, User-Agent, Logging, trabajo con formularios, etc.

De esta lista se han seleccionado los 7 que mejores características presentan para el tratamiento de tecnologías de lado cliente. Entre los OpenSource, se seleccionaron los siguientes:

- Nutch [8], crawler y buscador basado en Lucene. Desarrollado en Java y con arquitectura basada en Hadoop, lo que permite su implantación como crawler distribuido. Permite extraer enlaces tanto de código JavaScript como de Flash, aunque utilizando heurísticas muy sencillas.
- Heritrix [9], un crawler conocido por ser usado por Internet Archive[7]. En lo que refiere al tratamiento de JavaScript lo hace mediante el uso de expresiones regulares, de forma muy similar a como lo hace Nutch.
- Pavuk [28], crawler que destaca por permitir rellenar formularios de forma automática e intentar obtener enlaces en JavaScript mediante el uso de expresiones regulares.
- WebHTTrack, crawler escrito en C que permite analizar ficheros Java, Flash e intenta obtener enlaces en JavaScript a partir de diferentes heurísticas. A diferencia de los anteriores realiza comprobaciones en busca de cadenas tales como “Object.Write”, “document.location”, “window.replace”, etc.

De los crawlers comerciales se seleccionaron:

- Teleport [29], crawler con cierto reconocimiento sobre todo en la década de los 90, y que incluye soporte para análisis de JavaScript.
- Web2Disk [30], crawler desarrollado por la empresa Inspyder. Permite un análisis básico o avanzado de JavaScript con su correspondiente diferencia en resultados y en tiempo de cómputo.
- WebCopierPro [31], crawler que permite procesar eventos dinámicos, analizar código JavaScript y Flash en búsqueda de enlaces.

B. Resultados de crawlers OpenSource y comerciales

En primer lugar se analizan los resultados en función del contenido del sitio web que ha sido indexado por los diferentes crawlers OpenSource y comerciales. Esto tiene lugar analizando los diferentes repositorios generados por los crawler durante su operación. El crawler que mejores resultados obtiene es WebCopierPro (tabla de la izquierda de la Fig. 6) con un 57,14% de los niveles procesados, seguido de Heritrix con 47,14% y Web2Disk con 34,29%. Pocos de ellos obtienen valores por encima de 25% en la mayoría de los tipos de enlaces, no siendo capaces de obtener enlaces en muchos casos. Del mismo modo es importante observar los bajos resultados obtenidos en el apartado de redirecciones, sobre todo en el caso de WebCopierPro que no es capaz de tratar ninguna, cuando en niveles de mayor complejidad obtiene resultados del 100%. Ninguno de los crawlers alcanza el 100% en las redirecciones. Esto es debido a que ninguno de ellos ha sido capaz de procesar páginas con redirecciones incrustadas en Applets o Flash. Son capaces de descargar la página, pero no ejecutan el Flash o Applet que genera la redirección.

Analizando los resultados según el tipo de enlace, mostrados en la Fig. 7, se obtiene una visión sobre qué tipos de enlaces son procesados por un mayor número de crawlers. Sin prestar atención al 100% que consiguen para los enlaces de texto se observa que: Consiguen atravesar entre el 35% y el 40% de los niveles de Href=”javascript ; “document.write()” ; Menu links ; Links with # y VbScript. Muy por debajo está el 7% conseguido en los enlaces de ficheros class o Java y aquellos generados mediante AJAX. Ninguno ha conseguido en enlaces de Flash, puede ser debido a la poca atención de los crawlers sobre estos tipos de enlaces o a la dificultad que entraña obtenerlos.

Como se muestra en la Fig. 8, debido a que la mayoría de los crawlers trabajan buscando URL’s con expresiones regulares, el porcentaje de enlaces encontrados cae de un 42.52% en los considerados enlaces estáticos a un 27.38% en el caso de enlaces generados mediante la concatenación de cadenas y finalmente a un 15.18% en el caso de enlaces generados con funciones. Se concluye que la probabilidad de encontrar enlaces mediante expresiones regulares o tratamiento de texto es inversamente proporcional a la dificultad con la que se genera el enlace.

Resumiendo los datos de la Fig. 6 se puede decir que sólo una tercera parte de los enlaces generados con tecnologías

[2]http://www.google.es/addurl/
[3]http://www.bing.com/webmaster/SubmitSitePage.aspx
[4]http://siteexplorer.search.yahoo.com/submit
[5]http://www.picsearch.com/menu.cgi?item=FAQ
[6]http://www.gigablast.com/addurl
[7]http://www.archive.org/

	REPOSITORIOS												LOGS			
Tipo de enlace	Heritrix		Nutch		Pavuk		Teleport		Web2Disk		WebCopierPro		WebHTTrack		Teleport		WebHTTrack	
Texto	2	100,00%	2	100,00%	2	100,00%	2	100,00%	2	100,00%	2	100,00%	2	100,00%	2	100,00%	2	100,00%
Href="javaScript:..."	6	50,00%	3	25,00%	0	0,00%	3	25,00%	5	41,67%	12	100,00%	3	25,00%	4	33,33%	3	25,00%
Document.write	6	50,00%	3	25,00%	0	0,00%	2	16,67%	4	33,33%	12	100,00%	2	16,67%	3	25,00%	2	16,67%
Menu	6	50,00%	3	25,00%	0	0,00%	2	16,67%	4	33,33%	12	100,00%	2	16,67%	3	25,00%	2	16,67%
Flash	0	0,00%	0	0,00%	0	0,00%	0	0,00%	0	0,00%	0	0,00%	0	0,00%	0	0,00%	2	50,00%
Applet	2	50,00%	0	0,00%	0	0,00%	0	0,00%	0	0,00%	0	0,00%	2	50,00%	0	0,00%	2	50,00%
Redirecciones	6	60,00%	6	60,00%	2	20,00%	6	60,00%	4	40,00%	0	0,00%	6	60,00%	6	60,00%	6	60,00%
Class/Java	0	0,00%	0	0,00%	0	0,00%	0	0,00%	0	0,00%	2	50,00%	0	0,00%	0	0,00%	0	0,00%
Ajax	0	0,00%	1	50,00%	0	0,00%	0	0,00%	0	0,00%	0	0,00%	0	0,00%	0	0,00%	0	0,00%
Mediante #	2	50,00%	2	50,00%	0	0,00%	2	50,00%	2	50,00%	0	0,00%	2	50,00%	3	75,00%	2	50,00%
VbScript	3	75,00%	3	75,00%	0	0,00%	3	75,00%	3	75,00%	0	0,00%	2	50,00%	3	75,00%	2	50,00%
Enlaces cadena estática	26	61,90%	19	45,24%	4	9,52%	18	42,86%	22	52,38%	16	38,10%	20	47,62%	18	42,86%	22	52,38%
Enlaces cadena concatenada		50,00%		16,67%		0,00%		8,33%		8,33%		100,00%		8,33%	1	8,33%	1	8,33%
Enlaces de funciones espiales		6,25%		12,50%		0,00%		6,25%		6,25%		75,00%		0,00%	5	31,25%	0	0,00%
Enlaces totales conseguidos	33	47,14%	23	32,86%	4	5,71%	20	28,57%	24	34,29%	40	57,14%	21	30,00%	24	34,29%	23	32,86%

Fig. 6. Resumen de resultados de crawlers OpenSource y comerciales. A la derecha en los repositorios y a la izquierda en los logs

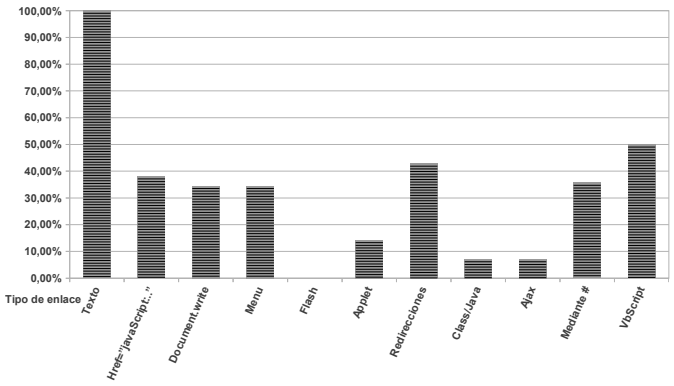


Fig. 7. Tipos de enlaces extraídos por los Crawlers OpenSource y comerciales

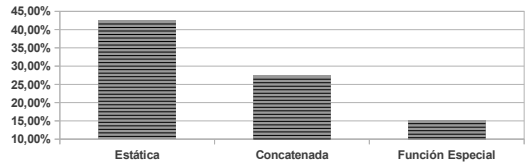


Fig. 8. Crawlers OpenSource y comerciales según el tipo de cadena

del lado cliente son tratados por los crawlers OpenSource o comerciales.

Como complemento a los resultados presentados analizando los índices, se han analizado las entradas que generaba cada crawler en los logs del servidor donde estaba almacenado el sitio Web. Con esto se pretende analizar más en detalle cómo ”intentan“, los crawlers, procesar cada nivel. La tabla de la derecha de la Fig. 6 muestra las diferencias encontradas con los resultados obtenidos al analizar los repositorios de cada crawler, mostrados en la tabla de la izquierda de la misma figura.

- WebHTTrack, intentó acceder a los niveles de Flash en que las direcciones están en el código HTML para ser pasadas a Flash mediante parámetros. No tuvo en cuenta que la estructura para pasar parámetros a Flash es ”parametro1=datos1&parametro2=datos2&....” por lo que considera toda la cadena como un único URL cuando no lo es.

```
"GET /∼mad/resources/projects/jstestingsite/1_test/link_relative=a_1000010010
0001000000000000_test_flash_link_html_relative.html&link_absolute=http://www.
tic.udc.es/∼mad/resources/projects/jstestingsite/1_test/a_10000100100001000000
000001_test_flash_link_html_absolute.html"
```

- Teleport intentó acceder a URLs absolutos generados mediante una función. Detecto “http...” y busco el correspondiente “.html” que indica el final de la dirección. No obstante, no detecta que por el medio se han definido variables, y por tanto el URL generado es incorrecto.

```
"GET /∼mad/resources/projects/jstestingsite/1_test/";var%20a6="a_1000100110000
0000000000001_test_document_write_function_special_embedded_absolute.html"
```

C. Resultados de crawlers de los principales buscadores web

Para realizar el análisis de crawlers utilizados por los principales buscadores web, se han esperado 70 días desde que se puso accesible el sitio web de prueba y se dio de alta en los diferentes buscadores. Tras este tiempo, se comprobó, en los logs del sitio web, qué crawlers de los contemplados habían accedido al menos a la página principal del sitio web. La Fig. 9 contiene los resultados obtenidos. No se muestran los resultados de Alexa, Ask, Bing, Gigablast y PicSearch, debido a que no indexaron el sitio web. Solo Google y Yahoo! indexaron el sitio web. Entre los factores que pudieron motivar que ciertos crawlers no indexaran el sitio web se puede considerar:

- Bajo PageRank.
- Alta profundidad en donde se encuentra alojado.
- Alto contenido de código, lo que implica que en base a una heurística un crawler determine la posibilidad de que el sitio web contenga Malware o bien determine que los enlaces tienen una alta dificultad para ser procesados.
- Contenido insuficiente o de mala calidad.

Google consiguió procesar un 44.29% de los enlaces, lo cual indica que no ha sido por el diseño, contenido o demás variables por las que el sitio no ha sido indexado por los demás. El crawler de Yahoo! también ha procesado algún enlace, pero de escasa dificultad.

Aunque GoogleBot consiguió procesar el 50% de muchos de los niveles propuestos, se observó que no logró un 100% debido a que en la otra mitad de los enlaces el código estaba almacenado en un fichero externo que no llegó a procesar. En caso contrario GoogleBot conseguiría resultados mucho mejores. Las tecnologías usadas en los enlaces que no procesó eran Flash, Applets y AJAX, del mismo modo que tampoco entró a analizar ficheros .class o .java.

En el caso de estos crawlers no se muestran por separado los resultados en los índices, es decir en sus sistemas de búsqueda, y en los logs del servidor del sitio web, ya que de ambas formas se obtienen resultados idénticos.

Tipo de enlace	Google		Yahoo	
Texto	2	100,00%	1	50,00%
Href="javaScript:..."	6	50,00%	0	0,00%
Document.write	6	50,00%	0	0,00%
Menu	6	50,00%	0	0,00%
Flash	0	0,00%	0	0,00%
Applet	0	0,00%	0	0,00%
Redirecciones	6	60,00%	2	20,00%
Class/Java	0	0,00%	0	0,00%
Ajax	0	0,00%	0	0,00%
Mediante #	4	100,00%	0	0,00%
VbScript	1	25,00%	0	0,00%
Enlaces cadena estática	17	40,48%	3	7,14%
Enlaces cadena concatenada	6	50,00%	0	0,00%
Enlaces de funciones espiales	8	50,00%	0	0,00%
Enlaces totales conseguidos	31	44,29%	3	4,29%

Fig. 9. Resumen resultados crawler de buscadores

D. Comparación de resultados

En la Fig. 10 se muestra un resumen comparativo de resultados. Claramente los crawlers OpenSource y comerciales obtienen en general mejores resultados. Es Google únicamente quien consigue unos números similares. Esto puede ser debido a que un crawler OpenSource o comercial está configurado por un usuario que indica el tipo de enlaces que debe seguir sin preocuparse de aspectos de rendimiento o seguridad, circunstancias que un crawler global de un buscador debe tener muy en cuenta.

En la Fig. 11 se comparan los resultados en base a la tecnología usada en la construcción del enlace. Nuevamente se observa un mejor comportamiento de los crawlers OpenSource. La curva que realiza cada grupo de crawlers es similar por lo que se puede concluir que a pesar de conseguir un menor número de enlaces de cada tecnología, sí muestran el mismo interés por procesar el mismo tipo de tecnologías.

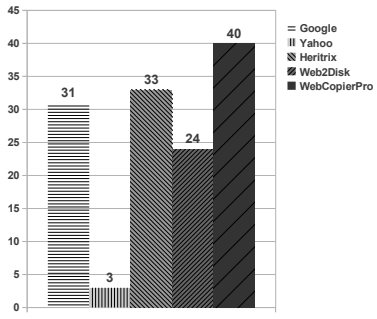


Fig. 10. Comparativa crawlers de motores de búsqueda y OpenSource

E. Clasificación de los crawlers en base a la escala definida

La Fig. 12 muestra el resultado de clasificar los distintos sistemas de crawling en base a la escala definida, según los diferentes métodos de evaluación propuestos.

- En media simple, WebCopier obtiene los mejores resultados, por delante de Heritrix y Google.
- Para el modelo de máximo nivel, Google se desmarca del resto consiguiendo procesar enlaces de nivel 8, seguido por WebCopier que obtiene un 7 y Heritrix un 6. Que Google haya alcanzado el nivel máximo según este modelo y no en otros, indica que podría tener capacidad para tratar cualquiera de los escenarios considerados, pero no lo hace debido a políticas internas.
- Para el método de la media ponderada, nuevamente WebCopier, seguido de Google, Heritrix y Nutch presentan los mejores resultados.
- En ocho niveles, Google cede los primeros puestos a Heritrix, Web2Disk y WebCopier. Esto indica que estos tres o bien han tratado gran cantidad de niveles de cada grupo o bien han atravesado enlaces que formaban parte de un grupo con pocos enlaces, lo cual da una alta puntuación a cada enlace procesado.

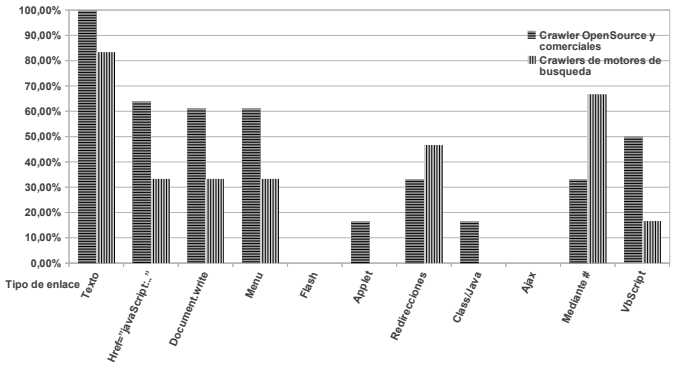


Fig. 11. Comparativa resultados de crawlers según el tipo de link

Se puede concluir que los crawlers que más y mejor tratan la Web Oculta del lado cliente son Google y WebCopier, seguidos por Heritrix, Nutch o Web2Disk. Es importante resaltar los resultados obtenidos para GoogleBot, por tratarse de un sistema de crawling orientado a toda la Web, con grandes requisitos de rendimiento y seguridad, y que no por ello ha descuidado el tratamiento de este tipo de tecnologías.

	Heritrix	Nutch	Pavuk	Teleport	Web2Disk	WebCopier	WebHTTRack	Google	Yahoo
Media simple	3,77	2,63	0,46	2,29	2,74	4,57	2,40	3,54	0,34
Máximo nivel	6,00	7,00	3,00	4,00	5,00	7,00	6,00	8,00	3,00
Media ponderada	1,63	0,99	0,14	0,75	1,03	3,42	0,86	2,34	0,11
Ocho niveles	6,00	4,30	1,20	4,00	4,55	4,55	3,40	3,70	0,70

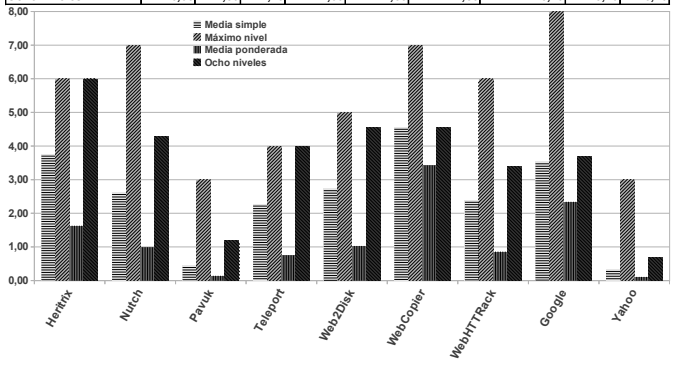


Fig. 12. Resultados en las escalas propuestas

VII. Conclusiones

Este artículo propone una escala que tiene en cuenta los diferentes niveles de dificultad existentes en los sitios de la Web Oculta del lado cliente. La definición de la escala permite clasificar los sistemas de crawling en base a su efectividad accediendo a dicha Web Oculta. La escala se ha definido en base a las diferentes tecnologías del lado cliente que

actualmente se usan por parte de los diseñadores para la creación de sitios web y a la dificultad que cada una de ellas presentan a los crawlers. Existen trabajos previos como el de M. Weideman y F. Schwenke en [12], pero en donde sólo se analizaban enlaces con JavaScript. En este trabajo se contempla un abanico de tecnologías mayor y más actuales. Por otro lado también se han incluido en el estudio crawlers OpenSource y comerciales con soporte para el tratamiento de las tecnologías del lado cliente.

Para realizar la clasificación de los diferentes sistemas de crawling, se ha creado un sitio web implementando las diferentes dificultades incluidas en la escala.

Los resultados obtenidos muestran que tanto en los niveles logrados como en los intentados, la mayor parte de las veces los crawlers tratan de descubrir los URL's procesando el código como texto, utilizando expresiones regulares. Es cierto que esto permite descubrir gran cantidad de escenarios y que el coste computacional es menor, pero se ha visto que la mayoría de las direcciones que forman parte de las tecnologías del lado cliente no son descubiertas. Los únicos para los que se puede concluir lo contrario son WebCopier y Google, que seguramente harán uso de algún intérprete que les permita ejecutar código.

Para tratar las dificultades presentadas por los niveles actualmente ignorados por los sistemas de crawling convencionales (tanto OpenSource como comerciales) existen diferentes aproximaciones en la literatura, que según los experimentos realizados no se utilizan en el práctica, seguramente por razones de eficiencia. Es el caso de la utilización de mini-navegadores web como componentes de crawling, como aparece descrito en [18], que permiten simular la navegación que realizaría un usuario al entrar en un sitio web.

Para finalizar, es importante resaltar que el tratamiento de la Web Oculta del lado cliente presenta desafíos como Redirection Spam o Cloacking. Una correcta detección evitaría perjuicios al usuario final y motivaría a que los crawlers presten atención a dichas páginas sin correr el riesgo de que éstas contengan algún tipo de Malware o redirección no deseada.

VIII. Trabajos futuros

Entre los estudios que se proponen como continuación de este trabajo está la mejora de los métodos de evaluación en base a la escala, para que tengan en cuenta el porcentaje de uso en la Web actual de los diferentes niveles de dificultad. Entre otras cosas permitirá determinar el volumen de información que se está quedando fuera del alcance de los crawlers, por no tratar algunos de los escenarios, y con ello determinar la importancia real del análisis de la Web Oculta del lado cliente.

También se plantea estudiar cómo afectan las diferentes características de un sitio web a su indexación y al análisis que los crawlers hacen de su contenido para localizar nuevos enlaces. De esta forma se podría determinar si la importancia, la temática y el número de visitas de un sitio afecta a cómo son tratados por los crawlers.

Por último, y debido a que los crawlers que han obtenido mejores resultados no tienen su código público, se podría considerar el diseño de algoritmos que permitan extraer enlaces de las tecnologías mostradas sin necesidad de utilizar mini-navegadores ni intérpretes completos, para que sean capaces de escalar a un crawling global.

Agradecimientos

Este trabajo de investigación ha sido financiado por el Ministerio de Educación y Ciencia de España y los fondos FEDER de la Unión Europea (Proyecto TIN2009-14203).

Referencias

[1] M. K. Bergman, "The deep web: Surfacing hidden value," 2000.
[2] Mozilla, "Javascript - mcd centro de documentacion."
[3] A. T. H. III, *Ajax: The Definitive Guide*. O'Reilly Media, 2008.
[4] Microsoft, "Vbscript user's guide," 2011. [Online; accessed 18-February-2011].
[5] Adobe, "application programming — adobe flash platform," 2011. [Internet; accessed 11-marzo-2011].
[6] K. Chellapilla and A. Maykov, "A taxonomy of javascript redirection spam," in *Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, AIRWeb '07, (New York, NY, USA), pp. 81–88, ACM, 2007.
[7] B. Wu and B. D. Davison, "Cloaking and redirection: A preliminary study," 2005.
[8] R. Khare and D. Cutting, "Nutch: A flexible and scalable open-source web search engine," tech. rep., 2004.
[9] G. Mohr, M. Kimpton, M. Stack, and I. Ranitovic, "Introduction to heritrix, an archival quality web crawler," in *4th International Web Archiving Workshop (IWAW04)*, 2004.
[10] "W3Techs - World Wide Web Technology Surveys." http://w3techs.com/, 2011. [Online; accessed 22-March-2011].
[11] "BuiltWith Web Technology Usage Statistics." http://trends.builtwith.com/, 2011. [Online; accessed 22-March-2011].
[12] M. Weideman and F. Schwenke, "The influence that JavaScript has on the visibility of a Website to search engines - a pilot study," *Information Research*, vol. 11, Jul 2006.
[13] S. Raghavan and H. Garcia-Molina, "Crawling the hidden web," in *Proceedings of the 27th International Conference on Very Large Data Bases*, VLDB '01, (San Francisco, CA, USA), pp. 129–138, Morgan Kaufmann Publishers Inc., 2001.
[14] J. Madhavan, D. Ko, L. Kot, V. Ganapathy, A. Rasmussen, and A. Halevy, "Google's deep web crawl," *Proc. VLDB Endow.*, vol. 1, pp. 1241–1252, August 2008.
[15] F. C. A. P. Manuel Ãlvarez, Juan Raposo, "Crawling web pages with support for client-side dynamism," (University of A Coruna, 15071 A Coruna, Spain), 2006.
[16] Mozilla, "Mozilla rhino javascript engine," 2011. [Online; accessed 18-February-2011].
[17] "v8 - V8 JavaScript Engine." http://code.google.com/p/v8/, 2011. [Online; accessed 21-March-2011].
[18] M. Á. Díaz, *Arquitectura para Crawling Dirigido de Información Contenida en la Web Oculta*. PhD thesis.
[19] F. C. A. P. Manuel Ãlvarez, Juan Raposo, "A task-specific approach for crawling the deep web," *Journal Engineering Letters. Special Issue:*.
[20] B. Wu and B. D. Davison, "Detecting semantic cloaking on the web," in *Proceedings of the 15th International World Wide Web Conference*, pp. 819–828, ACM Press, 2006.
[21] B. Wu and B. D. Davison, "Identifying link farm spam pages," in *Proceedings of the 14th International World Wide Web Conference*, pp. 820–829, ACM Press, 2005.
[22] Z. Gyongyi and H. Garcia-Molina, "Web spam taxonomy," 2005.
[23] "The w3 consortium the document object model." http://www.w3.org/DOM/, 2011. [Online; accessed 18-February-2011].
[24] Wikipedia, "Applet — wikipedia, the free encyclopedia," 2011. [Online; accessed 18-February-2011].
[25] "JSON." http://json.org/, 2011. [Online; accessed 22-March-2011].
[26] Microsoft, "The official microsoft asp.net site," 2011. [Online; accessed 18-February-2011].
[27] "The Web Robots Pages." http://www.robotstxt.org/, 2011. [Online; accessed 18-February-2011].
[28] "Pavuk Web page." http://www.pavuk.org/, 2011. [Online; accessed 18-February-2011].
[29] "Teleport Web page." http://www.tenmax.com/teleport/pro/home.htm, 2011. [Online; accessed 18-February-2011].
[30] "Web2Disk Web page." http://www.inspyder.com/products/Web2Disk/Default.aspx, 2011. [Online; accessed 18-February-2011].
[31] "Web Copier Pro Web page." http://www.maximumsoft.com/products/wc_pro/overview.html, 2011. [Online; accessed 18-February-2011].

Sesión 5.B
Modelado y análisis de prestaciones (II)

Implementación y evaluación del Path Computation Element Protocol

J.L. Añamuro, V. Lopez, J. Aracil
High Performance Computing and Networking group
Universidad Autónoma de Madrid
28049 Madrid, España.
jose.annamuro@estudiante.uam.es, {victor.lopez, javier.aracil}@uam.es

***Abstract*—Las redes troncales están migrando hacia un esquema de redes de nueva generación. Estas redes se utilizan para soportar todo tipo de servicios, por lo que deben ser capaces de soportar múltiples calidades de servicio. Las redes de nueva generación utilizan un plano de control automatizado que realiza las tareas de enrutado, reserva de recursos y gestión del estado de los enlaces. Sin embargo, esta integración de servicios sobre la misma red hace que el cálculo de las rutas sea cada vez más complicado. El Path Computation Element se define para tener un elemento en la red capaz de realizar el cálculo de rutas, descargando a los nodos del plano de control de realizarlo. En este trabajo se ha implementado el Path Computation Element y se evalúa su impacto en el plano de control de redes de nueva generación.**

***Index Terms*—Path Computation Element; plano de control; redes troncales; redes multi-capa; redes multi-dominio;**

I. Introducción

En los años recientes la explosión de conexiones de banda ancha ha impuesto un aumento sin precedentes en el tráfico en redes de telecomunicación con muy altas tasas de crecimiento anual. Un ejemplo de este enorme crecimiento de tráfico es el informe de Cisco [1] que prevé un tráfico IP anual de más de 700 hexabytes en 2014, cuatro veces superior que en 2009. Esto significa una tasa de crecimiento anual de 34%. Se espera que esta tendencia continúe en el futuro debido a la implementación de nuevas tecnologías de accesos de banda fija y móvil, el constante crecimiento en la capacidad de transmisión y la aparición de nuevos servicios de gran ancho de banda como son el Cloud Computing y la televisión de alta definición. En este ambiente de enorme demanda, los operadores de red están preocupados en la escalabilidad de las arquitecturas de red actuales para soportar el internet del futuro, especialmente en termino de coste por bit.

Con el objetivo de reducir los costes, los operadores están migrando a redes de próxima generación. Estas redes de próxima generación soportan múltiples servicios sobre una capa común IP con múltiples tecnologías de transporte [2]. En las redes troncales, los operadores de red utilizan esquemas basados en enrutadores IP/MPLS con la ayuda de una red de conmutación óptica (WDN, OTN, etc) [3].

En el pasado las redes troncales estaban gestionadas de forma estática y centralizada. Este sistema de gestión se encargaba de configurar cada equipo de la red por donde se encaminaba la conexión. Dos hechos cambiaron este paradigma de gestión centralizada en redes troncales. Por un lado, los elementos de red ópticos, como Optical Cross-Connects (OXC) o Reconfigurable Optical Add Drop Multiplexers (ROADM), avanzaron tecnológicamente para poder reconfigurarse de forma dinámica. Por otro lado, la introducción del paradigma Automatically Switched Optical Network (ASON) [4] y Generalized Multi-Protocol Label Switching (GMPLS) [5] creó un plano de control unificado que permite crear y destruir Label Switched Paths (LSP) de una forma automática. Al ser un plano de control unificado, se puede controlar múltiples dispositivos desde equipamiento Ethernet, IP/MPLS a equipamiento óptico.

Este cambio en la arquitectura crea una red de nueva generación con tres planos: plano de datos, de control y de gestión [6], tal y como se muestra en la Fig. 1. El plano de datos se encarga del transporte de la información de los usuarios. La ITU define el plano de gestión como la entidad que gestiona las funciones del plano de transporte, el plano de control y el sistema en su completo, así como asegura la coordinación entre ellos [4]. El plano de gestión se usa para las operaciones centralizadas de la red como son la facturación, la gestión de fallos o la monitorización de la calidad de servicio, entre otros. La ITU define el plano de control como la entidad que realiza se encarga de las conexiones, su establecimiento y liberación y la restauración [4].

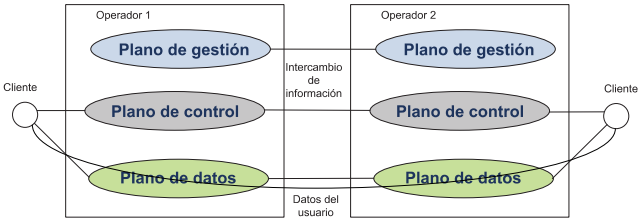


Fig. 1. Arquitectura de red de próxima generacióon

Un nuevo elemento de las redes de nueva generación es el Path Computation Element (PCE) [7] que surge como iniciativa del IETF para solucionar el problema de calcular una ruta óptima aplicando restricciones en escenarios complejos como son redes multicapa, multidominio y con diferentes áreas. En el caso de redes ópticas, el cálculo de rutas se puede volver muy complejo al tener que incluir información sobre los problemas de la capa física. [8] contiene un amplio resumen sobre los algoritmos con restricciones físicas para redes ópticas. El uso del PCE permite reducir los requisitos computacionales de los elementos del plano de control y con ello los costes de los mismos. Además permite optimizar el cálculo de las rutas al ser una entidad centralizada. El concepto de PCE como una entidad que calcula fue validado experimentalmente en [9]. Los autores

ISBN: 978-84-694-5948-5

en [10], [11] han estudiado las ventajas del agrupamiento de peticiones en términos de bloqueo y de carga al plano de control utilizando simulación. Sin embargo, en ningún trabajo anterior se ha estudiado la carga que introduce el protocolo PCE en el plano de control en su operación normal utilizando una implementación real del protocolo. Este trabajo tiene dos contribuciones: (1) implementación del protocolo PCE y (2) estudio del impacto en el plano de control del PCE en las redes de próxima generación.

Este trabajo está organizado del siguiente modo. La sección II presenta las distintas arquitecturas del PCE dentro de las redes de próxima generación, sus modos de configuración y cómo es su funcionamiento en cada una de ellas. En la sección III se describe en detalle el Path Computation Element Protocol (PCEP) , los mensajes y las diferentes situaciones que se dan en una comunicación entre un PCE y un Path Computation Client (PCC). La sección IV valida el protocolo implementado, analiza las prestaciones del mismo y su impacto en el plano de control. La sección V muestra las conclusiones de este trabajo y muestra el trabajo futuro.

II. Arquitectura del Path Computation Element

El PCE es una entidad (componente, aplicación, o nodo de red) capaz de calcular una ruta de red o una ruta basada en un gráfico de red aplicando restricciones [7]. Esta definición es amplia, por lo que se han definido una gran cantidad de modos de operación. A continuación se presenta la localización del PCE dentro de las redes de próxima generación, así cómo su integración en escenarios multicapa y multidominio.

A. Localización

El PCE puede estar en el plano de gestión, siendo parte de un sistema de gestión de red (Network Management System - NMS) tal que dada una petición de servicio, el NMS solicita una ruta al PCE. Para calcular la ruta más adecuada, el PCE requiere la información de estado de la red, la cual está almacenada en una base de datos de ingeniería de tráfico (TED). Una vez que el PCE provee una respuesta al NMS, la configuración es enviada a los elementos de red para configurar el servicio. Esta situación se muestra en la Fig. 2. En [12], se explica en mayor detalle como se puede poblar la TED con información desde el plano de gestión.

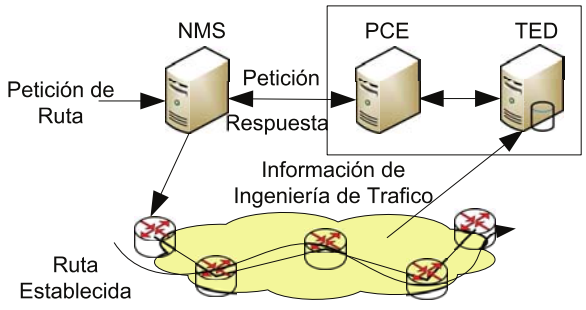


Fig. 2. Path Computation Element

En lugar de añadir la funcionalidad del PCE en el plano de gestión, puede estar en el plano de control. El equipo que solicita las rutas al PCE se denomina Path Computation Client (PCC). Usualmente el PCC es un enrutador GMPLS que puede computar una ruta de un camino distribuido usando el algoritmo de encaminamiento estándar GMPLS o puede solicitar una ruta al PCE cuando el algoritmo de encaminamiento no sea estándar. Para que el PCE pueda calcular la ruta debe conocer el estado de la red. Para ello se puede utilizar la información de inundación de OSPF y se almacena en una TED [12].

El PCE puede estar integrado en el mismo equipo que el PCC o puede estar separado del servidor. Ambos casos se muestran en la Fig. 3. La solución conjunta en un único servidor es fácil de implementar y no requiere un protocolo estándard de comunicaciones entre el PCC y PCE. En el otro caso, la solución de servidor separado usa el protocolo estándar basado en el esquema petición/respuesta y permite que una única entidad PCE de servicio a múltiples PCCs. Este protocolo de petición/respuesta se conoce como Path Computation Element Protocol (PCEP) y se detallará en la sección III.

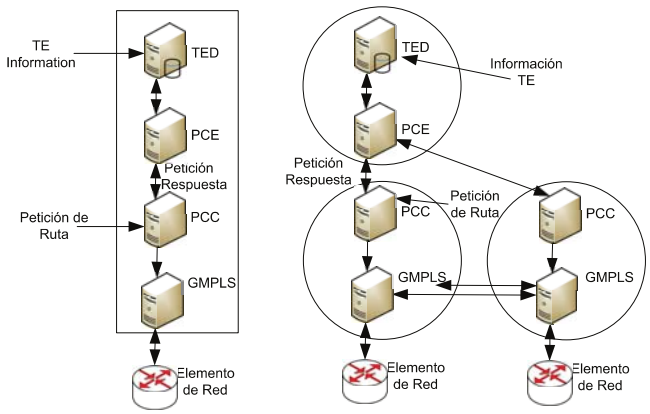


Fig. 3. Configuraciones separada o integrada del PCE

B. PCE en entornos multicapa

Se han definido esquemas del PCE para entornos multicapa. Una red multicapa es una red donde existen varias tecnologías de red. Un ejemplo son las redes IP sobre redes ópticas [3]. Para este tipo de redes existen, dos modelos de interconexión centralizados y dos modelos distribuidos. Los dos modelos centralizados son los siguientes:

- PCE único multicapa: Esta arquitectura tiene un PCE único capaz de almacenar la información de todas las capas de la red. Este PCE puede estar en cualquier lugar en un plano de control integrado o en un plano de gestión.
- PCE/VNTM (Virtual Network Topology Manager): El VNTM muestra una topología de red a la capa superior [13], tal y como se muestra en la Fig. 4. La capa superior (IP; PCE) puede pedir conexiones extras a la capa inferior. El VNTM puede cambiar las conexiones a la capa superior si sus políticas indican que es la mejor opción. El modo de operación de VNTM podría ser cualquier solución, incluso otro PCE.

En cuanto a las soluciones distribuidas se pueden distinguir las dos siguientes:

- Multicooperación de PCEs de capa única: Esta opción usa un PCE en cada capa y estos pueden intercambiar peticiones cuando estas son requeridas. Debido a este intercambio de peticiones la capa superior puede pedir

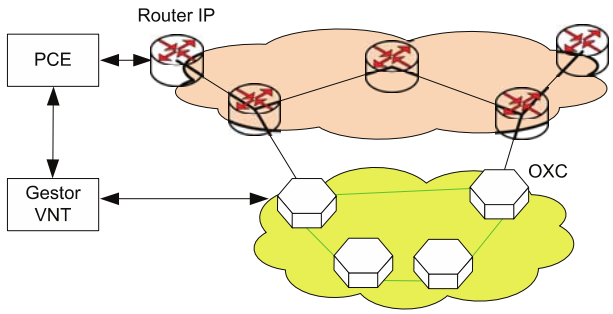


Fig. 4. Configuración PCE/VNTM

conexiones a la capa inferior para que puedan modificar la topología de la capa superior. Además como hay dos PCEs, la solución reduce la complejidad de cálculo en los algoritmos de cada PCE.

- Múltiples PCE multicapa: Esta arquitectura tiene múltiples PCEs multicapa de manera que los PCEs disponen información de cada capa de la red. Los PCCs pueden solicitar un cálculo de ruta a cualquiera de los PCEs, pero en general se realiza al más cercano. Una segunda opción es que el PCC envía consultas a un PCE el cual balancea las solicitudes a los otros PCE, lo que reduce el tiempo de cálculo.

C. *Interconexión multicapa y multidominio PCE*

La arquitectura del PCE encaja bien para abordar el problema del establecimiento de rutas multidominio. El modelo de interconexión para un escenario multidominio se muestra en la Fig. 5. Los autores de [14] proveen una visión general de los desarrollos en el área de ingeniería de tráfico basado en PCE en redes GMPLS además de un análisis detallado del enfoque de redes PCE en redes multidominio y comparan el rendimiento de las soluciones existentes. La Fig. 5 muestra que por lo menos hay un PCE en cada dominio y el modelo de interconexión entre los PCEs está basado en el protocolo de petición/respuesta.

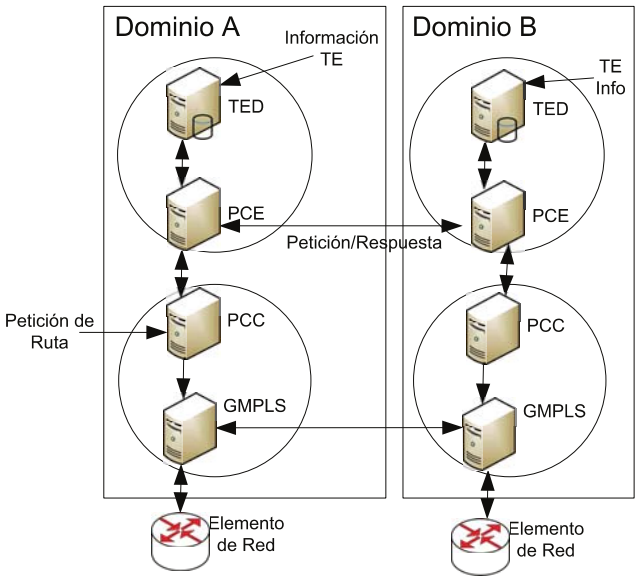


Fig. 5. PCE en un entorno de red multidominio

Cuando hay una petición del dominio A al dominio B, el PCC envía una petición al PCE A, el cual reenvía la petición al PCE B proporcionando una ruta o una fallo en el procedimiento. Este es el procedimiento general, pero hay tres métodos de cálculo diferentes:

1) PCE de cooperación simple: La configuración de cooperación PCE permite a los PCE intercambiar información para encontrar la mejor conexión de extremo a extremo. Cada PCE envía la mejor solución al siguiente PCE de la cadena, pero el PCE vecino puede sugerir otra conexión al PCE anterior. Sin embargo cada conexión se elige a nivel local lo cual significa que cuando la ruta óptima extremo a extremo no utiliza la ruta local óptima, entonces la solución global no puede ser encontrada (Fig. 6).

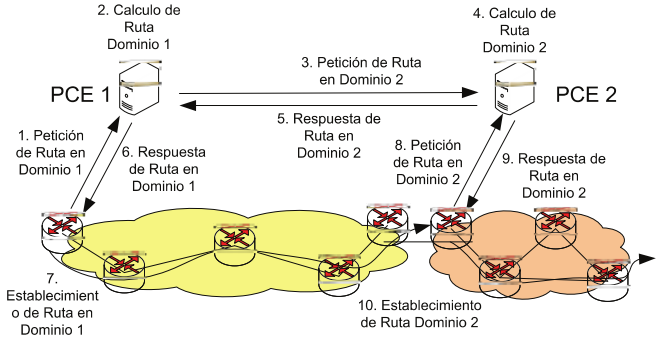


Fig. 6. PCEs de cooperación simple

2) Cálculo de ruta por dominio [15]: En este enfoque cada PCE calcula la ruta desde el router de ingreso hasta el router de salida en su dominio, tal y como se muestra en la Fig. 7. Por lo tanto, todos los dominios del origen al destino deben ser conocidos de antemano por el PCE en el dominio origen. Este es un problema, ya que no hay un mecanismo para escoger los dominios más adecuados del origen al destino. Por otra parte si hay múltiples conexiones entre dominios, el PCE1 puede proveer una ruta que es óptima a nivel local pero no a nivel global.

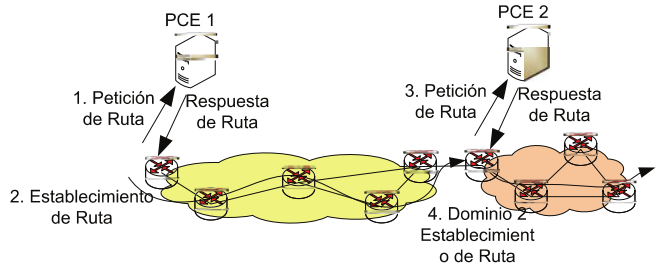


Fig. 7. Cálculo de ruta por dominio de un PCE

3) Backward recursive path computation (BRPC) [16]: El método BRPC se inicia en el dominio destino el cual envía a su vecino el coste desde el router del extremo al nodo de destino. En consecuencia el dominio vecino puede crear un árbol con sus nodos de origen y su nodo de destino. Este proceso continúa hasta el dominio

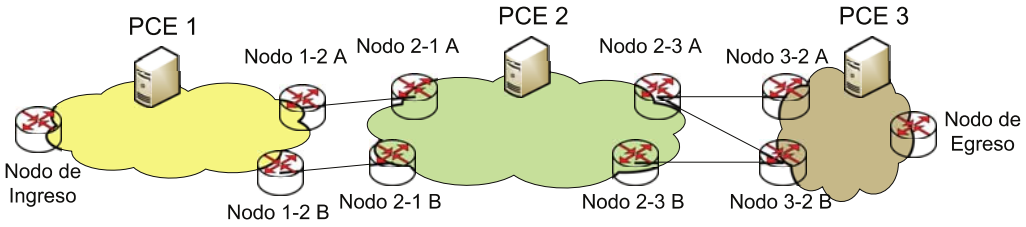


Fig. 8. Cálculo de ruta recursivo Backward

origen el cual selecciona la mejor ruta extremo a extremo. La Fig. 8 representa tres dominios conectados con un PCE por dominio. Usando BRPC, PCE1 envía una solicitud a PCE2 el cual se lo reenvía a PCE3. PCE3 responde con la distancia desde el nodo del borde con el dominio 2. PCE2 lleva a cabo la misma operación operación enviando el árbol con las combinaciones posibles de 1 a 3. Cuando múltiples dominios están interconectados intercambiar dicha información puede ser complicado.

III. Descripción detalla del Path Computation Element Protocol

Esta sección especifica el Path Computation Element Comunication Protocol (PCEP) para las comunicaciones entre un PCC y un PCE, o entre dos PCEs [17]. El PCEP es un protocolo basado en la arquitectura petición/respuesta que opera sobre el protocolo TCP. El protocolo consta de 7 posibles mensajes: `Open`, `Keepalive`, `Request`, `Response`, `Notify`, `Error` y `Close`.

A. Fase de Inicialización

La fase de inicialización consiste en dos pasos sucesivos, primero la creación de una conexión TCP y segundo el establecimiento de una sesión PCEP sobre TCP. Una vez que la conexión ha sido establecida, el PCC y el PCE (conocidos como pares PCEP) inician el establecimiento de una sesión PCEP en el que negocian varios parámetros que están establecidos dentro del mensaje `Open` que incluyen un temporizador `Keepalive` y un temporizador `Deadtimer`. También se permite el intercambio de las capacidades del PCE y PCC para saber qué tipo de solicitudes se pueden realizar para el cálculo de ruta al PCE (Fig. 9).

B. Sesión Keepalive

Una vez establecida la sesión, es necesario saber que el otro extremo está aún disponible. Se puede confiar en TCP para esta información, pero es posible que la función PCEP remota falle sin perturbar la conexión TCP. Con el fin de manejar esta situación, PCEP incluye un mecanismo de mantenimiento de la conexión basado en un temporizador `Keepalive`, un temporizador `Deadtimer` y un mensaje `Keepalive`.

Cada extremo de la sesión PCEP ejecuta un temporizador `Keepalive` que se reinicia cuando se envía un mensaje en la sesión. Una vez que el temporizador expira se manda un mensaje `Keepalive`.

Los extremos de la sesión ejecutan el temporizador `Deadtimer` y lo reinician cada vez que se recibe un mensaje en la sesión. Si un extremo de la sesión no recibe ningún

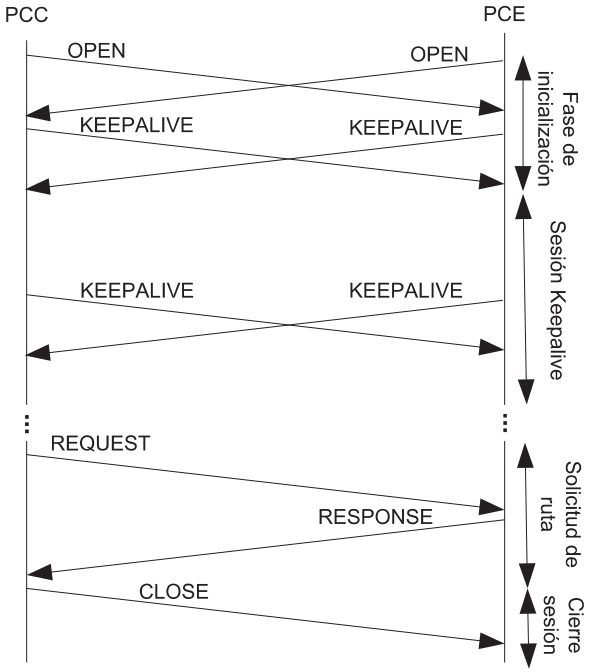


Fig. 9. Intercambio de paquetes en las diferentes sesiones del PCEP

mensaje y expirar el `Deadtimer` la sesión se declara muerta. Cada extremo de la sesión puede especificar (en el mensaje `Open` cuando se abre la sesión) el temporizador `Keepalive` que utilizará (es decir, con qué frecuencia se transmitirá un mensaje `Keepalive` si no hay otro tráfico), un `Deadtimer` que recomida utilizar a su par (cuánto tiempo debe esperar el par antes de declarar la sesión muerta si no recibe tráfico). Los extremos de la sesión podrán utilizar diferentes valores de temporizador `Keepalive`.

El valor mínimo para el temporizador `Keepalive` es de un segundo y el recomendado es de 30, puede deshabilitarse estableciendo el temporizador a cero. El valor recomendado para el `Deadtimer` es 4 veces el valor del `Keepalive` (Fig. 9).

C. Solicitud y respuesta de una ruta

Una vez que se ha establecido una sesión PCEP satisfactoriamente entre un PCC y un PCE, el primero envía una petición de ruta al PCE (mensaje `Request`) que contienen una variedad de objetos que especifican un conjunto de restricciones y atributos para calcular la ruta. Cada solicitud es única e identificada por un ID.

El PCE al recibir la solcititud acciona un cálculo de ruta y cuando lo resuelve retorna un mensaje `Response`. Este men-

saje puede contener las rutas calculadas al PCC solicitante, si el PCE resuelve el cálculo de ruta que satisfaga el conjunto de restricciones requeridas en la solicitud. O puede decir con el mensaje que no encontró una ruta adecuada. El PCC al recibir una respuesta negativa puede decidir volver a enviar la solicitud modificándola o tomar otra acción apropiada. Hay que remarcar que el PCEP soporta la capacidad de enviar en una solicitud simple el cálculo de más de una ruta.

D. Finalización de una sesión PCEP

Cuando uno de las pares PCEP desea terminar la sesión, este primero envía un mensaje PCEP `Close` y luego cierra la conexión TCP. Si el PCE finaliza la sesión, el PCC borra todos los estados relacionados con las peticiones pendientes previamente enviadas al PCE. Similarmente, si el PCC finaliza la sesión PCEP, el PCE borra todas las peticiones de ruta enviadas por el PCC (Fig. 9). Un mensaje `Close` solo pude ser enviado para finalizar la sesión PCEP si la sesión ha sido previamente establecida. Una sesión PCEP también puede finalizar si un PCE/PCC recibe un mensaje desconocido a una frecuencia elevada. En ese caso, el PCC o PCE debe enviar un mensaje `Close` con el valor close "Recepción de un número inaceptable de mensajes desconocidos".

E. Modos de funcionamiento del protocolo PCEP

Existen dos modos de funcionamiento del PCEP, el intermitente y el permanente. El modo intermitente abre y cierra sistemáticamente una sesión PCEP para cada petición de ruta, esta modalidad es aplicable cuando el envío de una petición es un evento raro. En el modo permanente, se mantiene establecida la sesión PCEP y su correspondiente conexión TCP por un intervalo de tiempo ilimitado, esta modalidad resulta apropiada cuando las peticiones de ruta se envían de forma frecuente. Este modo evita abrir y cerrar una conexión TCP para cada nueva petición reduciendo así la carga adicional. La sesión Keepalive (Fig. 9) únicamente aparece cuando existe el modo de funcionamiento permanente.

F. Otros mensajes del protocolo

Existen situaciones en las que un PCE quiere notificar un error o un evento especifico. El mensaje `Error` se usa cuando se cumple una condición de error del protocolo o cuando una petición no es compatible con la especificación del PCEP (recepción de un mensaje con un objeto obligatorio perdido, referencia de solicitud desconocida, violación de políticas o mensajes inesperados). El mensaje `Notify` se usa cuando un PCE alcanza una sobrecarga que podría dar lugar a tiempos de respuesta inaceptables, el PCE puede querer notificar a uno o mas PCCs que alguna de sus peticiones no puede ser realizada o experimentar retardos inaceptables. Sobre la recepción de tales notificaciones, el PCC puede decidir redirigir estas peticiones de ruta a otro PCE disponible. Similarmente un PCC puede notificar al PCE de un evento en particular como la cancelación de una petición pendiente.

IV. Evaluación de prestaciones del PCEP

Esta sección primero valida el funcionamiento de la implementación realizada y después analiza las prestaciones de dicha implementación en términos de número máximo de peticiones por segundo y ancho de banda que ocupa el protocolo PCE. Para estudiar el impacto en el plano de control, se analizarán los dos modos de operación intermitente y permanente y el modo de agrupamiento de peticiones.

Para realizar las pruebas se han utilizado dos equipos conectados en una red de área local (Fig. 10). El PCE estaba en en un servidor con un procesador de doble núcleo Intel(R) Xeon(R) CPU 3075 a 2.66 Ghz de 64 bits.

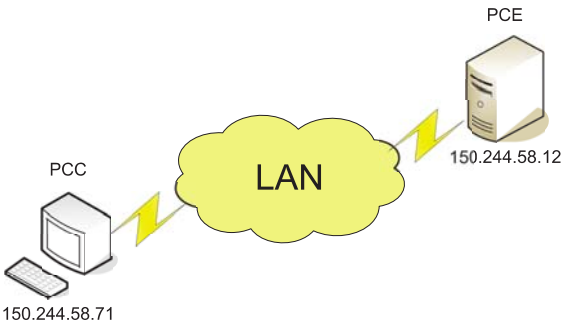


Fig. 10. Topología donde se evaluó el PCEP

A. Validación del funcionamiento del protocolo Path Computation Element

En este experimento se realizó una solicitud en modo intermitente para mostrar el correcto funcionamiento del protocolo PCE. En la Fig. 11, se muestra la captura con el intercambio de mensajes entre el PCC (`150.244.58.71`) y el PCE (`150.244.58.12`). Como se pude ver en la misma, el PCC inicia el establecimiento de una sesión enviando un mensaje `Open` que el PCE responde con otro mensaje `Open`, seguidamente intercambian mensajes `Keepalive` y queda establecida la sesión, luego un PCC envía una solicitud de cálculo de ruta (`Request` message) la cual el PCE procesa satisfactoriamente enviando de vuelta un (`Reply` message) y finalmente el PCC cierra la sesión.

B. Rendimiento en número de peticiones por segundo

La siguiente prueba consistió en determinar el número máximo de peticiones que se puede procesar entre un cliente y un servidor debido al retardo del protocolo en una red de área local. Para ello se ha medido el tiempo de duración de una petición cuando el PCE y el PCC están distribuidos según la topologia de la Fig. 10. Para evitar que el factor dominante sea el algoritmo de cálculo de rutas, se respondía que no se encontraba camino usando el objeto `NO-PATH` [17]. Para algoritmos de baja complejidad como son los de camino más corto en topologás de tamaño pequeño no cambia apenas los resultados. El retardo de la red era del orden de 0.1ms, puesto que se encontraban en una LAN.

En el experimento se lanzaron 10000 peticiones desde el PCC al PCE. El resultado del experimento (Fig. 12) revela que más del 3.5% de las peticiones tardan 3.5 milisegundos y la función de distribución acumulada (Fig. 13) muestra que el 99% que las peticiones tardan menos de 13.078 milisegundos. Acorde a estos valores, en el peor caso el servidor podrá soportar hasta 76, 7 peticiones.

Time	Source	Destination	Protocol	Info
0.0005	150.244.58.71	150.244.58.12	PCEP	OPEN MESSAGE
0.0007	150.244.58.12	150.244.58.71	PCEP	OPEN MESSAGE
0.0037	150.244.58.71	150.244.58.12	PCEP	KEEPALIVE MESSAGE
0.0039	150.244.58.12	150.244.58.71	PCEP	KEEPALIVE MESSAGE
0.0052	150.244.58.71	150.244.58.12	PCEP	PATH COMPUTATION REQUEST MESSAGE
0.0055	150.244.58.12	150.244.58.71	PCEP	PATH COMPUTATION REPLY MESSAGE
0.0058	150.244.58.71	150.244.58.12	PCEP	CLOSE MESSAGE

Fig. 11. `Wireshark` output: PCEP captured packets

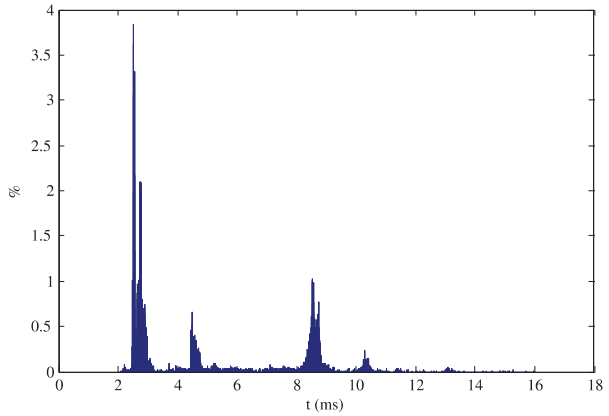


Fig. 12. Histograma de frecuencias relativas del retardo

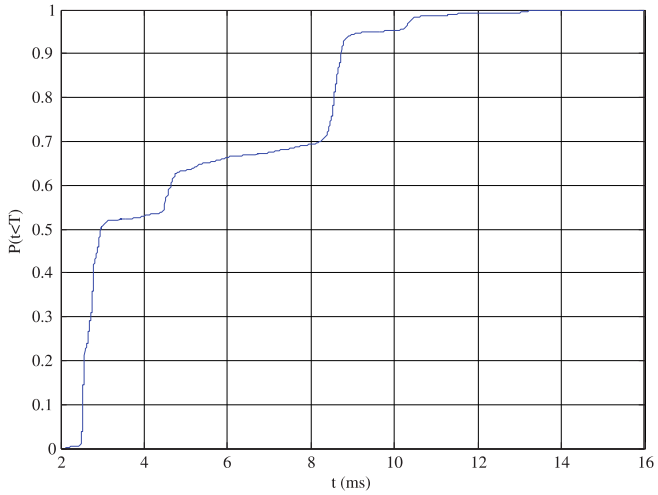


Fig. 13. Funcion de distribucion acumulada del retardo

C. Modo Intermitente vs. Permanente

Tal y como se explicó anteriormente, existen dos modos de operación del PCEP: permantente e intermitente. Este experimento pretende comparar ambos modos de funcionamiento, en términos de consumo de ancho de banda en el plano de control. Primero, se explicará el cálculo teórico de ancho de banda para cada modo y después se validará mediante un experimento que el protocolo implementado consume el mismo ancho de banda.

1) Modo Intermitente: Este modo inicia una sesión PCEP cada vez que sea necesario enviar una petición. Por lo tanto, para cada petición se envían dos mensajes `Open` (78bytes), dos mensajes `Keepalive` (70bytes), un mensaje `Request` (94bytes), un mensaje `Response` (90bytes) y un mensaje `Close` (78bytes). El tamanño de los mensajes `Request` y `Response` puede variar dependiendo del uso de IPv4 o IPv6 y de la topología de la red, ya que va incluida la información de cada salto. Para este experimento por cada petición se utilizan:

$$C_{inter} = 8 \times 558 \times N_{pet} \quad (1)$$

donde N_{pet} son las peticiones por segundo.

2) Modo Permanente: Este modo mantiene la sesión PCEP activa por un lapso de tiempo enviando un mensaje `Keepalive` cada `KeepTimer` segundos independientemente de si se envía otro mensaje o no. Además, por cada petición envía un mensaje `Request` (94bytes), un mensaje `Response` (90bytes). Se puede calcular los Kbps que injecta a la red con ls siguiente expresión:

$$C_{perm} = \frac{8 \times 140}{Keeptimer} + 8 \times 184 \times N_{pet} \quad (2)$$

donde $Keeptimer$ es el valor del temporizador `KeepTimer` en segundos y N_{pet} son las peticiones por segundo.

La Fig. 14 muestra el ancho de banda ocupado en el plano de control para $N_{pet} = \{1/120, 1/90, 1/60\}$ cuando se usa el modo intermitente. La carga es del orden de decenas de bps. En una red puede haber decenas de nodos y pueden atravesar varios saltos para llegar al PCE, pero aún así la carga en el plano de control es baja. Para estos valores usando el modo permanente se obtienen resultados similares.

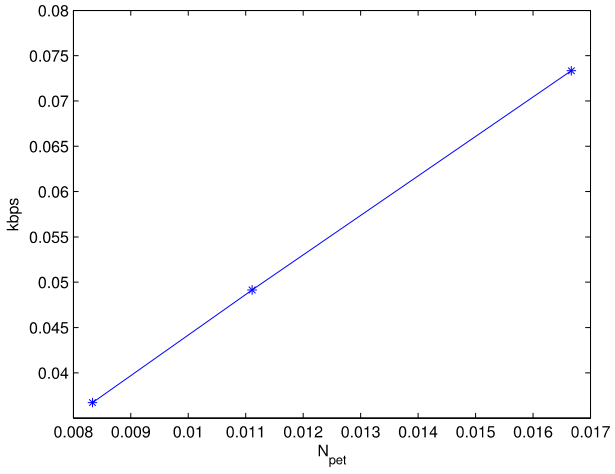


Fig. 14. Ancho de banda del PCEP

Keeptimer	kbps	Peticiones
10	0,112	0,03743
20	0,056	0,01872
30	0,0373	0,01248
40	0,028	0,00936
50	0,0224	0,00749
60	0,0186	0,00624

Tabla I
INTERMITENTE VS. PERMANENTE

A continuación se va a comparar qué modo de operación utiliza un mayor ancho de banda en base al número de peticiones por segundo y el valor del `KeepTimer`. La Fig. 15 compara el ancho de banda de acuerdo al número de peticiones lanzadas en el modo intermitente contra el modo permanente para varios valores del `KeepTimer`. En este experimento se han quitado la carga de meten los mensajes `Request` y `Response` que son comunes a ambos modos. Para calcular el número exacto N_{pet} donde se iguala el ancho de banda de ambos modos, se puede usar la siguiente expresión:

$$N_{pet} = \frac{140}{Keeptimer \times 374} \quad (3)$$

La Tabla 1 muestra el valor de N_{pet} para cada uno de los `KeepTimer` usados en el experimento anterior. Con este experimento se da una regla para decidir en qué modo se debe operar según los parámetros del protocolo. En base a los resultados, dependiendo del número de peticiones y del valor del temporizador `KeepTimer` se puede ver qué modo de operación envía más o menos tráfico al plano de control.

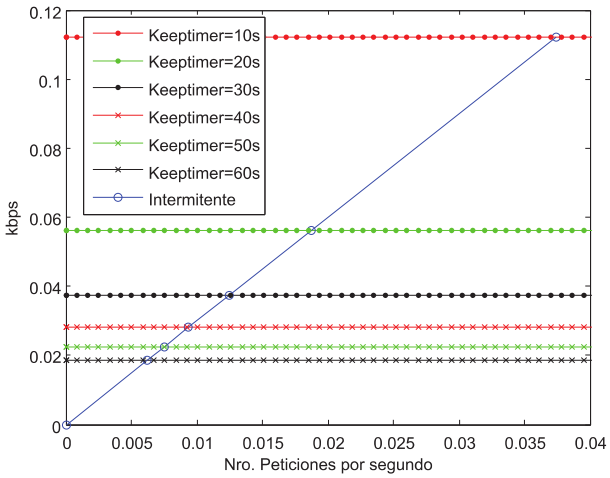


Fig. 15. Permanente vs. Intermitente

D. Agrupamiento de Peticiones

La definición de PCEP permite al PCC enviar al PCE más de una petición a la vez, es decir se pueden agrupar múltiples peticiones en una sola antes de enviarlas al PCE.

Las peticiones se agrupan para mejorar el proceso de optimización de red a expensas de un incremento en el retardo en la creación de la conexión. En el agrupamiento, se reúnen las peticiones usando un umbral de tiempo o de número máximo de peticiones. En el primer modo, se espera un temporizador para enviar la petición con todas las rutas llegadas antes que éste expire. En la segunda se espera una cantidad determinada de peticiones según el valor del umbral y se envía la petición.

En este experimento se usa el primer modo donde la llegada de peticiones sigue una distribución de Poisson con un valor de $N_{pet} = \{1/120, 1/90, 1/60\}$.

En la Fig. 16 se muestra como para una tasa mayor el ancho de banda usando agrupamiento se reduce significativamente debido al menor número de apertura y cierre de las conexiones. Sin embargo, cuando la carga es menor la reducción no es tan importante. Para intervalos de tiempo superiores la reducción de overhead no es muy significante.

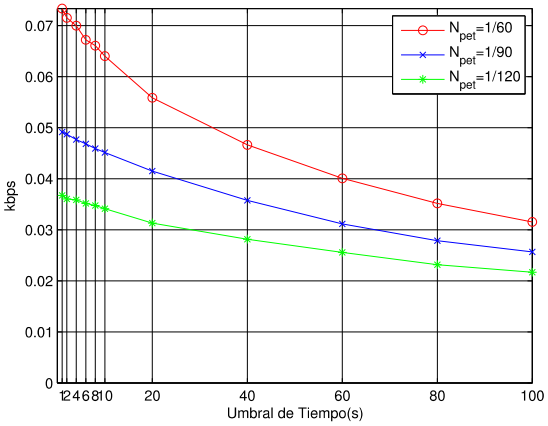


Fig. 16. Ancho de banda usando el agrupamiento

V. CONCLUSIONES

El Path Computation Element Protocol permite realizar el cálculo de rutas complejas descargando a los elementos del plano de control de realizar dicha tarea. Esto permite reducir el coste del equipamiento de la red mejorando el coste por bit de los operadores. El Path Computation Element se integra en las redes de próxima generación dentro de su arquitectura. En este trabajo se ha mostrado el rendimiento de una implementación del PCEP, estudiando en detalle los modos de operación del PCE y la sobrecarga introducida en el plano de control. Se ha demostrado que no introduce una gran cantidad de información en el plano de control.

Como trabajo futuro se extenderán las funcionalidades de esta implementación del PCE a entornos multidominio. Con esta extensión de funcionalidades, se puede estudiar el comportamiento de cada protocolo multidominio definido en distintos escenarios.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el Ministerio de Educación y Ciencia a través del proyecto *ANFORA* (TEC2009-13385).

REFERENCES

[1] Cisco and Associates, “Hyperconnectivity and the Approaching Zettabyte Era,” Cisco, Tech. Rep., 2010.

[2] J. Lobo and F. Jimenez, "Impact of GMPLS on an Integrated operator," in *WGN5: V Workshop in G/MPLS networks*, March 2006.

[3] J. Gabeiras, V. López, J. Aracil, J. Fernández Palacios, C. García Argos, O. González de Dios, F. Jiménez Chico, and J. Hernández, "Is Multilayer Networking Feasible?" *Optical Switching and Networking*, vol. 6, no. 2, pp. 129 – 140, 2009.

[4] ITU-T, "Architecture for the Automatically Switched Optical Network (ASON) - Rec. 8080/Y.1304," February 2003.

[5] E. Mannie, "Generalized Multi-Protocol Label Switching (GMPLS) Architecture," *IETF RFC 3945*, pp. 1–40, October 2004. http://tools.ietf.org/html/rfc3945.

[6] M. N. Ellanti, S. S. Gorshe, L. G. Raman, and W. D. Grover, *Next Generation Transport Networks: Data, Management, and Control Planes*, 1st ed. Springer, April 2005.

[7] A. Farrel, J.P. Vasseur, and J. Ash, "A path computation element (PCE)-based architecture," *IETF RFC 4655*, pp. 1–40, August 2006. http://tools.ietf.org/html/rfc4655.

[8] S. Azodolmolky, M. Klinkowski, E. Marin, D. Careglio, J. Pareta, and I. Tomkos, "A survey on physical layer impairments aware routing and wavelength assignment algorithms in optical networks," *Computer Networks*, vol. 53, no. 7, pp. 926–944, 2009.

[9] F. Cugini, F. Paolucci, L. Valcarenghi, and P. Castoldi, "Implementing a Path Computation Element (PCE) to encompass physical impairments in transparent networks," in *Proceedings of OFC/NFOEC*, 2007, pp. 1–3.

[10] J. Ahmed, P. Monti, and L. Wosinska, "LSP request bundling in a PCE-based WDM network," in *Proc. of IEEE/OSA Optical Fiber Communication Conference and Exposition (OFC), 2009.*, March 2009.

[11] J. Ahmed, C. Cavdar, P. Monti, and L. Wosinska, "An Optimal Model for LSP Bundle Provisioning in PCE-based WDM Networks," in *Proc. of IEEE/OSA Optical Fiber Communication Conference and Exposition (OFC), 2011.*, March 2011.

[12] V. López, B. Huiszoon, Ó. González de Dios, J. Fernández Palacios, and J. Aracil, "Path Computation Element in Telecom Networks: Recent Developments and Standardization Activities." in *Optical Networking Design and Modeling (ONDM)*, February 2010.

[13] K. Shiomoto, D. Papadimitriou, J.L. le Roux, M. Vigoureux, and D. Brungard, "Requirements for GMPLS-based multi-region and multi-layer networks (MRN/MLN)," *IETF RFC 5212*, pp. 1–28, July 2008. Online (Dec. 2009): http://tools.ietf.org/html/rfc5212.

[14] S. Dasgupta, J.C. de Oliveira, and J.P. Vasseur, "Path-computation-element-based architecture for interdomain MPLS/GMPLS traffic engineering: Overview and performance," *IEEE Network*, vol. 21, no. 4, pp. 38–45, July 2007.

[15] J.P. Vasseur and A. Ayyangar and R. Zhang, "A Per-Domain Path Computation Method for Establishing Inter-Domain Traffic Engineering (TE) Label Switched Paths (LSPs)," *IETF RFC 5152*, pp. 1–20, February 2008. http://tools.ietf.org/html/rfc55152.

[16] J.P. Vasseur and R. Zhang and N. Bitar and JL. Le Roux, "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths," *IETF RFC 5441*, pp. 1–16, April 2009. http://tools.ietf.org/html/rfc5441.

[17] J.P. Vasseur and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)," *IETF RFC 5440*, pp. 1–40, March 2009. http://tools.ietf.org/html/rfc5440.

Análisis y modelado en redes de sensores inalámbricas

Vicente Casares-Giner, Diego Pacheco-Paramo, David Todolí-Ferrandis
Departamento de Comunicaciones
Universitad Politécnica de Valencia
Camino de Vera, s/n. 46022 Valencia.
vcasares@dcom.upv.es, diepacpa@posgrado.upv.es, datofer@teleco.upv.es

***Resumen*—Una red de sensores inalámbrica (WSN) consiste en una agrupación de dispositivos de muy pequeño tamaño con la función básica de captar información de un determinado entorno y su posterior envío a un nodo destino denominado sumidero. Los dispositivos, llamados nodos o sensores, tienen una reducida capacidad de auto-funcionamiento por lo que una optimización del consumo energético es de suma importancia para una prolongada vida de la WSN. En una estructura plana, los nodos cercanos al sumidero, -éste ubicado en el centro de un área circular-, cursan más tráfico que los situados en la periferia del área; teniendo así un mayor consumo energético, efecto conocido como *Energy Hole Problem*. Por otra parte, la información ha de fluir desde los nodos más alejados hacia el sumidero, con tiempos de transferencia lo más corto posibles. El presente trabajo aporta un modelo de Markov que modela un sistema TDMA y permite estudiar y analizar el compromiso existente entre los dos factores anteriores: consumo energético y retardo de transferencia de la información.**

***Palabras Clave*—Wireless Sensor Networks.**

I. Introducción

El ahorro energético en las redes de sensores inalámbricas (WSN, de *Wireless Sensor Networks*) es un aspecto de suma importancia. Los pequeños dispositivos que configuran la red, denominados nodos o sensores, tienen una reducida capacidad de autonomía, por lo que un consumo comedido de su energía favorece una mayor vida de la WSN. En una WSN, la función de los nodos es múltiple. Por una parte han de captar la información del pequeño entorno geográfico que los rodea. Por otra parte, la red que conforman ha de encauzar toda la información captada hacia el nodo central, el nodo sumidero. Es decir, el nodo de una WSN hace funciones de fuente de información y de encaminador o enrutador. Por lo tanto, es claro que nodos cercanos al sumidero consumirán con mayor rapidez su limitada energía, cuyo efecto resultante es el conocido *energy hole problem*. Estudios diversos han concluido que cuando los nodos más cercanos al sumidero han consumido su energía inicial, -por tanto dejando aislado el nodo sumidero-, todavía queda disponible un 90% del total de la energía incial de la WSN [1], [2]. De ahí la importancia en disponer de algoritmos que optimicen el uso energético de los nodos de una WSN. Entre otros, merecen citarse las estrategias consistentes en activar el nodo-sensor cuando el número de paquetes de datos en su memoria local alcanza cierto umbral o cuando se percata del envío de paquetes desde un nodo vecino. Tras vaciar su memoria de datos o tras la recepción de una ráfaga de paquetes, el nodo vuelve al estado de hibernación o *sleeping*. La transición entre los estados de *transmisión*, *recepción* e *hibernación* requiere de una mínima coordinación entre nodos-sensores, siendo la frecuencia de transiciones un importante parámetro que influye en el consumo.

Un segundo aspecto es el encaminamiento de la información. La transferencia hacia, y entrega de la misma al sumidero ha de realizarse con el mínimo tiempo posible. Apresurarse en el envío de datos supone un alto riego en colisiones, esto es, de un solape temporal, parcial o total, de dos o más paquetes de datos en el mismo canal de comunicación. Por lo que se precisa de cierto grado de coordinación entre nodos, hecho que inevitablemente añade retardo en la entrega, [3].

En el presente trabajo se supone que los nodos de la WSN pueden operar en los estados de Recepción (R), de Transmisión (T) y de hibernación o *Sleeping*(S). Igualmente suponemos que la WSN dispone de un único radio-canal y el acceso al mismo se ha implementado, por una parte con el protocolo TDMA (*Time Division Multiple Access*) para nodos con posibilidad de interferencia mutua y por otra parte mediante el reuso espacial de la radiofrecuencia cuando no haya posibilidad de interferencia mutua. El trabajo se ha organizado en varias secciones. Tras la presente introducción, en la sección II se describe el escenario en estudio. Las secciones III y IV tratan del modelado y formulación mediante herramientas de Markov, de la caracterización funcional de un nodo. En la sección V se formula una función de coste para afrontar el uso optimizado de la WSN. La sección VI algunos ejemplos ilustran la metodología de análisis propuesta. El trabajo finaliza en la sección VII con las conclusiones.

ISBN: 978-84-694-5948-5

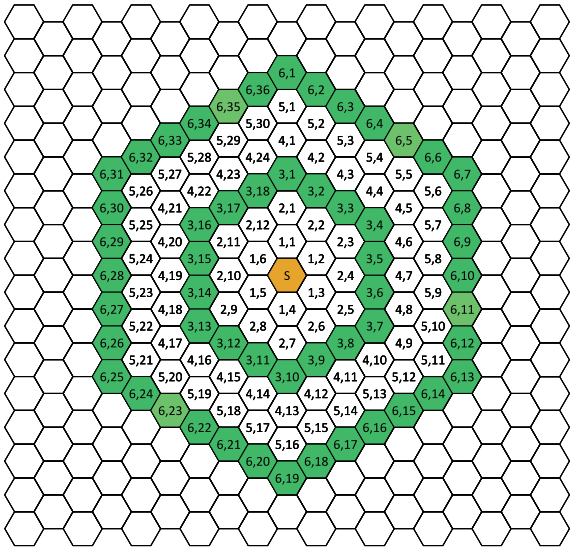


Fig. 1: Etiquetas identificativas de los nodos (i, j)=(anillo, orden dentro del anillo)

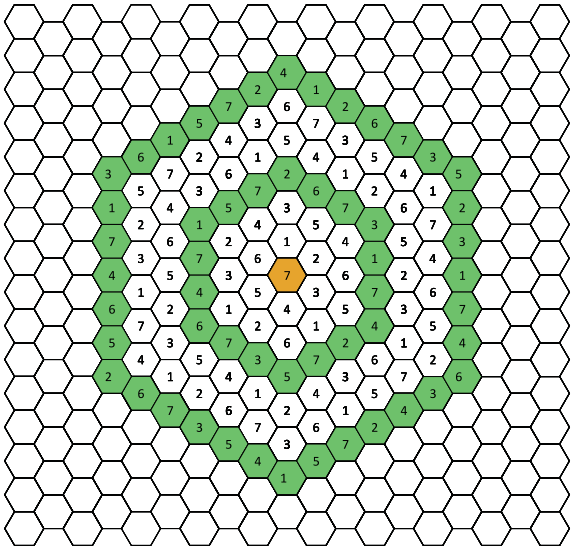


Fig. 2: Todos los nodos con la misma etiqueta, por ejemplo la 3, pueden transmitir durante la ranura temporal 3. El nodo central- etiqueta 7- es el nodo sumidero

II. Escenario de WSN

La Fig. 1 muestra un escenario regular hexagonal en donde hay ubicado un nodo-sensor en todos y cada uno de los centros de los hexágonos. Los nodos pertenecen al anillo 1, 2, ... en función del mínimo número de saltos precisos para alcanzar el nodo sumidero -anillo 0-, el cual se sitúa en el centro del área de servicio de la WSN. La potencia de emisión de cada uno de los nodos es la misma y queda limitada al alcance de cualquiera de sus seis vecinos; por lo tanto en presencia de saltos unitarios. Se supone que la WSN opera a una única frecuencia. Así pues, para evitar interferencias, cuando un nodo transmite, los demás vecinos no lo pueden hacer.

Sumidero	Anillo (# de nodos) 1 (6)	2 (12)	3 (18)	4 (24)	5 (30)	...
6+1	1					
18+1	3	1				
36+1	6	$^5/_2$	1			
60+1	10	$^9/_2$	$^7/_3$	1		
90+1	15	$^{14}/_2$	$^{12}/_3$	$^9/_4$	1	
120+1	21	$^{20}/_2$	$^{18}/_3$	$^{15}/_4$	$^{11}/_5$	...
⋮	⋮	⋮	⋮	⋮	⋮	⋱

Tabla I: Cargas por nodo, según posición en anillos. Se supone que cada nodo capta una tasa unitaria de su área de servicio y la transfiere en modo reparto de carga a nodos del anillo vecino interior.

El protocolo de encaminamiento consiste en transferir información hacia el algún nodo del anillo interior al que pertenece el nodo transmisor. Por consiguiente, las tasas de paquetes de datos que, en media, soporta cada nodo queda reflejada en la tabla I adjunta.

Admitiendo sincronización perfecta entre nodos, un adecuado plan de frecuencias sería arbitrar una multipexación temporal, consistente en una trama TDMA con 7 intervalos o *slots* temporales. En la Fig. 2, un nodo que tenga la etiqueta, i, $i = 1, 2, 3, 4, 5, 6, 7$, indica que puede transmitir en la ranura temporal i. Cuando un nodo no transmita podrá recibir datos de otro nodo perteneciente al anillo externo o alternativamente podrá estar en fase de hibernación o *sleeping*. Por lo tanto, para cada nodo de la WSN se distinguen tres estados: modo de recepción (estado R), de hibernación o *sleeping* (estado S) y en modo transmisión (estado T). La tabla II muestra el formato de trama configurada con 7 *slots*, con la secuencia de sicronización temporal de los tres estados por nodo, para los tres primeros anillos de la WSN. Por ejemplo el nodo $(2, 8)$ puede transmitir en el *slot* 2 y su transmisión puede ser recibida por los nodos $(1, 4)$ y $(1, 5)$. A su vez, el citado nodo $(2, 8)$ puede recibir información de los nodos $(3, 11)$ y $(3, 12)$ en los intervalos temporales, respectivamente 3 y 7. Durante los intervalos $1, 4, 5$ y 6 el nodo $(2, 8)$ tiene opción al estado S permitiendo así un ahorro energético.

III. Modelo de Markov con tres estados

La idea consiste en modelar la WSN mediante una red de colas, en donde cada nodo se representa por un único servidor y una cola de espera Fig. 3. Cada nodo puede alcanzar uno de los tres estados anteriormente descritos: R, S y T. En primera instancia abordamos el estudio viendo el comportamiento aislado de cada nodo. En este trabajo, como primera aproximación se ha supuesto un proceso de Markov con los tres estados (R, S, T), y con tasas de transición - a determinar-

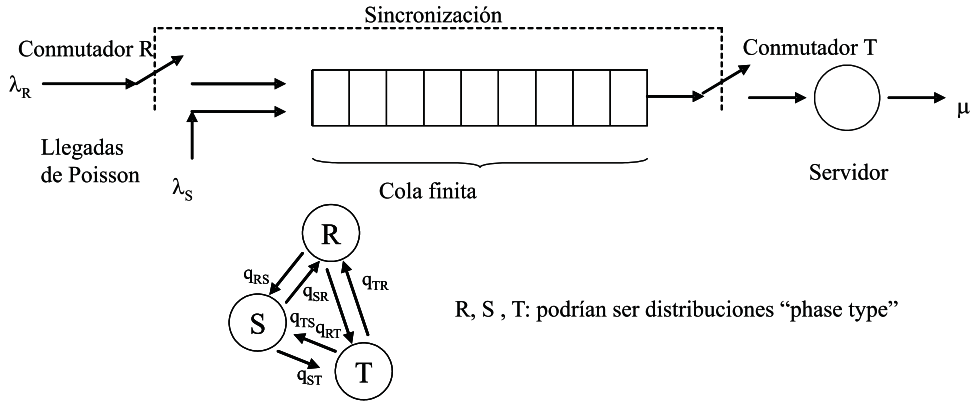


Fig. 3: Modelo de Markov de tres estados para un nodo de la WSN

	Número de "slot" en la trama						
Anillo, Nodo	1	2	3	4	5	6	7
1, 1	T		R	R	R		
1, 2		T		R	R	R	
1, 3	R		T		R	R	
1, 4	R	R		T		R	
1, 5	R	R	R		T		
1, 6		R	R	R		T	
2, 1		R	T			R	R
2, 2					T	R	R
2, 3	R		R	T			R
2, 4	R					T	R
2, 5		R		R	T		R
2, 6	T	R					R
2, 7			R		R	T	R
2, 8		T	R				R
2, 9	T			R		R	R
2, 10			T	R			R
2, 11	R	T			R		R
2, 12				T	R		R
3, 1	R	T		R	R		
3, 2	R			R		T	
3, 3	R	R					T
3, 4		R	T		R	R	
3, 5	T	R			R		
3, 6		R	R				T
3, 7	R		R	T		R	
3, 8		T	R			R	
3, 9			R	R			T
3, 10	R	R		R	T		
3, 11	R		T	R			
3, 12				R	R		T
3, 13		R	R		R	T	
3, 14		R		T	R		
3, 15					R	R	T
3, 16	T		R	R		R	
3, 17			R		T	R	
3, 18	R					R	T

Tabla II: Estructura TDMA para los tres primeros anillos alrededor del sumidero. Por claridad en la figura, mientras los estados R y T figuran etiquetados, el estado S aparece en blanco, sin etiqueta alguna.

dadas por $q_{i,j}$, $i \neq j \in (R, S, T)$. Es claro que la solución Markoviana ofrece tiempos exponenciales para la residencia en los estados R, S y T y no deterministas como indica la tabla II. No obstante, el presente estudio es una primera aproximación y la intención es captar el comportamiento de la WSN en los parámetros de retardo de transferencia y de consumo energético. Para ello, en nuestro modelo ajustaremos los valores medios exponenciales con los valores deterministas de la tabla II.

Para un nodo cualquiera, la Fig. 3 muestra el modelo de colas en consideración. El estado del nodo viene dado por la posición de los conmutadores R y T. Cada estado se identifica con los conmutadores en las posiciones según se muestra en la tabla III adjunta

Estado	Etiqueta	Conmutador R	Conmutador T
1	R	ON	OFF
2	S	OFF	OFF
3	T	OFF	ON
Estado no válido		ON	ON

Tabla III: Estados asociados a la posición de los conmutadores.

Por lo tanto el modo de trabajo de cualquier nodo viene dado por las siguientes pautas:

- El estado del nodo es irrelevante a efectos de captación de datos locales. Esto es, percibe información de su área de cobertura en cualquier instante de tiempo, con independencia de la posición de los conmutadores R y T.
- Cuando las posiciones de los conmutadores R y T sean $(R, T) = (OFF, ON)$, un nodo situado en el anillo r podrá transmitir información a otro nodo perteneciente al anillo $r - 1$ siempre y cuando la fase $T = ON$ de nuestro nodo coincida con la fase $R = ON$ del referido nodo del anillo $r - 1$.
- Cuando las posiciones de los conmutadores R y T sean $(R, T) = (ON, OFF)$, un nodo situado en el anillo r podrá recibir información de otro nodo ubicado en el anillo $r+1$ siempre y cuando

la fase $R = ON$ de nuestro nodo coincida con la fase $T = ON$ del referido nodo del anillo $r + 1$.

- Cuando las posiciones de los conmutadores R y T sean $(R, T) = (OFF, OFF)$, el nodo se encontrará en el estado S y sólo percibirá la información que se genere en modo local.

En el diagrama de transición de estados de la Fig. 3 identificamos (R, S, T) y $q_{i,j}$. Las probabilidades en régimen permanente, $\{\pi_i\}, i = R, S, T$,vienen dadas por:

$$\begin{aligned}\pi_R &= B[q_{S,R}q_{T,R} + q_{T,S}q_{S,R} + q_{S,T}q_{T,R}]\\ \pi_S &= B[q_{T,S}q_{R,S} + q_{R,T}q_{T,S} + q_{T,R}q_{R,S}]\\ \pi_T &= B[q_{R,T}\lambda_{S,T} + q_{S,R}q_{R,T} + q_{R,S}q_{S,T}]\end{aligned} \quad (1)$$

con

$$\begin{aligned}B^{-1} &= [q_{S,R}q_{T,R} + q_{T,S}q_{S,R} + q_{S,T}q_{T,R}]+\\ &+[q_{T,S}q_{R,S} + q_{R,T}q_{T,S} + q_{T,R}q_{R,S}]+\\ &+[q_{R,T}q_{S,T} + q_{S,R}q_{S,T} + q_{R,S}q_{S,T}]\end{aligned} \quad (2)$$

El tiempo medio entre visitas consecutivas a un determinado estado es una v.a. que denotamos por $c_i(t), \quad i \in [R, S, T]$. Las respectivas transformadas de Laplace, $c_i^*(s)$, resultan:

$$\begin{aligned}&c_R^*(s) =\\ &p_{R,S}\left(\frac{s_S^*(s)\left[p_{S,R} + s_T^*(s)p_{S,T}p_{T,R}\right]}{1 - s_S^*(s)s_T^*(s)p_{S,T}p_{T,S}}\right)s_R^*(s)+\\ &+p_{R,T}\left(\frac{s_T^*(s)\left[p_{T,R} + s_S^*(s)p_{T,S}p_{S,R}\right]}{1 - s_S^*(s)s_T^*(s)p_{S,T}p_{T,S}}\right)s_R^*(s)\\ &c_S^*(s) =\\ &p_{S,T}\left(\frac{s_T^*(s)\left[p_{T,S} + s_R^*(s)p_{T,R}p_{R,S}\right]}{1 - s_T^*(s)s_R^*(s)p_{T,R}p_{R,T}}\right)s_S^*(s)+\\ &+p_{S,R}\left(\frac{s_R^*(s)\left[p_{R,S} + s_T^*(s)p_{R,T}p_{T,S}\right]}{1 - s_T^*(s)s_R^*(s)p_{T,R}p_{R,T}}\right)s_S^*(s)\\ &c_T^*(s) =\\ &p_{T,R}\left(\frac{s_R^*(s)\left[p_{R,T} + s_S^*(s)p_{R,S}p_{S,T}\right]}{1 - s_R^*(s)s_S^*(s)p_{R,S}p_{S,R}}\right)s_T^*(s)+\\ &+p_{T,S}\left(\frac{s_S^*(s)\left[p_{S,T} + s_R^*(s)p_{S,R}p_{R,T}\right]}{1 - s_R^*(s)s_S^*(s)p_{R,S}p_{S,R}}\right)s_T^*(s)\end{aligned} \quad (3)$$

en donde las probabilidades $p_{i,j}$ y las funciones de variable compleja $s_i^*(s)$ resultan dadas por:

$$p_{i,j} = \frac{q_{i,j}}{q_{i,j} + q_{i,k}} = \frac{q_{i,j}}{q_i}, \quad p_{i,j} + p_{i,k} = 1, \quad i \neq j \neq k \in [R, S, T]. \quad (4)$$

$$s_i^*(s) = \frac{q_i}{s + q_i}, \quad i \in [R, S, T] \quad (5)$$

Derivando en (3) y particularizando para $s = 0$ obtenemos los tiempos medios de ciclo:

$$\begin{aligned}\overline{c}_R &= \frac{1}{q_R\pi_R} = \overline{s}_R + p_{R,S}\left(\frac{\overline{s}_S + p_{S,T}\overline{s}_T}{1 - p_{S,T}p_{T,S}}\right)+\\ &+p_{R,T}\left[\frac{\overline{s}_T + p_{T,S}\overline{s}_S}{1 - p_{T,S}p_{S,T}}\right]\\ \overline{c}_S &= \frac{1}{q_S\pi_S} = \overline{s}_S + p_{S,T}\left(\frac{\overline{s}_T + p_{T,R}\overline{s}_R}{1 - p_{T,R}p_{R,T}}\right)+\\ &+p_{S,R}\left[\frac{\overline{s}_R + p_{R,T}\overline{s}_T}{1 - p_{R,T}p_{T,R}}\right]\\ \overline{c}_T &= \frac{1}{q_T\pi_T} = \overline{s}_T + p_{T,R}\left(\frac{\overline{s}_R + p_{R,S}\overline{s}_S}{1 - p_{R,S}p_{S,R}}\right)+\\ &+p_{T,S}\left[\frac{\overline{s}_S + p_{S,R}\overline{s}_R}{1 - p_{S,R}p_{R,S}}\right]\end{aligned} \quad (6)$$

Al igual que en (3), las probabilidades p_{ij} de (6) vienen dadas en (4) y

$$\overline{s}_i = \frac{1}{q_i} = \frac{p_{i,j}}{q_{i,j}}, \quad i \neq j = R, S, T. \quad (7)$$

Las probabilidades $p_{i,j}$ pueden obtenerse al inspeccionar las transiciones entre estados R, S (en blanco) y T en la tabla II. Dichas probabilidades se muestran en la tabla IV para los tres primeros anillos de la WSN.

Los valores medios de estancia en los estados (R, S, T), también pueden obtenerse por simple inspección de la tabla II. Los valores normalizados a la duración del estado T, $\overline{s}_3 = \Delta$, esto es, $\overline{s}_i/\Delta$, se muestran en las columnas de la derecha de la tabla IV.

IV. Estudio analítico de un nodo con cola

La Fig. 4 muestra el correspondiente diagrama de estados para un nodo cualquiera de la WSN. Los subíndices en los estados R, S y T corresponden al número de paquetes que hay en cada nodo. En las siguientes líneas procedemos a su estudio.

A. Modelo de Markov

La Fig. 4 muestra un proceso QBD (*Quasi-Birth-Death*), [4]. Admitiendo cola infinita en todos y cada uno de los nodos, el generador infinitesimal nos viene dado por:

$$\mathbf{Q} = \begin{bmatrix}\mathbf{B}_0 & \mathbf{A}_0 & 0 & 0 & \dots\\ \mathbf{B}_1 & \mathbf{A}_1 & \mathbf{A}_0 & 0 & \dots\\ 0 & \mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_0 & \dots\\ 0 & 0 & \mathbf{A}_2 & \mathbf{A}_1 & \dots\\ \vdots & \vdots & \vdots & \vdots & \ddots\end{bmatrix} \quad (8)$$

Las matrices de (8) satisfacen $[\mathbf{B}_0 + \mathbf{A}_0]\mathbf{e} = \mathbf{0}$, $[\mathbf{B}_1 + \mathbf{A}_0 + \mathbf{A}_0]\mathbf{e} = \mathbf{0}$, y $[\mathbf{A}_2 + \mathbf{A}_0 + \mathbf{A}_0]\mathbf{e} = \mathbf{0}$, siendo $\mathbf{e}$ un vector columna con todos sus elementos a 1, y

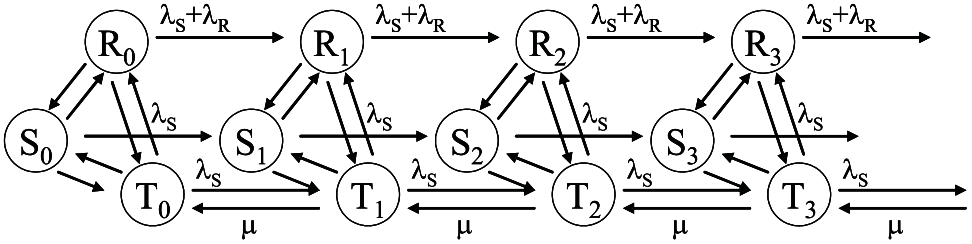


Fig. 4: Proceso QBD en un nodo de la WSN con tamaño de buffer infinito

	$\overline{s}_1 q_{1,2}$	$\overline{s}_2 q_{2,3}$	$\overline{s}_3 q_{3,1}$	$\overline{s}_1/\Delta$	$\overline{s}_2/\Delta$	$\overline{s}_3/\Delta$
1, 1	1	1/2	0	3	3/2	1
1, 2	1	1/2	0	3	3/2	1
1, 3	1	1/3	0	3/2	1	1
1, 4	1	1/3	0	3/2	1	1
1, 5	1	1/2	0	3	3/2	1
1, 6	1	1/2	0	3	3/2	1
2, 1	1/2	0	0	3/2	3/2	1
2, 2	1	1	1	2	4	1
2, 3	1/2	0	0	3/2	3/2	1
2, 4	1	1	1	2	4	1
2, 5	2/3	0	0	1	1	1
2, 6	1/2	0	1	1	4	1
2, 7	2/3	0	1	1	3/2	1
2, 8	1	1/2	1	1	2	1
2, 9	1/2	0	0	3/2	3/2	1
2, 10	1	1/2	1	1	2	1
2, 11	1/2	1	0	3/2	3/2	1
2, 12	1	1/2	1	1	2	1
3, 1	1/2	0	0	3/2	3/2	1
3, 2	1	1/3	0	1	3/2	1
3, 3	1	1	1	2	4	1
3, 4	1/2	0	0	3/2	3/2	1
3, 5	1	1/2	1	1	2	1
3, 6	1	1/2	0	2	2	1
3, 7	2/3	0	0	1	1	1
3, 8	1	1/2	1	1	2	1
3, 9	1	1/2	0	2	2	1
3, 10	1/2	0	0	3/2	3/2	1
3, 11	1	1/2	1	1	2	1
3, 12	1	1/2	0	2	2	1
3, 13	1/2	0	0	3/2	3/2	1
3, 14	1	1/2	1	1	2	1
3, 15	0	0	0	2	4	1
3, 16	1	1/3	0	3/2	1	1
3, 17	1	1/2	1	1	2	1
3, 18	1/2	0	1	1	4	1

Tabla IV: Columna 1: (anillo, nodo). Columnas 2-4: Probabilidades de transición entre estados. Columnas 4-7: tiempos de estancia en los estados, normalizados al de estancia en el estado T, Δ unidades de tiempo.

$$\mathbf{B}_0 = \begin{bmatrix} * & q_{R,S} & q_{R,T} \\ q_{S,R} & * & q_{S,T} \\ q_{T,R} & q_{T,S} & * \end{bmatrix} \quad \mathbf{B}_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \mu \end{bmatrix} \tag{9}$$

$$\mathbf{A}_2 = \mathbf{B}_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \mu \end{bmatrix} \tag{10}$$

$$\mathbf{A}_1 = \begin{bmatrix} * & q_{R,S} & q_{R,T} \\ q_{S,R} & * & q_{S,T} \\ q_{T,R} & q_{T,S} & * \end{bmatrix} \tag{11}$$

$$\mathbf{A}_0 = \begin{bmatrix} \lambda_R + \lambda_S & 0 & 0 \\ 0 & \lambda_S & 0 \\ 0 & 0 & \lambda_S \end{bmatrix} \tag{12}$$

en donde el valor de cada uno de los astericos, $*$, en $\mathbf{B}_0$ y en $\mathbf{A}_1$ es negativo, de valor absoluto igual a la suma de todos los restantes elementos de la misma fila en $\mathbf{Q}$.

La matríz $\mathbf{B}_0 + \mathbf{R}\mathbf{B}_1$ es un generador infinitesimal y $\mathbf{R}$ es la *ratio matrix* la cual satisface la ecuación cuadrática:

$$\mathbf{R}^2\mathbf{A}_2 + \mathbf{R}\mathbf{A}_1 + \mathbf{A}_0 = \mathbf{0} \tag{13}$$

Entonces, la obtención del vector de probabilidades estacionarias, $\boldsymbol{\pi} = [\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \ldots,]$, con $\boldsymbol{\pi}_i = (\pi_{R,i}, \pi_{S,i}, \pi_{T,i})$, $i = 0, 1, 2, \ldots$, satisface $\mathbf{0} = \boldsymbol{\pi}\mathbf{Q}$ y puede llevarse a término mediante:

$$\boldsymbol{\pi}_i = \boldsymbol{\pi}_0 \mathbf{R}^i; \quad i = 0, 1, 2, ... \tag{14}$$

$$\mathbf{0} = \boldsymbol{\pi}_0(\mathbf{B}_0 + \mathbf{R}\mathbf{B}_1) \tag{15}$$

$$\boldsymbol{\pi}_0(\mathbf{I} - \mathbf{R})^{-1}\mathbf{e} = 1 \tag{16}$$

La matrix $\mathbf{R}$ puede calcularse forma iterativa:

$$\mathbf{R}_{j+1} = -\left[\mathbf{R}_j^2\mathbf{A}_2 + \mathbf{A}_0\right]\mathbf{A}_1^{-1} \quad \text{con} \quad \mathbf{R}_0 = \mathbf{0}. \tag{17}$$

Comentar que la matríz $\mathbf{A} = \mathbf{A}_2 + \mathbf{A}_1 + \mathbf{A}_0$ es un generador infinitesimal. $\mathbf{A}$ es coincidente con el proceso de Markov de tres estados de la Fig. 3 por lo que el vector estocástico solución de $\mathbf{0} = (\pi_R, \pi_S, \pi_T)\mathbf{A}$ es el dado en (1). También conviene resaltar la siguiente identidad:

$$\pi_i = \sum_{n=0}^{\infty} = \pi_{i,n}, \quad i = R, S, T. \tag{18}$$

La condición de ergodicidad viene dada por, [4]:

$$(\pi_R, \pi_S, \pi_T)\mathbf{A}_0\mathbf{e} < (\pi_R, \pi_S, \pi_T)\mathbf{A}_2\mathbf{e} \quad \rightarrow \quad \lambda_R\pi_R + \lambda_S < \mu\pi_T. \tag{19}$$

Denotando por ρ_L la carga ofrecida a un nodo, la condición de estabilidad (19) equivale a

$$\rho_L = \frac{\rho_R \pi_R + \rho_S}{\pi_T} < 1 \quad \text{con} \quad \rho_R = \frac{\lambda_R}{\mu}, \quad \rho_S = \frac{\lambda_S}{\mu} \tag{20}$$

B. Parámetros de prestaciones

En el presente estudio nos detendremos en el número medio de unidades en la cola, $\overline{m}_q$, y en la frecuencia de visitas a los estados R, S y T. Las razones se explican en la sección V. El primer parámetro, $\overline{m}_q$, viene dado por:

$$\overline{m}_q = \sum_{m=1}^{\infty} (m-1)\boldsymbol{\pi}_m \mathbf{e} = \boldsymbol{\pi}_0 \sum_{m=1}^{\infty} (m-1)\mathbf{R}^m \mathbf{e} = \boldsymbol{\pi}_0 \mathbf{R}^2 (\mathbf{I}-\mathbf{R})^{-\mathbf{2}} \mathbf{e} \tag{21}$$

siendo $(\mathbf{I}-\mathbf{R})^{-\mathbf{2}}$ el cuadrado de la matriz $(\mathbf{I}-\mathbf{R})^{-\mathbf{1}}$.

Mediante la relación de Little, [5], el tiempo medio de espera de un paquete elegido al azar, viene expresado por:

$$\overline{W}_q = \frac{\overline{m}_q}{\lambda_R \pi_R + \lambda_S} = \frac{\boldsymbol{\pi}_0 \mathbf{R}^2 (\mathbf{I}-\mathbf{R})^{-\mathbf{2}} \mathbf{e}}{\lambda_R \pi_R + \lambda_S} \tag{22}$$

Normalizando con respecto al tiempo de transferencia de un paquete, $1/\mu$, tendremos:

$$\overline{W}_{q,n}(\Delta, \mu) = \mu \overline{W}_q = \frac{\boldsymbol{\pi}_0 \mathbf{R}^2 (\mathbf{I}-\mathbf{R})^{-\mathbf{2}} \mathbf{e}}{\rho_R \pi_R + \rho_S}; \tag{23}$$

Sea F_i el número de visitas por unidad de tiempo al estado i, $i = R, S, T$, frecuencia que viene dada por la inversa del tiempo medio de ciclo $F_i = 1/\overline{c}_i = q_i \pi_i$. Normalizando F_i con respecto a la tasa de servicio μ, tendremos.

$$F_{R,n}(\Delta, \mu) = \frac{1}{\overline{c_R}\mu} = \frac{q_R \pi_R}{\mu} = \theta_R \pi_R \tag{24}$$

$$F_{S,n}(\Delta, \mu) = \frac{1}{\overline{c_S}\mu} = \frac{q_S \pi_S}{\mu} = \theta_S \pi_S \tag{25}$$

$$F_{T,n}(\Delta, \mu) = \frac{1}{\overline{c_T}\mu} = \frac{q_T \pi_T}{\mu} = \theta_T \pi_T \tag{26}$$

V. Proceso de optimización

Resulta intuitivo observar que valores pequeños de $\overline{s}_3 = \Delta$ implican tiempos de ciclo pequeños ($\overline{c}_1, \overline{c}_2, \overline{c}_3$, (6)) al tiempo que una frecuente disposición del servidor en estado de trasmisión T, lo que supone retardos bajos en la entrega de datos. Por el contrario, altos valores de Δ implican ciclos de mayor duración, con una menor frecuencia de disponibilidad del transmisor y en consecuencia altos retardos en la entrega de datos. Por otra parte, la WSN ha de optimizarse en cuanto a consumo energético por unidad de tiempo. Para ello definimos una medida del consumo. Para un nodo cualquiera, sean C_R, C_S y C_T los consumos por unidad de tiempo en los estados R, S y T, respectivamente. Sean C_{sw-R}, C_{sw-S} y C_{sw-T} los consumos asociados a frecuencia de activación de los estados R, S y T. Hacemos notar que para el estado S es lógico suponer que $C_{sw-S} = 0$. Teniendo en cuenta (24), (25) y (26), la función del coste energético la definimos como:

$$\begin{aligned} F_n(\Delta, \mu) &= \sum_{n=0}^{\infty} [C_R \pi_{R,n} + C_S \pi_{S,n} + C_T \pi_{T,n}] + \\ &+ C_{sw-R} F_{R,n}(\Delta, \mu) + C_{sw-T} F_{T,n}(\Delta, \mu) = \\ &= C_R \pi_R + C_S \pi_S + C_T \pi_T + \\ &+ C_{sw-R} \theta_R \pi_R + C_{sw-T} \theta_T \pi_R = \\ &= (C_R + C_{sw-R} \theta_R) \pi_R + C_S \pi_S + \\ &+ (C_T + C_{sw-T} \theta_T) \pi_T = \\ &= \boldsymbol{\pi}_0 (\mathbf{I}-\mathbf{R})^{-1} \big[(C_R + C_{sw-R} \theta_1) \mathbf{e}_R + \\ &+ C_S \mathbf{e}_2 + (C_T + C_{sw-T} \theta_T) \mathbf{e}_3 \big]. \end{aligned} \tag{27}$$

siendo $\mathbf{e}_k$ un vector columna con un 1 en la posición k-ésima y los restantes elementos iguales a 0, $k = R, S, T$.

Definimos una función global de coste como una suma ponderada de (23), y de (27), esto es:

$$C(\Delta, \mu) = C_W \overline{W}_{q,n}(\Delta, \mu) + C_F F_n(\Delta, \mu) \tag{28}$$

VI. Resultados ilustrativos

A efectos ilustrativos hemos supuesto una WSN de cuatro anillos, con una carga de tráfico normalizada a $\rho_L = 0.8$, ecuación (20), para los nodos del primer anillo. Por consiguiente, según la tabla I a los nodos de los anillos 2, 3 y 4 les corresponden cargas de tráfico ρ_L iguales a 0.36, $0,186\hat{6}$ y 0.08 respectivamente. En referencia a las constantes de (27) se han elegido valores de $C_R = 10$, $C_S = 1$, $C_T = 10$; $C_{sw-R} = 40$, $C_{sw-S} = 0$, $C_{sw-T} = 40$, esto es valores similares a los considerados en [6]. Para la ponderación de (28), hemos tomado $C_W = 0.05$ y $C_F = 1$.

Las Fig. 5, 6 muestran los costes de retardo de transferencia, los costes de consumo energético y los costes totales para los nodos $(1,1)$ y $(1,4)$. Los costes obedecen, respectivamente, a la expresión (23) ponderada por el parámetro C_W, a la expresión (27) ponderada por el parámetro C_F y la suma ponderada de (23) y de (27), ecuación (28). Los mismos valores se ofrecen en las gráficas 7, 8, 9, 10, 11, y 12, respectivamente para los nodos $(2,1)$, $(2,6)$, $(3,3)$, $(3,7)$, $(4,3)$ y $(4,7)$. Es interesante observar el alto

consumo energético de los nodos del primer anillo en comparativa con los nodos del cuarto anillo, -los de menor consumo, como cabía esperar-. También resulta interesante apreciar la variación de los consumos energéticos con respecto al parámetro $\Delta\mu$. En el primer anillo obtenemos valores óptimos de $\Delta^*\mu$ entre 2 y 3. Para el segundo anillo, los valores óptimos de $\Delta^*\mu$ están entre 7 y 9 aproximadamente; valores que se incrementan ligeramente para los nodos del tercer y cuarto anillo. En consecuencia, hemos de anotar cierta discrepancia -no muy grande- entre los valores óptimos de cada anillo. No obstante, dado el elevado consumo energético de los nodos del primer anillo con respecto a los demás, cabe esperar que los valores óptimos del primer anillo sean de mayor influencia en un adecuado diseño de la WSN desde el punto de vista de eficiencia energética.

VII. Conclusiones

El presente trabajo trata de contribuir al estudio del problema EHP (*Energy Hole Problem*), problema muy común en las actuales WSN. Con el fin de paliar el consumo energético superfluo, se ha aportado un modelo de tres estados, modelo $R-S-T$ para cada nodo, correspondiente a recepción, *sleeping*y transmisión. Un modelo de Markov nos ha servido para poder captar los comportamientos esenciales en cuanto a costes energéticos debido a la frecuencia de cambios entre estados, los costes energéticos asociados al funcionamiento en los propios estados, y el coste asociado al retardo de entrega de datos al nodo sumidero. Los resultados muestran un claro compromiso entre los costes citados, y se han vislumbrado duraciones de trama óptimos que minimizan los costes totales ponderados. El trabajo de futuro, ya en curso, consiste en, por una parte extender los modelos Markovianos a otros semi-Markovianos que contemplen distribuciones no exponenciales más cercanas a la realidad de una WSN. Por otra parte se prevé la simulación de entornos reales, con el fin de corroborar las tendencias de los resultados aquí obtenidos y los posibles que se deriven al utilizar modelos extendidos al propuesto.

Agradecimientos

Los autores agradecen el soporte y la financiación recibida a través de los proyectos nacionales TIN2010-21378-C02-02, TEC2010-12250-E y de la red europea Euro-NF (FP7, IST 216366).

Referencias

[1] A. Wadaa, S. Olariu, L. Wilson, M. Eltowissy, K. Jones, "Training a wireless sensor networks," Mobile Networks and Applications", vol. 10, issue 1-2, pp. 151-168, Feb. 2005.

[2] J. Lian, K. Naik, G. Agnew, "Data capacity improvement of wireless sensor networks using non- uniform sensor distribution," Inter. Journal of Distributed Sensor Networks, vol. 2. no. 2, pp. 121-145, Apr.-Jun. 2006.

[3] J. L. Hammond, P. J. P. O'Reilly, " Performance analysis of local computer networks", AddisonWesley, 1986.

[4] M. F. Neuts, "Matrix-geometric solutions in stochastic models". Johns Hopkins University Press, 1981.

[5] L. Kleinrock, "Queueing theory. Volume 1: Theory", John Wiley, 1975.

[6] Fuu-Cheng Jiang, Hsiang-Wei Wu, Der-Chen Huang, Chu-Hsing Lin, "Life time security improvement in Wireless Sensor Network using queue-based techniques". Proceedings of the Int. Conf. on Broadband, Wireless Computing, Communication and Applications, pp. 469-474, 2010.

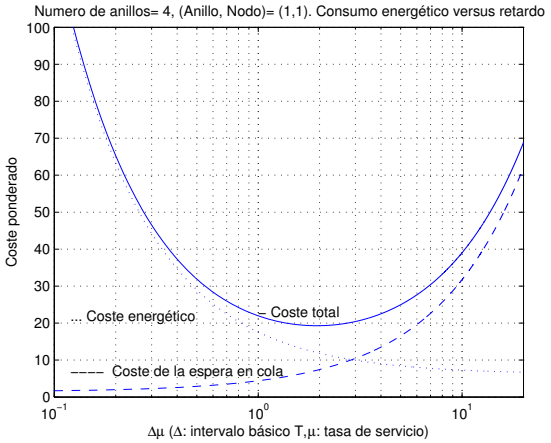


Fig. 5: WSN con 4 anillos. Coste total ponderado para el nodo $(1,1)$.

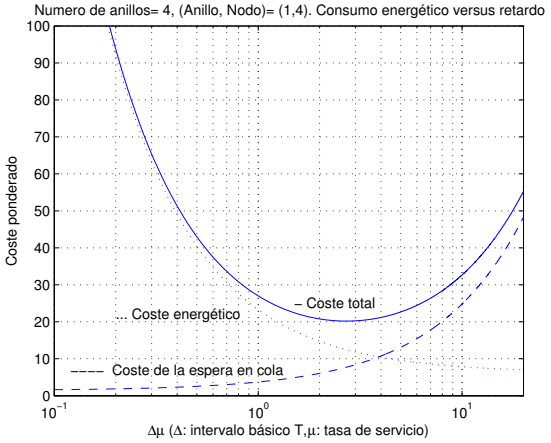


Fig. 6: WSN con 4 anillos. Coste total ponderado para el nodo $(1,4)$.

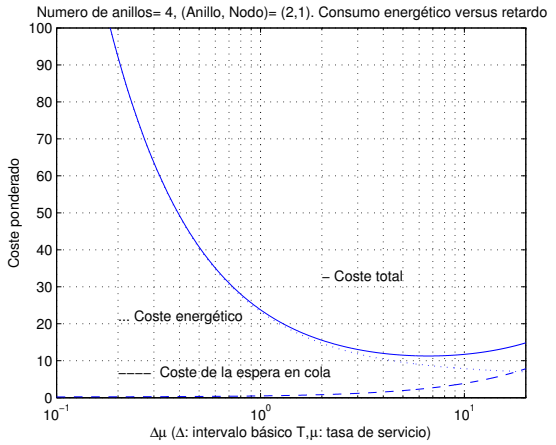


Fig. 7: WSN con 4 anillos. Coste total ponderado para el nodo $(2, 1)$.

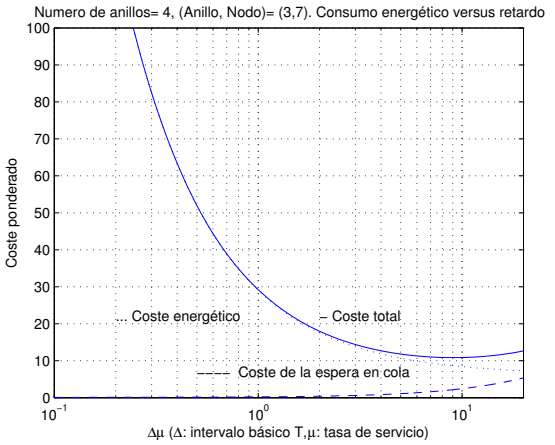


Fig. 10: WSN con 4 anillos. Coste total ponderado para el nodo $(3, 7)$.

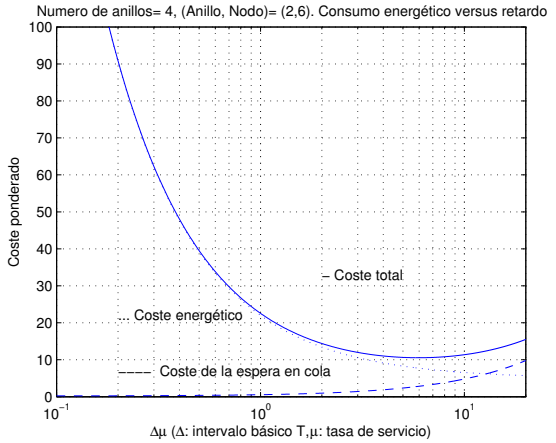


Fig. 8: WSN con 4 anillos. Coste total ponderado para el nodo $(2, 6)$.

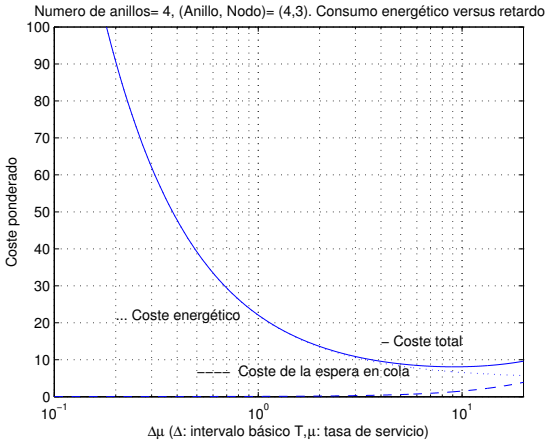


Fig. 11: WSN con 4 anillos. Coste total ponderado para el nodo $(4, 3)$.

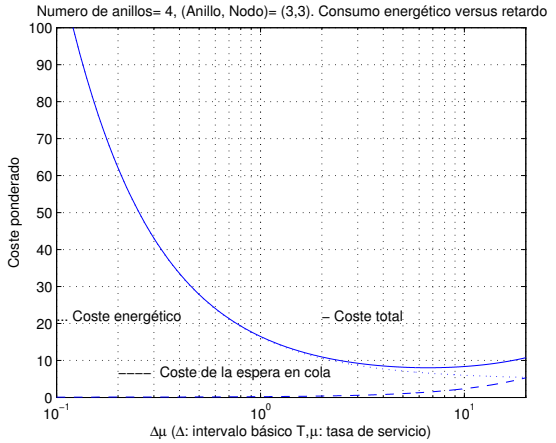


Fig. 9: WSN con 4 anillos. Coste total ponderado para el nodo $(3, 3)$.

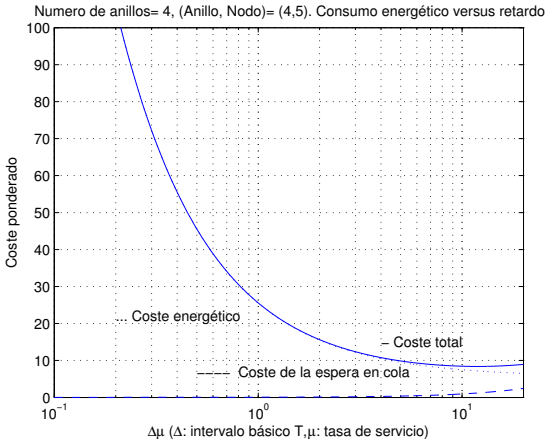


Fig. 12: WSN con 4 anillos. Coste total ponderado para el nodo $(4, 5)$.

Truetime Extendido: Un marco de simulación para el estudio de sistemas WNCS

Juan Jiménez*, Rafael Estepa*, Francisco R. Rubio†, Fabio Gómez-Estern† and Antonio Estepa*
*Departamento de Ingeniería Telemática
Universidad de Sevilla
Email: {juanjimenez, rafa, aestepa}@trajano.us.es
†Departamento de Ingeniería de Sistemas y Automática
Universidad de Sevilla
Email: rubio@esi.us.es, fgs@cartuja.us.es

***Resumen*—Mediante la mejora del paquete Matlab TrueTime, el presente trabajo propone un marco de simulación para el análisis y diseño de sistemas de control a través de redes inalámbricas, conocidos como WNCS (*Wireless Networked Control Systems*). Este entorno proporciona herramientas matemáticas útiles para el diseño de controladores, ausentes en simuladores de red como Opnet o Ns-2, al tiempo que ofrece, como se demostrará, un nivel de precisión similar. Se ha refinado y ampliado el simulador Truetime para mejorar su precisión en el cálculo de las pérdidas, retardo y consumo energético de la red. Para validar el entorno propuesto se han realizado múltiples simulaciones, comparando los resultados de Truetime con los proporcionados por un simulador aceptado comúnmente como Opnet.**

***Palabras Clave*—WNCS; Truetime; simulación; IEEE 802.11; control en tiempo real**

I. Introducción

En los últimos años cada vez hay más sistemas de control que se apoyan en redes 802.11 para la transmisión de información en tiempo real. Actualmente este tipo de redes constituyen un sistema de comunicación de alta velocidad (802.11n), energéticamente eficiente (modo de ahorro de energía) y capaz de cumplir exigencias de calidad de servicio (802.11e) que se ajusta perfectamente a las necesidades de los de sistemas de control en tiempo real. Sin embargo, el uso de esta tecnología aún conlleva ciertos riesgos, ya que la aleatoriedad inherente a los sistemas de comunicación inalámbricos, causada por la variabilidad del canal de transmisión y por el mecanismo de acceso compartido, dificulta el diseño de sistemas de control eficientes y robustos.

Para afrontar estos problemas, los diseñadores de sistemas de control a menudo recurren a herramientas de simulación que les permiten estudiar el comportamiento de los sistemas bajo múltiples condiciones de funcionamiento. Desgraciadamente, la mayoría de estos simuladores basados en Matlab, no cuentan con modelos de red 802.11 y el comportamiento de la red se suele caracterizar con un modelo probabilístico muy simplificado que se utiliza para estimar las métricas de red que influyen en el sistema de control: el retardo y las pérdidas. La falta de precisión de estos modelos de red compromete la fiabilidad de sus resultados y en consecuencia, la eficacia del diseño. Por otro lado, existen herramientas específicas de simulación del ámbito de la telemática como Ns-2 u Opnet, que cuenta con modelos de red 802.11 precisos, pero que no se adaptan bien a las necesidades de los sistemas de control, pues no permiten realizar de forma sencilla operaciones matemáticas como manejo de matrices, cálculo de transformadas o resolución de ecuaciones diferenciales que son fundamentales en el diseño de sistemas de control. Para superar estos inconvenientes es posible considerar como alternativa a la la cosimulación ([1], [2]) con Matlab. Sin embargo, este escenario requiere una elevada curva de aprendizaje para los investigadores del ámbito de control, que precisarían modificar diversos módulos en los simuladores de red para realizar la comunicación con Matlab.

Para facilitar el estudio de sistemas de control a través de redes 802.11, en este trabajo proponemos un marco de simulación basado en el paquete TrueTime[3] de Simulink (Matlab). Este entorno proporciona herramientas matemáticas útiles para el diseño de sistemas de control a la vez que ofrece un simulador de redes 802.11. No obstante, existen algunas deficiencias en la implementación del estándar IEEE 802.11 que realiza TrueTime y que pueden comprometer la fiabilidad de los resultados. Con el objetivo de aumentar la precisión de paquete Truetime, se han llevado a cabo una serie de modificaciones sobre el modelo original que permiten mejorar la precisión de los resultados y ofrecen algunas mejoras como la estimación de consumo energético en las estaciones y el modelado de pérdidas debidas a ruido en el canal.

Esta nueva implementación permite caracterizar con fiabilidad tres parámetros del rendimiento en la red: retardo, pérdidas y consumo de energía, que influyen directamente en el rendimiento de los sistemas de control. Además, se han añadido tres nuevas funcionalidades que aumenta el campo de aplicación del simulador.

- Selección del tamaño de la cola MAC en las estaciones.
- Control en tiempo de ejecución de parámetros de configuración de la capa MAC, en concreto, el tamaño de ventana de contienda y el número de reintentos.
- Mecanismo de estimación de la probabilidad de retransmisión de paquete y la probabilidad de que el canal esté libre, que identifica el nivel de congestión de la red. Estas probabilidades son muy útiles a la hora de diseñar mecanismos de adaptación de red.

Estas características implementadas hacen de la nueva versión de Truetime, que denominaremos Truetime Extendido, una herramienta útil para diseñar mecanismos de adaptación que permitan a la red a ajustarse a las necesidades del sistema

ISBN: 978-84-694-5948-5

de control. Para validar el nuevo entorno de simulación se han realizado una serie de simulaciones con Opnet y se han comparado los resultados obtenido por ambos simuladores.

El resto de este artículo se organiza de la siguiente manera: en la sección II se ofrece una visión general del entorno de simulación propuesto y en la sección III se describen una serie de ampliaciones realizadas sobre la implementación original de Truetime. En la sección IV se muestra una comparativa de simulación de nuestro entorno frente al simulador Opnet para confirmar la precisión de nuestra propuesta. En la sección V se muestra la utilidad del marco de simulación a través de un ejemplo de diseño de WNCS. En la sección VI se resumen las conclusiones del trabajo y se proponen algunas líneas de investigación futuras.

II. Descripción del entorno de simulación: Simulink y Truetime

Truetime es un simulador de redes 802.11 orientado a eventos, escrito en C++ y desarrollado bajo el entorno Simulink de Matlab. Es gratuito, abierto y está en continuo desarrollo. Su objetivo es ayudar al diseño de sistemas de control a través de red, incorporando los efectos que ésta provoca en el comportamiento de los controladores.

Los sistemas de control en tiempo real son muy sensibles a las pérdidas y al retardo, por lo que estimar estos valores de forma precisa es fundamental para diseñar sistemas de control robustos. Otra cuestión clave es el consumo de energía, ya que los dispositivos de red normalmente están alimentados por baterías y tienen un tiempo de vida limitado. La monitorización del consumo ayuda a estimar con precisión el tiempo de vida de de las estaciones y facilita la logística de reposición de baterías. Truetime considera estos tres aspectos claves y proporciona una forma sencilla de incorporar sus efectos a los sistemas de control[1].

A. Bloques de Truetime

Siguiendo la filosofía modular de Simulink, Truetime modela cada elemento de red como un bloque independiente. De los siete bloques que define nos centraremos en los tres utilizados en este estudio: *Truetime Wireless Network*, *TrueTime Send* y *Truetime Receive*.

1) Bloque Truetime Send: El bloque *TrueTime Send* representa una fuente de tráfico simple con una cola de salida. Muestrea la señal que recibe por el puerto de entrada data cada vez que la señal del puerto de entrada *trigger* cambia de valor, generando un paquete que incluye la información de la señal muestreada y poniéndolo en una cola de espera. Por lo tanto este bloque no representa una estación 802.11 sino sólo una fuente de tráfico, toda la lógica del estándar se concentra en el bloque *Truetime Wireless Network* que se verá más adelante. La interfaz de configuración de este bloque permite configurar detalles como el tamaño y el destino (bloque *Truetime Receive*) del paquete de datos.

[1]Los tres parámetros descritos: pérdidas de paquetes, retardo y limitación de consumo son comunes a cualquier escenario con estaciones alimentadas por batería que intercambien tráfico de tiempo real, ya sea para control de procesos, comunicaciones de VoIP o servicios de streaming.

2) Bloque Truetime Receive: El bloque *Truetime receive* representa un sumidero de tráfico. Cada vez que recibe un paquete genera un salida con la información útil (señal *data* de entrada del bloque *Truetime Send*) y la marca de tiempo del paquete. Al igual que *Truetime Send*, este bloque no representa una estación 802.11 y no implementa la monitorización del medio.

3) Truetime Wireless Network: Es el bloque que implementa el funcionamiento del estándar 802.11. Modela todas las características básicas: monitorización de canal, algoritmo de backoff, tiempos de espera entre tramas, colisiones, ruido externo, etc. Es el encargado de recoger los paquetes de las colas de bloques *Truetime Send* y de entregarlos en los bloques *Truetime Receive* que corresponda.

Permite modelar el comportamiento del canal radio estableciendo la probabilidad de error de paquete (probabilidad de que un paquete quede corrupto debido al ruido externo) y la función de pérdidas de propagación. También soporta movilidad en los terminales a través de los puertos de entrada x e y que indican la posición de las estaciones en cada momento.

B. Ejemplo

A continuación se ilustrará con un ejemplo sencillo el papel de cada bloque en el escenario de simulación. Supongamos un escenario donde se desea controlar un proceso industrial de forma remota a través de una red 802.11 de sensores. El sistema de control está formado por una planta, que representa el comportamiento dinámico del sistema a controlar, un controlador y dos sensores. La Fig. 1 muestra el aspecto en Simulink del escenario descrito. El sensor 1 lee periódicamente la salida de la planta y envía los datos al sensor 2 que es el encargado de cerrar el lazo de control. En este ejemplo, el sensor 1 no va a recibir datos a través de la red por lo que conectamos la salida de su bloque receptor *receive* 1 a un sumidero. El sensor 2 no necesita enviar datos a través de la red por lo que conectamos la entrada su bloque emisor *send* 2 una entrada nula. La comunicación entre sensores se realiza a través del bloque simulador de red. Este ejemplo

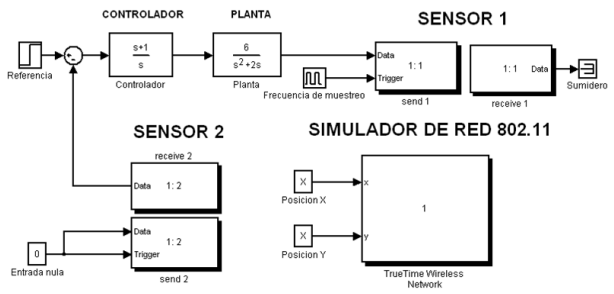


Fig. 1. Escenario de ejemplo que muestra el uso de Truetime dentro de Simulink.

tiene un objetivo meramente didáctico, sin embargo sirve como base para escenarios más complejos. Podemos pensar en escenarios donde la comunicación entre el controlador y la planta también se realice a través de la red o situaciones donde existan varias plantas y por lo tanto se requiera más sensores.

C. Limitaciones de TrueTime

En este apartado detallaremos las limitaciones y fallos de implementación encontrados en la versión original de Truetime. Nos limitaremos a solucionar los fallos que provoquen imprecisiones en los resultados que afectan al diseño de WCNS, es decir: pérdida de paquetes, retardo y consumo de energía. Hay que recalcar que este trabajo pretende conseguir una herramienta de simulación con una precisión similar a la ofrecida por simuladores como Opnet, Ns-2 u Omnet++, y que además permita la integración en Matlab de forma cómoda. A continuación enumeramos las limitaciones y problemas de implementación de 802.11 más relevantes:

- No se implementa el modo infraestructura: Truetime sólo implementa el modo ad-hoc y no contempla el uso de puntos de acceso.
- No se modela el envío de tramas gestión: El modelado de este tipo de tramas no tiene influencia sobre el retardo ni las pérdidas que sólo conciernen a las tramas de datos. En lo que se refiere al consumo de energía, su impacto puede considerarse despreciable.
- No se modela el envío de tramas de control: Esto incluye tramas RTS/CTS, y asentimientos. La ausencia de tramas RTS/CTS no es ningún problema para este estudio ya que asumimos el uso de modo básico de funcionamiento DCF. Esta elección se basa en el tamaño de los paquetes de las aplicaciones de control (menores a 100 bytes). Se ha demostrado que a medida que disminuye el tamaño de los paquetes, el mecanismo RTS/CTS se vuelve más ineficiente. En cuanto a las tramas de asentimiento, su ausencia sí supone un problema ya que afectan a todos los factores de interés. Por ese motivo se resolverá esta inconsistencia de modelado y se incluirá en la nueva implementación de Truetime.
- El modelo de consumo es muy simple: sólo considera el consumo asociado a la transmisión de paquetes. Para solucionar este inconveniente se ha desarrollado un modelo más completo que considera tres estados de consumo: transmisión, recepción y reposo. Las potencias de consumo de cada estado has sido obtenidos de [4].
- El modelado de pérdidas por ruido es incorrecto: en la implementación original de Truetime los paquetes marcados como erróneos son transmitidos con potencia nula para que siempre sean descartados en el receptor a causa de la SNR. Sin embargo, este comportamiento permite que las estaciones perciban el canal libre a pesar de que este esté ocupado con una transmisión de potencia nula. Este error ha sido corregido con una pequeña modificación del código. Además ha sido puesto en conocimiento de los autores, que incluirán la solución propuesta en la próxima versión de Truetime.

Todas estas modificaciones deben realizarse dentro del bloque *Truetime Wireless network*.

III. Ampliación de Truetime original: Truetime extendido

Además de las mejoras de implementación mencionadas en el apartado anterior, hemos realizado dos modificaciones adicionales que aumentar aún más la utilidad de del simulador.

A. Selección del tamaño de cola

Primero, hemos modificado el bloque *TrueTime Send* para establecer el tamaño de su cola como un parámetro de configuración a través su interfaz gráfica ya que, originalmente, cuenta con una cola de tamaño infinito. El tamaño de la cola influye directamente sobre las pérdidas y el retardo por lo que parece razonable poder ajustarlo en función de las características del tráfico. Para tráfico CBR como el generado por aplicaciones de control, está demostrado que las colas provocan un aumento del retardo y no disminuyen las pérdidas de paquetes [5], [6].

B. Ajuste dinámico de parámetros de configuración MAC

En el modelo original de *Truetime Wireless Network* los parámetros de configuración de la capa MAC se establecen estáticamente y no pueden ser modificados a lo largo de la simulación. A pesar de que este es el comportamiento por defecto que describe el estándar, supone una limitación a la hora de responder ante situaciones de congestión. Existen muchos trabajos que proponen mecanismos de adaptación distintos a los definidos en el estándar y consiguen mejorar los resultados de rendimiento de la red. Por tanto resulta interesante que los parámetros de configuración puedan establecerse dinámicamente a lo largo de la simulación. Para ello se han añadido dos nuevos puertos de entrada al bloque *Truetime Wireless Network* que permiten recibir la configuración de red de forma dinámica. Por un lado, el puerto CW recibe el valor de la ventana de contienda y por otro lado, el puerto r recibe el número de reintentos. Por último añadiremos el puerto de entrada P_e, que representa la probabilidad de error de paquete por ruido con el objetivo de variar esta probabilidad durante la simulación de forma más cómoda.

C. Estimación de probabilidades

La estimación de la probabilidad de retransmisión de paquete y la probabilidad de canal libre son utilizadas en la mayoría de mecanismos de adaptación para medir el nivel de congestión de la red [7], [8], [9], [10], [11], por lo que resultan muy útil disponer de ellas. Para hacerlas accesibles se han añadido dos puertos de salida al bloque *Truetime Wireless network*; P_{eq} *estimation* que representa la probabilidad de retransmisión y *idle* que representa la probabilidad de canal libre.

La probabilidad de que un paquete tenga que ser retransmitido P_{eq} se obtiene a partir de la proporción entre paquete retransmitidos y paquete recibidos. Si denominados R al número de paquetes recibidos por una estación durante un cierto periodo de observación y S al número total de paquete recibidos en ese mismo periodo. Entonces podemos estimar la probabilidad de retransmisión como:

$$P_{eq} = \frac{R}{S+R} \tag{1}$$

De acuerdo con [12], la probabilidad de que el canal esté libre puede ser estimada haciendo uso de la relación:

$$P_{idle} = \frac{n_{idle}}{n_{idle}+1} \tag{2}$$

Done n_{idle} representa el número de slots consecutivos que una estación a detectado libre el canal.

Para mejorar la estimación de P_{eq} y P_{idle} aplicamos un filtro de media móvil con ventana de tamaño M sobre últimas medidas realizadas, de manera que la estimación final de las probabilidades se calcula según la fórmula:

$$\widehat{p}_n = \frac{1}{M} \sum_{i=n}^{n+M-1} \widetilde{p}_i \quad (3)$$

Donde $\widehat{p}_n$ representa la n-ésima estimación de la probabilidad bajo estudio y $\widetilde{p}_i$ representa su i-ésima medida.

A partir de estas dos probabilidades podemos estimar la probabilidad de error del canal P_e siguiendo un procedimiento similar al realizado en [13], [14]. Como dijimos anteriormente, contar con una estimación de la probabilidad de error del canal P_e es muy útil para diseñar algoritmos de adaptación.

Las mejoras de implementación y las ampliaciones mostradas en las secciones anteriores, se encuentran disponibles en [15]. Como veremos en la próxima sección, true-time extendido ofrece un entorno de simulación que proporciona un nivel de precisión similar al ofrecido por el simulador Opnet.

IV. Validación: Truetime extendido vs Opnet

Para la validación de nuestra herramienta de simulación hemos llevado a cabo múltiples simulaciones tanto con nuestro simulador como con el simulador comercial Opnet Modeler 14.5, que será utilizado como referencia. Además incluimos los resultados del Truetime original[2] para destacar el aumento de precisión conseguido. El escenario de simulación utilizado se compone de N sensores distribuidos de forma circular que generan tráfico de forma periódica con una frecuencia f_s. La comunicación entre los sensores se realiza por pares, de forma que todos envían y reciben el mismo volumen de tráfico. Se establece una la potencia de transmisión y sensibilidad de recepción que evitan el problema de los nodos ocultos. Los parámetros de simulación utilizados se enumeran en la Tabla I. Los parámetros de capa MAC se han elegido conforme a lo establecido por el estándar 802.11b. En primer lugar analizamos el comportamiento de la red en función del número de sensores conectados. Como muestra la Fig. 2 los resultados obtenidos con ambos simuladores se encuentran muy próximos para todas las métricas estudiadas.

Los resultados de pérdidas y retardo de Truetime extendido son ligeramente superiores a los de Opnet, lo que se traduce en un mayor consumo de energía. En definitiva, la congestión de red es algo superior en Truetime extendido. Este comportamiento puede estar motivado por las diferentes estrategias de encolado que utilizan ambos simuladores. En Opnet, un paquete no se saca de la cola hasta que se recibe el asentimiento. Por el contrario, en Truetime un paquete se saca de la cola en el momento que empieza a ser transmitido. Esta diferencia provoca que Truetime encole paquetes que Opnet descarta, dando lugar a una carga de tráfico algo superior. Las discrepancias producidas por este efecto son más apreciables en los resultados de retardo. A pesar de que existen diferencias, los resultados de ambos simuladores se pueden considerar cualitativamente similares, confirmando la utilidad de nuestra alternativa de simulación.

[2]Se ha utilizado la versión 2.0 (beta 6) de Truetime, pero incluyendo la característica de selección de tamaño de cola para poder comparar los modelos en igualdad de condiciones.

Tabla I
Parámetros de simulación

Parámetro	Valor
Tiempo de simulación en Opnet	50 s
Tiempo de simulación en Truetime	50 s
Tamaño de cola	1 paquete
Régimen binario (R_b)	1 Mb/s
Tamaño de paquete (PL)	80 bytes
Cabecera MAC	28 bytes
Duración PLCP	192 μs
EIFS	364 μs
Tamaño mínimo de ventana (CW_{min})	32
Tamaño máximo de ventana (CW_{max})	1024
Límite de reintentos (r)	5
Potencia de consumo en transmisión (ρ_{tx})	2.5 W
Potencia de consumo en recepción (ρ_{rx})	0.9 W
Potencia de consumo en reposo (ρ_{idle})	0.11 W

Los resultados de Truetime original confirman los defectos de implementación encontrado en la evaluación del código. Por un lado, la ausencia de tramas de asentimiento disminuye el tiempo de espera de las estaciones, que se traduce en menores pérdidas y retardo. Por otro lado, la consideración de consumo de energía exclusivamente en estado de transmisión subestima el consumo real.

También estudiamos el comportamiento de la red para distintos valores de probabilidad de error del canal P_e. En la Fig. 3 se muestra una comparativa entre los simuladores en términos de pérdidas, retardo y consumo de potencia por estación. Los niveles de precisión son especialmente considerables en los resultados de pérdidas y retardo, donde las gráficas son prácticamente indistinguibles. Por último estudiamos el comportamiento de la red antes distintas frecuencias de muestreo f_s. La Fig 4 muestra cómo crecen las discrepancias en los resultados de retardo a medida que aumenta la frecuencia de muestreo. Cuando el sistema se acerca a la condición de saturación, el retardo tiende asintóticamente a un retardo máximo de 22 ms., en el caso de Opnet y de 26 ms. en el de Truetime extendido. En el proceso de diseño de nuestro sistema de control esta diferencia no tiene ninguna relevancia, más aún si tenemos en cuenta que vamos a trabajar lejos de la zona de saturación, donde la precisión de nuestro simulador es mayor.

V. Caso de uso

En esta sección mostramos un ejemplo de uso del entorno de simulación. Vamos a estudiar el comportamiento de un sistema de control con función de transferencia:

$$C(s) = \frac{s+5}{s}$$

que se aplica sobre una planta modelada como un sistema de segundo orden con función de transferencia:

$$P(s) = \frac{25}{s^2+5s+25}$$

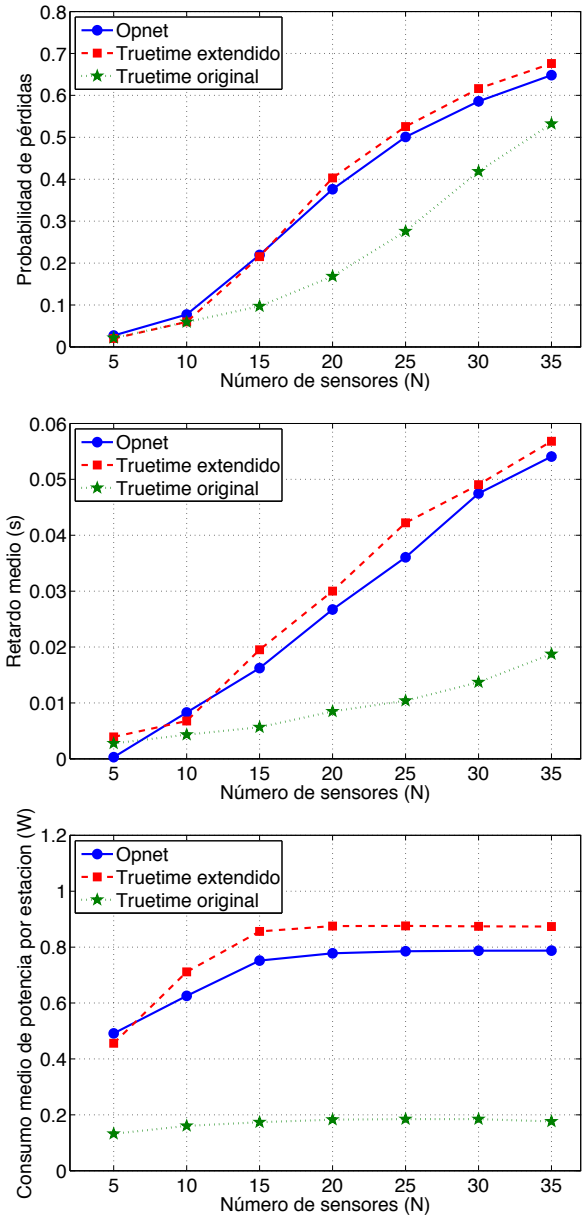


Fig. 2. Resultados de pérdidas de paquetes, retardo y consumo de potencia por estación para escenarios con distinto número de sensores. La frecuencia de muestreo de los sensores se establece a f_s=25 paquetes/s y la probabilidad de error de canal a P_e=0.5.

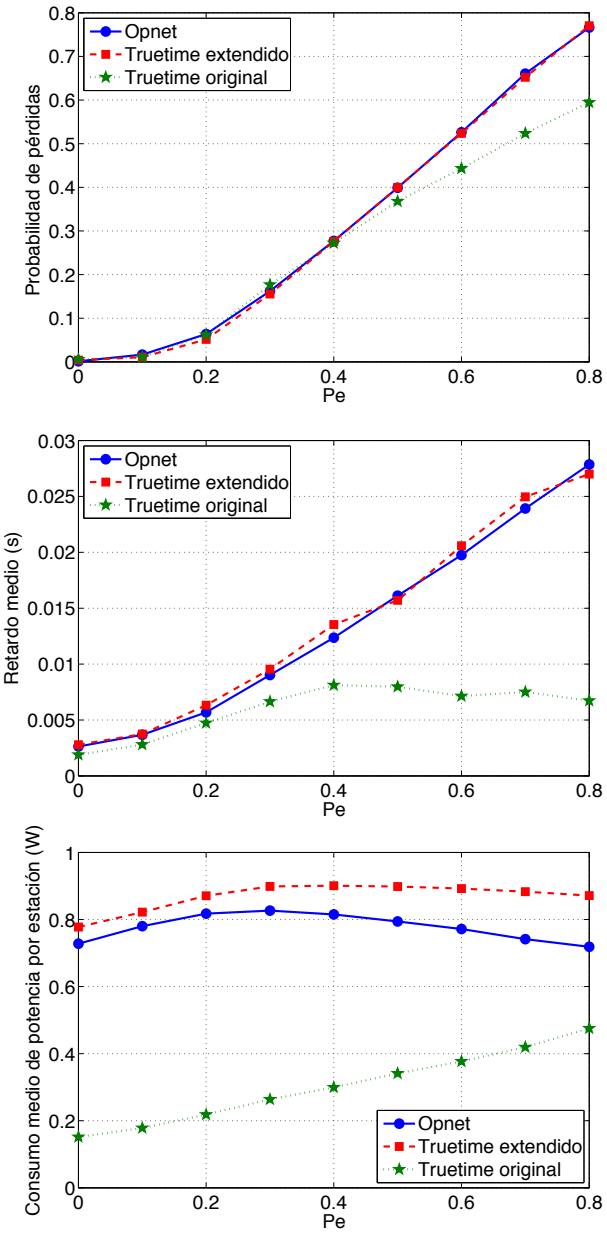


Fig. 3. Resultados de pérdidas de paquetes, retardo y consumo de potencia por estación para escenarios con distinta probabilidad de error del canal P_e. El número de sensores es N=10 y a frecuencia de muestreo de los sensores se establece a f_s=50 paquetes/s.

La red está compuesta por 10 sensores que envían información periódicamente con una frecuencia f_s=50 paquetes/s. La probabilidad de error de canal vale P_e=0.2 hasta t=5 segundos y a partir, pasa a P_e=0.6 hasta el fin de la simulación. La Fig. 5 muestra la estimación de P_e obtenida a partir de la estimación de las probabilidades de retransmisión P_{eq} y canal libre P_{idle}. Como puede observarse, el seguimiento de la referencia es correcto, aunque también lento. Se puede mejorar la velocidad de seguimiento ajustando adecuadamente el filtro de media móvil o aplicando otras técnicas de estimación más sofisticadas [14]. En la Fig. 6 se muestra la salida del sistema a lo largo del tiempo. Se observan las pérdidas de información así como el retardo de la respuesta. En este caso, el sistema mantiene la estabilidad a pesar de la congestión de la red pero la calidad de control se ve afectada. En este ejemplo no se ha aplicado ningún mecanismo de configuración MAC dinámico pero su uso sería inmediato.

VI. Conclusiones y trabajo futuro

En este trabajo hemos desarrollado un entorno de simulación basado en Matlab y el simulador Truetime útil para el diseño de sistemas WCNS. Este entorno proporciona un marco de trabajo único que facilita la integración de las disciplinas de control y telemática. El uso de una herramienta común permite establecer más fácilmente las relaciones entre calidad del control y rendimiento de red, facilitando así al diseño de sistemas de control robustos. También facilita el diseño de sistemas de control que consideren el ajuste de parámetros de red (tamaño de ventana de contención, límite de reintentos o régimen binario) en sus estrategias. El simulador

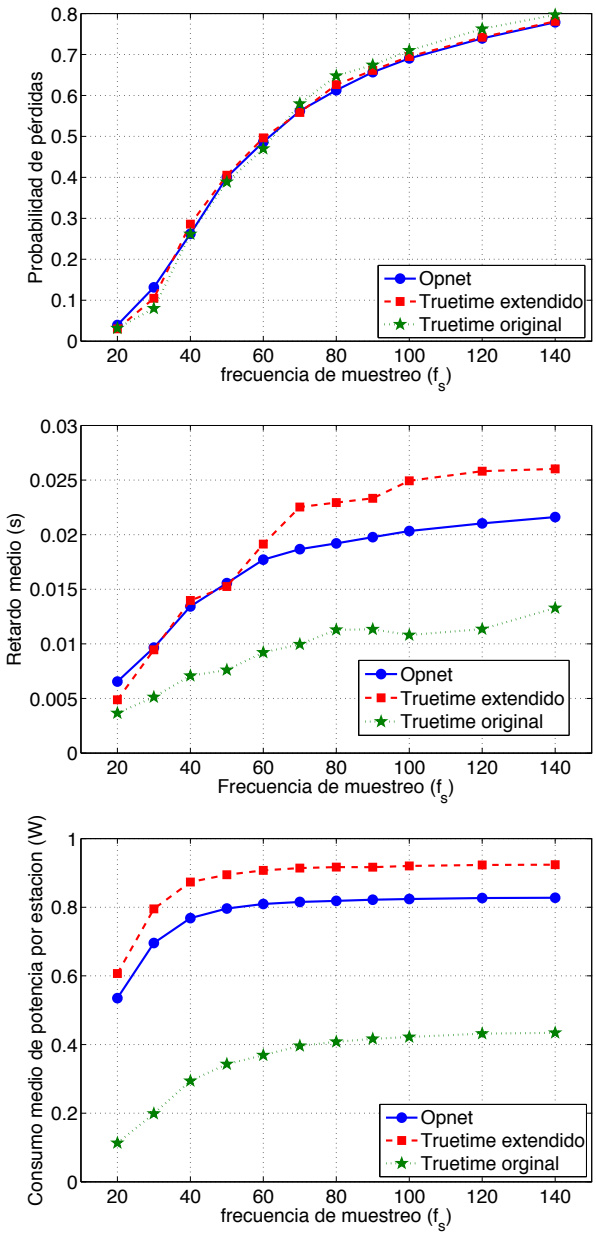


Fig. 4. Resultados de pérdidas de paquetes, retardo y consumo de potencia por estación para escenarios con distinta frecuencia de muestreo f_s. El número de sensores es N=10 y probabilidad de error en el canal se establece a P_e=0.5.

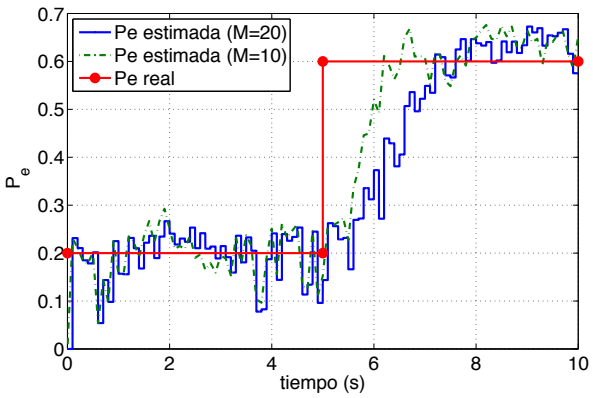


Fig. 5. Estimación de probabilidad de error del canal P_e con distintos tamaños de ventana. El número de sensores es N=10 y la frecuencia de muestreo se fija a f_s=50 paquetes/s.

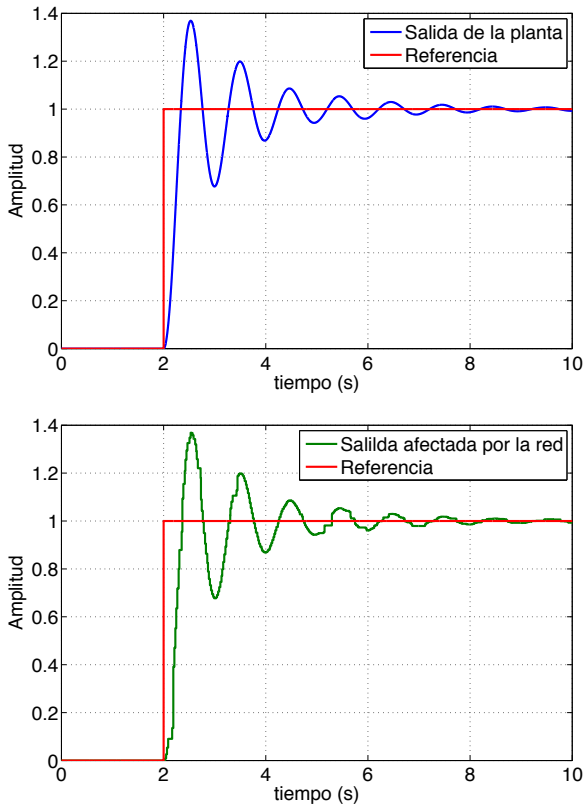


Fig. 6. Respuesta del sistema de control. El número de sensores es N=10 y la frecuencia de muestreo se fija a f_s=50 paquetes/s.

resultante denominado Truetime extendido ha sido validado frente a otro simulador de precisión contrastada da como Opnet, mostrando niveles de fiabilidad muy cercanos a los ofrecidos por éste último.

Como líneas de trabajo futuras, estamos interesados en investigar los beneficios del co-diseño de sistemas de control. Actuando sobre parámetros de red, este tipo de diseño puede asegurar una cierta calidad del control mientras satisface requisitos adicionales relativos a consumo energético, retardo o pérdidas.

AGRADECIMIENTOS

Este trabajo ha sido financiado en parte gracias al proyecto CICYT DPI2010-19154, y al proyecto European Commission (EC) (FeedNetBack Project, grant agreement 223866).

REFERENCIAS

[1] C. Harding, A. Griffiths, and H. Yu, "An interface between matlab and opnet to allow simulation of wncs with manets," in *2007 IEEE International Conference on Networking, Sensing and Control, ICNSC'07*, 2007, pp. 711–716.

[2] M. S. Hasan, H. Yu, A. L. Griffiths, and T. C. Yang, "Co-simulation framework for networked sontrol systems over multi-hop mobile ad-hoc networks," in *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 17, 2008.

[3] A. Cervin, D. Henriksson, B. Lincoln, J. Eker, and K. E. Arzen, "How does control timing affect performance? analysis and simulation of timing using jitterbug and truetime," *Control Systems Magazine, IEEE*, vol. 23, no. 3, pp. 16–30, 2003.

[4] J. Ebert, S. Aier, G. Kofahl, A. Becker, B. Burns, and A. Wolisz, "Measurement and simulation of the energy consumption of an WLAN

interface," *Technical University of Berlin, Telecommunication Networks Group, Tech. Rep. TKN-02-010*, 2002.

[5] J. Jimenez, R. Estepa, F. R. Rubio, and F. A. E. Gomez-Estern, "Networked control system: 802.11 performance analysis," in *Proceeding 9th. Portuguese Conference on Automatic Control, CONTROLO '2010*, September 2010 2010, pp. 83–90.

[6] R. P. Liu, G. Sutton, and I. B. Collings, "A 3-d markov chain queueing model of ieee 802.11 dcf with finite buffer and load," in *IEEE International Conference on Communications*, 2009.

[7] F. Calì, M. Conti, and E. Gregori, "Dynamic tuning of the ieee 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Transactions on Networking*, vol. 8, no. 6, pp. 785–799, 2000.

[8] L. Bononi, M. Conti, and E. Gregori, "Runtime optimization of ieee 802.11 wireless lans performance," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 1, pp. 66–80, 2004.

[9] C. Wang and W. Tang, "A probability-based algorithm to adjust contention window in ieee 802.11 dcf," in *2004 International Conference on Communications, Circuits and System*, vol. 1, 2004, pp. 418–422.

[10] D. J. Deng, C. H. Ke, H. H. Chen, and Y. M. Huang, "Contention window optimization for ieee 802.11 dcf access control," *IEEE Transactions on Wireless Communications*, vol. 7, no. 12, pp. 5129–5135, 2008.

[11] A. L. Toledo, T. Vercauteren, and X. Wang, "Adaptive optimization of ieee 802.11 dcf based on bayesian estimation of the number of competing terminals," *IEEE Transactions on Mobile Computing*, vol. 5, no. 9, pp. 1283–1296, 2006.

[12] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, "Idle sense: An optimal access method for high throughput and fairness in rate diverse wireless lans," in *Computer Communication Review*, vol. 35, 2005, pp. 121–132.

[13] D. Malone, P. Clifford, and D. J. Leith, "Mac layer channel quality measurement in 802.11," *IEEE Communications Letters*, vol. 11, no. 2, pp. 143–145, 2007.

[14] I. Tinnirello and A. Sgora, "A kalman filter approach for distinguishing channel and collision errors in ieee 802.11 networks," in *IEEE Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008.*, 2008, pp. 1–5.

[15] J.Jimenez, 2011. [Online]. Available: http://masai.us.es/research/truetime

Comparación de Algoritmos para el Mapeo de Redes Virtuales

Juan Felipe Botero, Xavier Hesselbach
Departamento de Ingeniería Telemática
Universitat Politècnica de Catalunya
C/Jordi Girona, 1 i 3 Mòdul C3 - Campus Nord 08034 Barcelona.
jfbotero@entel.upc.edu, xavierh@entel.upc.edu

***Resumen*—La virtualización de red es reconocida como una de las principales tecnologías que formará parte del Internet del futuro. Mediante ella, la actual inhabilidad de Internet para desarrollar e implementar nuevos servicios, causada por la falta de coordinación entre los proveedores de servicio, podrá ser superada. Aplicar virtualización a los recursos de red nos lleva al problema de asignación de recursos consistente en asignar (mapear) de manera óptima los recursos de la red física a los nodos y enlaces de las redes virtuales, comúnmente este problema se ha denominado "problema de incrustar o mapear redes virtuales". En este artículo se muestran las principales estrategias que han sido propuestas para resolver de manera óptima este problema y se comparan teniendo en cuenta varias métricas.**

***Palabras Clave*—Virtualización de red, estrategias de mapeo, VNE, optimización, programación lineal**

I. Introducción

La virtualización de recursos de red ha sido identificada como una tecnología clave para la investigación en el Internet del futuro [1] y es usada activamente en los proyectos actuales que prueban futuras arquitecturas de red [2], [3]. Mediante la virtualización de los recursos en nodos y enlaces de una red física (también conocida como red sustrato), múltiples topologías de red con diferentes características pueden ser creadas y hospedadas en el mismo hardware físico. Además, la abstracción introducida mediante la virtualización de recursos permite que los operadores de red puedan administrar y modificar sus redes de una manera flexible y dinámica. Una red virtual (RV), en este sentido, es una combinación de elementos de red activos y pasivos (nodos y enlaces de red) funcionando sobre una red sustrato (RS). Los nodos virtuales están conectados a través de enlaces virtuales, formando una red que puede ser representada por un grafo dirigido, donde los nodos virtuales corresponden a los nodos en el grafo y los enlaces virtuales corresponden a los arcos.

La flexibilidad ganada a través de la virtualización de red puede ser usada para incrementar la sostenibilidad (en términos de costo-rendimiento) del hardware de red. Mediante la asignación dinámica de recursos virtuales en el hardware físico, el beneficio obtenido del hardware existente puede ser maximizado. Sin embargo, esto conduce a un problema de optimización: Para alcanzar una operación efectiva de las redes virtuales es indispensable desarrollar algoritmos que distribuyan los recursos virtuales en la infraestructura física de una manera óptima. Los parámetros para definir si una asignación es o no óptima pueden ser varios, desde la distribución de cargas, pasando por la eficiencia energética hasta la seguridad requerida por las redes virtuales. El problema de mapear un conjunto de redes virtuales en una red sustrato de una manera óptima es conocido como "problema de mapear redes virtuales" (VNE por sus siglas en inglés). Varios algoritmos han sido propuestos para resolver este problema. Comparar estos algoritmos no es fácil, dado que típicamente cada algoritmo optimiza un subconjunto de todos los posibles parámetros de red.

En este artículo, se propone extender la comparación entre diferentes algoritmos a una nueva dimensión. Debería ser posible comparar los algoritmos en términos iguales. Es decir, los algoritmos no deberían ser evaluados teniendo en cuenta solamente su propio objetivo de optimización, sino también considerando objetivos de optimización definidos por otros algoritmos. De este modo, se hace posible evaluar si un algoritmo que provee soluciones óptimas a una métrica será capaz de obtener soluciones aceptables teniendo en cuenta otras métricas (lo cual puede ser muy importante para el operador de red). Además, los algoritmos pueden ser comparados considerando métricas que no han sido consideradas como objetivos de optimización. Esto podría indicar áreas en las que se necesitan realizar modificaciones a los algoritmos existentes o crear nuevos algoritmos.

II. Definición del problema

A. Perspectiva

El problema de mapear redes virtuales se ocupa de la asignación (mapeo) eficiente de un grupo de peticiones de red virtual (PRV) a los nodos y enlaces de la red sustrato. Una PRV es un grupo de nodos virtuales que deben ser mapeados a un grupo de nodos sustrato con suficiente capacidad para cumplir las demandas, y un grupo de enlaces virtuales que deben ser mapeados a un conjunto de **caminos** en la red sustrato. Este mapeo debe optimizar la asignación de los recursos que pertenecen a la red física. Los mapeos pueden ser optimizados considerando el desempeño (e.g. capacidad de CPU, ancho de banda del enlace), la eficiencia energética (e.g. uso de potencia en un nodo), seguridad (e.g. fiabilidad en nodos, codificación en enlaces), u otros parámetros. Por otra parte, algunas restricciones pueden existir: Por ejemplo los mapeos fijos en nodos pueden influenciar la asignación.

En la Fig. 1 se muestran dos redes virtuales mapeadas en una red sustrato. Un nodo en la red sustrato puede alojar varios nodos virtuales. El mapeo de enlaces virtuales es más complejo, un enlace virtual es mapeado en un **camino** en la red sustrato (i.e. varios recursos físicos son combinados para formar un recurso virtual). El mapeo de un enlace virtual en un

ISBN: 978-84-694-5948-5

Tabla I
TERMINOLOGÍA USADA EN ESTE ARTÍCULO

Término	Descripción
$G = (V, A)$	G es un gráfico que representa la red física. Está compuesto de un grupo de vértices (V) y un grupo de arcos A que conectan los vértices.
$G^k = (V^k, A^k)$	G^k es el grafo de la k-ésima petición de red virtual. Como G, está formado por un grupo de vértices (V^k) que están conectados con por un grupo de arcos (A^k).
$f : V^k \rightarrow V$	f es la función que mapea nodos virtuales en nodos de la red sustrato.
$P = \{G' \mid G' \subset G\}$	P es el grupo de todos los caminos dirigidos G' en G
$g : A^k \rightarrow 2^P \setminus \emptyset$	g es la función que mapea enlaces virtuales a un grupo de caminos dirigidos (un elemento del conjunto de todos los caminos 2^P) en la red sustrato.
$HH : A^k \times V \rightarrow \mathbb{R}$	HH es la función que define la demanda de un enlace virtual $(i^k, j^k) \in A^k$ que le corresponde a un "nodo oculto" $l \in V$.

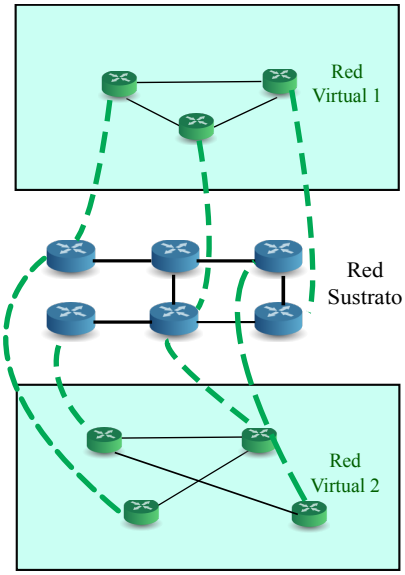


Fig. 1. Dos Redes Virtuales Mapeadas en una red sustrato

camino de la red sustrato usa, obviamente, recursos de los enlaces físicos que conforman el camino. Sin embargo, también hay nodos físicos en el camino que deben ser cruzados por el enlace virtual. Estos nodos son comúnmente llamados nodos ocultos [4]. Cada enlace virtual tiene también una demanda de recursos para todos los nodos ocultos que hacen parte del camino que se utiliza para hacer el mapeo. Además, varios enlaces virtuales pueden usar el mismo enlace sustrato (i.e. un recurso físico puede ser dividido en varios recursos virtuales). Para el mapeo de nodos y enlaces virtuales, varios algoritmos pueden ser considerados. Debido a las diferencias en el mapeo de nodos y enlaces, la mayoría de los algoritmos realizan el mapeo de nodos y enlaces en diferentes fases. Puede ser demostrado que el problema de mapeo de redes virtuales es $\mathcal{NP}$-completo. Por esta razón, los algoritmos prácticos usan heurísticas para obtener un valor cercano al óptimo.

Los algoritmos para resolver el problema se pueden evaluar de dos diferentes maneras: *En línea* y *fuera de línea*. Los algoritmos *fuera de línea* tienen como entrada un grupo de PRVs junto con una red sustrato y obtienen una solución cercana al óptimo para ese grupo de PRVs. Este tipo de enfoque puede obtener buenos resultados con respecto a la optimización, pero no considera procesos de llegada de PRVs dinámicos (i.e. cada PRV llega al sistema en un tiempo diferente y debe ser mapeado en tiempo real). Los algoritmos *en línea*, por otra parte, atienden los PRVs de una manera FIFO, redistribuyendo los recursos virtuales conforme llegan los PRVs. Mientras que el último enfoque encaja mejor con ambientes dinámicos, tiende a generar soluciones menos óptimas. Las cosas pueden complicarse más si se consideran redes sustrato con recursos heterogéneos. La búsqueda de algoritmos de adaptación que puedan considerar este tipo de redes son aún un tema abierto.

B. Descripción formal del problema

Formalmente, el problema de mapeo de redes virtuales puede ser descrito con la terminología que se presenta en la Tabla I. Una red física es definida como un grafo $G = (V, A)$ dónde los vértices representan los nodos en la red y los arcos representan los enlaces entre nodos. Sobre esta red, un conjunto de redes virtuales, cada una descrita por su propio grafo dirigido $G^k = (V^k, A^k)$, son mapeadas mediante la asignación de un nodo físico para cada nodo virtual y un camino físico para cada enlace virtual.

La función que se encarga de describir esta operación es la función de mapeo que es la salida de los algoritmos que resuelven el problema. Esta función puede ser dividida en mapeo de nodo y enlace. La función de mapeo de nodo $f : V^k \rightarrow V$ se encarga de asignar nodos de la PRV en nodos de la red sustrato. De manera similar, la función de mapeo de enlaces es definida como $g : A^k \rightarrow 2^P$, dónde 2^P es el grupo de conjuntos de todos los caminos dirigidos en la red sustrato. Si g se mapea a un grupo que tiene más de un elemento, la función permite considerar algoritmos que resuelven el problema mediante el encaminamiento multicamino. Si no es así, el elemento único del conjunto es el camino usado para mapear el enlace virtual. Ambas funciones no pueden exceder los recursos de los nodos o enlaces en la red sustrato. Un mapeo óptimo es entonces una función que satisfaga todas las restricciones de capacidad y que además satisfaga una función objetivo determinada (e.g. maximizar los recursos sobrantes en la red sustrato).

A continuación, se describirán los principales algoritmos para resolver el problema de mapear redes virtuales. Los algoritmos más importantes de este grupo serán evaluados en este artículo con relación a las métricas más utilizadas.

III. ALGORITMOS PARA MAPEAR REDES VIRTUALES

Antes de entrar a describir en detalle el funcionamiento de los algoritmos, se ha definido una taxonomía para clasificar

sus parámetros más importantes. Esta taxonomía se presenta en la Tabla II, a continuación los principales algoritmos son descritos.

A. Algoritmo Basado en Estrés

Una de las primeras estrategias para resolver el problema de mapeo en redes virtuales fue presentada en [5]. El objetivo de este algoritmo es mantener un *estrés* balanceado tanto en enlaces como en nodos de la RS. El estrés de un nodo sustrato se define como el número de nodos virtuales mapeados sobre él, mientras que el estrés de un enlace sustrato se define como el número de enlaces virtuales que lo contienen. Para lograr el objetivo propuesto, la suma del máximo estrés de nodo y de enlace es minimizada. Se propone un algoritmo heurístico que divide la PRV en un número determinado de sub-RV conectadas entre sí, cada una con una topología en estrella. Después de la división, el mapeo de nodos y enlaces virtuales es realizado para cada sub-RV. El mapeo de nodos es resuelto mediante un algoritmo voraz y luego, se resuelve el mapeo de enlaces usando una solución de camino más corto entre los nodos mapeados.

B. Algoritmo Basado en Multicamino

La maximización de la *ganancia* promedio, es decir, la suma de las demandas de ancho de banda y la CPU de un PRV, es el objetivo del algoritmo propuesto en [7]. Este algoritmo introduce en una cola el grupo de PVRs que llegan al sistema, ordenados de manera decreciente por ganancia, durante una ventana de tiempo predefinida y trata de realizar una asignación eficiente de los PRVs. Si algún PRV no es aceptado, inmediatamente se envía a la cola a la espera de que algunos recursos de la red sustrato sean liberados, si pasado un tiempo de expiración determinado el PRV no ha podido ser mapeado, se descarta.

El mapeo de nodos y de enlaces se realiza de manera independiente. La asignación de nodos es realizada usando un algoritmo voraz que mapea los nodos virtuales con mayor ganancia a los nodos sustrato con mayores recursos disponibles. El mapeo de enlaces es realizado de dos maneras: La primera es mediante multicamino con la que, por medio de programación lineal, se obtiene una solución óptima que evita la complejidad inherente al problema, la segunda manera utiliza caminos más cortos, cumpliendo las restricciones de capacidad entre las demandas de los enlaces virtuales y las capacidades de los enlaces sustrato que hacen parte de los caminos escogidos.

C. Algoritmo Coordinando Mapeo de Nodos y Enlaces

Un algoritmo, evaluado en línea, que propone una nueva solución para el mapeo de nodos es presentado en [8]; la etapa de mapeo de enlace virtual es realizada de igual manera que en [7].Un nuevo grupo de restricciones en los nodos se añaden a los trabajos anteriores en la etapa de mapeo de nodos: Restricciones de ubicación. Estas restricciones introducen una distancia no negativa por cada PRV indicando cuán lejos puede estar un nodo sustrato, para ser considerado como candidato, de la ubicación demandada de un nodo virtual.

La asignación de nodos se realiza definiendo un grafo aumentado sobre la red sustrato; se introduce un grupo de nuevos nodos (llamados metanodos), uno por cada nodo virtual, y conectado a un cluster de nodos sustrato candidatos que cumplen las restricciones de ubicación y de capacidad. El algoritmo resuelve el problema de mapeo, formulándolo como un problema de programación entera mixta (*Mixed Integer Programming*). Su objetivo es minimizar el *costo* de mapear una PRV, es decir, la suma del ancho de banda y la CPU que se asigna en la red sustrato para cumplir las demandas de la PRV. Para evitar la complejidad (los problemas de programación entera son $\mathcal{NP}'$-completos), el problema se relaja para convertirlo en un problema de programación lineal (sus variables enteras se convierten en reales), y la solución se redondea de dos maneras diferentes: determinista o aleatoria.

D. Algoritmo Usando la Detección de un Isomorfismo de Subgrafo

El algoritmo propuesto por Lischka y Karl [9] se basa en el problema de la detección de un isomorfismo de subgrafo (DIS). En teoría de grafos, un isomorfismo entre dos grafos G and H ($G \simeq H$) es una biyección entre los grupos de vértices de G y H, $m : V(G) \rightarrow V(H)$, de tal forma que cualquier par de vértices i y j de G son adyacentes en G si y sólo si $m(i)$ y $m(j)$ son adyacentes en H. El problema $\mathcal{NP}'$-completo de detección de un isomorfismo de subgrafo trata de encontrar un subgrafo G_{sg} de G ($G_{sg} \subset G$) tal que $G_{sg} \simeq H$.

El problema de mapeo de redes virtuales puede ser reducido al DIS. En [9] se propone un algoritmo heurístico que consiste en encontrar un isomorfismo de subgrafo (representando la PRV), que cumpla además todas las demandas, dentro de la red sustrato. La contribución principal de esta propuesta es la realización en la misma etapa del mapeo de nodos y de enlaces. Esta característica reduce considerablemente el tiempo de ejecución del algoritmo.

E. Algoritmo Considerando la Topología y Reoptimizando las Asignaciones

Dos contribuciones importantes para mejorar la tasa de aceptación de PVRs son propuesta en [10]:

Mapeo considerando la topología: Existen nodos y enlaces sustrato que son críticos y que contribuyen a la *partición* de la red cuando un mapeo es realizado; un enlace o nodo es susceptible de *partición* cuando su eliminación divide la red sustrato en dos redes diferentes. Estos recursos deben ser considerados en la función objetivo de los algoritmos existentes, el parámetro para medir cuán crítico es un recurso se llama índice de criticidad (IC).

Reoptimización de mapeos con cuellos de botella: Como las PRVs son mapeadas con un tiempo de vida asociado, los valores arbitrarios del mapeo pueden causar que el mapeo en la red sustrato se vaya tornando ineficiente. Los esquemas de reconfiguración periódica no funcionan muy bien para situaciones reales (debido a la sobrecarga incurrida en la reubicación de nodos y enlaces virtuales). La solución propuesta es actuar cada vez que una PRV se rechaza. Se identifican los enlaces virtuales que están causando el rechazo de la PRV y se procede a reubicarlos en regiones que no son críticas en la red sustrato.

Tabla II
TAXONOMÍA DE LOS ALGORITMOS (EXTENDIDA DE [6])

Parámetro de Clasificación		Descripción
Tipo de Evaluación	En línea	Las PVRs llegan al sistema de manera dinámica y no son conocidas con anterioridad.
	Fuera de línea	Las PVRs son conocidas con anterioridad de manera que el orden en que son atendidas puede ser organizado previo a su mapeo
Tipo de Algoritmo	Dinámico	Después de que varios PVRs han sido mapeados, el algoritmo propone mecanismos para reoptimizar los mapeos que ya han sido realizados
	Estático	No se realiza reoptimización
Mapeo de Enlace Virtual	Multicamino (MC)	Múltiples caminos de la RS som considerados para realizar el mapeo de un enlace virtual link
	Único camino (UC)	Cada enlace virtual es mapeado solamente a un camino de la RS
Nodos Ocultos	Considerados	Las demandas de nodo oculto son consideradas cuando se realiza el mapeo
	No Considerados	No se consideran las demandas de nodo oculto

F. Algoritmo Considerando los Nodos Ocultos

El concepto de nodos ocultos, que tiene una aplicación realista a el mapeo de recursos en redes virtuales fue introducido en [4]. Este trabajo consideró la evaluación fuera de línea del problema. El mapeo de nodos virtuales no es considerado, mientras que el mapeo de enlaces virtuales es realizado mediante el uso de k-caminos más cortos para cada enlace virtual, incorporando las demandas de los nodos ocultos cuando cada enlace virtual es mapeado.

Este trabajo propone realizar el mapeo con demandas de CPU en nodos y de ancho de banda en enlaces, además, se consideran PRV que no tengan demandas explícitas de recursos. El algoritmo propuesto mapea los PRVs con demandas explícitas en primer lugar tratando de maximizar los recursos sobrantes en la red sustrato (equivalente a minimizar el costo). Luego, las PRVs sin demandas explícitas son mapeados distribuyendo los recursos sobrantes de manera justa entre ellas.

G. Resumen

Para resumir esta sección, se ha construido la Tabla III que muestra las características de los algoritmos previamente presentados. Es fácil observar que, básicamente, CPU en nodos y ancho de banda en enlaces son los parámetros considerados por casi todos los algoritmos.

IV. COMPARACIÓN Y EVALUACIÓN DE ALGORITMOS

Esta sección está dedicada a evaluar y comparar el comportamiento de los principales algoritmos para mapear redes virtuales en redes sustrato. Se presentará en primer lugar la metodología utilizada para la creación de escenarios de evaluación seguido por los resultados obtenidos.

A. Metodología para la Creación de los Escenarios

Para la comparación entre los algoritmos, se usa una metodología para crear escenarios con una carga ρ y parámetros determinados.

1) Creación de Topologías: Para la creación de topologías tanto sustrato como virtuales, se usa el algoritmo de Waxman [11]. Las coordenadas de los nodos son distribuidas uniformemente en un área.

El generador de Waxman recibe los parámetros α y β para determinar la probabilidad de que haya un arco entre los nodos u y v usando la siguiente ecuación.

$$P(u,v) = \alpha \cdot e^{\frac{-d(u,v)}{\beta \cdot L}}, \tag{1}$$

donde $0 < \alpha,\ \beta \leq 1$, d es la distancia euclidiana entre los nodos u y v y L es la máxima distancia euclidiana entre cualquier par de nodos. Si el parámetro α se incrementa, la probabilidad de que haya arcos entre cualquier par de nodos crece, mientras que un aumento de β conduce a una mayor proporción de arcos largos que de arcos cortos. Para este estudio se realizó esta generación de topologías para la red sustrato y para un número k^{max} de redes virtuales.

2) Generación de Recursos y Demandas: Después de que las topologías sustrato y virtuales han sido creadas, se realiza la generación de recursos y demandas en dos pasos.

Primero, se generan los recursos para la red sustrato distribuyendo uniformemente los recursos de CPU nodo X en un intervalo $(0, \mathrm{NR}_{\mathrm{X}}^{\max}]$ en cada nodo sustrato y de la misma manera para cada recurso de ancho de banda de enlace Y e un intervalo $(0, \mathrm{LR}_{\mathrm{Y}}^{\max}]$ para cada enlace sustrato. Esta es la manera usual en las propuestas existentes de distribuir las capacidades de recursos.

En segundo lugar, se generan las demandas en todos las PRVs y se trata de lograr un cierto porcentaje de carga de cada recurso. La creación de demandas de nodo y de enlace que cumplen los requerimientos de carga conllevan algunos retos.

Como el número de nodos es fijo en las generaciones de topología de Waxman, se puede fácilmente calcular el recurso medio en un nodo sustrato.

$$\mathrm{E}[\mathrm{NR_X}] = \frac{0 + \mathrm{NR}_{\mathrm{X}}^{max}}{2} = \frac{\mathrm{NR}_{\mathrm{X}}^{max}}{2} \tag{2}$$

de la misma manera se puede calcular la demanda media para un nodo virtual dada una carga ρ

$$\mathrm{E}[\mathrm{ND_X}] = \rho \cdot \mathrm{E}[\mathrm{NR_X}] \cdot \frac{|V|}{|V^k| \cdot k^{max}} \tag{3}$$

esto resulta en una máxima demanda de recursos de

$$\mathrm{ND}_{\mathrm{X}}^{\max} = 2 \cdot \mathrm{E}[\mathrm{ND_X}] \tag{4}$$

debido a la distribución uniforme de demandas entre $(0, \mathrm{ND}_{\mathrm{X}}^{\max}]$.

Como la topología Waxman es probabilística teniendo en cuenta la creación de enlaces, el número de enlaces en una red Waxman no es fijo y sólo puede estar dado por probabilidad.

Tabla III
CARATERÍSTICAS DE LOS PRINCIPALES ALGORITMOS PARA RESOVER EL MAPEO DE REDES VIRTUALES (EXTENDIDA DE [6])

Referencia	Objetivo de Optimización					Restricciones de Nodo y Enlace		Taxonomía				Heurística de Enlace	Heurística de Nodo
	Estrés Balanceado	Ganancia	Costo	Mapeo Posible	Costo + IC	Sin Restricciones	CPU + Anncho de Banda	Tipo de Evaluación	Tipo de Algoritmo	Mapeo de Enlace Virtual	Nodos Ocultos		
[5]	**X**					**X**		FDL	EST/DIN	UC	NC	CMC	EV
[4]			**X**				**X**		EST	UC	C	CMC	NC
[7]		**X**					**X**	EL	EST	MC	NC	PFM	VRD
[8]			**X**				**X**		EST	MC	NC	PFM	R-PME
[9]				**X**			**X**		EST	UC	NC	DIS	DIS
[10]					**X**		**X**		DIN	MC	NC	PFM	R-PME

EST:Estático UC: Único Camino NC: No Considerado CMC: Camino más Corto EV: Estrés Voraz
DIN:Dinámico MC: Multicamino EL: En Línea C: Considerado FDL: Fuera de Línea DIS: Detección de Isomorfismo de Subgrafo
VRD: Voraz Recursos Disponibles R-PME: Programación Mixta Entera-Redondeada PFM: Problema de Flujos por Multi-Commodities

Para lograr la carga ρ requerida para los recursos del enlace, se considera el número promedio de enlaces en una red Waxman mediante la estimación de la media $E[p]$ de crear un enlace entre cualquier par de nodos. Por lo tanto, se dice que el número medio de enlaces $\mathrm{E}[|A|]$ en un grafo dirigido es:

$$\mathrm{E}[|A|] = \mathrm{E}[p] \cdot |V| \cdot (|V| - 1). \tag{5}$$

De esta manera podemos calcular el recurso medio en la red sustrato como

$$\mathrm{E}[\mathrm{LR_Y}] = \frac{0 + \mathrm{LR_Y^{max}}}{2} = \frac{\mathrm{LR_Y^{max}}}{2} \tag{6}$$

y

$$\mathrm{E}[\mathrm{LD_Y}] = \rho \cdot \mathrm{E}[\mathrm{LR_Y}] \cdot \frac{\mathrm{E}[|A|]}{\mathrm{E}[|A^k|] \cdot k^{max}} \tag{7}$$

que al final redunda en:

$$\mathrm{LD_Y^{max}} = 2 \cdot \mathrm{E}[\mathrm{LD_Y}]. \tag{8}$$

Sin embargo, nos tenemos que asegurar que $\mathrm{LD_Y^{max}} \leq \mathrm{LR_Y^{max}}$ se cumpla. Especialmente, se necesita asegurar que $\mathrm{E}[|A|] < \mathrm{E}[|A^k|] \cdot k^{max}$ in En la ecuación (7) se cumpla para $0 \leq \rho \leq 1$.

Con esta finalidad, la siguiente restricción debe cumplirse siempre:

$$|V|^2 < k^{max} \cdot |V^k|^2. \tag{9}$$

Si esta restricción es violada, la anterior aproximación tendrá una carga mayor que la carga ρ deseada para los recursos de enlaces y podría resultar incluso en $\mathrm{LD_Y^{max}} > \mathrm{LR_Y^{max}}$, lo cual nunca podrá ser asignado.

B. Evaluación

1) Escenarios: Como se describe en la sección IV-A, se ha usado el algoritmo de Waxman para la generación de topologías. Para realizar la evaluación, hemos usado $\alpha = \beta = 0.5$ y distribuido las coordenadas de los nodos uniformemente en un área cuadrada de 1×1. Estudios empíricos han obtenido para los parámetros anteriores, una distancia promedio entre cualquier par de nodos de $\mathrm{E}[d] \approx 0.5$ y una distancia máxima de $L = \sqrt{2}$. Por lo tanto, teniendo en cuenta la ecuación (1), la probabilidad media de crear un enlace entre 2 nodos es de $\mathrm{E}[p] \approx 1/4$.

En este trabajo, se ha considerado CPU como recurso en nodos, denotado por $\mathrm{NR_{CPU}}$, y ancho de banda como recurso de enlace, denotado por $\mathrm{LR_{BW}}$ en la red sustrato. Para la distribución uniforme de estos valores se han escogido los siguientes máximos $\mathrm{NR_{CPU}^{max}} = 100$ y $\mathrm{LR_{BW}^{max}} = 100$ (siguiendo los valores de trabajos anteriores).

El número de nodos sustrato elegido fue $|V| = 50$ y el número de nodos virtuales por red fue $|V^k| = 20$ para balancear el tiempo de ejecución de algunos algoritmos y la realidad de los escenarios. Para explorar el impacto de la consolidación de redes virtuales, se consideraron diferentes números de redes virtuales $k^{\max} \in \{10, 15, 20\}$.

Se consideraron también escenarios para cargas diferentes $\rho \in \{0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$.

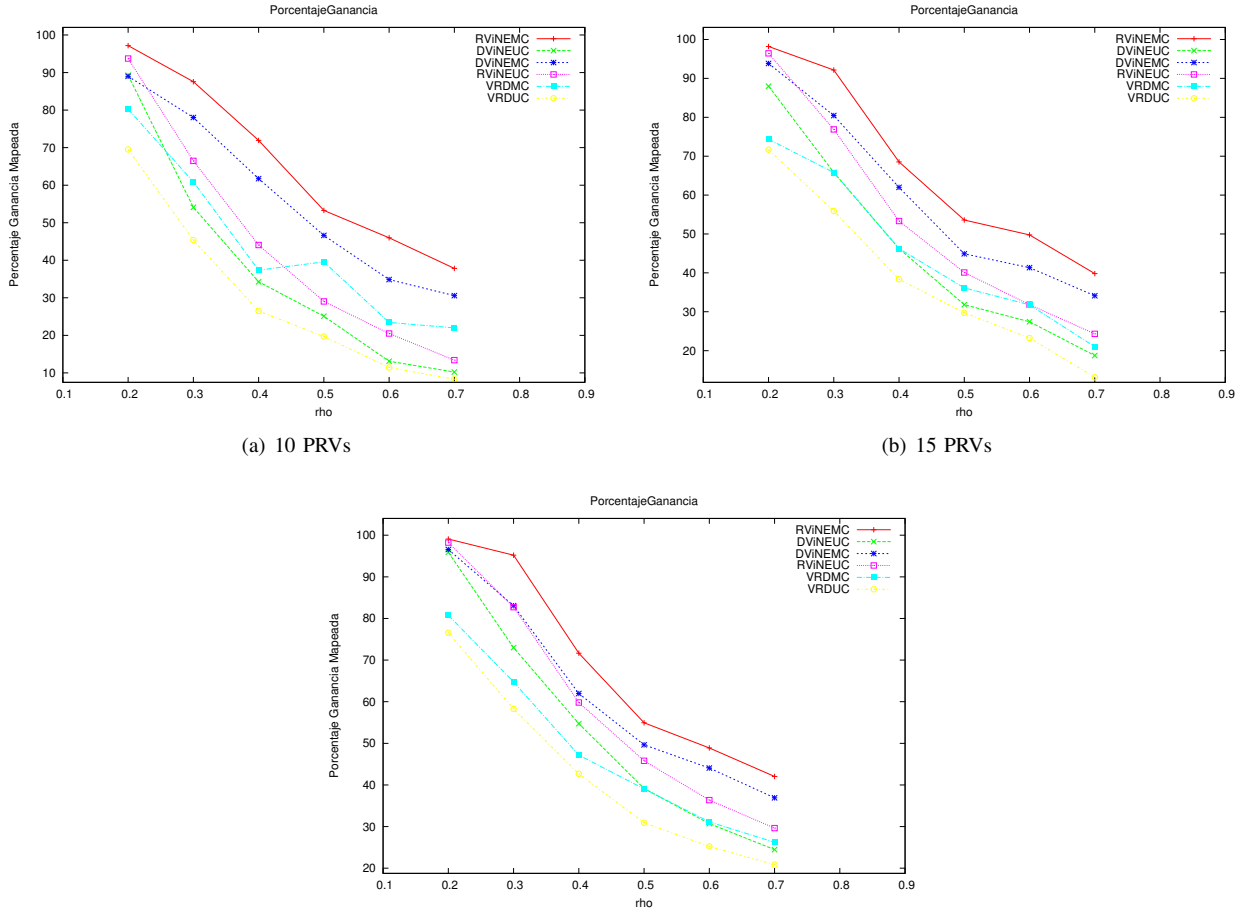
Como la generación de topologías Waxman es probabilística, se realizaron $N = 10$ ejecuciones para cada escenario. Para algunos algoritmos probabilísticos como **RViNEMC, RViNEUC**, se realizaron adicionalmente $M = 10$ ejecuciones para cada escenario.

2) Algoritmos Evaluados: En este trabajo se evaluaron los que, a nuestro parecer, son los seis algoritmos principales en el mapeo de redes virtuales en la actual literatura. Los algoritmos evaluados se muestran en la Tabla IV. Estos algoritmos vienen de una combinación entre dos técnicas para el mapeo de nodos (ver subsecciones III-B y III-C) y dos técnicas para mapeo de enlace (único camino y multicamino). Es importante agregar que en las implementaciones de los algoritmos no se tuvo en cuenta la demanda de los nodos ocultos.

3) Métricas: Dos métricas de evaluación serán utilizadas para comparar los algoritmos:

Porcentaje Ganancia: Esta métrica mostrará cuál es el porcentaje de ganancia (sobre la ganancia total) obtenida al mapear el grupo de redes virtuales por cada algoritmo.

Relación costo-ganancia: Se define como el cociente entre



(a) 10 PRVs

(b) 15 PRVs

(c) 20 PRVs

Fig. 2. Ganancia Obtenida después del Mapeo

Tabla IV
ALGORITMOS EVALUADOS

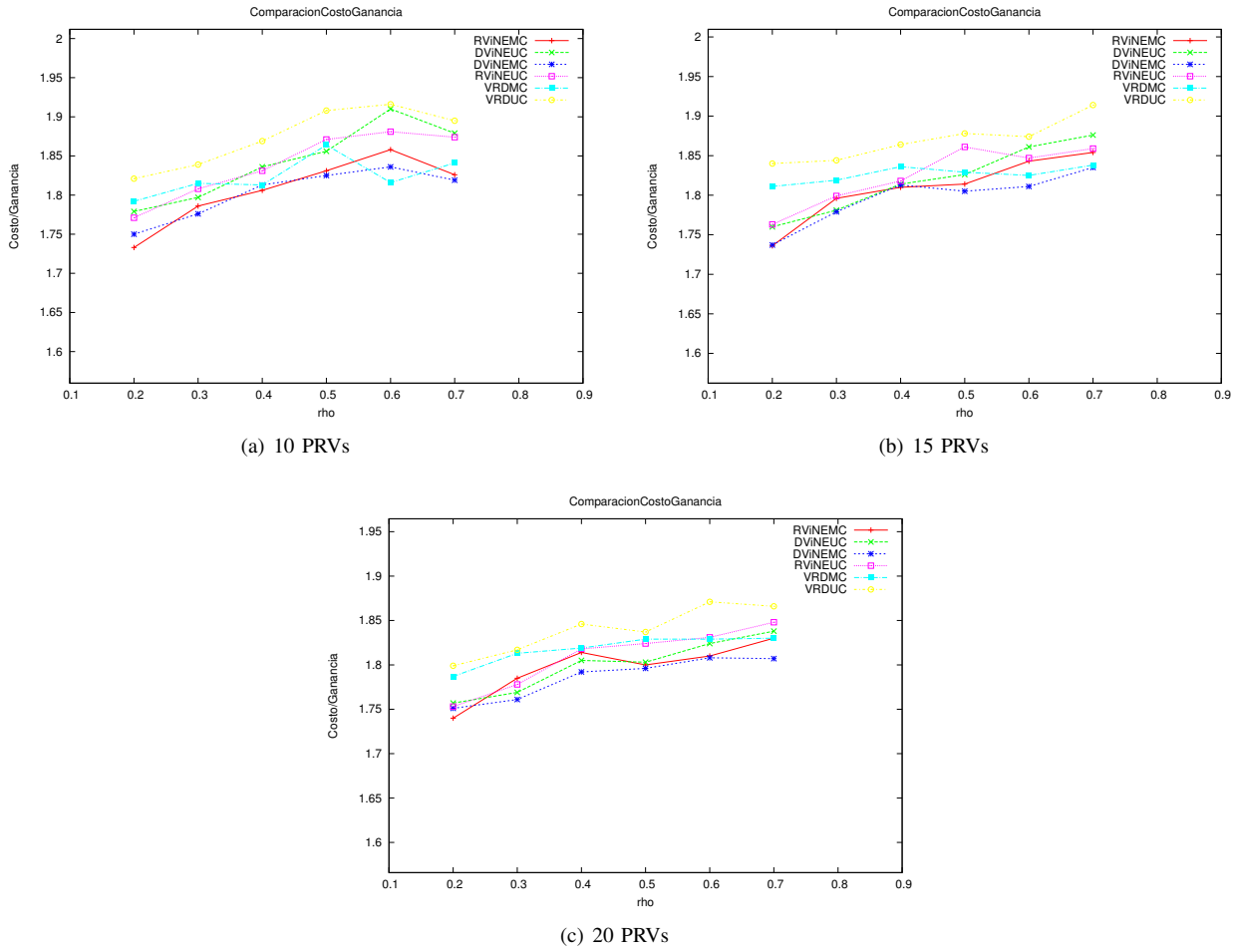
Notación	Descripción del Algoritmo
VRDMC	Algoritmo de asignación voraz de recursos disponibles en el mapeo de nodos y multicamino en el mapeo de enlaces (ver subsección III-B)
VRDUC	Algoritmo de asignación voraz de recursos disponibles en el mapeo de nodos y único camino en el mapeo de enlaces (ver subsección III-B)
DViNEMC	Algoritmo de coordinado con redondeo determinista en mapeo de nodos y multicamino en el mapeo de enlaces (ver subsección III-C)
DViNEUC	Algoritmo de coordinado con redondeo determinista en mapeo de nodos y único camino en el mapeo de enlaces (ver subsección III-C)
RViNEMC	Algoritmo de coordinado con redondeo aleatorio en mapeo de nodos y multicamino en el mapeo de enlaces (ver subsección III-C)
RViNEUC	Algoritmo de coordinado con redondeo aleatorio en mapeo de nodos y único camino en el mapeo de enlaces (ver subsección III-C)

el costo y la ganancia. Esta relación muestra la ganancia que ofrecen los algoritmos teniendo en cuenta el costo que tiene hacer el mapeo en la red sustrato. Se puede decir que a menor relación costo-ganancia mejor es el mapeo.

4) Resultados: Para evaluar los algoritmos se utilizó el software ALEVIN [6] (realizado en trabajo previo) que implementa los algoritmos anteriormente mencionados.

En la Fig. 2 se muestran los resultados obtenidos, en la métrica de ganancia, para los diferentes algoritmos cuando hay 10, 15 y 20 redes virtuales. Las conclusiones que se pueden obtener de estos resultados son las siguientes:

- Como se ha mostrado en los trabajos previos, RViNE-MC obtiene los mejores resultados. Es muy importante tener en cuenta que en este algoritmo, se restringe el grupo de nodos candidatos que pueden ser mapeados por cada nodo virtual.
- Es importante observar que, conforme el número de redes virtuales crece, el porcentaje de ganancia aceptada crece también. Esto indica que los algoritmos funcionan mejor cuando la carga es distribuida en flujos más pequeños.
- Es notorio que la forma en que los nodos se mapean es muy importante, en algunos casos (Figs. 2(b) y 2(c)) el algoritmo de único camino DViNEUC mejora el funcionamiento del algoritmo multicamino pero con asignación de nodos voraz teniendo en cuenta los recursos disponibles (VRDMC).
- Parece extraño observar que aunque se ha diseñado un mecanismo para que la carga de la red sustrato quede distribuida entre las redes virtuales (ver subsección IV-A), el porcentaje de ganancia mapeada desciende a unos niveles muy bajos. Con una carga de 0.7 y 20 redes virtuales, el peor y mejor algoritmo logran unos porcentajes de ganancias de 20% y 45% respectivamente (Fig. 2(c)).



(a) 10 PRVs

(b) 15 PRVs

(c) 20 PRVs

Fig. 3. Relación Costo-Ganancia después del Mapeo

Esto se debe a que los recursos de la red sustrato que son distribuidos entre las PRVs no se ocupan de forma 1:1. Es decir, las demandas de enlace de las PRVs se ocupan utilizando uno o varios enlaces consecutivos (caminos) de la red sustrato y por eso la asignación no obtiene siempre el 100% de ganancia mapeada.

En la Fig. 3 se muestran los resultados obtenidos, en la métrica de la relación costo-ganancia, para los diferentes algoritmos cuando hay 10, 15 y 20 redes virtuales. Se puede concluir que:

- Los algoritmos que usan multicamino para realizar el mapeo de enlaces, tienen una mejor relación costo-ganancia. Esto se debe a que, al poder utilizar múltiples caminos, el costo de mapear las redes virtuales se reduce y por eso la relación disminuye.
- Es muy importante apreciar esta métrica teniendo en cuenta el porcentaje de ganancia mapeado por cada algoritmo. Podría parecer, a la luz de los resultados que, en algunos casos, el algoritmo VRDUC obtiene buenos valores en la relación costo-ganancia, sin embargo, hay que tener también en cuenta, que este algoritmo no ha tenido altos porcentajes de ganancia (lo que implica muy bajos costos), por lo que su relación costo-ganancia disminuye.
- Los algoritmos que usan mapeo coordinado de nodos y enlaces (RViNEMC, RViNEUC, DViNEUC, DViNEMC) presentan una buena relación costo ganancia, teniendo en cuenta el porcentaje de ganancia que mapean. Esto se ve claramente cuando la carga está distribuida entre varias redes (Figs. 3(c) y 2(c)).

V. Conclusiones y Trabajo Futuro

Se ha introducido en este trabajo un estado del arte de los principales algoritmos utilizados para resolver el problema de mapeo o incrustación de redes virtuales, comúnmente llamado VNE por sus siglas en inglés (Virtual Network Embedding). Algunas de estas propuestas se han evaluado con respecto a las métricas de ganancia y de relación costo-ganancia.

Se ha encontrado que los algoritmos multicamino de mapeo coordinado de nodo y enlace (RViNEMC y DViNEMC) son los que obtienen mejores resultados con respecto a las métricas. Sin embargo, es importante aclarar que estos algoritmos escogen un grupo de nodos sustrato candidatos por cada nodo virtual para realizar el mapeo (definido por restricciones de ubicación). Sería muy interesante analizar que pasa cuando todos los nodos de la red sustrato son candidatos para cada nodo de la red virtual.

En la implementación de los algoritmos no se han considerado las demandas inherentes a los nodos ocultos, en un futuro esta característica se debe incorporar. También la implementación de nuevos objetivos como eficiencia energética y una ampliación en la gama de parámetros de red considerados

(no solamente CPU y ancho de banda) se debe considerar en los futuros algoritmos para mapeo de redes virtuales.

AGRADECIMIENTOS

Los autores expresan su agradecimiento a Michael Duelli y Daniel Schlosser de la Universidad de Wuerzburg (Alemania) y a Andreas Fischer de la Universidad de Passau (Alemania) por su contribución a la implementación del software ALEVIN, desarrollado dentro del proyecto VNREAL, con el que los resultados obtenidos en este artículo han sido generados.

Este trabajo ha sido parcialmente financiado por el gobierno español, MICINN, bajo la ayuda de investigación TIN2010-20136-C03, el "Comissionat per a Universitats i Recerca del DIUE" de la Generalitat de Catalunya y el fondo social europeo. También ha sido financiado por el programa FP7 de la comunidad europea ([FP7/2007-2013] [FP7/2007-2011]) en el contexto de la red de excelencia "Euro-NF" (grant agreement no. 216366) a través del proyecto "Specific Joint Developments and Experiments Project" "Virtual Network Resource Embedding Algorithms" (VNREAL)

REFERENCIAS

[1] A. Berl, A. Fischer, and H. de Meer, "Virtualisierung im future internet - virtualisierungsmethoden und anwendungen," *Informatik-Spektrum*, vol. 33, no. 2, pp. 186–194, Apr. 2010.

[2] J. Carapinha and J. Jiménez, "Network virtualization: a view from the bottom," in *Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures*, ser. VISA '09. New York, NY, USA: ACM, 2009, pp. 73–80.

[3] D. Schwerdel, D. Günther, R. Henjes, B. Reuther, and P. Müller, "German-lab experimental facility," *Future Internet Symposium (FIS) 2010*, 9 2010.

[4] J. Botero, X. Hesselbach, A. Fischer, and H. de Meer, "Optimal mapping of virtual networks with hidden hops," *Telecommunication Systems*, pp. 1–10, 2011, 10.1007/s11235-011-9437-0. [Online]. Available: http://dx.doi.org/10.1007/s11235-011-9437-0

[5] Y. Zhu and M. Ammar, "Algorithms for assigning substrate network resources to virtual network components," in *Proc. IEEE INFOCOM*, Apr. 2006, pp. 2812–2823.

[6] A. Fischer, J. F. Botero, M. Duelli, D. Schlosser, X. Hesselbach, and H. De Meer, "ALEVIN - a framework to develop, compare, and analyze virtual network embedding algorithms," *Electronic Communications of the EASST*, vol. [NA], p. [NA], 2011.

[7] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: Substrate support for path splitting and migration," *ACM SIGCOMM CCR*, vol. 38, no. 2, pp. 17–29, Apr. 2008.

[8] N. M. M. K. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," in *Proc. IEEE INFOCOM*. IEEE Infocom, Apr. 2009.

[9] J. Lischka and H. Karl, "A virtual network mapping algorithm based on subgraph isomorphism detection," in *Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures*. New York, USA, Aug. 2009, pp. 81–88.

[10] N. F. Butt, N. M. Chowdhury, and R. Boutaba, "Topology-awareness and reoptimization mechanism for virtual network embedding," in *Networking*, 2010, pp. 27–39.

[11] B. M. Waxman, "Routing of multipoint connections," *IEEE Journal of Selected Areas in Communication*, vol. 6, no. 9, pp. 1617–1622, Dec. 1988.

Integrating probabilistic techniques for indoor localization of heterogeneous clients

Antonio J. Ruiz-Ruiz, Oscar Canovas
Department of Computer Engineering
University of Murcia
Murcia, Spain
antonioruiz@um.es, ocanovas@um.es

***Abstract*—This work is a survey of well-known proposals for indoor location of wireless devices using signal strength on commodity hardware. During the last years, remarkable contributions have been made by the research community to enable location-aware services for indoor scenarios. Location fingerprinting has been proved to be a promising technique of exploiting already existing infrastructures based on IEEE 802.11. In this paper, we integrate several approaches in order to design a location estimator which is able to provide good accuracy and performance for different hardware devices, such as laptops, smart phones and wireless tags. Some of the techniques that we have implemented are: error estimation, clustering, probabilistic inference to estimate the location of a device, hidden Markov model, handling of heterogeneous hardware through the least-squares method, and path-restricted location. Our selection has been made after an exhaustive analysis of the existing proposals, pursuing a good balance between accuracy and performance. The experimental testbed has an area of 1050 squared meters, with several corridors, offices and labs. Our main intention is to determine whether this set of techniques can be used to build a ready-to-use location service for specific scenarios and to investigate the need of integrating other sensors that would enhance the results. For example, signal strength can be used as a powerful tool to determine a cluster of physical points, or zone, where the device seems to be. This coarse-grained estimation can be further refined by means of other sensors, such as accelerometers and cameras, included in commodity handset devices.**

***Keywords*—Wireless networks, 802.11, probabilistic techniques**

I. Introduction

The widespread adoption of devices like smart phones is confirming the essential role of location-based applications. For a diverse set of areas including tracking, geographic routing or entertainment, location-sensing systems have been an active research field. Though the Global Positioning System (GPS) is the predominant outdoor positioning system, it suffers from several obstacles blocking the radio signals indoor.

However, wireless devices, like those based on IEEE 802.11, include the hardware necessary to measure the received signal strength intensity (RSSI) of transmitted packets. Using this widely-deployed off-the-shelf hardware, several previous works claim to obtain a significant accuracy by means of location fingerprinting techniques each associated with distinct tradeoffs between accuracy and scalability.

Nowadays, the increasing number of sensors on mobile devices presents new opportunities for localization [1][5][20]. In-built accelerometers or cameras may be useful in inferring coarse-grained user motion and the nature of particular places, respectively. Nevertheless, some of the most promising techniques for indoor location based on images, like Scale Invariant Feature Transform (or SIFT) [19], require significant computing resources and large databases of descriptors to check against. An initial coarse-grained estimation based on RSSI might constrain the search space to a particular area, enabling the integration of these additional sensors which are available in most of the existing handset devices.

Our long-term goal, out of the scope of this paper, is to define a multi-sensor architecture for indoor localization of smart phones. Thus, in order to accomplish our work, we have performed an extensive survey of previous existing and well-known proposals for indoor location based on RSSI fingerprinting. Therefore, the primary contribution of this paper is an engineering approach for the analysis, implementation and integration of several existing techniques for location estimation. For our particular scenario, we wanted to know which technique provides a higher accuracy, how to improve the performance, how to support different devices, and how the scenario may be optimized when path restrictions apply. As we show, we have mainly focused on location techniques based on Bayesian inference. We find especially interesting the obtained balance between accuracy and performance, which constitutes a solid basis to integrate other sensors.

The rest of this paper is structured as follows. Section II gives an overview of the techniques that inspired our work. Section III describes our experimental setup. Section IV presents the way we have managed different devices. Section V provides the results we obtained with different estimation techniques. Section VI introduces the system model based on Markov. Section VII analyzes how we can improve performance in terms of locations per second. Section VIII describes a method for obtaining better accuracy when path is restricted. Section IX depicts that clustering favors the integration of multiple sensors. Section X provides information about the accuracy provided by our system when using several devices in real time estimations. Finally, Section XI presents our main remarks and future directions.

II. Related Work

Indoor positioning is a research field that has been addressed by many different authors and disciplines. Several types of signals (radio, light, sound) and methods have been used to infer location. Each method has specific requirements as to what types of measurements are needed. Different methods make use of the propagation speed of signals in order to collect distance-related measurements. Lateration methods

[23], such as Time-Of-Flight (TOF) and Time-Difference-Of-Arrival (TDOA), estimate positions from distance-related measurements to fixed sensors with known positions. Angle-Of-Arrival (AOA) methods work by observing what angle a signal from a sensor arrives in. Both lateration and angulation require special sensors or hardware to be installed in the covered area. However, most of the pattern recognition methods, like fingerprinting, estimate locations by recognizing position-related patterns in measurements using commodity hardware. Fingerprinting is based on radio maps containing patterns of RSSIs, which are obtained using 802.11, Zigbee, Bluetooth or any other widespread wireless technology. Maps can be manually obtained by collecting signal samples or can be derived from radio propagation models[8][22]. Compared to other types of positioning methods, fingerprinting is not able to provide the centimetre accuracy realized with other proposals, which is not necessary for most location-based applications. As we will see in this paper, we can obtain an accuracy ranging from 0.5 to 3 meters using fingerprinting.

Fingerprinting can be classified into two main categories: deterministic techniques and probabilistic techniques. Deterministic techniques [2] represent the signal by a scalar value and use some pattern-matching method to estimate the user location, for example by means of nearest neighbour. However, probabilistic techniques [7][25] store information about the signal strength distributions from the access points and represent user positions as probability vectors. For example, one of the main methods to infer location is the Bayesian inference. In this paper we are going to analyze the results obtained using both set of techniques.

There are several options to implement location systems using 802.11, depending on the division of responsibilities between wireless clients, access points, and servers. The three main categories are network-based, client-assisted and client-based, and they differ in who sends out beacons, who makes measurements and who stores the radio map and estimates locations. Network-based systems [2][13] offer better support for limited wireless clients, since measurements are collected by access points and forwarded to location servers. Most fingerprinting systems were built client-assisted or client-based [3][24], which are more suitable to support privacy since clients measure RSSI and might estimate locations using the radio map. As we show in this work, our system is both network-based and client-assisted, depending on the type of client we are using (802.11 wireless tags, smart phones or laptops).

III. Experimental environment

A. Physical environment

The testbed where our experiments were conducted is located on the third floor of our Faculty where several students, researchers and professors move around constantly. The dimension of the testbed is 35 meters by 30 meters, and includes 26 rooms. We have selected 94 cells where the users could be located, spaced out 1.5 meters, according to recommendations made by King et al. in [9]. These cells are representative of places where users are both mainly still (such as offices and laboratories) or in motion (like corridors).

Our location system works in two phases. First, an off-line or training phase is performed to build the radio signal map and to obtain the signal distribution models. Then, it is during the on-line phase when we are able to estimate the user location.

As we will show, in order to compare the accuracy of our system depending on the number of access points and the number of samples used to build the radio map, we carried out several tests using two different testbed configurations. Initially, we distributed four access points along our dependencies (in Figure 1 they are indicated as red dots). During the corresponding training phase we collected 60 observations[1] at each cell. Later, we carried out a second set of tests, by adding two more access points (blue squares in Figure 1) and collecting 250 observations at each cell. There are

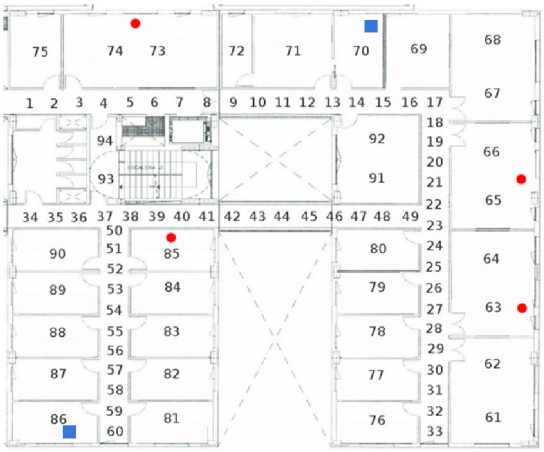


Fig. 1. Experimental environment map

several works proposing how to automate the training phase. Chen et al. [4] or LaMarca et al. [17] provide techniques for the automatic generation of fingerprinting maps. The former approach was developed using RFID sensors, while the latter studies the pattern of WiFi signals. Though we have not integrated these proposals within our testbed, as it is relatively small, they should be considered in order to improve the scalability for bigger scenarios.

B. Hardware and software

Our experiments were carried out using several hardware devices. The training observations were captured with an Asus Eee 1201 laptop with a Realtek TRL8191SE Wireless LAN 802.11n card. In addition, during the online phase we have also used a HP iPAD hx2400 series using Windows Mobile 2005, a HTC Desire smart phone with Android and Aeroscout T2 802.11 wireless tags. Tags transmit beacons in a regular basis (three per second) to the APs whilst the rest of devices perform a passive scanning. With the exception of wireless tags, we developed the appropriate software client for each device in order to collect RSSIs and to send them to a repository. Applications were programmed in C++ and Java, depending on the requirements imposed by each device. Furthermore, we implemented several estimation techniques in Java. According to the different nature of our devices, the system was designed to support both a client-assisted and a

[1] An observation is a set of RSSIs collected from all the reachable access points at the same cell and during a particular scan.

network-based infrastructure, that is, RSSI can be collected by the end-user devices or by the 802.11 access points.

We have used Linksys WRT54G access points with 802.11abg support working on channel 5 (since it was the least congested). Their locations were chosen so as to provide consistent coverage throughout the entire scenario, guaranteeing that each cell is covered by at least four access points. In addition, the firmware was modified to work in monitor mode, thus providing support for special devices with limited computing resources, like the already mentioned wireless tags. Nevertheless, for those scenarios where wireless tags are not used there is no need to modify the firmware of the access points and our experiments can be conducted with similar results.

IV. Calibration

Besides accuracy or performance, one of the imposed requirements of our proposal is the support for heterogeneous hardware clients. Due to the wide range of devices on the market, we do not want to restrict the performance of our location system to specific hardware. However, different devices provide different intensity readings, depending on antennas, transmission power and many other factors. Gwon and Jain proposed in [6] a calibration-free location algorithm that eliminates offline RSSI measurements. However, mean error distance is about 5.4 meters. Several proposals such as [7][10][11] provide calibration mechanisms improving this distance error.

On the one hand, Haeberlen et al. [7] proposed a calibration function based on the following linear relationship:

$$c(i) = c_1 \cdot i + c_2 \quad (1)$$

where i is the observed signal intensity value by the new device and c(i) is the value that would have been observed by the training device. Computing the least-squares fit between the observations obtained by the new device on the calibration cells[2] and the corresponding values from the sensor map, we can obtain the parameters c_1 and c_2. The authors proposed several methods for manual, quasi-automatic and automatic calibration. On the other had, Kjaergaard [11] proposed a Hyperbolic location fingerprinting to solve the signal-strength difference problem and an automatic technique [10] for adapting an indoor localization system based on signal strength to the specific hardware and software of a wireless network client. In relation to our scenario, the best calibration parameters were obtained with the proposal from Haeberlen et al. As Figure 2 shows, unadjusted RSSIs do not fit to the training laptop signal. Nevertheless, once we have calibrated all the devices, Figure 3, signals are quite similar.

The calibration of the smart phones was performed while they were hand held and while they were inside the pocket, with no meaningful differences in the final results.

V. Analysis of Estimation Methods

As we mentioned before, our first intention was to explore the accuracy of the system as we varied the amount of access points and the number of observations used to build the

[2]A set of cells previously established to get a heterogeneous set of observations.

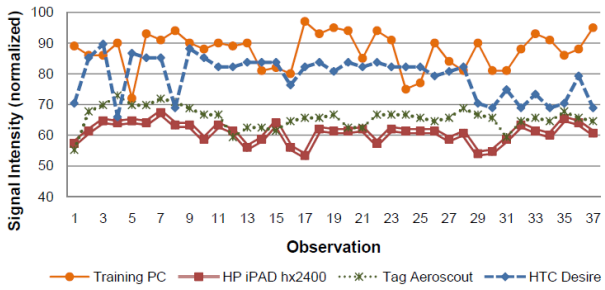


Fig. 2. Signal intensity before calibration

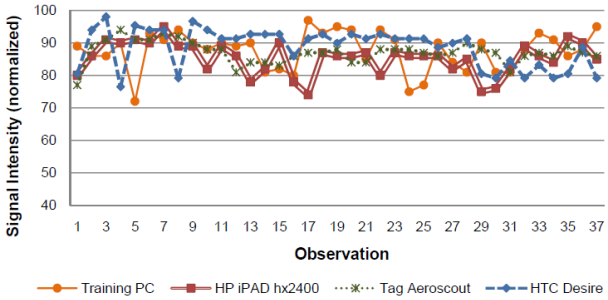


Fig. 3. Signal intensity after calibration

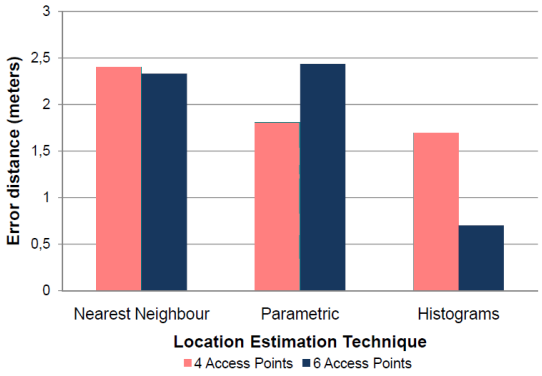


Fig. 4. Mean error for different configurations

radio maps. We have accomplished several tests using two different techniques in order to compare their results. On the one hand we have used a deterministic technique based on nearest neighbour and Euclidean distance of RSSIs. We implemented the proposal from [2] and it is able to estimate location with a mean error distance between 2 and 3 meters, about the size of a typical office room. On the other hand, we have also represented the position as a probability distribution using a Bayesian inference technique discussed in [7], [16]. This algorithm estimates posterior distributions and can be applied in the case of sensors that have non-Gaussian noise distributions, such as our signal strength sensor.

We have to take into account that signal propagation in an indoor environment is noisy since it is affected by reflection, diffraction, and scattering of radio waves caused by structures within the building. These dynamic environmental influences can cause the observed signal strength to vary considerably and this makes very difficult to estimate the location using a single signal observation. So, using historical information about the previous locations of the user, we may get better

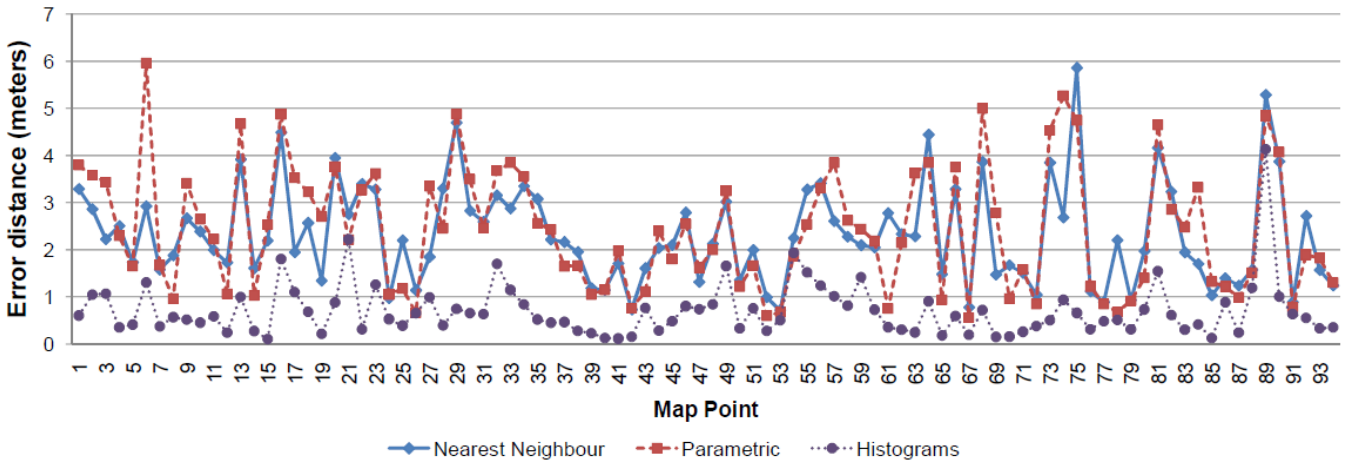


Fig. 5. Mean error at each cell

results by means of probabilistic methods, as you can see in Figure 4. Hereinafter, we will focus our tests on probabilistic methods since they offer several possibilities to improve the performance and accuracy of our system.

Being $C = \{c_1, .., c_m\}$ the set of cells that make up the finite space state, n the amount of measurements in the current observation and π a probability distribution vector over each cell, for each observation O_j, the probability to take a measurement from the access point a_β at reference cell c_i with a signal strength λ_β can be expressed by the conditional probability:

$$Pr(O_j|c_i) = \prod_{\beta=1}^{n} Pr(\lambda_\beta|a_\beta, c_i) \quad (2)$$

These conditional probabilities are used to update the probability vector π by applying Bayes' Rule:

$$\pi'_i = \frac{\pi_i Pr(O_j|c_i)}{\sum_{\alpha=1}^{m}(\pi_\alpha Pr(O_j|c_\alpha))} \quad (3)$$

We also compare taking a Gaussian fit of signal strength to using the full histogram of signal strength. Parametric-based distribution is built by modelling the signal intensity as a normal distribution defined at each cell and for every base station by its mean and standard deviation. Histograms represent the sensor model explicitly.

As you can see in Figure 4, there are some techniques that perform better using four access points but it is clear that the result obtained using the histogram-based probabilistic method and six access points provides the higher accuracy. WiFi signals have a very unpredictable behaviour so the main cause of histograms to perform better than parametric is because signals are not fit to a parametric based probability distribution, therefore using a histogram based probability distribution it is easier to obtain a correct probability estimation. Therefore, we are going to analyze the results obtained from this configuration of 6 base stations and 250 training observations. Additionally, in order to provide more detailed information, Figure 5 shows the mean error for each cell after estimating the user position.

The shape of the histogram sometimes is particularly sensitive to the number of bins[3]. In order to find the right number of bins there are several aspects that we have to take into account. If the bins are too wide, important information might get omitted. However, if the bins are too narrow, what seems to be meaningful information may be due to random variations that show up because of the small range in a bin. In conclusion, there is no best number of bins since different bin sizes can reveal different features of the data. So, to determine whether the bin width is set to an appropriate size, different bin widths should be tested to determine the sensitivity of the histogram shape with respect to it size. It is worth mentioning that every bin is considered to contain at least one sample, in order to discard zero probability.

Figure 6 shows the error distance obtained using a distribution model based on histograms varying the number of bins. When a greater number of bins is used, accuracy improves since each bin is formed by a lower range of samples, giving more importance to those that are more representative, and lowering the error down to 0.7 meters.

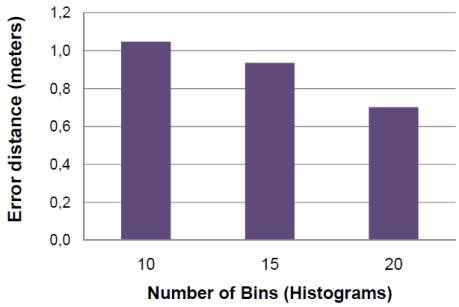


Fig. 6. Accuracy depending on bins

VI. System Model

Until now, the way we have used RSSIs is not rich enough to track the location of a mobile device since we should include additional information to infer motion. Considering that, at this stage, we have not integrated inertial sensors (like

[3]Each of the disjoint categories in which a Histogram is divided.

an accelerometer) into our system, we might take into account several proposals integrating sensor readings over time to track mobile users. Krumm and Horvitz [15] measure the variance of the signal strength of the strongest access point to infer whether the user is still or moving. Muthukrishnan et al. [21] present an inference system based on euclidean distances between signals. Despite both proposals are good motion estimators, we implemented an algorithm which takes the output of the estimation method as a stream of observations and stabilizes the distribution by modelling the usual behaviour of users within our scenario.

This algorithm is based on a Hidden Markov Model (HMM) [12] and it has been used in several proposals such as [7][16], where it has been proved as a good system model. Given a user position, this method spreads probability over those points that are reachable during the next interval of time. The performance we can obtain from HMM depends on the design of the Markov chain, which encodes assumptions about how the user can move from state to state, referring to a state as a cell in our scenario. This chain specifies the probability of remaining still at a cell or moving to a nearby one. One of the more critical points of using HMM is to define the matrix describing how the system being modelled evolves with time. In order to create the chain that best fits to our environment, we took into account several considerations. We have designed a matrix A that encodes the HMM chain considering the normal behaviour of users around our scenario. As we saw in equation 3, if π is a probability distribution vector over C, then $\pi' = A\pi$ will be the probability distribution vector at the following instant time.

Definition of A will be carried out only once. Taking into account that our scenario is mainly formed by offices and laboratories where people usually stays static, probability of moving should be lower than remaining at the same point. Also, we assume that people do not exceed a speed of 2 meters per second. Using our graph-based space model and the adjacency between cells, we can automatically build matrix A by means of spreading the likelihood of moving from one cell to those which are reachable during the period of time between two consecutive estimations.

Once the HMM matrix is designed, we carried out some tests using histograms and 20 bins in order to check whether better results are obtained. Figure 7 confirms that the best location estimation technique is based on histograms and including HMM, since it usually reduces the error down to 0.41 meters on average.

VII. Performance Analysis

In order to reduce the computational cost of our location estimation system, to minimize the number of operations per location estimation, and thus to get a greater number of locations per second, we studied the contributions made by Youssef et al. in [24][25], paying more attention to the *Incremental Triangulation (IT)* clustering technique. This technique is based on the idea that the strongest signals come from the nearest access points. Therefore we can assume that those signals are more stable and more reliable. So, when we estimate the location of a user using the received signals ordered by their intensity, it means that we evaluate the signals ordered by their usefulness. During the location estimation process we use the access points iteratively, one after the other, starting with the first access point. Therefore, we restrict our search space to the cells covered by this access point. In reduced scenarios, like ours, this might not suppose any important improvement since we do not discard so many cells. Nevertheless, in bigger spaces with a higher number of access points, like an airport or a hospital, this can suppose a huge time reduction. As it is presented in [24], given a sequence of observations from each access point, we start by sorting the access points in descending order according to received intensity. If the probability of the most probable location is meaningfully higher (threshold) than the probability of the second most probable location, we return that most probable location as our location estimation, and we do not take into consideration the next access points. According to equation 2, when we calculate the probability of being at each cell we reduce the number n of access points. This provides an additional advantage since some outliers providing weak signals might add noise to the localization process, but now they would be filtered.

Before analyzing the *IT* results, we would like to note that our main intention is to compare performance in relative terms. However, for the sake of completeness we provide the details of the used computing platform: CPU Pentium(R) Dual-Core E5300(2M Cache, 2.60 GHz, 800 MHz FSB), 2GB RAM memory and Windows XP Professional.

IT Threshold	Error Distance	Access Points	Locations/sec	Optimization Locations/sec	Improvement
No IT	0,419	6,0	1772	-	-
0.1	1,224	2,3	2645	2670	33,64%
0.2	0,854	2,8	2219	2437	27,31%
0.3	0,668	3,2	2240	2337	24,19%
0.4	0,571	3,5	2102	2180	18,72%

Tabla I
PERFORMANCE ANALYSIS

Table I summarizes the obtained results applying this algorithm. For lower threshold values (1st column), the decision is taken quickly after examining a small number of access points, no more than 3 access points on average (3rd column). As the threshold value increases, a higher number of access points has to be evaluated. Consequently, as the number of considered access points increases, the number of operations increases, which reduces the number of location estimations per second (4th column), but the average accuracy increases (2nd column). We have carried out this test using the histogram-based probability distribution technique with HMM. As we can see, using *IT* we can obtain similar results in terms of accuracy to those obtained previously. Using a threshold of 0.4 we are able to reduce the number of analyzed access points, from 6 to 3.5 on average. This involves a speed-up of 15.73% on average. System accuracy is adversely affected by a few centimetres, from an error distance of 0.41 m. to 0.57 m., what is acceptable to estimate the location of a user into our scenario.

Despite we obtained a good improvement with *IT*, we designed a further optimization. This optimization tries to improve system performance without compromising accuracy. We avoid to evaluate unnecessary cells (at each iteration of

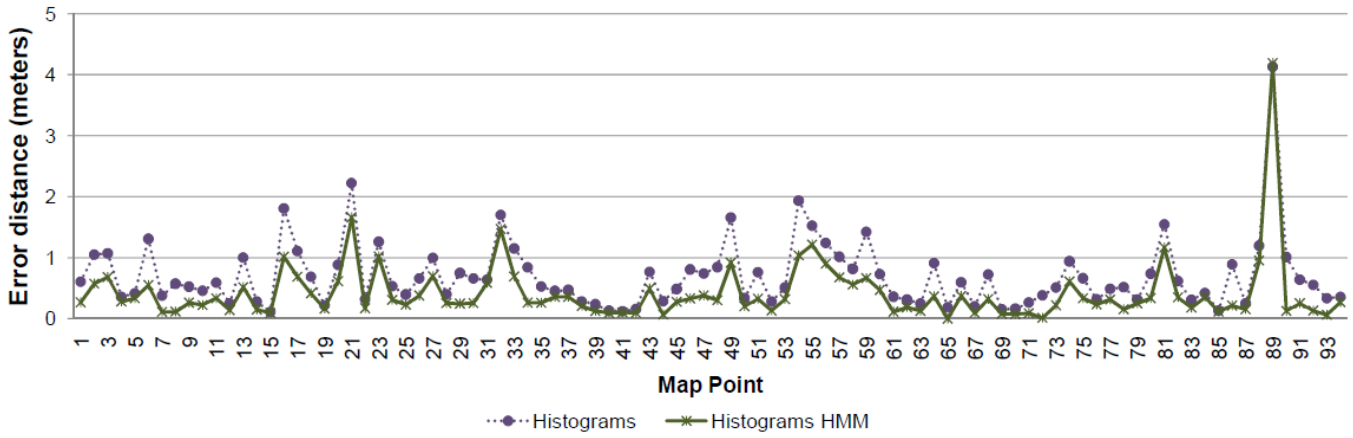


Fig. 7. Histogram-based position estimation error with HMM at each cell

the *IT* algorithm) where probability is meaningfully low. For example, if using the signal received by one of the strongest access points the cell probability is under a threshold, we will not evaluate this cell again using the next access points. This threshold is determined by the minimum density of histogram distributions.

This optimization reduces the required cells in vector π (equation 3), and therefore the number of locations per second increases. The 5th column in table I shows the results of applying this optimization to the *IT* technique, always offering better results. The 6th column shows the speed-up of using *IT* in relation to the absence of any improvement (1st row). As we can see, we can improve our performance up to 18.72% without having an adverse effect on accuracy.

VIII. Path-Restricted Location

There are scenarios where users have restricted access to some dependencies[4]. To reflect these restrictions, we have to discard those points where the user cannot be located. One approach is to label each cell indicating its access level. Since our scenario is within a Faculty, we have conducted some tests assuming two different types of users: professors and students. Usually students will move primarily along the corridors, so the cells belonging to those dependencies are labeled as public. The rest of dependencies are labeled as private, and only professors can gain access to them.

Consequently, we propose another optimization with the aim of minimizing the number of cells where a user could be located. We called it Path-Restricted Location (PRL). IT-based and PRL optimizations may be complementaries, but we prefer to show them in an independent way. We carried out some tests assuming that the user was a student and therefore he had no access to private rooms. To carry out these tests we have used the histogram-based probability distribution with 20 bins and HMM. The path we have covered during the test goes from cell 33, through 49 and 50, to cell 60 (Figure 1).

As you can see in Figure 8, PRL still improves accuracy due to average error is reduced to 0.31 meters. This makes sense if we think that the number of cells analyzed is lower than in the previous tests, around 33%, discarding the possibility of being in private dependencies.

[4]Referred to a set of cells that form a corridor, a laboratory, an office, etc.

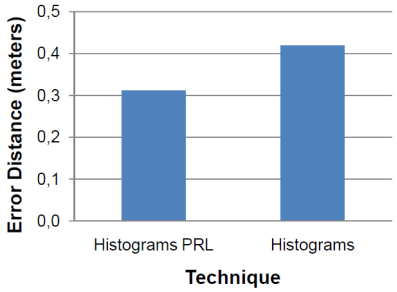


Fig. 8. Accuracy using PRL

IX. Clustering

Clustering techniques have been applied in several ways. On the one hand, Youssef et al. [25] propose the Joint Clustering algorithm that uses joint probability distributions of the RSSI of different access points to find the most probable user location. They try to reduce the computational cost by grouping the cells into clusters according to the access points providing coverage, at the expense of loosing accuracy. Each cell belonging to a cluster has in common the order in which signals are received, according to their intensity, from those of the strongest visible access points q choose for clustering. This technique is further applied during the online phase, using the q strongest access points to select the cluster of cells that will be analyzed to determine the most probable location. A similar proposal was made by Krumm and Hinckley [14] to obtain a coarse-grained proximity between users.

On the other hand, Lemelson et al. propose in [18] four algorithms to estimate the position error that is inherent to 802.11-based positioning systems. One of them, Fingerprint Clustering algorithm, makes use of the training RSSIs to find clusters. It is based on the idea that the signal collected in nearby cells tend to cover only a limited range of the possible values. So, if we find an area with similar signal properties, the position estimation error will be higher because of the number of similar fingerprints. However, we have a high probability to estimate the location of a user within those areas. We are especially interested in this second proposal, since it will help us to define medium-sized zones, joining adjacent clusters, where WiFi-based location might be further

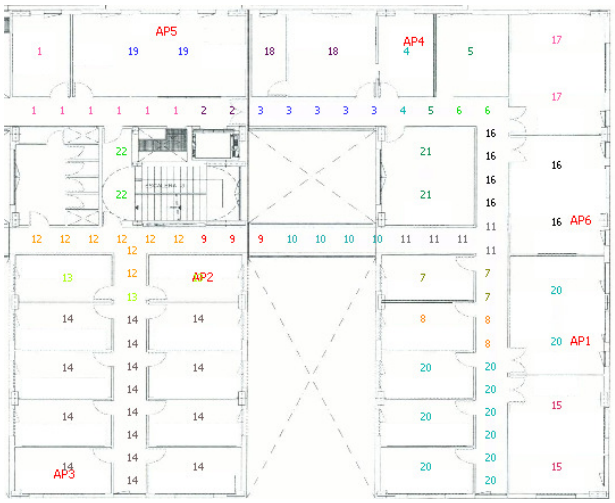


Fig. 9. Clusters

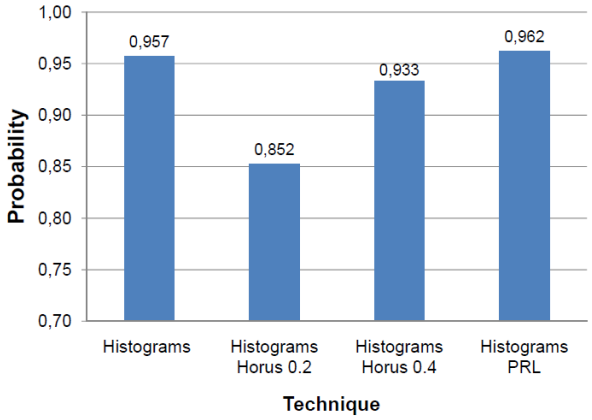


Fig. 10. Cluster hit probability

refined using other sensors. We can define the clusters once the training phase have been finished, and it does not require a high computational effort.

In order to show how this proposal can be applied to our interest, we have calculated the clusters, shown in Figure 9, and then we have carried out several tests. The clusters hit probabilities obtained from those tests are shown in Figure 10. As you can see, most of the already-analyzed techniques obtain a high cluster hit percentage, up to 93%. These results will have important implications for our future work.

X. Tracking tests

Previous sections have presented analytic results that were calculated using the observations obtained during the training phase as inputs to our location system. Nevertheless, in order to validate our location system, we realized that we have to demonstrate its accuracy carrying out real time test, i.e. trying to estimate a user location using the RSSI captured while the user is walking around the scenario. The estimation method we used is also histogram-based with HMM.

Therefore, we performed several walks around our scenario carrying three different devices: PDA, smart phone and laptop (the one used during the training phase). We took one observation at each cell along the path. Figure 11 presents the results on average. There are several conclusions we can derive from

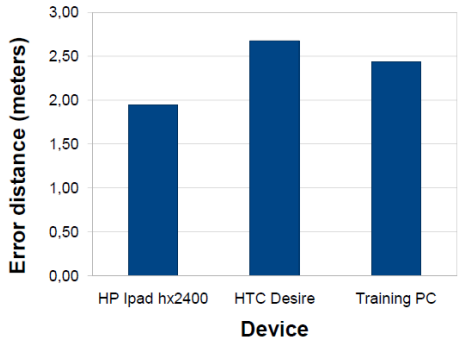


Fig. 11. Mean error while tracking

those tests. Firstly, we show that the error distance is lower than 2.5 meters on average. This accuracy is quite similar to that obtained in referenced proposals. Secondly, both Figures highlight the importance of the calibration process, since the results obtained with both the PDA and the smart phone are very similar to those obtained with the training laptop.

XI. Summary and Future work

In this paper, we have analyzed widely known research works for indoor location in order to evaluate them and to design a system for heterogeneous clients. This heterogeneity is given by the possibility of using a wide range of devices. The results shown in the calibration section allow us to be optimistic about it, as we have been able to adapt the signals collected by different devices to those of the training laptop.

We modelled the signal strength distributions received from access points using deterministic and probabilistic techniques (by means of parametric and non-parametric based probability distributions). This allowed us to demonstrate that probabilistic methods fit better to signal behaviour since they reduces the effect of temporal variations. Therefore, we decided to use the Bayesian inference technique using a 20 bins histogram-based probability distribution as default algorithm for our next experimental tests, because it reduces the error down to 0.7 meters on average. Thereinafter, we have added several optimizations to our location estimation system that offer better accuracy and performance results.

The integration of HMM, discussed in section VI, improves the accuracy of our system. In the absence of inertial sensors, the HMM allows us to estimate user movements. Using this widely-deployed technique we have improved the accuracy of the system up to 40% on average, reducing the error to 0.41 meters.

In addition, we have analyzed several proposals in order to improve the system performance, and we have carried out our own tests to validate their benefits within our scenario. On the one hand, in section VII we have analyzed the *Incremental Triangulation (IT)* clustering technique, that allows us to reduce the number of required operations to infer the user location. Furthermore, we proposed an optimization to *IT* that improves the system performance up to 18.72% without having an adverse effect on accuracy. On the other hand, test results from section VIII, where we discussed the Path-Restricted Location optimization, show that using environment information we avoid the evaluation of unnecessary cells, and we are able to improve the average accuracy error.

This results demonstrate the need for an appropriate context model, whose design we are already defining.

From previous sections, we are able to state the degree of accuracy a WiFi sensor can offer. Indeed, considering a cluster level accuracy around 93% on average, we will concentrate our efforts on integrating several sensors within our location system. Some proposals, like Azizyan et al. [1], introduced several methods to join data from different sensors of existing smart phones. Our future direction to exploit the sensor fusion goes in a different way.

In order to check if our system works properly in real time conditions we carried out some tracking test, discussed in section X. After analyzing this tests results we can get some conclusions. We confirm that selected position estimation technique gets good results to locate a user in motion. Moreover, we demonstrate that it has been able to adapt the signals collected by different devices since all of them have similar behaviour, so it means that calibration works properly.

Despite the techniques are working properly in a specific environment, we are aware that different algorithms may work better in a different setup. However, a broadly generalizable conclusion is that we have obtained a reasonable integration of techniques that is able to provide an acceptable cluster-grained localization estimation.

As a statement of direction, we have performed some initial tests using the camera of the smart phone and the Scale-Invariant Feature Transform (or SIFT) algorithm proposed by Lowe [19]. The use of clustering algorithms, such as Fingerprint Clustering, reduces the number of cells to be analyzed to those contained in the cluster and is helping us to reduce the set of images required to perform a fine-grained localization, improving scalability.

XII. Acknowledgements

This work has been supported by the Spanish project CICYT-TIN2009-14475-C04.

References

[1] M. Azizyan, I. Constandache, and R. Roy Choudhury. SurroundSense: Mobile Phone Localization via Ambience Fingerprinting. In Proceedings of the 15th Annual International Conference on Mobile Computing and Networking, MOBICOM, pages 261-272, Beijing, China, September 2009.

[2] P. Bahl and V. N. Padmanabhan. RADAR: An In-Building RF-based User Location and Tracking System. In Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, volume 2, pages 775-784, Tel Aviv, Israel, March 2000. IEEE INFOCOM.

[3] G. Borriello, M. Chalmers, A. LaMarca, and P. Nixon. Delivering Real-World Ubiquitous Location Systems. Communications of the ACM, 48:36-41, March 2005.

[4] Y.-C. Chen, J.-R. Chiang, H.-h. Chu, P. Huang, and A. W. Tsui. Sensor-Assisted Wi-Fi Indoor Location System for Adapting to Environmental Dynamics. In Proceedings of the 8th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM, pages 118-125, Montreal, Quebec, Canada, 2005.

[5] S. Diverdi and T. Höllerer. Groundcam: A Tracking Modality for Mobile Mixed Reality. In Proceedings of IEEE Virtual Reality Conference, VR, pages 75-82, 2007.

[6] Y. Gwon and R. Jain. Error Characteristics and Calibration-free Techniques for Wireless LAN-based Location Estimation. In Proceedings of the 2nd International Workshop on Mobility Management and Wireless Access Protocols, MobiWac, pages 2-9, Philadelphia, PA, USA, 2004.

[7] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki. Practical Robust Localization over Large-Scale 802.11 Wireless Networks. In Proceedings of the 10th Annual International Conference on Mobile Computing and Networking, MOBICOM, pages 70-84, Philadelphia, PA, USA, 2004.

[8] H. Hashemi. Indoor Radio Propagation Channel. In Proceedings of the IEEE, volume 81,7, pages 943-968, Washington, DC, USA, August 1993. IEEE Computer Society.

[9] T. King, T. Haenselmann, and W. Effelsberg. Deployment, calibration, and Measurement Factors for Position Errors in 802.11-based Indoor Positioning Systems. In Proceedings of the 3rd International Conference on Location-and Context-Awareness, LoCA, pages 17-34, Oberpfaoenhofen, Germany, 2007.

[10] M. Kjaergaard. Automatic Mitigation of Sensor Variations for Signal Strength Based Location Systems. In M. Hazas, J. Krumm, and T. Strang, editors, Location and Context Awareness, volume 3987 of Lecture Notes in Computer Science, pages 30-47. Springer Berlin / Heidelberg, 2006.

[11] M. B. Kjaergaard and C. V. Munk. Hyperbolic Location Fingerprinting: A Calibration-Free Solution for Handling Diferences in Signal Strength. In Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications, PerCom, pages 110-116, Hong Kong, China, 2008. IEEE Computer Society.

[12] K. Konolige and K. Chou. Markov Localization using Correlation. In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI, pages 1154-1159, Stockholm, Sweden, 1999. Morgan Kaufmann Publishers Inc.

[13] P. Krishnan, A. Krishnakumar, W.-H. Ju, C. Mallows, and S. Ganu. A System for LEASE: Location Estimation Assisted by Stationary Emitters for Indoor RF Wireless Networks. In Proceedings of the Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE INFOCOM, Hong Kong, China, 2004.

[14] J. Krumm and K. Hinckley. The NearMe Wireless Proximity Server. In Proceedings of the Sixth International Conference on Ubiquitous Computing, UbiComp, pages 283-300, Nottingham, England, September 2004. Springer.

[15] J. Krumm and E. Horvitz. LOCADIO: Inferring Motion and Location from Wi-Fi Signal Strengths. In First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, Mobiquitous, pages 4-13, Boston, MA, USA, August 2004.

[16] A. M. Ladd, K. E. Bekris, A. Rudys, G. Marceau, L. E. Kavraki, and D. S. Wallach. Robotics-Based Location Sensing Using Wireless Ethernet. In Proceedings of the 8th Annual International Conference on Mobile Computing and Networking, MOBICOM, pages 227-238, Atlanta, Georgia, USA, 2002.

[17] A. Lamarca, J. Hightower, I. Smith, and S. Consolvo. Self-Mapping in 802.11 Location Systems. In Proceedings of the Seventh International Conference on Ubiquitous Computing (Ubicomp 2005), Lecture Notes in Computer Science, pages 87-104, Tokyo, Japan, 2005.

[18] H. Lemelson, M. B. Kjaergaard, R. Hansen, and T. King. Error Estimation for Indoor 802.11 Location Fingerprinting. In Proceedings of the 4th International Symposium on Location and Context Awareness, LoCA '09, pages 138-155, Tokyo, Japan, 2009.

[19] D. G. Lowe. Object Recognition from Local Scale-Invariant Features. In Proceedings of the International Conference on Computer Vision, Volume 2, ICCV '99, pages 1150-1157, Kerkyra, Greece, September 1999. IEEE Computer Society.

[20] A. Mulloni, D. Wagner, I. Barakonyi, and D. Schmalstieg. Indoor Positioning and Navigation with Camera Phones. IEEE Pervasive Computing, 8:22-31, 2009.

[21] K. Muthukrishnan, M. Lijding, N. Meratnia, and P. Havinga. Sensing motion using spectral and spatial analysis of WLAN RSSI. In Proceedings of the 2nd European Conference on Smart Sensing and Context, EuroSSC, pages 62-76, Kendal, England, 2007.

[22] T. S. Rappaport. Wireless Communications-Principles and Practice. Prentice Hall Communications Engineering and Emerging Technologies Series, 2 edition, 2005.

[23] M. Vossiek, L. Wiebking, P. Gulden, J. Wieghart, C. Hoffman, and P. Heide. Wireless local positioning. IEEE Microwave magazine,Volume 4, No. 4, pages 77-86, 2003.

[24] M. Youssef and A. Agrawala. The Horus WLAN Location Determination System. In Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services, MobiSys, pages 205-218, Seattle, Washington, 2005.

[25] M. A. Youssef, A. Agrawala, and A. U. Shankar. WLAN Location Determination via Clustering and Probability Distributions. In Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, PERCOM, pages 143-150, Dallas-Fort Worth, Texas, USA, March 2003.

Generador de tráfico sintético para la evaluación del rendimiento de cachés

F.J. González-Cañete, Raúl Jiménez-Jiménez, E. Casilari
Departamento de Tecnología Electrónica,
Universidad de Málaga
ETSI de Telecomunicación, Campus de Teatinos, 29071, Málaga.
fgc@uma.es, ruljimenez@yahoo.es, ecasilari@uma.es

***Resumen*- En este artículo se presenta un generador de tráfico telemático para la evaluación del rendimiento de cachés tanto en redes cableadas como redes inalámbricas con acceso a Internet. Para ello, el trabajo se basa en los estudios previos sobre las características del tráfico Web para implementar un generador de tráfico con dichas características. El generador se configura para tener en cuenta parámetros como el número de documentos referenciados una única vez, la popularidad, la localidad temporal, la correlación entre tamaño de documentos y su frecuencia de acceso, el modelado del tamaño de los documentos y el tipo de los mismos. La aplicación ha sido validada para comprobar que las características del tráfico generado coinciden con los valores ideales.**

***Palabras Clave*- generador de tráfico, caché**

I. INTRODUCCIÓN

En los últimos años se ha producido un considerable aumento de la popularidad de las páginas Web, que se han introducido en todos los ámbitos de nuestra sociedad. Muchas razones explican este incremento, entre las que destacamos: la disponibilidad de interfaces gráficas de usuario para navegar, la existencia de editores y herramientas de soporte para la creación y publicación de documentos Web; una tendencia de investigadores, instituciones educacionales y públicas, y organizaciones comerciales para usar la Web como diseminador de información y de enlace con sus clientes. Este crecimiento ha llevado consigo la aparición de problemas tales como: grandes congestiones en la red, un bajo ancho de banda, altas latencias al momento de extraer un documento, sobrecarga en servidores y accesos simultáneos masivos a un servidor, entre otros.

Por otro lado, también ha habido un gran auge de las redes inalámbricas móviles. Estas redes se caracterizan por tener unas importantes restricciones en lo referente a ancho de banda y capacidad de los terminales. Por lo tanto, cuando este tipo de redes intentan acceder a través de pasarelas hacia Internet, se produce un problema de calidad de servicio aún mayor que en las redes cableadas debido a las restricciones anteriormente comentadas.

Muchos han sido los intentos para paliar estos efectos, pero sin duda, una de las soluciones más utilizadas es el uso de cachés, cuya misión es la de almacenar aquellos documentos que han sido solicitados por los usuarios con mayor frecuencia. Cuando se produce la petición de un documento ya consultado, éste será servido directamente por la caché disminuyendo, por tanto, el tiempo de servicio, la sobrecarga de servidores y la sobrecarga de la red. Estas cachés pueden estar localizadas tanto en los clientes (navegadores Web) como en servidores *proxy*. En el caso de las cachés en los navegadores, el tráfico generado es nulo o muy pequeño, ya que, cuando el usuario solicita un documento que ya tiene almacenado en la cachés, éste es servido directamente por ella o, como mucho, se realizará una consulta al servidor para comprobar que el documento sigue siendo válido. Por otro lado, los servidores *proxy* son cachés intermedias que se sitúan entre los usuarios y los servidores de forma que almacenan aquellos documentos que son más frecuentemente accedidos por una comunidad de usuarios. Al encontrarse los *proxies* situados más cerca de los usuarios que los propios servidores Web, el tiempo de respuesta se reduce, así como la carga en los servidores Web.

Por tanto, el uso de cachés es una buena solución para agilizar la atención de peticiones a documentos Web tanto en redes cableadas como móviles, pero el estudio de estas cachés necesita de tráfico real en la red. Aunque existen muestras de tráfico real de carácter público del tráfico Web que soportan determinados *proxies* de Internet [1], no es así para redes inalámbricas. Es aquí donde surge la necesidad de la creación de un generador de tráfico sintético que sirva para evaluar esquemas de caché tanto en redes cableadas como inalámbricas. La ventaja de usar un generador de tráfico sintético radica en la posibilidad de ir modificando las diferentes características del tráfico para así analizar el comportamiento de las cachés.

Actualmente, existen diferentes simuladores de tráfico sintético como ProWGen [2], TrGen [3] y GenSyn [4]. Estos simuladores son capaces de crear muestras de tráfico sintético a partir de la introducción de una serie de parámetros o bien mediante la introducción de muestras de tráfico real, que son analizadas y posteriormente se simula su comportamiento. Sin embargo, solo ProWGen es capaz de generar tráfico sintético para evaluar el rendimiento de cachés, ya que es el único que genera tráfico a nivel de documentos Web. Desafortunadamente, ProWGen no es capaz de distinguir el tipo de los documentos que generan sus muestras. Por lo tanto, el objetivo del presente trabajo es la creación de un generador de muestras de tráfico telemático a nivel de documentos Web que cumpla todas las características que se consideran necesarias para dicho tipo de tráfico. El generador a implementar será capaz, además, de diferenciar los tipos de los documentos.

El resto de este artículo está estructurado como sigue: en el capítulo II se analizan los fundamentos matemáticos de las características del tráfico a generar y se especifica la forma

de implementar dichas características; en el capítulo III se valida el generador implementado comprobando que las muestras sintéticas generadas coinciden con lo esperado; en el capítulo IV se describe el entorno de usuario de la aplicación generada; finalmente en el capítulo V se comentan las principales conclusiones del presente trabajo.

II. IMPLEMENTACIÓN

Se ha creado una herramienta sobre el entorno MATLAB que tiene como objetivo la generación de muestras de tráfico para su posterior procesado en simuladores de cachés. El resultado de la aplicación es la creación de ficheros donde se especifican estas muestras de tráfico que serán las que utilicen posteriormente dichos simuladores.

La implementación de la aplicación se divide en dos partes, la primera es la generación en sí de las muestras de tráfico a partir de una serie de parámetros, y la segunda es la interfaz gráfica con el usuario donde se interactúa en la elección de los parámetros deseados.

Existen dos posibles enfoques para sintetizar muestras de tráfico [5]. La primera de ellas consiste en muestrear o permutar muestras de tráfico reales reordenando las peticiones de modo que se genere una nueva carga de tráfico diferente a partir de la carga de tráfico original. A este enfoque se le denomina *basado en traza real*. La segunda aproximación, denominada *aproximación analítica*, usa modelos matemáticos para modelar las características más interesantes del tráfico, y utiliza la generación de números aleatorios para producir tráfico que estadísticamente coincide con dichos modelos. Dado que la aproximación analítica ofrece la flexibilidad de modificar los parámetros de los modelos matemáticos para regular las características del tráfico a generar, éste ha sido el método empleado.

El generador de tráfico incorpora cinco características de tráfico seleccionadas, las cuales han sido identificadas como importantes en estudios previos de muestras de tráfico en servidores [2] [5] [6] [7]: los documentos referenciados una sola vez, la popularidad de los documentos, la distribución del tamaño de los documentos, la correlación entre el tamaño de los documentos y su popularidad y, finalmente, la localidad temporal.

A. Documentos referenciados una sola vez

Muchos estudios de tráfico Web en servidores y proxy han mostrado que gran parte de las peticiones hacia un servidor o un *proxy* son realizadas una sola vez, independientemente de la duración del acceso [8] [9]. A este tipo de documentos se les denomina *one-timers*. Es evidente, que no tiene ningún beneficio almacenar en caché este tipo de documentos puesto que nunca más van a ser requeridos. De hecho, sería deseable la existencia de algoritmos ubicados en las cachés para discriminar a estos documentos de modo que no inunden la caché reduciendo así su efectividad.

El enfoque utilizado para modelar las referencias a documentos *one-timers* es determinar cuántos de los documentos distintos en la traza de tráfico a crear deberían ser *one-timers*. Utilizando el porcentaje de *one-timers* como un parámetro, se permite al usuario especificar el valor deseado. Una vez especificado, se puede hallar con facilidad el número de *one-timers*, con lo que las referencias a estos documentos se fijan a uno.

B. Popularidad

Una característica común en el tráfico Web es la irregular distribución de las referencias a los distintos documentos [10] [11]. En muchos casos se aplica la ley de Zipf [12] para modelar la popularidad de los documentos [13] [14] [15]. La ley de Zipf expresa una relación exponencial entre la popularidad P (número de veces que se repite la petición de dicho documento) y su ranking r (que se extrae de la ordenación de los documentos según su popularidad). Esta relación se expresa en la ecuación (1):

$$P = \frac{c}{r^{\alpha}} \tag{1}$$

donde c es una constante y α es un valor entre cero y uno.

Algunos investigadores han encontrado que el valor de α es cercano a la unidad [13] [15], precisamente siguiendo la ley de Zipf. Otros autores [2] [4] [14] han encontrado que el valor de α es menor que la unidad, y que la distribución puede ser descrita "como la de Zipf" con el valor de α variando dependiendo del tráfico. Este comportamiento típico puede describirse como una línea recta de pendiente negativa α si representamos en unos ejes logarítmicos P frente a r. Este ajuste lineal suele ser casi perfecto para la parte principal del cuerpo de la distribución. Sin embargo, suele desajustarse tanto para los elementos más populares como para los menos populares (debido a los *one-timers*) [6].

Para determinar la popularidad de los restantes documentos, el primer paso es calcular la constante de proporcionalidad c de la fórmula de Zipf en la ecuación (1) usando para ello los parámetros conocidos como el número de referencias (N), número de documentos distintos (a), y el número de *one-timers* (b). Puesto que todos los *one-timers* tienen un número de referencias igual a uno, el ranking r del primer *one-timer* es $a–b+1$, y el ranking del último *one-timer* es a, la constante c puede ser calculada usando el *one-timer* situado en el centro del rango de los *one-timers*, $r=a–(b/2)$ y con popularidad $P=1$. Sustituyendo estos valores en la fórmula de Zipf, obtenemos la ecuación (2).

$$1 = \frac{c}{(a - b/2)^{\alpha}} \tag{2}$$

Despejando c de la ecuación (2) obtenemos la ecuación (3).

$$c = (a - b/2)^{\alpha} \tag{3}$$

Una vez que se ha calculado el valor de c, la popularidad de cada uno de los documentos que faltan por hallar (aquellos que ocupan el ranking desde uno hasta a-b), puede ser calculada fácilmente directamente de la fórmula de Zipf.

Desafortunadamente, después de determinar la popularidad de cada uno de los documentos como se ha indicado anteriormente, la suma de las popularidades de todos los documentos no siempre coincide exactamente con N (que es el número total de peticiones a generar). Hay dos razones principales para ello: el cálculo de la constante c no es exacto, y que todas las popularidades han sido redondeadas al entero más próximo. Para resolver este problema, se ha añadido un paso adicional para ajustar más la popularidad de cada documento. Para ello se ha escalado la

popularidad de los no *one-timers* (aquellos que ocupan los puestos de popularidad del uno al *a–b*), manteniendo en todo momento la pendiente *α*. El factor de escala *s* se determina usando la ecuación (4).

$$s = (N-b) \Big/ \sum_{i=1}^{a-b} P_i \tag{4}$$

Siguiendo con estos cálculos, el número total de peticiones es muy cercano a *N* (típicamente dentro del 1-5% y casi siempre dentro del 10%).

C. Distribución del tamaño de los documentos

Estudios previos de tráfico Web [6] [9] han demostrado que la distribución de los tamaños de los documentos de tráfico Web es de cola pesada. Este tipo de distribuciones implica que relativamente pocos documentos de tamaño muy grande acumulan un porcentaje elevado del volumen total de datos del tráfico Web. Por tanto, la distribución del tamaño de los documentos afecta significativamente en el diseño de estrategias de caché. Almacenando en caché sólo documentos pequeños se puede reducir el número de peticiones enviadas a los servidores de origen, y puede convertirse en una tasa elevada de aciertos de documentos en caché, pero también se traduce en un bajo número de tasa de aciertos por byte. En el otro extremo, almacenando en caché documentos de gran tamaño provoca una elevada tasa de acierto por byte a costa de la tasa de aciertos por documentos.

Para una evaluación efectiva de la gestión de las estrategias en caché, la distribución de cola pesada de los tamaños de documentos debe ser incorporada en la generación de muestras de tráfico, en particular, la "pesadez" de la cola debe ser ajustable de modo que su impacto en las cachés pueda ser evaluado.

En la aplicación desarrollada, el modelado de la distribución del tamaño de los documentos se ha dividido en tres partes:

1. Se modela la cola de la distribución usando una distribución de Pareto.
2. Se modela el cuerpo de la distribución usando una distribución logarítmica normal.
3. Se unen ambas distribuciones.

La distribución de Pareto de doble exponencial es un ejemplo de distribución de cola pesada y ha sido la utilizada para modelar la cola de la distribución del tamaño de los documentos. Su función densidad de probabilidad se muestra en la ecuación (5) y su función de distribución acumulada en la ecuación (6).

$$p(x) = \alpha \cdot k^{\alpha} \cdot x^{-\alpha-1} \qquad \alpha, k > 0,\ x \geq k \tag{5}$$

$$F(x) = P(X \leq x) = 1 - \left(k/x\right)^{\alpha} \tag{6}$$

El parámetro *α*, llamado índice de cola, determina la "pesadez" de la cola de la distribución. La distribución tiene varianza infinita y si $\alpha \leq 1$, entonces la distribución tiene media infinita. Esto implica que pequeños valores de *α* representan colas pesadas (la mayoría del volumen estaría presente en la cola de la distribución). *k* es un parámetro que determina dónde empieza la cola de la distribución al representa el valor más pequeño posible que puede tomar la variable aleatoria de la distribución de cola pesada.

Los parámetros *α* y *k* que caracterizan la distribución del tamaño de los documentos pueden ser determinados usando una representación logarítmica de la distribución complementaria [4] [9] [15]. Si trabajamos con la función de distribución acumulada complementaria, se puede comprobar que se verifica la ecuación (7).

$$\frac{d \log \overline{F(x)}}{d \log x} = -\alpha, \quad \alpha > k \tag{7}$$

La pendiente *α* puede ser estimada al exhibir un comportamiento lineal. Este método es empleado para estimar *α* en los estudios de tráfico sintético puesto que *α* y *k* son proporcionados como parámetros en el proceso de generación de muestras.

Incorporar el modelo de cola pesada en la aplicación comienza distinguiendo entre los documentos en dos grupos: aquéllos que se encuentran en el cuerpo de la distribución y aquéllos que se encuentran en la cola. El porcentaje de documentos en la cola es especificado como un parámetro de entrada del generador de tráfico.

Para generar el tamaño de los documentos se despeja en la distribución de la cola de Pareto en función de los parámetros de entrada. El objetivo es obtener, para un número aleatorio uniforme *y* entre (0,1), el correspondiente valor de *x*. Despejando en la ecuación (6) obtenemos la ecuación (8).

$$x = k \Big/ \left(1-y\right)^{\frac{1}{\alpha}} \tag{8}$$

Si *y* es una variable aleatoria uniforme (0,1) entonces *1–y* es también un valor aleatorio uniforme (0,1). Por tanto, la ecuación (8) quedaría como se muestra en la ecuación (9).

$$x = k \Big/ \left(y'\right)^{\frac{1}{\alpha}} \quad \text{con } y' = (1-y) \tag{9}$$

Así, una vez que *α* y *k* han sido especificados en la aplicación, los valores de cola pesada pueden ser generados usando la ecuación (9).

Para modelar el cuerpo de la distribución del tamaño de los documentos se usa una distribución logarítmica normal [16]. La distribución logarítmica normal tiene la propiedad que si $X \approx N(\mu, \sigma^2)$, es decir, *X* está normalmente distribuida con media *μ* y varianza σ^2, entonces e^X tiene una distribución logarítmica normal con parámetros *μ* y σ^2, denotado por $LN(\mu,\sigma^2)$. La función de densidad de probabilidad de una distribución logarítmica normal se muestra en a la ecuación (10).

$$p(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{\frac{-(\ln x - \mu)^2}{2\sigma^2}} \tag{10}$$

Aunque la distribución logarítmica normal no tiene una forma cerrada y, por tanto, no tiene una función de distribución acumulada, hay una aproximación sencilla que explota la propiedad anterior de la distribución logarítmica normal para generar variables logarítmicas normales. Así, para generar una variable logarítmica normal es suficiente

con generar una variable aleatoria *x* normal *N(μ,σ2)*, y entonces devolver e^X como una variable logarítmica normal. Sin embargo, μ y σ^2 son la media y la varianza de una distribución normal, por lo que se debe, a partir de dichos valores, obtener la media y la varianza de una distribución logarítmica normal. Esos valores son (11) y (12):

$$\mu_l = e^{\mu+\sigma^2/2} \tag{11}$$

$$\sigma_l^2 = e^{2\mu+\sigma^2}\left(e^{\sigma^2}-1\right) \tag{12}$$

De este modo, si se quiere generar una variable aleatoria logarítmica normal con un determinado μ_l y σ_l^2, primero se deben resolver las ecuaciones (13) y (14).

$$\mu = \ln\left(\frac{\mu_l^2}{\sqrt{\sigma_l^2+\mu_l^2}}\right) \tag{13}$$

$$\sigma^2 = \ln\left[\left(\sigma_l^2+\mu_l^2\right)/\mu_l^2\right] \tag{14}$$

Una vez que se tienen estos valores, se puede generar una variable aleatoria normal $N(\mu,\sigma^2)$, con valores provenientes de una distribución logarítmica normal, y con esa variable aleatoria devolver e^X, con lo que ya se habrá generado una variable aleatoria logarítmica normal de media μ_l y varianza σ_l^2.

Puesto que se modela tanto la cola de la distribución como el cuerpo de la misma de forma separada, se deben tener en cuenta ciertas restricciones necesarias a la hora de unir ambas distribuciones. Primero, los valores que se encuentren en el cuerpo de la distribución no pueden encontrarse también en la cola. Segundo, el caso contrario tampoco debe producirse. Tercero, cuando se represente la función de distribución acumulada, debe producirse una transición suave desde el cuerpo hasta la cola.

Un primer paso para lograr estos objetivos sería limitar superiormente el tamaño de los documentos generados para el cuerpo de la distribución. Es decir, si se genera un valor mayor que *k*, se descarta y se genera un nuevo valor. Los efectos colaterales de esta restricción son:

1. Después de la generación de las variables logarítmicas normales, la media y la desviación típica de los valores resultantes son menores que los especificados en los valores de entrada.
2. La distribución del tamaño de los documentos resultantes puede presentar una obvia discontinuidad donde se unen ambas distribuciones.

Otra posibilidad sería eliminar la restricción de tamaño superior, resolviendo estos dos problemas, pero a cambio, se incrementaría el porcentaje de documentos en la cola de la distribución. En la aplicación se ha optado por la segunda opción, no obstante, en el código del mismo se deja abierta la posibilidad de la primera opción.

D. Correlación entre el tamaño de los documentos y su popularidad

Numerosos estudios de tráfico Web en *proxies* muestras que muchos de los documentos transferidos en la Web tienen un tamaño pequeño [5] [6] [14]. Una cuestión natural que se plantea al respecto es si existe alguna correlación estadística entre la frecuencia de acceso a un determinado documento y su tamaño. Algunos estudios [15] [17] han mostrado que hay una muy pequeña correlación entre su frecuencia de acceso y su tamaño, aunque esta cuestión es todavía tema de debate.

En la aplicación desarrollada, para otorgarla de mayor flexibilidad, se ofrece la posibilidad de que el tráfico generado tenga correlación positiva, negativa o cero entre la popularidad de los documentos y su tamaño. Una correlación positiva implica que los documentos de mayor tamaño tienen mayor popularidad y correlación negativa implica que los documentos de menor tamaño tengan más probabilidad de ser requeridos (mayor popularidad). Mientras que correlación cero no otorga ninguna correlación entre la popularidad de los documentos y su tamaño. El hecho de permitir la existencia o no de correlación radica en la posibilidad de explorar distintos algoritmos de caché según cada una de estas opciones.

Modelar e incorporar estas características de correlación dentro de la generación de muestras de tráfico se realiza en tres etapas:

1. Generar un conjunto de popularidades de documentos usando la aproximación comentada en la sección *B*.
2. Generar un conjunto de tamaños de documentos usando la aproximación de la sección *C*.
3. Usar una técnica de mapeo para introducir correlación positiva, negativa o cero entre la popularidad de los documentos y su tamaño.

El algoritmo a seguir es el siguiente:

1. Se genera una lista *P* de popularidades para *n* documentos distintos usando la aproximación explicada en la sección *B*. Se ordena la lista *P* en orden ascendente.
2. Se genera una lista *S* de tamaños de documentos para *n* documentos distintos usando la aproximación explicada en la sección *C*. Se ordena la lista *S* en orden ascendente.
3. Se calculan los valores de la función de distribución acumulada para los elementos de la lista *P*. Se construye una nueva lista P_{nueva}, con valores de la forma (v_i, p_i), donde v_i es un valor único de popularidad de la lista *P* y p_i representa su probabilidad acumulada.
4. Se calculan los valores de la función de distribución acumulada para los elementos de la lista *S*. Se construye una nueva lista S_{nueva}, con valores de la forma (v_i, p_i), donde v_i es un valor único de tamaño de documento de la lista *S* y p_i representa su probabilidad acumulada.
   a) Se repiten los siguientes pasos n veces para generar una nueva lista L que represente la popularidad y el tamaño de los documentos distintos. La entrada j ($1 \le j \le n$) en L es de la forma (f_j, s_j), donde f_j representa la popularidad y s_j el tamaño del documento. La entrada (f_j,s_j) es determinada de la siguiente forma: Se genera un número aleatorio r_1 dentro una distribución aleatoria uniforme de rango (0, 1).
   b) Se busca en la lista ordenada P_{nueva} el primer elemento i que satisface $r_1 \le p_i$. Si $p_i=r_1$ ó i=1, entonces se hace $f_j=v_i$. En otro caso, se interpola linealmente usando la ecuación (15).

$$f_j = v_{i-1} + \frac{(r_1 - p_{i-1}) \times (v_i - v_{i-1})}{(p_i - p_{i-1})} \quad (15)$$

c) En función de la correlación:
   i. Si se desea correlación positiva, se usa r_1 para buscar en la lista S_{nueva} exactamente del mismo modo que se ha hecho en el paso b) para buscar el valor del tamaño de documento s_j.
   ii. Si se desea correlación negativa, se usa $1\text{-}r_1$ para buscar en la lista S_{nueva} exactamente del mismo modo que se ha hecho en el paso b) para buscar el valor del tamaño de documento s_j.
   iii. Si se desea que no haya ningún tipo de correlación, se genera otro número aleatorio r_2 dentro de una distribución aleatoria uniforme de rango (0, 1) y se emplea para buscar en la lista S_{nueva} exactamente del mismo modo que se ha hecho en el paso b) para buscar el valor del tamaño de documento s_j.

d) Se añade el par de valores (f_j,s_j) a la lista *L* como la popularidad y el tamaño del documento único *j*.

5. La lista *L* ahora representa una nueva lista de *n* documentos distintos, los cuales tienen su propia popularidad y tamaño de documento con el deseado valor de correlación introducido.
6. Se normaliza para ajustar la popularidad de los documentos al número total de peticiones a generar. Pero se va a seguir manteniendo el número de *one-timers* como aquél que se ha introducido como parámetro, eludiendo el hecho de que al ser elegidos aleatoriamente según su función de distribución acumulada saldrían muchos menos *one-timers*, ya que suponen un porcentaje menor sobre la función de distribución acumulada de popularidad.

E. Localidad temporal

La localidad temporal hace referencia a la tendencia de documentos referenciados en el pasado a volver a ser nuevamente referenciados en el futuro. El número de veces que cada documento debería aparecer en las muestras de tráfico (la popularidad) ya se conoce. Sin embargo, dado que una referencia a un documento es generada en un instante de tiempo t_o, no está claro cuándo se va a producir la próxima referencia a dicho documento. Es por esto, que la presencia de la localidad temporal en la generación de tráfico tiene un efecto muy importante para las pruebas con cachés.

La aproximación utilizada para el modelado de la localidad temporal está basada en el modelo de pila finita LRU (*Least Recently Used*). Una pila LRU es una lista de todos los documentos ordenados según hayan sido referenciados recientemente [13], esto es, el último que haya sido referenciado estará en primer lugar de la pila y el que fue referenciado hace más tiempo estará en la última posición. La pila es actualizada dinámicamente cada vez que se procesa una referencia. En muchos casos, esta actualización implica el tener que añadir un nuevo elemento en la cima de la pila empujando el resto hacia abajo, en otros casos, implica extraer un elemento existente en el interior de la pila y trasladarla a la cima de la misma, desplazando al resto de los elementos hacia abajo.

Una pila LRU de tamaño finito es una pila LRU que sólo puede almacenar un número *m* de documentos. Experimentos realizados por Mahanti [6] sugieren que $m=1000$ es un número adecuado para capturar la presencia de localidad temporal en cargas de tráfico Web. No obstante, para la realización de todo tipo de pruebas en cachés, se permite que el tamaño de la pila sea especificado por el usuario como un parámetro.

El aspecto más importante en una pila LRU es que cada posición en la pila tiene asociada una probabilidad de referencia. Las probabilidades son asociadas a la posición de la pila y no a los documentos. Las probabilidades de las posiciones de la pila pueden ser proporcionadas por modelos u obtenidas de analizar muestras de tráfico reales.

Por ejemplo, supongamos que las popularidades de los distintos documentos en la carga de tráfico están representadas por la ecuación (16):

$$D = \{x_1,\ x_2,\ \ldots.,\ x_n\} \qquad x_1 \geq x_2 \geq \ldots \geq x_n \quad (16)$$

Entonces, las probabilidades a_i pueden ser calculadas para cada documento *i* usando la ecuación (17):

$$a_i = x_i \Big/ \sum_{j=1}^{n} x_j \quad \text{para } i = 1,\ 2,\ \ldots.,\ n \quad (17)$$

Hay dos formas posibles de utilizar esas probabilidades: estática y dinámica. En la aproximación estática, si la pila finita tiene un tamaño fijo *m* de modo que $m \leq n$, la probabilidad acumulada y_i, se calcula como se muestra en (18):

$$y_i = \sum_{j=1}^{i} a_j \quad (18)$$

La probabilidad acumulada calculada para cada posición de la pila es asignada al comienzo de la generación de muestras y no puede ser cambiada durante dicho proceso. Esta técnica genera una localidad temporal estadística homogénea para todos los documentos en la traza.

En la aproximación dinámica, cada vez que la pila LRU es modificada (ya sea al mover un documento desde otra posición de la pila a la cima de la misma o bien por traer un nuevo documento a la pila), las probabilidades acumuladas para cada posición de la pila son recalculadas usando los valores a_i de los documentos que actualmente ocupan cada posición de la pila. Esta aproximación puede modelar propiedades de localidad temporal heterogéneas. Por ejemplo, si en el instante temporal t_o, las dos primeras posiciones de la pila contienen los documentos con identificadores 1 y 2 respectivamente, entonces sus estados pueden ser representados como $St_o=\{a_1,a_2\}$, donde a_1 es la probabilidad de referenciar el documento 1 y a_2 es la probabilidad de referenciar el documento 2. Por lo tanto, la probabilidad acumulada de referenciar esas dos posiciones de la pila son a_1 y a_1+a_2, respectivamente. Asumiendo que en el instante t_1, el documento en la posición dos de la pila es movido a la posición uno, esta acción provocará que haya que recalcular las probabilidades acumuladas asociadas a cada posición de la pila. En ese caso, los estados de la pila se convierten en $St_1=\{a_2,a_1\}$. Así, las probabilidades

acumuladas para referenciar estas dos posiciones de la pila son ahora a_2 y a_2+a_1, respectivamente.

El proceso de generación de referencias comienza con la pila LRU vacía. Si se elige la aproximación estática, cada posición de la pila tiene asociada una probabilidad acumulada de referenciar a esa posición. En el caso de la aproximación dinámica, las probabilidades están sin inicializar. El proceso de generación de referencias comienza generando un número aleatorio x_i de una distribución aleatoria uniforme de rango (0,1). Después se debe comprobar si ese documento seleccionado está en la pila o no (si $x_i \leq y_i$). Pueden producirse tres circunstancias:

a) La pila está vacía.
b) La pila no está vacía pero el próximo documento a referenciar no está en la pila.
c) La pila no está vacía y el próximo documento a referenciar ya está en la pila.

En los casos a) y b), se selecciona un documento al azar del conjunto de documentos distintos que quedan aún por referenciar. Se genera una referencia para el documento seleccionado y el contador de referencias que le faltan por generar (es decir, su popularidad) es decrementada en una unidad. Si no quedan más referencias por generar de este documento seleccionado, entonces el documento es eliminado del conjunto de documentos distintos que quedan por referenciar. En caso contrario, el documento es movido hasta la cima de la pila, desplazando a los otros documentos hacia abajo en la pila si es el caso b). En el caso c), se busca desde el comienzo de la pila al mayor elemento de la misma que verifique $x_i \leq y_i$. Una vez encontrado se generará una referencia para ese documento y se decrementará en una unidad el contador de referencias que le quedan por generar (popularidad). Si ya no le quedan más referencias por generar, el documento es eliminado de la pila y todos los documentos situados debajo de él suben una posición en la pila. En otro caso, el documento es movido a la cima de la pila, desplazando el resto de elementos, si los hubiera, una posición hacia abajo.

En el modelo de pila dinámica LRU, cada uno de estos casos provocará nuevos cálculos de las probabilidades acumuladas asociadas con esas posiciones de la pila.

F. Tipos de documentos

Los documentos existentes en la Web se clasifican en función de su tipo en: aplicaciones, audio, imágenes, mensajes, texto y video. La aplicación desarrollada permite seleccionar el porcentaje de cada tipo de documentos que se desean generar. Además, permite la generación de muestras de tráfico para cada uno de los tipos de documentos especificando todos los parámetros anteriormente mencionados y posteriormente mezclar las muestras generadas para generar una muestra conjunta.

III. VALIDACIÓN

La aplicación desarrollada requiere de doce parámetros de entrada para generar muestras de tráfico. Según diversos estudios de las características del tráfico Web [5][6][8], los parámetros para la generación de tráfico más usuales son los representados en la Tabla 1. Además, hay que especificar el número total de peticiones que se desea generar y la correlación existente entre el tamaño de los documentos y su frecuencia de acceso.

Se ha generado una muestra de tráfico con las características de tráfico presentes en la Tabla 1, que es la comentada en el resto del epígrafe. Además, se han generado múltiples muestras modificando cada uno de los parámetros por separado y se ha comprobado que, para todos los casos, las características de la muestra generada coinciden con las esperadas.

Parámetro	Valor
Porcentaje de documentos distintos	30%
Porcentaje de *one-timers*	70%
Pendiente de Zipf	0.75
Índice de cola-pesada	1.2
Comienzo de la cola en bytes, k	10000
Porcentaje de documentos en la cola pesada	20%
Media de la distribución logarítmica normal (μ)	7000
Varianza de la distribución logarítmica normal (σ)	11000
Modo de la pila para la localidad temporal	Dinámico
Tamaño de pila para la localidad temporal	1000
Correlación entre tamaño y popularidad	0

Tabla 1. Parámetros usuales para la generación de tráfico

En la gráfica mostrada en la Fig. 1 se puede verificar que las muestras de tráfico generadas tienen un comportamiento "como el de Zipf", esto es, al trazar en ejes logarítmicos, la popularidad de los documentos frente a su ranking producido al ordenar dicha popularidad, se observa como sigue una distribución lineal con una pendiente igual a -0.76045, mientras el caso ideal sería -0.75. Mencionar que tanto la muestra generada como la muestra ideal están casi superpuestas y es sólo en los valores extremos donde se puede apreciar una mínima diferencia.

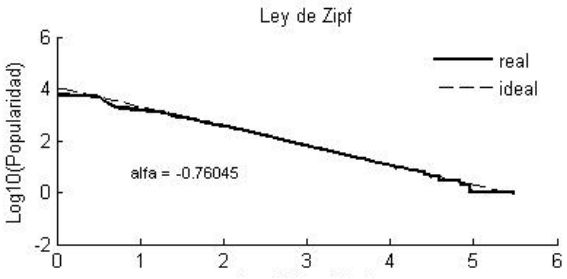


Fig. 1. Análisis del comportamiento de la popularidad

En la Fig. 2 se aprecia el comportamiento de la correlación entre el tamaño de los documentos y su frecuencia de acceso. En ella se observa como no hay ningún tamaño de documento que predomine sobre el resto, es decir, todos los tamaños de documento tienen el mismo peso. El valor de correlación obtenido es de 0.00029 pero el resultado que se expone es 0, puesto que sólo se distingue entre -1, 0 y 1. Por tanto, el resultado coincide con el valor de correlación de entrada.

La Fig. 3 representa la función de distribución acumulada de los tamaños de los documentos en el cuerpo de la distribución, es decir, si el valor de k es de 10.000 bytes, Es decir, cómo se distribuyen los tamaños de los documentos

para valores menores que *k*. Es quizás, la característica que más se aleja al valor ideal. Se explica en el amplio rango de valores aleatorios que se pueden generar y en los errores de aproximación de los modelos matemáticos utilizados tanto para la generación de muestras como para su posterior análisis. Puesto que para la generación de muestras se debe llegar a un compromiso entre todas las variables de entrada, se ha considerado que sea en el tamaño de los documentos donde recaiga el mayor margen de error. No obstante, sí que se puede apreciar que el comportamiento de las distribuciones de tamaño de las muestras son bastantes similares.

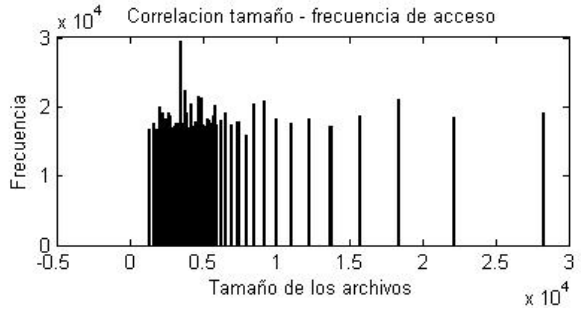


Fig. 2. Correlación entre el tamaño y la frecuencia de acceso

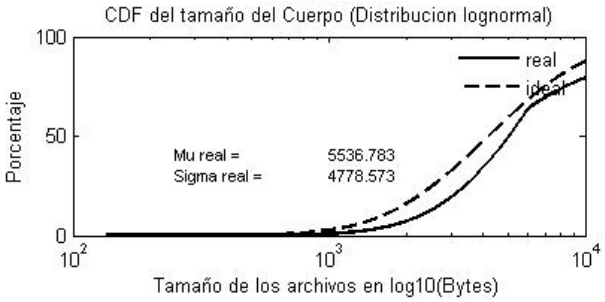


Fig. 3. Análisis de la función de distribución acumulada

En la Fig. 4, se muestra la función de densidad de probabilidad de los tamaños de los documentos tanto en el cuerpo como en la cola de la distribución. Esta gráfica nos proporciona una idea muy intuitiva del tamaño de los documentos.

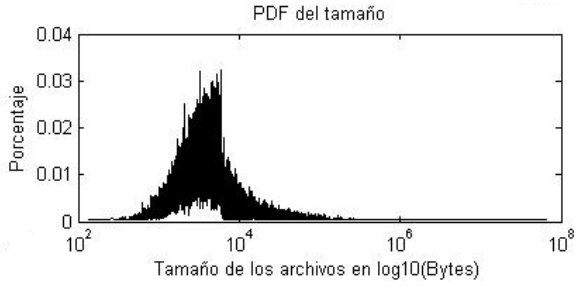


Fig. 4. Función de densidad de probabilidad de los tamaños

En la Fig. 5 se muestra la pendiente de la cola de Pareto, tanto para la muestra obtenida como para la muestra ideal, que se pasa como parámetro de entrada. El valor de inicio del comportamiento de cola de Pareto lo marca el valor de *k*. Se puede observar que ambos conjuntos de muestras son prácticamente coincidentes.

La Fig. 6 representa la localidad temporal presente en las referencias a los documentos. Es decir, cuando se produce una referencia a un documento es relativamente más probable que en un instante corto de tiempo se vuelva a producir una nueva referencia a dicho documento. Éste fenómeno queda patente en la mencionada gráfica.

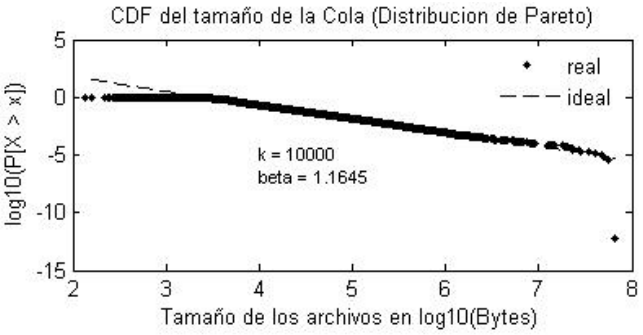


Fig. 5. Análisis de la pendiente de cola de Pareto

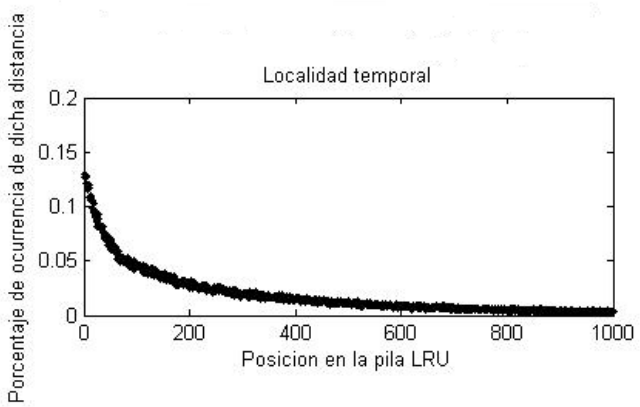


Fig. 6. Análisis de la localidad temporal en modo dinámico

IV. El entorno de usuario

A la herramienta de generación de tráfico desarrollada se le ha provisto de un entorno gráfico para facilitar la configuración de los parámetros de generación del tráfico. La ventana principal de la aplicación puede verse en la Fig. 7.

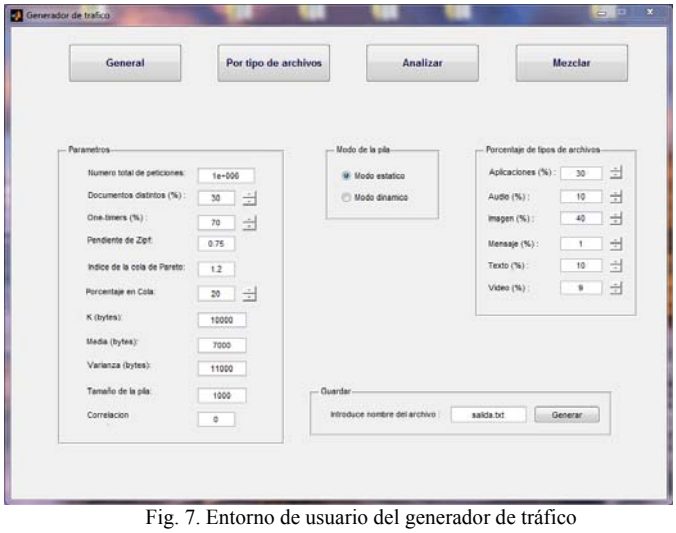


Fig. 7. Entorno de usuario del generador de tráfico

La aplicación se ha dividido en cuatro secciones a las que se accede mediante los botones situados en la parte superior de la ventana principal:

- General: En esta sección se introducen todos los parámetros necesarios para la generación de las muestras de tráfico sintético comentadas en apartados

previos. Además se especifica el nombre del archivo en el que se generará la muestra.

- Por tipos de archivos: Esta sección es muy similar a la anterior, la principal diferencia es que antes de introducir los parámetros para la generación de las muestras de tráfico, se deberá indicar de qué tipo de documentos serán las muestras que se van a generar. Una vez elegido el tipo de documento, se introducirán los parámetros necesarios y el nombre del archivo a generar.
- Analizar: Esta sección permite verificar el correcto funcionamiento de la aplicación, ya que muestra información relativa a las muestras que contiene el archivo seleccionado, como la Ley de Zipf, la función de probabilidad acumulada del cuerpo de la distribución de tamaño, la función de probabilidad acumulada de la cola de la distribución de tamaño, la correlación entre el tamaño de los documentos y su frecuencia de aparición, la función de probabilidad de los tamaños de los documentos y la localidad temporal de las repeticiones de las peticiones de los documentos comparadas con sus respectivas gráficas ideales.
- Mezclar: Esta sección permite mezclar muestras generadas en distintos archivos. A priori, se realiza una mezcla aleatoria uniforme, dejando abierta la posibilidad de incluir cualquier otro tipo de distribución para dicha mezcla.

Una vez que se han introducido todos los parámetros y se ejecuta la generación de muestras, la aplicación genera un archivo con el nombre indicado por el usuario. Este archivo consta de una lista en la que cada fila hace referencia a un documento. Cada fila del fichero tiene tres campos: el primer campo es el identificador de archivo, especificado mediante un número para facilitar su procesamiento; el segundo campo indica el tamaño del archivo en bytes; y el tercer campo indica el tipo de archivo generado: aplicación, audio, imágenes, mensajes, texto y video.

V. CONCLUSIONES

El presente artículo describe una aplicación desarrollada en Matlab para la generación de muestras sintéticas de tráfico para la evaluación del rendimiento de redes cableadas o inalámbricas con caché. La aplicación permite configurar los parámetros típicos a tener en cuenta en una caché Web, como son los documentos referenciados una única vez (*one-timers*), la popularidad de los documentos, la distribución del tamaño de los documentos, la correlación entre el tamaño de los documentos y su popularidad, la localidad temporal y los tipos de los documentos. Para cada una de las mencionadas características se ha especificado la forma en que han sido implementadas en la aplicación, comentando además los fundamentos matemáticos de las mismas.

La aplicación ha sido validada generando muestras de tráfico con diferentes características y comprobando que los valores generados coinciden con los ideales.

Finalmente, se ha procurado crear un entorno de usuario que permite, de forma sencilla e intuitiva, la generación de las muestras sintéticas de tráfico.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado mediante el proyecto TEC2009-13763-C02-01

REFERENCIAS

[1] Proyecto IRCaché: http://www.ircache.net/
[2] M.Busari, C. Williamson, “ProWGen: A Synthetic Workload Generation Tool for the Simulation Evaluation of Proxy Caches”, Computer Networks, pp. 779-794. Jun. 2002.
[3] F.J. Ridruejo, A. Gonzalez, J. Miguel-Alonso. "TrGen: a Traffic Generation System for Interconnection Network Simulators", International Conference on Parallel Processing, 2005. 1st. Int. Workshop on Performance Evaluation of Networks for Parallel, Cluster and Grid Computing Systems (PEN-PCGCS'05). ICPP 2005 Workshops. 14-17 Junio 2005.
[4] Poul E. Heegaard, “GenSyn - a Java based generator of synthetic Internet traffic linking user behaviour models to real network protocols”, Presentation at ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management, Monterey (EEUU), Septiembre 2000.
[5] G. Abdulla, E. Fox, M. Abrams, y S. Williams, “WWW Proxy Traffic Characterization with Application to Caching” Technical Report TR-97-03, Computer Science Department, Virginia Tech., Marzo 1997.
[6] A. Mahanti y C. Williamson, “Web Proxy Workload Characterization” Technical Report, Department of Computer Science, University of Saskatchewan, Feb. 1999.
[7] P. Badford y M. Crovella, “Generating Representative Web Workloads for Network and Server Performance Evaluation”, 1998 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, pp.151-160, Julio 1998.
[8] A. Mahanti, Web Proxy Workload Characterization and Modelling, M.Sc. Thesis, Department of Computer Science, University of Saskatchewan, Septiembre 1999.
[9] M. Arlitt and C. Williamson, “Internet Web Servers: Workload Characterization and Performance Implications”, IEEE/ACM Transaction on Networking, vol. 5, no 5, pp. 631-645, Oct. 1997.
[10] M. Arlitt, L. Cherkasova, J. Dilley, R. Friedrich, y Tai Jin, “Evaluating Content Management Techniques for Web Proxy Caches”, 2nd Workshop on Internet Server Performance, Atlanta, Georgia, Mayo 1999.
[11] C. Roadknight, I. Marshall y D. Vearer, “File Popularity Characterization”, 2nd Workshop on Internet Server Performance (WISP 99), Atlanta, Georgia, Mayo 1999.
[12] Referencias a la ley de Zipf: http://www.nslij-genetics.org/wli/zipf/
[13] V. Almeida, A. Bestavros, M. Crovella, y A. Oliveira, “Characterizing Reference Locality in the WWW”, in Proceedings of the 1996 International Conference on Parallel and Distributed Information Systems (PDIS 96), pp. 92-103, Dic. 1996.
[14] L. Breslau, P. Cao, L. Fan, G. Phillips, y S. Shenker, “Web Caching and Zipf-like Distributions : Evidence and Implications”, IEEE Infocom ’99 Conference, New York, NY, Marzo 1999.
[15] M. Crovella and A. Bestavros, “Self-Similarity in World Wide Web Traffic: Evidence and Posible Causes”, IEEE/ACM Transactions on Networking, vol. 5, no.6, pp.835-846, Dic. 1997.
[16] A. Law y W. Kelton, “Simulation Modeling and Analysis”, Segunda Edición, Ed. Mc-Graw-Hill, 1991
[17] A. Mahanti, C. Williamson, y D. Eager, “Traffic Analysis of a Web Proxy Caching Hierarchy”, IEEE Network, vol. 14, no. 3, pp. 16-23, Mayo/Junio 2000.

Sesión de posters

Arquitecturas de generación de contenido colaborativo para sistemas basados en realidad aumentada móvil

Daniel Gallego Vico, Iván Martínez Toro y Joaquín Salvachúa Rodríguez.
Departamento de Ingeniería de Sistemas Telemáticos
Escuela Técnica Superior de Ingenieros de Telecomunicación, Universidad Politécnica de Madrid
Avenida Complutense nº 30, "Ciudad Universitaria", 28040, Madrid, España
{dgallego, imartinez, jsalvachua}@dit.upm.es

***Resumen*- La evolución actual de los terminales móviles ha propiciado el surgimiento de un nuevo campo de investigación relacionado con las aplicaciones móviles colaborativas basadas en Realidad Aumentada. Debido a su inmadurez, es necesaria una conceptualización de términos que aclaren un entorno hasta el momento complejo y poco estructurado. Este artículo propone una nueva taxonomía llamada "Pirámide de Generación de Contenido Colaborativo" que clasifica este tipo de aplicaciones en tres niveles: aisladas, sociales y en tiempo real. Dicha clasificación describe las diferentes arquitecturas que se deben tener en cuenta para conseguir sistemas de cada uno de estos niveles, teniendo en cuenta la forma en que el contenido aumentado es generado y cómo se lleva a cabo la colaboración. Por tanto, el principal objetivo es clarificar terminología relativa a este nuevo paradigma, a la vez que se propone un marco para identificar y clasificar futuras investigaciones relativas a este entorno.**

***Palabras Clave*- Realidad Aumentada, Móvil, Colaboración, Arquitecturas, Localización, Generación de contenido, Taxonomía, Experiencia de usuario.**

I. INTRODUCCIÓN

En los últimos años, el mundo móvil ha experimentado una evolución increíblemente rápida potenciada aún más gracias al surgimiento de los *smartphones*. La disponibilidad de grandes capacidades técnicas tanto en las infraestructuras de red como en los terminales, unido al esfuerzo puesto en integrar grandes redes sociales como Facebook o Twitter, ha permitido la creación de lo que se ha venido a llamar colaboración móvil [1].

Por otro lado, es importante destacar que recientemente hemos contemplado el renacimiento de la Realidad Aumentada (RA) hasta convertirse en un tema de actualidad [2] gracias en gran medida a la evolución de los terminales móviles. De esta forma, las propiedades que un sistema basado en RA debe cumplir según Azuma et al. [3], están soportadas perfectamente por los actuales dispositivos que presentan potencias de cómputo y capacidades multimedia acordes a los requisitos. En consecuencia, y como se ha visto en los últimos años en el mercado móvil, los desarrolladores han encontrado que la combinación entre las técnicas asociadas al área de la RA y los dispositivos móviles ha propiciado un fértil campo de investigación y desarrollo donde experimentar y crear innovadoras aplicaciones. Además, desde el punto de vista de los usuarios, la inclusión de la RA en el entorno móvil ha cambiado radicalmente la forma en que interactuamos con el mundo, ya que a través de estos sistemas estamos enriqueciendo la realidad con un contenido aumentado o virtual que nos provee de nuevos tipos de información y modelos de visualización, modificando por tanto la experiencia de usuario final.

Por estas razones, el presente artículo está enfocado en la colaboración móvil basada en RA (una unión consecuencia directa de los avances en ambos campos) que se apoya en teléfonos móviles que no utilizan tecnologías marcadoras, ropa inteligente o cualquier otro tipo de tecnología que usualmente no está presente en un s*martphone*. Se ha elegido este enfoque ya que la aceptación de los teléfonos móviles es un hito ya conseguido hace años, mientras que el de tecnologías más experimentales de *tracking* se encuentra en un estado más exploratorio que real, y por tanto lo expuesto en estas líneas estará dirigido a terminales al alcance de toda la sociedad, que cumplen unos criterios de comodidad, transparencia óptica, precio medio y atractivo acordes al uso de teléfonos móviles tal y como Feiner expone en [4].

Estudiar este tipo de sistemas y las arquitecturas relativas a ellos, así como los diferentes parámetros que se ven involucrados será por tanto el objetivo de este artículo, en el que presentaremos una taxonomía denominada *Pirámide de Generación de Contenido Colaborativo* que mediante tres niveles (Aislado, Social y Real) nos permitirá clasificar las diferentes aplicaciones móviles colaborativas basadas en RA.

Por tanto, gracias a esta taxonomía veremos como analizar y reconocer las ventajas y desventajas de cada una de las diferentes arquitecturas, además de permitirnos clasificar las aplicaciones que actualmente podemos encontrar en el mercado móvil.

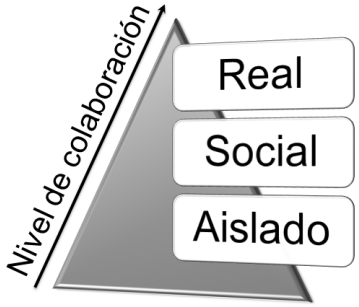


Fig. 1. Pirámide de Generación de Contenido Colaborativo.

ISBN: 978-84-694-5948-5

II. Pirámide de Generación de Contenido Colaborativo

La RA es hoy en día un área de investigación y desarrollo muy activa, y debido a su rápido crecimiento y relativa inmadurez sufre la falta de cierta conceptualización. Es por ello que varias de sus interesantes propiedades no son recogidas en ninguna taxonomía existente, como por ejemplo las diferencias entre aplicaciones en cuanto a generación de contenido. Las definiciones, así como las taxonomías, son herramientas esenciales que llevan a la optimización de los procesos de investigación y desarrollo. Una muestra de esta necesidad es el esfuerzo que está llevando a cabo el *World Wide Web Consortium* (W3C) para definir estándares que gestionen los recursos involucrados en la RA, como pueden ser la geolocalización [5] o el acceso a dispositivos como la cámara [6].

Durante el estudio del entorno de la RA y las aplicaciones que coexisten en el mismo, hemos apreciado la existencia de una característica concreta que difiere entre distintas aplicaciones. Esta característica define la manera en que se genera el contenido a ser mostrado en forma de capas de RA. La capacidad de colaboración permitida por cada aplicación está directamente relacionada con esta característica.

Para cubrir la falta de una taxonomía general que represente la característica anteriormente introducida, proponemos una nueva categorización: la Pirámide de Generación de Contenido Colaborativo (ilustrada en la Figura 1). Esta taxonomía clasifica el entorno de las aplicaciones móviles colaborativas basadas en RA tomando como criterio principal la característica explicada anteriormente, pero también tiene en cuenta otros conceptos estrictamente relacionados como la tecnología y la arquitectura necesarias, y el impacto percibido por el usuario final.

Además, la estructura piramidal de la clasificación proviene del hecho de que cada nivel es soportado por la arquitectura y tecnología de los niveles inferiores. Luego, siguiendo estas ideas hemos definido tres niveles diferenciados (Aislado, Social y Real), que serán explicados en detalle en las siguientes secciones aportando ejemplos de aplicaciones reales en Apple Store o en Android Market para validar su uso como herramienta de clasificación.

A. Primer nivel: colaboración aislada

Este nivel incluye cualquier aplicación que utilice una gestión y generación de contenido centralizada. Esto es, cualquier aplicación en la que la información que formará las diferentes capas sobre el mundo real es creada o recopilada por el equipo que soporta la propia aplicación y almacenada y gestionada únicamente en los servidores de la misma. No hay ningún tipo de contribución por parte del usuario final al repositorio de contenidos, siendo el usuario únicamente un consumidor de información, sin participar en la generación ni la mejora de la misma. Como resultado se obtiene que la colaboración en este nivel es mínima.

Sin embargo, este es un nivel que es importante estudiar no solo por la existencia de un gran número de aplicaciones que concuerdan con sus características, sino también por ser la base tecnológica y arquitectural del resto de niveles superiores.

Fig. 2. Arquitectura general del nivel de colaboración aislado.

La arquitectura que soporta este tipo de sistemas es en la mayoría de los casos similar a la mostrada en la Figura 2. La parte principal del esquema es el servidor de la aplicación, que contiene todo el contenido a ser mostrado en forma de capas de información. Denominamos este nivel Aislado debido a que el servidor no se relaciona con otros servicios, siendo por tanto la única fuente de información para la parte cliente. El cliente suele ser un *smartphone* que cuenta con diferentes dispositivos de captura de contexto físico como la cámara, la brújula o el acelerómetro, y que es capaz de acceder a recursos externos como el GPS. Diversas aplicaciones se valen de distintos subconjuntos de estos dispositivos para generar las capas de información de RA.

Las aplicaciones pertenecientes a este nivel están principalmente basadas en información sobre el contexto del usuario final. Utilizan información del entorno del usuario para mejorar los mecanismos de la generación de RA, optimizar su rendimiento [7], y estar alerta de todo lo que ocurre en el entorno para reaccionar de forma acorde al contexto del usuario [8].

Como ejemplo de aplicaciones de este nivel, hemos seleccionado dos: Nearest Tube y Theodolite, ambas completamente basadas en técnicas de captación de contexto y en las que sólo se utilizan los servidores de la propia aplicación para generar las capas de RA.

B. Segundo nivel: colaboración social

El nivel Social está localizado en la parte media de la pirámide y se refiere a aquellas aplicaciones que presentan capas de información generadas a partir de diferentes fuentes de contenido. No sólo grandes medios como Wikipedia, o la propia aplicación comparten sus contenidos, sino también los propios usuarios finales generan y comparten información a través de redes sociales (Facebook, Twitter, Blogger, etc.) o subiéndola directamente al servidor de la aplicación. Cada usuario es capaz de seleccionar un Punto De Interés (PDI) y adjuntar cierta información al mismo, que será almacenada y mostrada al resto de usuarios. El entorno social de cada usuario colabora de forma dinámica para crear contenido, lo que se traduce en generación de contexto social que acompaña al físico ya obtenido en el nivel anterior.

Fig. 3. Arquitectura general del nivel de colaboración social.

En este caso la arquitectura es más compleja que en el nivel inferior ya que ha de soportar la colaboración entre diferentes participantes, tal y como puede observarse en la Figura 3. El servidor de aplicación tiene que permitir la adición dinámica de contenido, mientras que la parte cliente es similar a la descrita para el nivel Aislado, con la capacidad extra de subir contenido al servidor y de aceptarlo de diversas fuentes.

Se deduce de la descripción anterior que la colaboración móvil en RA comienza en este nivel. El potencial de las redes sociales combinado con el de la RA y las características de los terminales móviles, permiten la creación de aplicaciones verdaderamente colaborativas en las que el usuario final es quien crea la información más interesante. Otros usuarios consumirán dicha información en forma de capas sobre el mundo real. Este nivel también genera información de contexto social que puede ser utilizado para hacer posible un abanico de aplicaciones colaborativas de distintos tipos [9].

La mayoría de las aplicaciones que están siendo desarrolladas y puestas en producción hoy en día se enmarcan dentro de este nivel de la pirámide. Gracias a su creciente importancia y aceptación, nuevas formas de colaboración están siendo habilitadas. De este modo, tenemos ejemplos reales como Layar o Junaio que permiten añadir piezas simples de información (comentarios, valoraciones, etc.) sobre PDIs existentes, o como Wikitude, WhereMark y Sekai Camera que permiten la creación y edición de PDIs.

C. Tercer nivel: colaboración en tiempo real

El nivel superior de la pirámide corresponde a las aplicaciones que comparten capas de información generadas en tiempo real. Específicamente, en este tipo de aplicaciones dos o más usuarios se conectan entre ellos para contribuir en la generación en vivo del contenido aumentado en forma de capas sobre el PDI capturado por uno de los usuarios. El contenido generado puede ser compartido de cara a estar disponible para usos futuros aprovechando la conexión con redes sociales o servidores propios del nivel anterior.

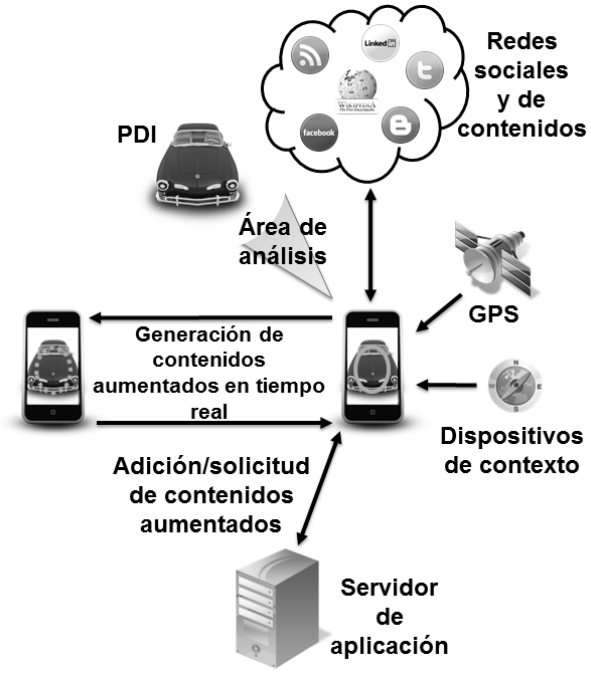


Fig. 4. Arquitectura general del nivel de colaboración en tiempo real.

Para dar soporte a este tipo de aplicaciones es necesario contar con una arquitectura completamente distribuida. En la Figura 4 está representada la abstracción de una posible arquitectura, mostrando la interacción entre usuarios y la generación de contenido sobre el área de análisis de uno de ellos, así como la forma en que se comparte dicho contenido. Las aplicaciones basadas en este nivel requieren toda la tecnología ofrecida por los niveles inferiores y, además, una solución tecnológica para la comunicación en tiempo real.

Este nivel supone un área de investigación con un futuro prometedor. Esto es así debido a que hoy en día no es posible encontrar aplicaciones comerciales capaces de interconectar varios usuarios para generar contenido aumentado en tiempo real de manera colaborativa. Por esta razón, a continuación describimos brevemente dos posibles casos de uso enmarcados en diferentes áreas de conocimiento. El primero, en educación, se basaría en impartir una clase al completo siguiendo un modelo de conexión de 1 a N usuarios, con 1 profesor y N estudiantes recibiendo las enseñanzas a través las pantallas de sus teléfonos móviles y permitiendo además la posibilidad de añadir notas en una capa de RA y compartirlas con el resto de la clase. El segundo caso, esta vez relacionado con la medicina, consiste en conectar a pacientes situados en zonas geográficamente aisladas o de difícil acceso con el médico de familia, o permitir exámenes médicos a distancia realizados por uno o más médicos de manera simultánea y colaborativa.

III. Conclusiones y Trabajos Futuros

A lo largo de este artículo hemos propuesto una serie de definiciones con el objetivo de conceptualizar y clasificar el paradigma actual de las aplicaciones móviles basadas en RA, mostrando además las arquitecturas generales que se encuentran detrás de este tipo de sistemas. Por ello, llegados a este punto es necesario analizar diferentes cuestiones importantes relativas a la propuesta de una nueva taxonomía.

En primer lugar, es necesario describir el rol que juega dicha taxonomía frente a otras planteadas anteriormente, que

como se ha comentado ya, eran escasas y poco enfocadas a esta área. Así, debemos comparar la "Pirámide de Generación de Contenido Colaborativo" con la clasificación "*Interaction Techniques and User Interfaces*" descrita en [10]. Específicamente, en dicha aportación se define la colaboración remota basada en RA como la capacidad para integrar múltiples usuarios que poseen diferentes dispositivos y que se encuentran en contextos dispares, incrementando la interacción entre ellos en tiempo real gracias a una mejora de su experiencia de usuario. Si nos detenemos un instante a pensar en esta idea, comprobamos que el tercer nivel de nuestra taxonomía (nivel Real) es la evolución lógica de esta definición propuesta en 2008. Esto es así, ya que nuestra propuesta intenta establecer la creación de un espacio compartido entre usuarios que permita colaborar mediante la generación de contenido virtual sobre un entorno mixto que proporcione un contexto colaborativo avanzado.

Otro punto importante a analizar es cómo ayuda la taxonomía descrita en este artículo a evaluar el entorno de las aplicaciones móviles basadas en RA. Una respuesta rápida es que nos permite clasificarlas de manera adecuada y con unos criterios claros basados en algo fácilmente identificable como es la arquitectura del sistema y los métodos de generación de contenido colaborativo, poniendo orden por tanto en un entorno bastante caótico que ha evolucionado sin una línea clara en los últimos años. Sin embargo, hay una respuesta si cabe más interesante: ofrece a diseñadores y desarrolladores la oportunidad de trabajar en un marco común claramente definido y conocido por todos, que además, ayuda a entender más fácilmente que tipo de aplicación se quiere implementar y qué módulos son necesarios. Así, identificar por ejemplo en el mercado móvil los competidores que se tiene a la hora de desarrollar una nueva aplicación resulta más sencillo.

Por otro lado, y enfocando ahora nuestro análisis en detalles de índole más tecnológica, creemos que existen significativas líneas de estudio que pueden ser explotadas en años venideros, siendo algunas de las más importantes las que detallamos a continuación.

La primera que queremos remarcar está relacionada con uno de los mayores problemas que podemos encontrar cuando se generan capas de información de RA en tiempo real. Concretamente nos referimos al renderizado de gráficos (muchas veces en 3D) en tiempo real, que es necesario para visualizar las capas aumentadas de este tipo de aplicaciones. Estos procesos son pesados en lo que a procesamiento se refiere, y aunque los actuales *smartphones* son dispositivos muy potentes, es evidente que descargar a los clientes móviles de este tipo de operaciones daría mayor libertad para dedicar esos recursos a otros aspectos. Por ello, creemos que iniciativas como OnLive [11] son altamente interesantes, ya que actualmente ofrece un portal de videojuegos que se ejecuta en la nube, siendo necesarios únicamente una conexión de banda ancha y no un ordenador potente para ejecutar dichos juegos, pues el renderizado gráfico se realiza en sus servidores. Luego, aplicar este tipo de técnicas de *cloud computing* [12] al entorno móvil permitiría que no sólo los últimos modelos de *smartphones* pudiesen soportar aplicaciones de los niveles superiores de la pirámide, sino también aquellos menos potentes.

Cambiando de enfoque, y poniendo nuestra atención ahora en la poca estandarización que este entorno posee actualmente, sería sumamente interesante crear un API o formato común para la definición de PDIs, de manera que una aplicación pudiese integrar PDIs creados en cualquier otra aplicación, evitando así la replicación y redundancia de contenido que actualmente existe entre aplicaciones. Con esto en mente, hemos seguido por un lado lo propuesto en el *W3C Workshop: Augmented Reality on the Web* donde Reynolds et al. [13] propusieron usar Linked Data [14] para definir PDIs en aplicaciones de RA, y por otro lado el proyecto actual de definición de PDIs en el que W3C está trabajando [15].

Por último, y enmarcado en un trabajo futuro relacionado con la propia definición de la taxonomía, sería interesante añadir una nueva dimensión a la misma que tuviese en cuenta el impacto social de usar las aplicaciones de los diferentes niveles, como por ejemplo el número máximo de usuarios que es viable que colaboren a la vez o cómo este tipo de sistemas afectan a la experiencia de usuario final.

REFERENCIAS

[1] F. Reynolds, "Web 2.0-In Your Hand". *IEEE Pervasive Computing*, vol. 8, no. 1, pp. 86-88, Jan. 2009.

[2] S.J. Vaughan-Nichols, "Augmented Reality: No Longer a Novelty?" *Computer*, vol. 42, no. 12, pp. 19-22, Dec. 2009.

[3] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier and B. MacIntyre, "Recent advances in Augmented Reality". *IEEE Computer Graphics and Applications*, vol. 21, no. 6, pp. 34-47, Nov/Dec. 2001.

[4] S.K. Feiner, "The importance of being mobile: some social consequences of wearable augmented reality systems". In Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality, pp. 145-148, San Francisco, CA, USA, Oct. 1999.

[5] W3C, "Geolocation API Specification", 2010. [Online]. Available: http://www.w3.org/TR/geolocation-API/ [Accessed: March 10, 2011].

[6] W3C, "HTML Media Capture", 2010. [Online]. Available: http://www.w3.org/TR/capture-api/ [Accessed: March 10, 2011].

[7] W. Lee and W. Woo, "Exploiting Context-Awareness in Augmented Reality Applications". In Proceedings of the International Symposium on Ubiquitous Virtual Reality, pp. 51-54, Gwangju, South Korea, Jul. 2008.

[8] T. Hofer, W. Schwinger, M. Pichler, G. Leonhartsberger, J. Altmann and W. Retschitzegger, "Context-awareness on mobile devices - the hydrogen approach". In Proceedings of the 36th Annual Hawaii International Conference on System Science, pp. 292-301, Big Island, Hawaii, Jan. 2003.

[9] E. Prasolova-Forland, M. Divitini and A.E. Lindas, "Supporting Social Awareness with 3D Collaborative Virtual Environments and Mobile Devices: VirasMobile". In Proceedings of the Second International Conference on Systems, pp. 33-38, Sainte-Luce, Martinique, France, April 2007.

[10] Feng Zhou, H.B.-L. Duh and M. Billinghurst, "Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR". In Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, pp. 193-202, Cambridge, UK, Sep. 2008.

[11] OnLive, 2010. [Online]. Available: http://www.onlive.com/ [Accessed: March 10, 2011].

[12] X. Luo, "From Augmented Reality to Augmented Computing: A Look at Cloud-Mobile Convergence". In Proceedings of the International Symposium on Ubiquitous Virtual Reality, 29-32, Guangju, Korea, Jul. 2009.

[13] V. Reynolds, M. Hausenblas, A. Polleres, M. Hauswirth and V. Hegde, "Exploiting Linked Open Data for Mobile Augmented Reality". In Proceedings of the W3C Workshop: Augmented Reality on the Web, Barcelona, Spain, Jun. 2010.

[14] Linked Data, Connect Distributed Data Across the Web, 2010. [Online] Available: http://linkeddata.org/ [Accessed: March 10, 2011].

[15] W3C, Points of Interest (POI) Working Group, 2010. [Online] Available: http://www.w3.org/2010/POI/ [Accessed: March 10, 2011].

AFICUS: Una arquitectura para contenidos generados por el usuario en la Internet del Futuro

Luis López Fernández,
Departamento de Sistemas Telemáticos y Computación
Universidad Rey Juan Carlos
C/ Tulipán S/N, 28933 Móstoles (Madrid)
llopez@gsyc.es

Diego González Martínez y David Lozano Llanos
Service Layer Arquitecture (SLA) Area. Neo SDP Initiative
Telefónica Investigación y Desarrollo
Boecillo (Valladolid), España
diegog@tid.es, dll@tid.es

Carlos Esteban Baz Hormigos
Departamento de Teoría de la Señal y Comunicaciones
Universidad Carlos III de Madrid
Avda.de la Universidad, 30, 28911-Leganés (Madrid)
cebaz@tsc.uc3m.es

Carlos Maestre Terol
Departamento de I+D+i
Amaris España
C/ Miguel Yuste, 12, 1ª, 28037 Madrid
carlos.maestre@amaris.es

***Resumen*- En este artículo presentamos AFICUS (Arquitectura para la Futura Internet de Contenidos de USuario) una arquitectura e implementación de una plataforma cuyo objetivo es facilitar la evolución del usuario de servicios de contenidos desde el papel de consumidor, que mayoritariamente desempeña en nuestros días, hacia el papel de productor/consumidor. Para ello, AFICUS parte de una hipótesis básica: si logramos que generar contenidos sea tan sencillo como usar un teléfono móvil, entonces tendremos más de dos mil millones de potenciales generadores de contenidos, uno por cada usuario de telefonía móvil que existe en la actualidad. Partiendo de esta hipótesis básica, AFICUS integra los dominios de generación y consumo de contenido más habituales (Telco, Web, IPTV, etc.) y los complementa con un conjunto de funcionalidades adicionales de enriquecimiento, adaptación y recomendación. De este modo, AFICUS hace posible desarrollar aplicaciones de manera sencilla mediante una API coherente de acceso a todas esas capacidades.**

***Palabras Clave* – User Generated Content (UCG), interconexión de dominios, Internet del Futuro.**

I. INTRODUCCIÓN

Los servicios para la creación, el almacenamiento y distribución de contenidos generados por el usuario se han convertido en uno de los protagonistas principales en el escenario de la Internet del Futuro. Servicios como Youtube, Vimeo o Last.fm son claros ejemplos que muestran la enorme capacidad de estas tecnologías a la hora de atraer la atención de millones de usuarios y a la hora de posibilitar nuevos modelos de negocio.

Sin embargo, este tipo de tecnologías no han logrado todavía universalizar el proceso de producción/consumo de contenidos. Un análisis superficial de alguna de las múltiples estadísticas de uso [1] muestra que los usuarios tienen un claro sesgo hacia el consumo de UGC (User Generated Content) siendo minoritarios los que realmente producen (y suben) UGC. Así, apenas un 30% de los usuarios de Internet declara tener los conocimientos suficientes como para subir un vídeo a Youtube. Esta sensación se ve reforzada por estudios adicionales [2] que muestran que el número total de vídeos subidos a Youtube en 2008 rondaba los 70 millones, mientras que en ese mismo año el número total de usuarios de telefonía móvil celular sobrepasaba con creces los 2.000 millones [1].

A partir de esta información podemos establecer la hipótesis fundamental de este trabajo: si logramos que el perfil de consumidor de UGC se convierta en un perfil productor/consumidor, entonces será posible multiplicar la capacidad de generación de tráfico (y la capacidad de generación de negocio) del UGC en la Internet del Futuro.

ISBN: 978-84-694-5948-5

Para lograrlo proponemos una estrategia sencilla. Como acabamos de ver, miles de millones de personas saben cómo utilizar un teléfono móvil para generar contenido (tanto vocal para llamadas como audiovisual a través de las capacidades adicionales que ofrecen los *smartphones*). Hagamos entonces que la experiencia de usuario para generar UGC sea la misma que para usar un teléfono móvil y, en ese momento, tendremos potencialmente miles de millones de productores de contenidos.

Partiendo de esta idea, la iniciativa AFICUS (Arquitectura para una Futura Internet de Contenidos de Usuario) trata de dar un paso más en el estado del arte para simplificar la transición desde el perfil masivo de usuario consumidor de UGC hacia el de usuario consumidor/productor de UGC. AFICUS propone una arquitectura cuyos detalles se especifican en la sección siguiente..

II. Arquitectura AFICUS

Para comprender en detalle qué es AFICUS y cómo es posible utilizarlo, lo mejor es comenzar observando la Fig. 1, en la que se muestran los elementos esenciales de la arquitectura.

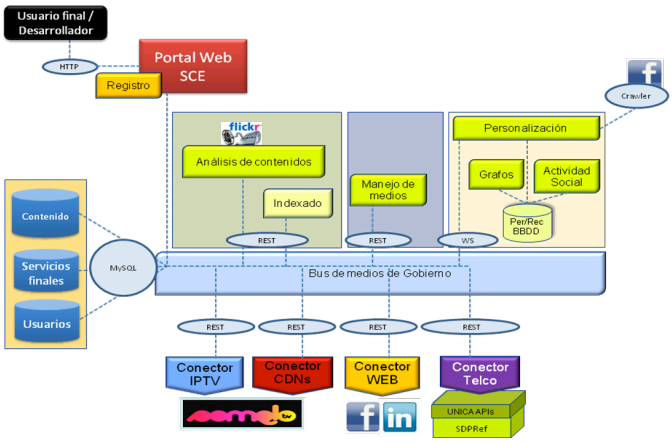


Fig. 1. Diagrama con los elementos esenciales de la Arquitectura AFICUS

De manera sintética, podemos decir que AFICUS es una plataforma que ofrece un conjunto de capacidades al desarrollador de servicios, que puede utilizarlas y combinarlas de manera flexible para proporcionar una experiencia unificada a la hora de producir y consumir UGC. Para lograrlo, la plataforma utiliza una arquitectura que se compone de los siguientes elementos fundamentales:

A. *El Gobierno Extremo a Extremo*

Es la parte de la arquitectura que contiene la lógica de orquestación de servicios. Es decir, el gobierno puede ser configurado para que ante un evento determinado, se ejecuten un conjunto de capacidades que, combinadas, permiten la prestación de un servicio. El núcleo del gobierno, es un motor de control de flujos que realiza la orquestación de las acciones y la comunicación con el resto de los elementos de la arquitectura.

El Gobierno es, por tanto, un *hub* que conoce todos los eventos internos del sistema y que permite definir la semántica de esos eventos desde el punto de vista de la arquitectura. Es decir, admite ser programado para ejecutar secuencias de acciones específicas ante eventos específicos. Para simplificar la labor del desarrollador, el Gobierno contiene un conjunto de flujos de ejecución predefinidos asociados a comportamientos habituales de los servicios. Así, por ejemplo, la secuencia de acciones para recibir una fotografía desde un terminal móvil, etiquetar la fotografía de acuerdo a su contexto y su contenido, y almacenar la fotografía en el repositorio de *media*, es un comportamiento que se puede lograr mediante una sola llamada al gobierno.

B. *Conectores de dominio*

La arquitectura AFICUS cuenta con 4 conectores que permiten la interoperabilidad (con las adaptaciones oportunas) de redes Telco, de servicios Web preexistentes, de infraestructuras IPTV y de sistemas CDN (Content Distribution Networks) basados en P2P. La mayor parte de estos conectores hacen labores complejas de adaptación de protocolos y formatos.

La plataforma AFICUS se ha concebido para ser fácilmente extensible, motivo por el que sus interfaces deben poder evolucionar de una manera estándar. Con este fin, el conector Telco ha incorporado UNICA, una especificación común de APIs utilizadas en el Grupo Telefónica para exponer diferentes tipos de servicios y capacidades Telco. Por tanto, podemos afirmar que UNICA es un estándar corporativo de Telefónica que actualmente se aplica para la exposición de capacidades tanto a nivel local como a nivel global. De este modo, UNICA aproxima la filosofía SOA al mundo del operador para permitir la construcción de servicios distribuidos de manera sencilla. En el contexto de AFICUS, las APIs UNICA se han extendido mejorando el *binding* RESTful. Estas mejoras se están utilizando actualmente como parte del conector Telco de la arquitectura AFICUS y han servido como *guideline* para el desarrollo de otras interfaces internas de otros módulos, como la que implica al sistema de enriquecimiento o al de manejo de contenidos.

Adicionalmente, y en parte gracias a las aportaciones de AFICUS, UNICA ha asimilado otras influencias de la industria, pero al mismo tiempo ha influido ella misma en la evolución de ciertos estándares como GSMA OneAPI [3] y OMA ParlayRest v1.0 [4]. Siguiendo estas directrices, y para proveer un mecanismo homogéneo para la definición de interfaces, UNICA ofrece lo siguiente:

- Un conjunto de principios tecnológicamente agnósticos para consolidar la homogeneidad de las interfaces independientemente del tipo de capacidades a la que permiten acceder.
- Un conjunto de guías y buenas prácticas que deben ser respetadas cuando se diseñan e implementan APIs para tecnologías específicas, tales como SOAP, REST, o tecnologías RPC ligeras.

Por último, es interesante destacar que UNICA no está asociada a ningún mecanismo de representación de datos particular. Así, dependiendo de las características de cada servicio, es posible usar contenedores SOAP, mecanismos REST (RESTful o light RPC) o ambos al mismo tiempo. De este modo, HTTP se convierte en el protocolo principal que puede transportar XML y también JSON mediante REST. Este esquema proporciona a AFICUS una enorme capacidad

para exponer y utilizar las diferentes facilidades de la infraestructura Telco.

C. Servicio de Gestión (Manejo de medios)

Este módulo de la arquitectura es el responsable de proporcionar las capacidades de adaptación y composición dinámica de contenidos que son utilizadas por los servicios AFICUS. En este sentido, su objetivo es poner a disposición del desarrollador herramientas de fusión, mezcla y enriquecimiento de los contenidos siguiendo un esquema que podríamos definir como de "edición avanzada". Esta capacidad es un habilitador de modelos de negocio dado que permite realizar tareas como la combinación de contenidos profesionales y de usuario, la inserción de *banners* y otro tipo de publicidad en los contenidos, etc.

D. Servicio de Personalización

Como ya hemos adelantado, una de las novedades que introduce AFICUS es la capacidad de integrar información contextual combinada, incorporando para ello los datos que los usuarios proporcionan en el contexto de sus redes sociales. Este objetivo se logra gracias a dos subsistemas que realizan labores complementarias: el *crawler*, o sistema de extracción de información de información social AFICUS y el módulo de personalización propiamente dicho. Vamos a analizar brevemente el funcionamiento de cada uno de ellos:

El sistema de extracción de información social de AFICUS consiste, en la actualidad, en una aplicación Facebook que el usuario se instala y que dota a la plataforma de la capacidad de acceder de manera estructurada a la mayor parte de la información que la red social tiene del usuario. Una vez que la información ha sido extraída e incorporada dentro del CCU (Contexto Combinado de Usuario), esta puede ser utilizada para la realización de las labores de Personalización y Recomendación. Desde el punto de vista de AFICUS, entendemos la personalización como la adaptación de servicios, interfaces y aplicaciones a las características precisas de un usuario. Por otro lado, entendemos la recomendación como un servicio a través del cual un conjunto de ítems (contenidos, documentos, usuarios, etc.) se ordenan de acuerdo a las características y preferencias concretas de un usuario.

Para realizar la labor de personalización y recomendación a partir del CCU es posible utilizar diversas técnicas que van desde la inteligencia artificial a los heurísticos pasando por sistemas expertos o técnicas bayesianas. En este momento, el sistema contiene una única implementación basada en los mecanismos clásicos de cálculo matricial por similitud en coseno [6]. Esta implementación tiene algunas limitaciones, como la carencia de aprendizaje, pero tiene la ventaja de no requerir ninguna clase de datos de entrenamiento ni parametrización iniciales, por lo que puede ser usada desde el primer momento sin necesidad de acumular un volumen inicial de muestras.

E. Servicio de Enriquecimiento de Contenidos

El servicio de Enriquecimiento de Contenidos es uno de los aspectos en los que AFICUS sobresale con respecto a otras iniciativas previas similares por un doble motivo: por un lado, porque utiliza las más modernas técnicas de visión artificial de reconocimiento de patrones y clasificación; por otro, porque estas técnicas se ven complementadas con la información social y el CCU del usuario del que dispone la plataforma AFICUS. Por este motivo, vamos a invertir unos párrafos para explicar con un poco más de detalle su funcionamiento interno.

Desde el punto de vista social, es conocido que los usuarios suben millones de imágenes a sitios web como "*Flickr"* o "*Picasa"*. Estas imágenes a menudo llevan asociadas diferentes etiquetas que contienen información sobre el contenido, hora, fecha, cámara y más recientemente, geolocalización. Empleando esta información AFICUS implementa un sistema capaz de ofrecer automáticamente etiquetas relevantes ante nuevo contenido generado por usuario. Además, para la anotación de vídeo, se ha explotado la particular estructura de movimientos de cámara que contienen los vídeos generados por usuario para realizar una segmentación en fotogramas clave, que puedan ser etiquetados y utilizados para generar vídeo resúmenes y contenido multimedia a la carta.

A partir de la información de contexto del contenido a anotar, se recupera de las bases de datos disponibles un conjunto de imágenes relevantes. Posteriormente, tiene lugar un proceso de *matching* visual que analiza la similitud de cada una de las imágenes recuperadas con el contenido de usuario, ya sea la fotografía tomada o los fotogramas clave extraídos del vídeo. Mediante un procesado de etiquetas, los conceptos que presentan una mayor frecuencia de aparición en dichas imágenes son propuestos para anotar el contenido UGC.

Adicionalmente, para contenidos de vídeo, se realiza una estimación del movimiento global y, empleando una ventana deslizante, se construye un espacio de vectores con características como velocidad y aceleración medias, varianza de la aceleración, número medio de cambios de dirección en el eje vertical y horizontal y movimiento radial estimado. La segmentación se realiza mediante el entrenamiento de clasificadores SVM multiclase implementados con una estrategia de SVMs binarias en la forma 'uno contra uno' y 'votación' [9]. Finalmente, se seleccionan un conjunto de fotogramas clave correspondientes a fragmentos de vídeo con movimiento *Pan, Zoom* o *Still*.

La tarea de *matching* visual se puede descomponer en dos etapas: representación de imágenes y evaluación de similitud. Estudios recientes han demostrado que una combinación de múltiples características ofrece una representación más precisa. Por ello, en el sistema se emplean los siguientes descriptores: SIFT *(Scale-Invariant Feature Transform)* [10], HOG *(Histogram of Oriented Gradient)* [11] y los tres primeros *Momentos de color*.

La relación visual entre imágenes se establece según el modelo de Bolsa de Palabras o *Bag of Words* [12], generando un vocabulario representativo de palabras visuales a través de métodos de *clustering*. Posteriormente el descriptor asociado a cada región local se proyecta sobre el vocabulario asignándosele la palabra más próxima. Finalmente se construye un histograma normalizado de palabras con el que se calcula la similitud entre imágenes.

En la integración de las características extraídas, se emplea la siguiente combinación lineal:

$$d_{TOTAL} = k_{SIFT} \cdot d_{SIFT} + k_{HOG} \cdot d_{HOG} + k_{CM} \cdot d_{CM}$$

Donde d representa distancia entre histogramas normalizados y k factor de ponderación. Las etiquetas de cada imagen recuperada son pesadas en función de su distancia visual con la referencia o fotograma clave, generando histogramas de frecuencia ponderada de aparición. Adicionalmente, se realiza un proceso de normalización de etiquetas mediante la distancia *Levenshtein*, o distancia de edición.

F. Portal Web

El Portal Web es el punto de acceso a la plataforma para desarrolladores y usuarios de modo que, dependiendo del perfil de cada uno, lo que se visualiza en el portal permite acceder y personalizar sus aplicaciones (en el caso de los usuarios), o bien utilizar herramientas que les ayuden en la creación y configuración de nuevos servicios y aplicaciones (en el caso de los desarrolladores). En la actualidad, las facilidades orientadas a desarrolladores están siendo definidas e implementadas, siendo los únicos aspectos realmente funcionales los relativos a usuarios finales.

En este último caso, el portal (que es accesible en la URL http://www.aficus.org de manera restringuida) permite el registro de usuarios y el acceso a las aplicaciones que ya están implementadas. A modo de ejemplo, un usuario puede registrarse y, mediante la utilización de códigos QR [13], descargar e instalar en su móvil la aplicación de captura de fotografías con automatizaciones de almacenamiento, geolocalización y enriquecimiento por parte de la plataforma.

En el futuro, el Portal Web debería convertirse en la puerta de entrada a todo el ecosistema AFICUS, permitiendo a los usuarios navegar entre sus contenidos y seleccionar las aplicaciones que permiten su uso.

III. CONCLUSIONES

En este artículo hemos presentado AFICUS, la arquitectura e implementación de una plataforma de desarrollo de aplicaciones UGC (User Generated Content) que integra múltiples dominios de interconexión y los enriquece.

Tal y como hemos visto, AFICUS aporta novedades con respecto al estado del arte en diversos ámbitos. Desde el punto de vista de la interconexión entre dominios, AFICUS es capaz de alcanzar de manera muy profunda las capacidades del operador a través del uso de una API basada en UNICA, el estándar unificado de Telefónica de acceso a las capacidades de red. Lo que dota a la arquitectura de una potencia especial gracias a su interoperabilidad con dominios habituales para la distribución de UGC, tales como IPTV, P2P y servicios Web 2.0. Más relevante todavía es el hecho de que AFICUS no se comporta simplemente como un *pipe* de transporte de contenidos, sino que tiene capacidad para enriquecerlos, indexarlos, almacenarlos y recuperarlos mediante un mecanismo recomendación que se basa en un contexto combinado de usuario.

Aunque AFICUS todavía tiene que recorrer (y mejorar) los aspectos relativos a simplificar el desarrollo de aplicaciones, creemos que sí es posible afirmar que abre un novedoso abanico de posibilidades para la creación de aplicaciones y modelos de negocio innovadores en el ámbito de la Internet del Futuro.

AGRADECIMIENTOS

Este artículo y los trabajos descritos en el mismo han sido cofinanciados por el Ministerio de Industria Turismo y Comercio dentro del Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica 2008-2011, cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) bajo el proyecto AFICUS con referencia TSI-020110-2009-103 y por la Comunidad de Madrid bajo el proyecto CLOUDS con referencia S2009-TIC1692.

REFERENCIAS

[1] La Sociedad de la Información en España 2010. Report by Fundación Telefónica available at http://sociedadinformacion.fundacion.telefonica.com/seccion=1190&idioma=es_ES&id=&activo=13.do

[2] http://ksudigg.wetpaint.com/page/YouTube+Statistics

[3] http://www.gsmworld.com/oneapi/

[4] RESTful bindings for Parlay X Web Services., Open Mobile Alliance. http://member.openmobilealliance.org/ftp/Public_documents/ARCH/Permanent_documents/

[5] http://developers.facebook.com/

[6] G. Adomavicius, ad A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions", IEEE Transactions on Knowledge and Data Engineering, Vol. 17(6), pp. 734-749.

[7] A. Golnaz, C.M. Taskiran, P. Zygmunt, and E.J. Delp, "Camera Motion-Based Analysis of User Generated Video", IEEE Transactions on Multimedia, Vol. 12, No. 1. (January 2010), pp. 28-41.

[8] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analisys and automated cartography", Comm. ACM, 24 (1981), pp. 381-395.

[9] C. Chang and C.-J. Lin. "LIBSVM: a library for support vector machines",2001. Software available at http://www.csie.ntu.edu.tw/-cjlin/libsvm.

[10] David G. Lowe, "Distinctive image features from scale-invariant keypoints" *International Journal of Computer Vision,* vol. 60, pp. 91-110, 2004.

[11] Navneet Dalal and Bill Triggs, "Histogram of oriented gradients for human detection", *Proceedings of IEEE Computer Society Conference on CVPR,* 2005, pp. 886-893.

[12] Y.-G. Jiang, C.-W. Ngo, and J. Yang. "Towards optimal bag-of-features for object categorization and semantic video retrieval" *Proc. of ACM Int'l Conf. on Image and Video Retrieval*, 2007.

[13] http://es.wikipedia.org/wiki/C%C3%B3digo_QR

QMoES: una herramienta de estimación de BW en arquitecturas QoE de banda ancha

José I. Aznar, Eduardo Viruete, Julián Fernández-Navajas, José Ruiz-Mas, José Mª Saldaña,
Grupo de Tecnologías de las Comunicaciones (GTC)
Dpto. de Ingeniería Electrónica y Comunicaciones (DIEC)–Instituto de Investigación en Ing. de Aragón (I3A)
Universidad de Zaragoza
DIEC, Edificio Ada Byron, Centro Politécnico Superior de la Universidad de Zaragoza
Email: {jiaznar, eviruete, navajas, jruiz, jsaldana}@unizar.es

***Resumen*- El actual panorama de las comunicaciones multimedia está evolucionando hacia servicios que implican estrictos requerimientos tecnológicos. El actual escenario *Triple-Play* resulta insuficiente para adaptarse a cada servicio solicitado y gestionarlo a través de Internet con Calidad de la Experiencia (QoE, *Quality of Experience*) garantizada. La iniciativa RUBENS (*Rethinking the Usage of Broadband Access for Experience-optimized Networks*) define un nuevo modelo de arquitectura que aúna los mecanismos de QoS más relevantes a fin de mejorar la QoE de usuario. En este contexto, presentamos el mecanismo QMoES (*Quality Monitoring and Estimation*): una herramienta de estimación y monitorización de ancho de banda que colabora en el mantenimiento de la QoE. Nuestro estudio explica la implementación y evaluación del mecanismo QMoES,**

***Palabras Clave*- QoS, QoE, Estimación Activa de BW, Simulación OPNET, Arquitecturas NGN.**

I. INTRODUCCIÓN

La evolución exponencial de servicios conlleva un considerable incremento del tráfico (ej. flujos de vídeo, P2P (*Peer-to-Peer*), juegos on-line, etc.), al mismo tiempo que los usuarios finales están limitados por configuraciones de red basadas en *Best-Effort*, las cuales no proporcionan garantías de QoS. El serio incremento de contenido multimedia combinado con la necesidad de personalización de los servicios establece nuevos parámetros y requerimientos que las nuevas arquitecturas de red basadas en QoE deben gestionar. Por un lado, la provisión de servicios a través de Internet permite a los operadores y proveedores de servicio ofrecer un elevado grado de flexibilidad. Por otro lado, las redes basadas en *Triple-Play* proporcionan calidad no personalizada a determinadas aplicaciones que no requieren configuraciones dinámicas [1]. Sin embargo, existen diversos factores que implican que resulte complicado garantizar QoE extremo a extremo en la provisión de servicios multimedia.

Una posible solución podría consistir en un sobre-dimensionamiento de redes y recursos. Una solución más interesante desde el punto de vista de la investigación, considera que los bloques funcionales de una red son responsables de la gestión de recursos y la provisión de QoE a través de señalización cruzada entre capas y técnicas de ingeniería de red. Esta alternativa supone una solución fundamental para el problema de provisión de contenidos y para alcanzar niveles de QoE y personalización aceptables.

El proyecto RUBENS (*Rethinking the Usage of Broadband Access for Experience-optimized Networks*) [1] propone la definición y evaluación de una arquitectura de red que ofrece QoE personalizada para una gran variedad de aplicaciones, modelos de distribución y equipamiento de red. RUBENS incluye diversos mecanismos que optimizan dinámicamente la experiencia de los usuarios finales a través de la coordinación entre la red y las aplicaciones. Uno de esos mecanismos clave consiste en la provisión de medidas de estimación de ancho de banda disponible de un determinado enlace extremo a extremo y su monitorización. La utilización de herramientas de estimación de ancho de banda (BW, *BandWidth*) permite adaptar dinámicamente las sesiones de usuario y reaccionar a inesperadas degradaciones de calidad en la red RUBENS.

En este trabajo proponemos una nueva herramienta de estimación y monitorización de BW denominada QMoES (*Quality Monitoring and Estimation*) adaptada a los requerimientos específicos que la arquitectura de red y los servicios precisan. Con este fin, se presenta en primer lugar una descripción general de la arquitectura RUBENS y se define el rol específico de QMoES. Además, se explican los requerimientos de RUBENS en términos de las métricas más relevantes que precisan ser mejoradas en el proceso de implementación de herramientas de estimación. Adicionalmente, se exponen los resultados de simulación que cuantifican la bondad de QMoES, así como las principales conclusiones y líneas de trabajo futuras.

El trabajo está organizado de la siguiente manera: Una revisión del estado del arte de las herramientas de estimación de BW se ofrece en la sección II. La sección III describe la arquitectura RUBENS y los bloques funcionales de gestión de la QoE. En la sección IV se detalla la plataforma de simulación basada en OPNET. La sección V explica los resultados de simulación más relevantes. La sección VI concluye este trabajo.

II. REVISIÓN DE HERRAMIENTAS DE ESTIMACIÓN DE BW

Una de las clasificaciones más aceptadas divide las herramientas activas de estimación de BW en dos grupos

principales: PGM (*Probe Gap Models*), las cuales basan su estimación en la dispersión temporal entre dos paquetes consecutivos en el receptor, y PRM (*Probe Rate Models*), cuyas estimaciones se basan en el envío de trenes de paquetes (más de dos paquetes) a diferentes tasas.

La eficacia y eficiencia de las herramientas de estimación de BW ha sido ampliamente estudiada. En esta sección proponemos una revisión de las herramientas más recientes e identificamos las principales limitaciones que tanto los modelos PGM como PRM presentan.

Las implementaciones de mecanismos de estimación de BW recientes han sido argumentadas de forma genérica, considerando diversas hipótesis de partida que no representan condiciones reales en redes actuales: por ejemplo, es común a todos los métodos PGM la suposición de que el *Narrow-link* (enlace que presenta una menor capacidad total) coincide con el *Tigh-link* (enlace que presenta un menor ancho de banda disponible). Sin embargo, se ha probado [2] que esta hipótesis no funciona en escenarios de alta capacidad en los que los enlaces presentan varios saltos entre los extremos de la comunicación y el tráfico es no persistente [3]. Spruce [4] y ABwE [5] constituyen mecanismos recientes de estimación de BW que se apoyan en este concepto. Por este motivo, los métodos basados en PGM y sus características no han sido considerados para la implementación de QMoES, centrando nuestro esfuerzo en herramientas tipo PRM.

Otra limitación observada está relacionada con el modelo de tráfico interferente utilizado para interaccionar con el tráfico de prueba propio de las estimaciones de BW. Por ejemplo, en [5] el modelo de tráfico se compone de paquetes de 700 bytes de tamaño fijo con tasa de bit constante, el cual no representa condiciones de tráfico real. También la configuración de los *test-bed* propuestos para la validación de las herramientas de estimación de BW presenta cierta obsolescencia. Así, en [6] los autores presentan un estudio basado en un modelo de Markov que utiliza un *test-bed* con una capacidad máxima de 10 Mbps. Adicionalmente, algunas de las herramientas propuestas se presentan de una forma muy generalista, ignorando la posibilidad de incluir en los estudios aplicaciones concretas y arquitecturas en las que pudieran ser integradas.

En definitiva, no es evidente que los mecanismos de estimación propuestos más actuales sean adecuados para las redes de nueva generación. En las sucesivas secciones presentamos la implementación de QMoES en el contexto de la arquitectura RUBENS. Existen varias diferencias entre los mecanismos previamente propuestos y el que se presenta en este trabajo. En primer lugar, particularizamos QMoES a los requerimientos específicos de RUBENS, mejorando las métricas de interés. En segundo lugar, no se ha asumido un modelo de tráfico interferente, sino que se ha utilizado la herramienta OPNET para configurar los servicios de voz y vídeo. Finalmente, hemos considerado un escenario con enlaces de alta capacidad y para situaciones de congestión media y elevada, que sí representan condiciones más realistas en las actuales redes de comunicaciones.

III. ARQUITECTURA RUBENS

El objetivo de RUBENS consiste en definir y validar una arquitectura de red de acceso que garantice la provisión de servicios con QoE. Los servicios considerados están basados en contenidos multimedia, especialmente Video bajo demanda (VoD, *Video on Demand*), puesto que representa uno de los servicios más atractivos tanto para usuarios como para proveedores. Considerando este principio, la arquitectura RUBENS presenta un nuevo escenario para la provisión de servicios de forma dinámica.

A. Perspectiva global de los bloques funcionales RUBENS

La arquitectura RUBENS ha sido implementada atendiendo a los servicios (principalmente VoD) que pretenden ser desplegados y especifica las funcionalidades que deben poseer los accesos de banda ancha para controlar la calidad con la que dichos contenidos son ofrecidos. A fin de definir las interacciones que se producen en el seno de la arquitectura, los mecanismos se agrupan en siete bloques:

El **Sistema Interfaz de Aplicación (SIA)** proporciona un enlace directo entre las características del nivel de aplicación y la función global de RUBENS. El **Sistema de Gestión de la Calidad (SGC)** habilita la modificación y ajuste dinámico de los parámetros de un servicio determinado e incrementa o disminuye el nivel de QoE con el que dicho servicio es distribuido. El **Sistema de Gestión de Ancho de Banda (SGAB)** es responsable de la asignación de ancho de banda de una clase de servicio.

El **Sistema de Gestión de Transporte (SGT)** gestiona los distintos mecanismos de transporte que utilizan los servicios. Esto incluye tanto la selección del método de transporte (*unicast/multicast, stream/progresive*, etc.) como la adecuación de los parámetros de transporte (p. ej. capacidad media de *buffering*, *caching*, etc.). El **Sistema de Control de Servicios y Usuarios (SCSU)** se encarga de contrastar las restricciones de provisión de servicios a través de la red RUBENS. Las políticas de usuario y servicio son consultadas en este bloque. Las **Funciones de Mediación** constituyen el núcleo de la arquitectura. Encapsulan la lógica para tomar decisiones en base a los datos proporcionados desde los demás bloques funcionales. La gestión ha sido dividida en dos funciones: El **Sistema de Gestión Principal (SGP),** que gestiona los servicios de forma individual, y el **Sistema de Gestión Inter-Servicio (SGIS),** encargado de la gestión de la totalidad de la QoS observada en una línea de acceso. Por último, el **Sistema de Monitorización de QoE (SMQoE)** es responsable de proporcionar estimaciones de ancho de banda disponible e informar al SGP y al SGIS sobre el rendimiento de un determinado enlace extremo a extremo, y así adaptar el funcionamiento de los bloques SGAB, SGT y SGC en caso de que la calidad percibida por el usuario experimente una cierta degradación.

B. Requerimientos y métricas fundamentales en RUBENS

Las metodologías utilizadas por las herramientas de estimación de BW establecen que debe haber un

compromiso entre una serie de métricas de referencia: *Precisión, Intrusividad y Tiempo de Estimación* [7], las cuales determinan la bondad de los métodos de estimación. Diversas investigaciones han propuesto interesantes soluciones basadas en estas métricas, buscando realizar una herramienta en la que todas las métricas fueran optimizadas simultáneamente. Sin embargo, tal y como se explica en [2], no existe ni la necesidad ni la oportunidad (al menos en redes con enlaces de capacidad elevada) de implementar una herramienta que mejore al mismo tiempo las tres métricas, debido a su carácter antagónico. En este trabajo se ha optado en primer lugar por caracterizar los requerimientos exigidos a la red RUBENS, a fin de identificar qué métricas sería deseable optimizar.

El objetivo de RUBENS consiste en proporcionar calidad a los contenidos personalizados ofrecidos a los usuarios finales. Por esta razón, la *Precisión* es una métrica significativa para mantener niveles de QoE adecuados con fiabilidad. Considerando que la red debe adaptarse dinámicamente en tiempo real a posibles degradaciones, el *Tiempo de Estimación* es también crucial para permitir al sistema reaccionar rápidamente a variaciones de calidad inesperadas que pudieran empeorar la percepción de servicio que tiene el usuario. Por último, la *Intrusividad* de las medidas es también una métrica a considerar. Dado que la red RUBENS trabaja con enlaces de alta capacidad, estas métricas no resultan tan críticas en el proceso de validación, a pesar de que minimizar su impacto es deseable también.

IV. PLATAFORMA DE SIMULACIÓN

Se ha propuesto un escenario de simulación que refleja las condiciones de la arquitectura RUBENS. La idea no consiste en implementar todas las funcionalidades de RUBENS, sino en imitar su comportamiento y determinar la bondad de las estimaciones del sistema QMoES. La plataforma de simulación ha sido dividida en dos partes fundamentales (Fig. 1): la primera consiste en un escenario basado en la herramienta OPNET *Modeler* que se encarga de obtener en el extremo receptor del enlace, medidas temporales de los paquetes de estimación inyectados desde el extremo origen del enlace. (*TRX mechanism* en Fig. 1). La segunda parte del proceso ha sido implementada con MATLAB y lleva a cabo la traslación de las medidas temporales en estimaciones de ancho de banda disponible (*RCV mechanism* en Fig. 1).

A. Plataforma OPNET y configuración de parámetros

La plataforma de simulación basada en OPNET simula la infraestructura RUBENS. Consiste en una red de acceso compuesta de varios nodos y un nodo central que representa el núcleo de red. Varios usuarios finales comparten vídeos multimedia a través de los enlaces que comprende la red RUBENS. El proceso de intercambio de vídeo ha sido configurado en las utilidades *Definición de Aplicaciones* y *Perfiles* disponibles en OPNET que permiten simular distintos tipos de contenido (Video, VoIP, etc.) y diversos niveles de congestión (medio y alto). Niveles de congestión media significa que el *Tight Link* del enlace de interés presenta un 40% de ancho de banda disponible. Un nivel de congestión alto presenta un 20% de ancho de banda disponible en el *Tight Link*. Los enlaces extremo a extremo configurados entre usuarios finales presentan capacidades del orden de Gigabits. Adicionalmente, existen una serie de parámetros que condicionan el comportamiento de QMoES y que es preciso tener en consideración. Cabe destacar:

1) *Modelo de tráfico de prueba (o estimación):* es el tráfico de estimación inyectado por la herramienta QMoES en uno de los extremos del enlace a monitorizar, característico de los modelos de estimación activos. El tráfico ha sido configurado en trenes de paquetes de tamaño 1500 bytes (MTU soportado por la mayoría de las redes basadas en Ethernet), y de la tasa más popular y utilizada en diversos trabajos de estimación de BW [8-10]. El número de paquetes por tren ha sido empíricamente establecido en 200. Se han estudiado dos tipos de configuración de trenes de paquetes: uniforme y sectorizada, las cuales son objeto de análisis en la sección V. El patrón de distribución uniforme genera trenes de paquetes a una tasa con distribución uniforme entre dos umbrales inferior y superior. El modelo sectorizado divide el espectro de tasas de estimación (mismos umbrales de tasa mínima y máxima) en diversos sectores con una probabilidad de ocurrencia asignada. Así, el número de trenes de paquetes configurados con la misma tasa varía en función de la probabilidad de ocurrencia.

2) *Modelo de colas.* El modelo de colas FIFO ha sido considerado como modelo de referencia en recientes implementaciones de *test-bed* y herramientas de estimación de ancho de banda.

B. Proceso de simulación

Los paquetes de estimación son inyectados en el nodo SRC, en donde se les añade una marca temporal. El tráfico

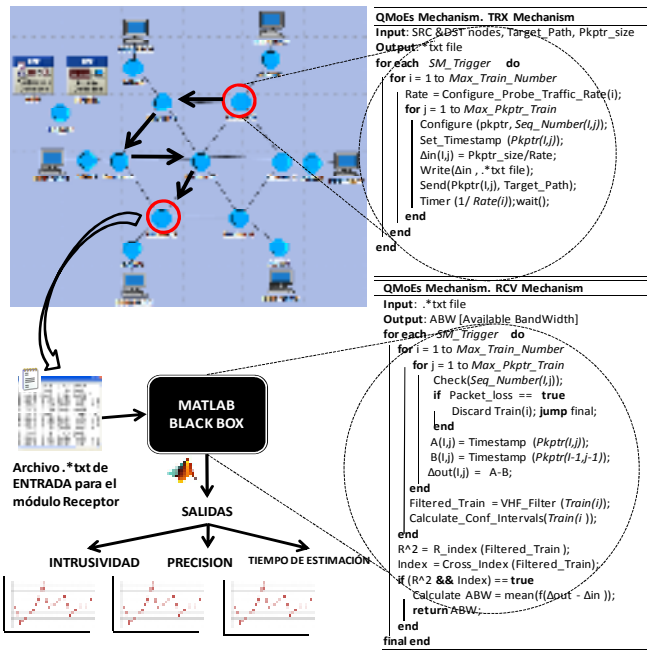


Fig. 1. Plataforma de simulación basada en OPNET y MATLAB para la validación de QMoES.

de estimación recorre el enlace cuyo ancho de banda disponible se quiere estimar y alcanza el nodo destino *DST,* nodo en el cual también recibe una marca temporal. La información temporal en origen y destino se recoge en un archivo que sirve como documento de entrada para la plataforma basada en MATLAB, donde el mecanismo de recepción (*RCV mechanism*) ha sido integrado. Este mecanismo lleva a cabo las siguientes funciones: Detección de pérdidas de paquetes de estimación, determinación de los intervalos de confianza, proceso de filtrado de las muestras temporales y traslación de dichas muestras temporales a estimación de ancho de banda. La Fig. 1 muestra un esquema de las dos partes de que consta el proceso junto con el pseudo-código ejecutado.

V. Resultados de simulación

Tal y como se ha explicado en secciones anteriores, las métricas de interés que requieren un mayor rigor en el proceso de integración de QMoES en RUBENS son la *Precisión* y el *Tiempo de Estimación*. RUBENS precisa de un *Tiempo de Estimación* pequeño para reaccionar de forma instantánea en caso de que un cierto enlace extremo a extremo experimente una degradación de la QoE durante la provisión de un determinado servicio. La *Precisión* es también necesaria para determinar los recursos disponibles en caso de que una nueva sesión o servicio sea solicitado.

La Fig. 2 representa el error relativo medio cometido (la *Precisión* propiamente dicha) únicamente para el tráfico de estimación sectorizado, dado que es el que presenta un mejor rendimiento. Se han considerado las mismas situaciones de congestión media y elevada en función del número de paquetes por tren (eje de abscisas). El número de paquetes por tren es realmente importante en métodos PRM, dado que determina la intrusividad del método de estimación. Los intervalos de confianza han sido establecidos al 95%. Puede apreciarse que el error relativo no excede el 10% para niveles de congestión media. Para una congestión elevada, las configuraciones de 175 y 200 paquetes de estimación por tren presentan un buen rendimiento con un error relativo por debajo del 15%, de modo que los requerimientos de RUBENS son satisfechos en términos de *Precisión*. El *Tiempo de estimación* medio de una medida son 2,301 segundos. Este resultado es también esperanzador, dado que el método de estimación puede proporcionar una estimación de BW al SGP de RUBENS de forma cuasi-inmediata y habilitar a la arquitectura para reaccionar a degradaciones repentinas de QoE en tiempo real. Como se ha explicado en el capítulo III, la *Intrusividad* no es un parámetro tan crítico como los otros dos vértices del llamado "triángulo de compromiso" [11], *Precisión* y *Tiempo de Estimación*.

VI. Conclusiones y líneas de trabajo futuro

Este trabajo introduce QMoES, una herramienta activa de estimación PRM basada en tasas de generación de tráfico variable. Los resultados de simulación, muestran que QMoES presenta valores adecuados en términos de *Precisión* y *Tiempo de estimación*, lo que permite validar la herramienta para la infraestructura RUBENS. La plataforma, basada en OPNET y MATLAB, tiene un diseño modular.

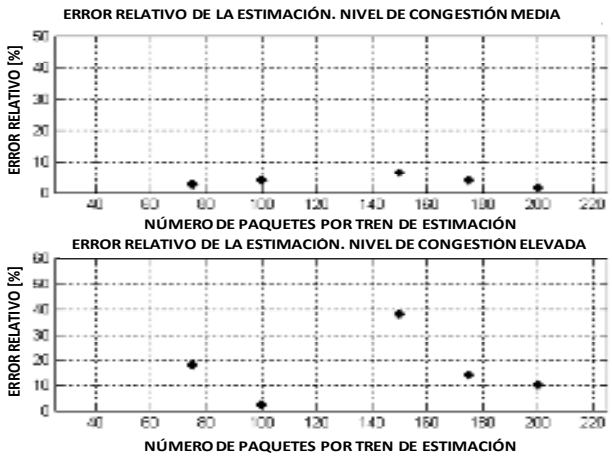


Fig. 2. Error relativo (*Precisión*) para niveles de congestión media y elevada y una distribución de paquetes de trenes sectorizada.

La evaluación de trabajos recientes muestra que las herramientas de estimación de ancho de banda están todavía lejos de presentar un rendimiento óptimo dada la naturaleza contradictoria de las tres métricas: una herramienta que realice medidas eficientes, precisas y no intrusivas sigue siendo a día de hoy un reto.

Agradecimientos

Este trabajo ha sido financiado parcialmente por el Proyecto CPUFLIPI (MICINN TIN2010-17298), el Proyecto MBACToIP, de la Agencia I+D del Gobierno de Aragón e Ibercaja Obra Social y por la Cátedra Telefónica de la Universidad de Zaragoza.

Referencias

[1] (2009) RUBENS project website. [Online]. Available: http://wiki-rubens.celtic-initiative.org/index.php/Main_Page.

[2] M. Jain and C. Dovrolis, "Ten fallacies and pitfalls in end-to-end available bandwidth estimation," in Proc. ACM IMC, 2004, pp. 272–277.

[3] L. Lao, C. Dovrolis, M.Y. Sanadidi, "The Probe Gap Model can Underestimate the Available Bandwidth of Multihop Paths," in ACM SIGCOMM CCR, 2006, vol. 36, no.5. pp. 29-34.

[4] J. Strauss, D. Katabi, F. Kaashoek, "A Measurement Study of Available Bandwidth Estimation Tools", in Proc AMC IMC SIGCOMM 2003.

[5] J. Navratil and R. L. Cottrell, "ABwE: A practical approach to available bandwidth," in Proc. of 4th PAM Workshop 2003.

[6] C.D. Guerrero, M.A. Labrador, "Traceband: A fast, low overhead and accurate tool for available bandwidth estimation and monitoring," Journal of Computer and Telecommunications Networking, vol.54, no. 6, pp.977-990, Apr. 2009.

[7] C. D. Guerrero, M. A. Labrador, "On the Applicability of Available bandwidth estimation techniques and tools, "Journal of Computer Communications. Vol. 33, no.1, pp. 11–22, Jan. 2010.

[8] J. Strauss, D. Katabi, and F. Kaashoek, "A Measurement Study of Available Bandwidth Estimation Tools," in Pror. IMC, 2003, pp.39-44.

[9] B. Melander, M. Bjorkman, and P. Gunningberg, "Regression-based available bandwidth measurements," in Proc. SPECTS 2002.

[10] "A scheme for measuring subpath available bandwidth," in Proc. Of IEEE LCN 2009, pp.1095-1101.

[11] J.I. Aznar, "Estimación Extremo a Extremo de Ancho de Banda Disponible para Redes de Alta Capacidad: Implementación y Evaluación de Herramientas," Tesis de Máster, Centro Politécnico Superior, Zaragoza, Spain, Sep. 2010.

Mejora de la calidad en un sistema de telefonía IP mediante el uso de técnicas de multiplexión

José Mª Saldaña, Julián Fernández-Navajas, José Ruiz-Mas, Jenifer Murillo, José I. Aznar, Eduardo Viruete Navarro, Luis Casadesus
Grupo de Tecnologías de las Comunicaciones – Instituto de Investigación en Ingeniería de Aragón
Dpt. IEC. Centro Politécnico Superior Universidad de Zaragoza
Edif. Ada Byron, 50018, Zaragoza
{jsaldana, navajas, jruiz, jenifer.murillo, jiaznar, eviruete, luis.casadesus}@unizar.es

***Resumen*- En la actualidad muchas empresas utilizan la Voz sobre IP *(Voice over Internet Protocol,* VoIP) en sus sistemas de telefonía. Las empresas con oficinas en distintos países y áreas geográficas pueden construir un sistema de telefonía centralizado para compartir las líneas de sus sucursales y así incrementar la probabilidad de admisión y ahorrar costes en llamadas internacionales. Por tanto, es conveniente introducir un sistema que permita asegurar una Calidad de Servicio *(Quality of Service,* QoS) mínima para las llamadas. Uno de estos sistemas es el Control de Admisión de Llamadas *(Call Admission Control,* CAC). En este trabajo se estudian las mejoras en cuanto a probabilidad de admisión y calidad de la conversación (Factor R) que se pueden obtener cuando se usan técnicas de multiplexión RTP, puesto que en este escenario habrá múltiples llamadas con un origen y destino comunes. Se han realizado simulaciones para comparar el uso habitual de RTP con el del protocolo TCRTP *(Tunneling Multiplexed Compressed RTP).* Los resultados cuantifican la mejora que se obtiene usando multiplexión en términos de probabilidad de admisión y calidad en la conversación.**

***Palabras Clave*- telefonía IP, Factor R, VoIP, QoS, centralita *software*, multiplexión RTP**

I. INTRODUCCIÓN

En los últimos años muchas empresas están sustituyendo sus antiguos sistemas de telefonía basados en Red Telefónica Conmutada (RTC) por otros nuevos que usan IP, de manera que las llamadas y las videoconferencias se puedan establecer a través de redes de datos. Uno de los objetivos es la reducción de costes, que se puede lograr aprovechando las conexiones a Internet existentes, para transmitir vídeo, voz y datos.

Muchas de estas empresas tienen sus recursos descentralizados, de manera que cada oficina o sucursal es independiente del resto y se encarga de gestionar su conexión a Internet y líneas telefónicas. Si se realizase una gestión centralizada de los recursos globales de la empresa, se podrían ahorrar costes realizando parte de las llamadas a través de Internet, manteniendo, e incluso aumentando, la Calidad de Servicio *(Quality of Service,* QoS) mediante el control de la probabilidad de admisión de las llamadas y de sus parámetros de calidad.

Como solución centralizada, se puede usar una centralita *(Private Branch eXchange,* PBX) *software* para unir los sistemas de telefonía de cada oficina, creando un sistema que cuenta con varias ventajas, como compartir las líneas de diferentes sucursales. Esto mejoraría los costes, ya que las llamadas internacionales se podrían establecer en dos tramos: uno utilizando Internet, hasta el país destino, y otro con llamada local desde la sucursal en ese país hasta el usuario final.

La Voz sobre IP (*Voice over Internet Protocol,* VoIP) es un servicio en tiempo real, que en muchos casos utiliza una red que fue diseñada para servicios *best-effort*. Sin embargo, los usuarios desean una calidad similar a la que acostumbran a tener utilizando los sistemas de telefonía RTC tradicionales. Este hecho ha llevado a los investigadores a buscar soluciones para añadir calidad a las redes IP. Una de las más utilizadas es el Control de Admisión de Llamadas *(Call Admission Control,* CAC), que acepta o rechaza las nuevas llamadas para evitar la degradación del servicio. Más específicamente, en este trabajo se usará el CAC basado en parámetros [1], que cuenta el número de llamadas que hay establecidas simultáneamente, y solamente acepta las nuevas llegadas si ese número está por debajo de un límite marcado por la calidad mínima aceptable para una llamada.

En el presente trabajo estimaremos la calidad de las conversaciones mediante el Factor R, propuesto por la norma ITU G.107 [2]. El Factor R va desde 0 hasta 100, y los valores que se consideran aceptables son R > 70. Se puede convertir fácilmente en MOS *(Mean Opinion Score)*, que va desde 0 (mala calidad) hasta 5 (muy buena).

En el escenario propuesto ocurrirá con frecuencia que varias llamadas tendrán el mismo origen y destino (Fig. 1). Por tanto, el uso de técnicas de multiplexión puede suponer una mejora. Si introducimos en el mismo paquete IP las muestras de diferentes llamadas, podremos ahorrar ancho de banda añadiendo pequeños retardos. En este trabajo usaremos el RFC 4170: TCRTP *(Tunneling Multiplexed Compressed RTP)* [3].

Para poder estudiar los beneficios de los sistemas de telefonía distribuidos, se construyó un escenario diseñado en trabajos previos [4], similar al de una empresa con sucursales

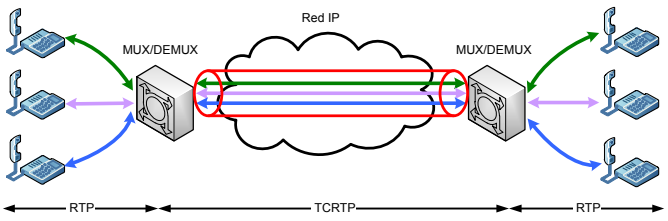


Fig. 1. Llamadas que comparten las mismas sucursales origen y destino

ISBN: 978-84-694-5948-5

en diferentes áreas o países. En sus primeras etapas se implementó mediante emulación en un *testbed*, pero este hecho introducía una limitación de tamaño causada por el número de máquinas que se requerían.

Para solucionar este problema se ha usado simulación, que puede ayudarnos a estudiar el comportamiento del sistema cuando las líneas se comparten entre un número de sucursales y se usa multiplexión RTP. Para evaluar los resultados nos centraremos en la probabilidad de admisión y el Factor R. Como parámetros de simulación se usarán los resultados obtenidos previamente en el *testbed,* especialmente retardos y pérdidas de paquetes, utilizando por tanto un método de pruebas híbrido, que integra simulación y emulación.

El presente trabajo está organizado de la siguiente manera: la sección II trata sobre los trabajos relacionados. La arquitectura del sistema se presenta en la sección III. La siguiente sección detalla la plataforma utilizada. La sección V presenta las pruebas que se han realizado y los resultados. Las conclusiones cierran el trabajo.

II. Trabajos Relacionados

En la actualidad, en entornos empresariales se está tendiendo a introducir nuevas soluciones como VoIP, en parte buscando disminuir costes. En [5] Intel publicó los resultados de un programa piloto en el que un grupo de empleados utilizó VoIP basada en SIP (*Session Initiation Protocol*) durante unos meses. La conclusión fue que esta tecnología es beneficiosa para las empresas, en términos de costes y también de productividad. En [6] se puede encontrar otro estudio que ilustra las mejoras obtenidas al usar VoIP en lugar de telefonía tradicional. Se obtenían ahorros en costes de equipos, aprovisionamiento, facturación, mantenimiento y servicio, y se recomendaba VoIP como la nueva solución de telefonía para las empresas.

Los sistemas CAC se pueden clasificar en dos categorías [7]: por un lado, los basados en medidas toman sus decisiones en función del estado de la red; por otro, los basados en parámetros requieren la realización de una serie de medidas durante la puesta en marcha del sistema, para obtener los parámetros que luego regirán su funcionamiento, como el número máximo de llamadas simultáneas.

Respecto a las políticas del *buffer* de salida de los *router* en el sistema, podemos decir que en los últimos años la regla usada tradicionalmente para dimensionarlos era el uso del producto del ancho de banda por el retardo. Pero esta regla ha sido cuestionada por el llamado *"Stanford model"*. En [8] se presentó una comparativa y se sugirió el uso de un *buffer* limitado en tiempo. En este trabajo usaremos esta política, ya que es muy adecuada para mantener los retardos por debajo de una cota superior.

III. Arquitectura del Sistema

Probaremos el sistema en modo *original*, en el que cada flujo RTP es independiente del resto, y en modo *multiplexión*, que multiplexa los flujos usando TCRTP. El escenario se corresponde con el de una empresa con sucursales en varias áreas geográficas. Este escenario es el equivalente al de algunas soluciones comerciales [9].

Como puede verse en la Fig. 2, cada oficina tiene un conjunto de usuarios, y se conecta a RTC mediante una pasarela o *gateway.* También dispone de un acceso a Internet, y las llamadas de VoIP basadas en SIP son controladas por un sistema CAC. En el centro de datos hay una PBX *software* que también está conectada a la red IP. Se utiliza Internet para el tráfico telefónico, evitando de esta manera los costes de las líneas dedicadas.

En cada sucursal existe un agente local que incluye un *proxy* SIP que implementa las decisiones de admisión de llamadas. Se realizan una serie de medidas durante la puesta en marcha del sistema, y como resultado se asigna un número máximo de llamadas simultáneas permitidas a cada sucursal.

Como se ha comentado anteriormente, se usa la técnica de multiplexión TCRTP. La Fig. 3 presenta el esquema de un paquete multiplexado. En primer lugar se usa ECRTP para comprimir las cabeceras IP/UDP/RTP. Posteriormente, se utiliza PPPMux para multiplexar varios paquetes dentro de otro, que se envía usando un túnel L2TP. ECRTP se usa extremo a extremo porque, al usar un túnel, no es necesario descomprimir y volver a comprimir las cabeceras en cada salto.

Cuando se usa el modo *multiplexión*, el ancho de banda ocupado por las llamadas será menor que el que ocuparían si se enviasen por separado. El sistema CAC debe tener esto en cuenta para tomar las decisiones de admisión, puesto que ya no puede contar las llamadas de la misma manera que en el modo *original*. Nuestra primera idea fue contar el ancho de banda de cada túnel según el número de llamadas que contiene, con lo que el sistema CAC limitaría el ancho de banda de cada sucursal en lugar del número de llamadas. De acuerdo con la Fig. 3, podemos obtener el ancho de banda de un túnel con *k* llamadas de la siguiente manera:

$$E[BW_k] = E[PS_k] / IPT = (CH + k\,(MH + E[RH] + S)) / IPT \qquad (1)$$

Donde:

- **CH:** Cabecera Común *(Common Header)*: Corresponde a las cabeceras IP/L2TP/PPP, que ocupan 25 bytes.
- **MH:** Cabecera de Multiplexión *(Multiplexing Header)*. Es la cabecera PPPMux (2 bytes).

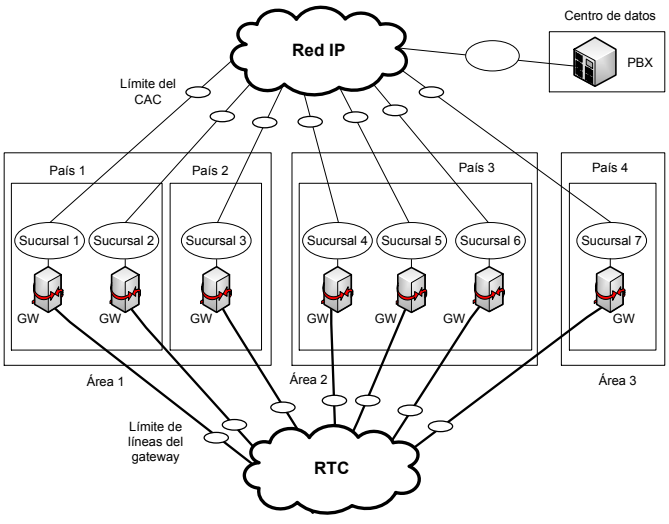


Fig. 2. Arquitectura del sistema

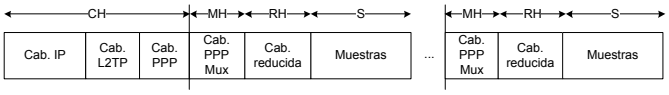


Fig. 3. Esquema de un paquete multiplexado

- **RH:** Cabecera Reducida *(Reduced Header)*. Precede a las muestras de cada flujo RTP. Los protocolos de ompresión producen cabeceras comprimidas de diferentes tamaños, por lo que *E[RH]* se calculará según la probabilidad de obtener una cabecera de cada tamaño [10].
- **S:** Tamaño de las muestras *(Samples)*. Tamaño en bytes de las muestras que lleva cada paquete RTP. En nuestro caso tendrá un valor de 20 bytes, al usar el *codec* G729a con 2 muestras por paquete.
- **IPT:** Tiempo entre paquetes *(Inter Packet Time)*.

Podemos ver que (1) crece linealmente con *k,* por eso finalmente se ha decidido que el sistema CAC cuente el número de llamadas establecidas, independientemente del ancho de banda de los túneles.

En trabajos previos este sistema se probó en un *testbed* utilizando emulación en tiempo real, pero en este trabajo utilizaremos también simulación. Una ventaja de la simulación es que el número de sucursales del escenario puede ser mayor, como ocurría en [1]. Por otro lado, la emulación en tiempo real permite obtener otros valores, como el Retardo en un Sentido *(One Way Delay,* OWD) y las pérdidas de paquetes, necesarios para conocer el máximo número de llamadas que se pueden establecer simultáneamente con una calidad aceptable. Con estos parámetros se puede calcular el Factor R utilizando el E-Model [2].

IV. Plataforma de Pruebas

En primer lugar, hemos utilizado el *testbed* presentado en [11] para implementar el escenario. Se usa un sistema híbrido para las pruebas, combinando emulación y simulación. Lo explicamos con más detenimiento a continuación. Una prueba se divide en las siguientes etapas, como se ve en la Fig. 4:

- Emulación, en la que se usan tres máquinas. En primer lugar, el generador de tráfico envía paquetes RTP y tráfico de fondo. Después, estos tráficos atraviesan el *router*, que emula diferentes políticas de *buffer*. Finalmente, una máquina recibe y almacena el tráfico.
- Procesado *offline*, en el que se añaden algunos retardos al tráfico capturado en la fase anterior.
- Simulación del escenario global y resultados finales. Se usa Matlab para simular el escenario y se utilizan los resultados anteriores para obtener la probabilidad de admisión y el Factor R de las llamadas.

A. Red de emulación

Se ha usado el *testbed* para implementar una versión reducida del sistema, y así obtener los parámetros que determinan la QoS, y que serán incluidos después en las simulaciones. Se utiliza un generador para enviar tráfico de fondo y saturar de distintas maneras el acceso de cada sucursal. Para limitar el ancho de banda del acceso se utiliza la herramienta Linux *traffic control (tc)*.

B. Procesado offline

Se han añadido al sistema los siguientes retardos: el de paquetización, que depende del *codec*; el tiempo de retención en el multiplexor, que se debe a que los paquetes deben esperar hasta que han llegado todos los paquetes a multiplexar. El retardo de encolado depende de la política del

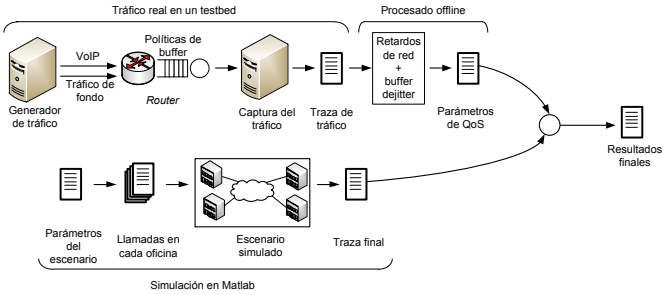


Fig. 4. Diagrama del sistema de medidas

buffer que se esté usando. También se ha incluido un *buffer* de *dejitter* en la máquina destino, que añade nuevos retardos.

C. Simulación

Se ha utilizado Matlab para generar realizaciones con diferentes parámetros: número de oficinas, usuarios, líneas de los *gateway*, países, zonas geográficas, retardos de establecimiento, etc. Las llamadas se generan según una distribución de Poisson. Su duración se ha modelado con una distribución Normal de media 180 seg. y varianza 30.

Posteriormente, cada realización se puede simular utilizando dos algoritmos. El primero, denominado *aislado* se usa como referencia. No implementa el sistema CAC, y cada sucursal es independiente del resto, por lo que el sistema no comparte los *gateway* y no se redirigen llamadas. El segundo algoritmo, denominado *compartido*, simula un sistema centralizado, que comparte todas las líneas de sus *gateway* entre las sucursales e implementa un sistema CAC, permitiendo al agente local redirigir llamadas a otras oficinas, buscando siempre la tarifa más barata.

El modo *compartido* se usa para intercambiar la probabilidad de bloqueo en los *gateway* por probabilidad de bloqueo en el acceso a Internet, ya que normalmente es más barato contratar ancho de banda que aumentar el número de líneas. Por tanto, es de esperar que la probabilidad de admisión aumente a causa de las redirecciones, y que se consiga ahorrar costes en llamadas internacionales. Sin embargo, incrementar la probabilidad de admisión también tiene una contrapartida, y es que se introduce más tráfico en el acceso a Internet, por lo que la QoS puede verse afectada.

V. Pruebas y Resultados

A. Probabilidad de admisión al compartir las líneas

La Fig. 5 muestra cómo al usar el modo *compartido*, a mayor número de oficinas se consigue una mayor probabilidad de admisión. Se han incluido 25 usuarios por sucursal. Esto se corresponde con la idea de la fórmula de Erlang de que se consigue más probabilidad de admisión cuantos más recursos se comparten. Naturalmente, si la tasa de llamadas por hora y usuario λ se hace muy grande, la probabilidad de admisión se reduce.

Hemos realizado otras pruebas variando el límite del CAC. En la Fig. 6 se puede ver que el modo *compartido* da mejores resultados para la probabilidad de admisión que el modo *aislado*, por lo que se confirma que compartir las líneas es beneficioso. Por otro lado, vemos que según aumenta el límite del CAC, en el modo *compartido* aumenta la probabilidad de admisión, mientras que en el modo *aislado* no varía.

B. *Factor R*

Se ha configurado un escenario con cuatro zonas en el mismo país, y una sucursal en cada una. Existen 25 usuarios en cada oficina. Se ha usado un tráfico de fondo que satura el acceso en un 80%. Los parámetros variables son la tasa de generación de llamadas y el límite del CAC de las sucursales.

La Fig. 7 compara los dos modos en términos de Factor R en función del límite del CAC. En primer lugar, puede verse que la multiplexión supone una mejora para los mismos valores de λ y del límite del CAC. Este hecho se debe a que la cantidad de tráfico enviado en el modo *multiplexado* es menor que en el *original*. Por tanto, usando multiplexión, el límite del CAC puede ser mayor si se desea obtener el mismo valor del Factor R.

Por otra parte, si mantenemos λ fija, la figura muestra que a mayor límite del CAC, peor será la calidad en términos de R. Obviamente, incrementar el número de llamadas simultáneas sin aumentar el ancho de banda hará que la calidad disminuya.

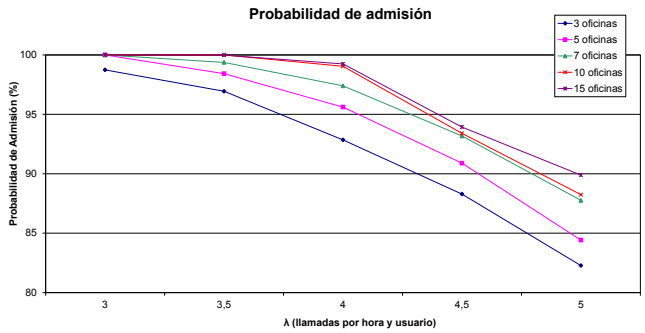


Fig. 5. Probabilidad de admisión en modo *compartido*

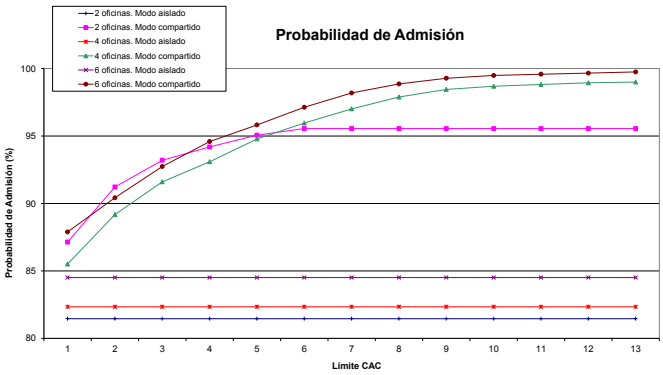


Fig. 6. Probabilidad de admisión en modos *aislado* y *compartido*

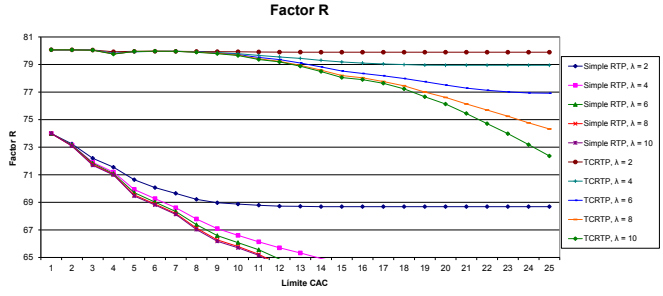


Fig. 7. Factor R en modos *original* y *multiplexado* en función del límite del CAC. No se representan los valores bajos del Factor R, porque se considera inaceptable por debajo de 70

Observamos que existe un compromiso entre la probabilidad de admisión y la calidad de las llamadas. Por tanto, se podría sacrificar parte del margen del Factor R que nos aporta la multiplexión, para lograr así mejores valores de probabilidad de admisión.

VI. Conclusiones

En este trabajo se ha estudiado un sistema de telefonía basado en VoIP y que usa el protocolo SIP. El escenario se corresponde con el de una empresa con oficinas en diferentes zonas. Se ha incluido un sistema CAC que limita el número simultáneo de llamadas. Se ha usado un sistema híbrido de pruebas, obteniendo en un *testbed* en primer lugar los parámetros de QoS, y después usándolos como entradas para las simulaciones del escenario completo.

Por un lado, se quería comprobar si compartiendo las líneas de las sucursales es posible mejorar la probabilidad de admisión. Por otro lado, queríamos comparar el comportamiento del sistema en términos de calidad de las llamadas, si se usa RTP simple o túneles TCRTP.

Se han realizado simulaciones del sistema en diferentes situaciones. Los resultados obtenidos muestran que la probabilidad de admisión aumenta si se comparten las líneas entre sucursales. También se observan mejoras en el Factor R cuando se usa multiplexión TCRTP, que se pueden traducir también en mejoras de la probabilidad de admisión.

Agradecimientos

Este trabajo ha sido financiado parcialmente por el Proyecto CPUFLIPI (MICINN TIN2010-17298), por el Proyecto MBACToIP, de la Agencia I+D del Gobierno de Aragón e Ibercaja Obra Social, y por el Proyecto NDCIPI-QQoE de la Cátedra Telefónica, de la Univ. de Zaragoza.

Referencias

[1] S. Wang, Z. Mai, D. Xuan, and W. Zhao, "Design and implementation of QoS-provisioning system for voice over IP," Parallel and Distributed Systems, IEEE Transactions on, vol.17, no3, pp. 276--288, 2006

[2] ITU-T Recommendation G.107, "E-model, a computational model for use in transmission planning," 2003

[3] B. Thompson, T. Koren, and D. Wing, "RFC 4170: Tunneling Multiplexed Compressed RTP (TCRTP)," 2005

[4] J. Saldana, J. Aznar, E. Viruete, J. Fernández-Navajas, and J. Ruiz-Mas, "QoS Measurement-Based CAC for an IP Telephony System," QShine 2009, The Sixth International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness. Las Palmas de Gran Canaria (Spain), 2009

[5] S. Sacker, M. Santaiti, and C. Spence, "The Business Case for Enterprise VoIP," Intel Corporation, 2006

[6] B. Athawal, "Replacing Centric Voice Services with Hosted VoIP Services: An Application of Real Options Approach"

[7] R. Solange, P. Carvalho, and V. Freitas, "Admission Control in Multiservice IP Networks: Architectural Issues and Trends," IEEE Communications, vol.45, no. 4, pp. 114--121, 2007

[8] A. Dhamdhere, and C. Dovrolis, "Open issues in router buffer sizing," Comput. Commun. Rev., vol. 36, no. 1 pp. 89--92, 2006

[9] VoIP Call Admission Control, http://www.cisco.com/en/US/docs/ios/solutions_doc/voip_solutions/ CAC.pdf

[10] G. Dimitriadis, S. Karapantazis, F.-N. Pavlidou, "Comparison of Header Compression Schemes over Satellite Links", In Proc. International Workshop on IP Networking over Next-generation Satellite Systems (INNSS'07), Budapest, Hungary, Jul 2007.

[11] J. Saldana, E. Viruete, J. Fernández-Navajas, J. Ruiz-Mas, and J. Aznar, "Hybrid Testbed for Network Scenarios," SIMUTools 2010, the Third International Conference on Simulation Tools and Techniques. Torremolinos (Spain), 2010

Selección distribuida y dinámica de portales en redes malladas inalámbricas

A. Triviño Cabrera, A. Ariza Quintana, E. Casilari Pérez
Departamento de Tecnología Electrónica, E.T.S.I. Telecomunicación
Universidad de Málaga
29071 Málaga, Spain.
{aarizaq,atc,ecasilari}@uma.es.

***Resumen*- Las redes inalámbricas malladas pueden estar equipadas con múltiples portales de acceso que ofrecen conectividad hacia otras redes externas como puede ser Internet. Estos elementos disponen, al menos, de dos interfaces: una cableada, que permite conectarse con otros portales y pasarelas, y otra para la conexión inalámbrica. Permitiendo la conexión de los portales en este trabajo resolvemos el problema de la selección dinámica del punto de acceso a Internet en una red mallada inalámbrica. Esta propuesta se fundamenta en la definición de grupos *anycast* formados por los portales. Cualquier trama al exterior de la red se envía a cualquiera elemento de este grupo. La ruta se selecciona dinámicamente en función del estado instantáneo de la red. Para conseguir que este proceso sea distribuido, se usa un mecanismo de selección de rutas y de envío de paquetes en el nivel de enlace. Por medio de simulaciones se ha podido constatar una ganancia en el rendimiento de la red al emplear la propuesta.**

***Palabras Clave*- Encaminamiento, redes inalámbricas malladas, *anycast*, selección de portales, 802.11s.**

I. INTRODUCCIÓN

Las comunicaciones inalámbricas se han convertido en un eficaz sistema para el intercambio de información entre máquinas. Esta expansión ha dado lugar a que cada vez más dispositivos estén equipados con varios interfaces radio: uno para conexión con el operador y otro para la conexión a través de las bandas sin licencias. De hecho, en la actualidad se están llevando diversas experiencias para proveer de una forma económica servicios de Internet mediante el uso de la tecnología 802.11 [1]. Dada esta popularidad, el IEEE ha desarrollado un nuevo protocolo (802.11s) para el desarrollo e implementación de redes inalámbricas malladas [2]. De acuerdo a este estándar, una red *mesh* inalámbrica está formada por una serie de nodos estáticos los cuales se encuentran interconectados entre sí mediante enlaces inalámbricos formando una especie de red troncal multi-salto. Estos nodos, que se denominan MP (*Mesh Points*) implementan dentro del nivel de enlace tanto el mecanismo de "*forwarding*" como el de búsqueda de camino. Para ello, estos nodos ejecutan un protocolo de encaminamiento basado en el nivel de enlace denominado HWMP (*Hybrid Wireless Mesh Protocol*), pudiendo este ser reemplazado opcionalmente por el protocolo OLSR [3]. Sin embargo, algunos MP pueden ofrecer funcionalidades adicionales. En este sentido, algunos MP pueden proporcionar servicios de punto de acceso a terminales móviles o estaciones (STA), denominándose en este caso MAP (*Mesh Access Points*), mientras que otros proporcionan acceso a otras redes, denominándose en este caso MPP (*Mesh Portals*). Para poder comunicarse con otras redes distintas de la formada por los nodos *mesh*, como puede ser Internet, los STA o los clientes móviles deben poder conectarse a cualquier MAP de la red o tener implementado en su nivel de enlace el protocolo 802.11s y posteriormente, encontrar un camino hacia uno de los MPP disponibles en la red.

En una red inalámbrica la selección del nodo MPP con el que conectarse hacia Internet tiene un impacto directo en el rendimiento del sistema. Analizando este efecto, se han publicado algunas políticas para el uso con redes 802.11s y redes de acceso IP. Para describir las peculiaridades de las propuestas existentes de una forma genérica, a los MPP y a los *Gateways* los llamaremos nodos de interconexión. Los nodos de interconexión proporcionan diferentes funcionalidades según las distintas propuestas. De forma básica podemos distinguir cinco características básicas para describir el método de acceso a los nodos de interconexión: (i) cómo se descubre el nodo de interconexión, (ii) si se descubre mediante procedimientos *anycast* (iii) posibilidad de usar múltiples nodos de interconexión simultáneamente, (iv) la selección del nodo de interconexión y (v) cómo los diferentes nodos de interconexión colaboraran entre sí en el encaminamiento dentro de la propia red mallada (tráfico *intra-mesh*). En lo relativo al procedimiento de transmisión *anycast*, la mayoría de los trabajos previos restringen esta funcionalidad exclusivamente a la fase de descubrimiento de los nodos de interconexión. Una vez que éstos son identificados, se selecciona uno y todas las conexiones se realizan siempre a través de éste. Este método restringe, pues, la flexibilidad de explotar, durante la transmisión de las tramas, el disponer de múltiples nodos de interconexión. En este trabajo proponemos un mecanismo de selección dinámica del portal. Para probar las prestaciones que ofrece nuestra propuesta hemos realizado diversas simulaciones en OMNeT++ [4] y los resultados confirman un considerable aumento del rendimiento.

El resto de este trabajo está organizado de la siguiente manera. En la Sección II se presentan diversos trabajos relativos al tema. En la Sección III se describe nuestra propuesta. El rendimiento de dicha propuesta se evalúa en la Sección IV. Finalmente, en la Sección V se presentan las conclusiones finales de este trabajo.

II. ESTADO DEL ARTE

Las propuestas para usar múltiples nodos de conexión propuestas pueden clasificarse de acuerdo a las siguientes características:

A. Descubrimiento de los nodos de interconexión. Este procedimiento hace referencia al método empleado para iniciar el proceso de descubrimiento de estos nodos y pueden clasificarse como reactivos, proactivos o híbridos. En el procedimiento reactivo el nodo que necesita acceder a Internet genera un mensaje de descubrimiento siempre que no disponga en sus tablas de rutas ninguna información válida, tal y como se muestra en [5]. Por otro lado, los sistemas proactivos usan una emisión periódica anunciando los nodos de interconexión. Los nodos adquieren de esta forma las rutas necesarias para la transmisión de sus datos hacia los nodos de interconexión. Las propuestas [6] y [7] usan este tipo de solución. Por último, los sistemas híbridos usan un aviso periódico dentro de una determinada área establecida por el número máximo de veces que los mensajes de anuncio pueden ser retransmitidos mientras que los nodos fuera de este are usan un mecanismo reactivo. Un ejemplo de este tipo de soluciones es [8].

B. Capacidad para soportar descubrimiento de rutas por procedimientos anycast. Existen varias propuestas que definen un grupo *anycast* al cual los nodos de interconexión se encuentran asociados. En este caso, los procedimientos de descubrimiento de ruta son enviados al grupo *anycast*. En redes IP esta capacidad está incorporada en [8]. Así mismo, '*One Laptop Per Child*' (OLPC) propone que el descubrimiento de los nodos MPP se realice usando una dirección MAC *anycast* reservada a tal efecto [9].

C. Capacidad de asociarse a múltiples nodos de Interconexión. Cuando varios nodos de interconexión están disponibles algunos investigadores proponen dividir los flujos de Internet entre ellos de forma simultánea. Un protocolo de encaminamiento multi-camino puede soportar esta funcionalidad. Los trabajos [8] y [10] son una muestra de este tipo de soluciones.

D. Selección del nodo de interconexión. En escenarios con múltiples nodos de interconexión, es necesario establecer un criterio de selección entre las distintas alternativas. El criterio de decisión puede considerar diversos parámetros como la capacidad binaria de los enlaces radio, la sobrecarga del nodo [11] o el número de saltos [12]. Una métrica multiparámetro que combina la carga del Gateway, la interferencia entre rutas y la calidad del enlace se presenta en [9]. Así mismo en [13] se presenta una formulación matemática que permite la construcción de árboles usando como raíces los *Gateways*. La selección del nodo de interconexión afecta a la retrasmisión de los mensajes de anuncio como se muestra en [10].

III. ANYCAST EN EL NIVEL DE ENLACE PARA REDES 802.11S

Nuestra propuesta está orientada a redes inalámbricas en las cuales los procedimientos de encaminamiento y "*forwarding*" se ejecutan en el nivel de enlace, como es el caso del estándar 802.11s. La propuesta proporciona los siguientes tres mecanismos fundamentales:

A. Emulación anycast para el descubrimiento de MPP. En la actualidad no hay definido dentro de los estándares un procedimiento *anycast* usando direcciones MAC. Como excepción, OLPC propone el uso de una dirección MAC particular para esto [14]. El problema de esta solución es que todos los MP de la red deben estar preparados para poder trabajar con esta propuesta. Para evitar este problema, y permitir que en la misma red puedan convivir MP con soporte *anycast* y MP sin este soporte, proponemos el uso de un novedoso mecanismo que emula el uso de direcciones *anycast* sobre 802.11s. Este mecanismo utiliza, al igual que el protocolo 802.11s, un procedimiento proactivo para el descubrimiento de los nodos MPP. Sin embargo, en nuestra propuesta, permitimos que los MPP se comuniquen y se conozcan entre sí a través de la red cableada. De esta forma, en lugar de que cada nodo se anuncie de forma separada, permitimos que uno de los MPP de forma periódica se erija como representante del grupo e inicie la emisión periódica de mensajes de notificación de los MPP a la red inalámbrica. Este mensaje contiene información acerca de todos los MPP presentes en el grupo. Adicionalmente, un MP puede iniciar un procedimiento para descubrir los MPP presentes. Para ello el nodo emite un mensaje solicitando esta información a los nodos vecinos.

La transmisión *anycast* se emula en este trabajo permitiendo que las direcciones MAC de los MPP sean intercambiables. Con esta intención, la gestión de las direcciones intercambiables se realiza con una estructura que denominamos *Equivalent Address Table* (EAT). Esta estructura la crean y la mantienen los nodos. En esta fase la información de encaminamiento no está completa y se necesita un protocolo de encaminamiento para encontrar las rutas y determinar el coste de cada ruta hacia cada uno de los MPP. Es el coste de las rutas lo que determina la elección del MPP. La presente propuesta no requiere de ninguna métrica específica para decidir el MPP a usar. En nuestros experimentos, por simplicidad, hemos usado el número de saltos como métrica.

Una de las principales ventajas de nuestra propuesta es que puede ser gradualmente incorporada a la red mallada, ya que funciona aunque no todos los nodos implementen la solución. Los nodos que no la implementen ignorarán la información contenida en los mensajes de anuncio de los MPP, y no trabajarán con la información contenida en las EAT.

B. Envió de información mediante la emulación Anycast. Una vez que la EAT está actualizada, los nodos se configuran para soportar la retransmisión de las tramas de datos que tengan como destino una de las direcciones presentes en la tabla, bien al nodo que tiene asignada dicha dirección, o bien a cualquier otro nodo cuya dirección se encuentre en la tabla. Así pues, aunque la fuente selecciona un MPP e incluye su dirección en el campo dirección de destino de la trama, los diversos nodos que forman parte de la ruta pueden decidir que un MPP que se encuentre en la lista EAT es más adecuado para la transmisión. En este caso, el nodo procederá al cambio de direcciones, tal como se muestra en la Fig. 1. En la figura, las líneas discontinuas representan los enlaces inalámbricos, mientras que las líneas continuas representan los enlaces cableados. En este caso, el MP-A genera una trama con destino Internet. El nodo mira en su tabla de encaminamiento y decide que el MPP3 es el MPP más adecuado así que rellena la trama con la dirección de destino del MPP3 y la envía al siguiente nodo del camino, en este caso el MP-B. Este nodo detecta que la dirección de destino se encuentra dentro de la AET. Por lo tanto, esta dirección es una dirección intercambiable y procede a evaluar los costes contenidos en su tabla de encaminamiento a los diferentes nodos contenidos en esta tabla EAT. Tras esta evaluación, determina que el MPP más adecuado es el MPP2 y procede a cambiar la dirección de destino de la

trama por la de éste y, a continuación, procede a la retransmisión de la trama, pero ahora con la nueva dirección de destino. La retransmisión basada en EAT es capaz de considerar los costes instantáneos de transmisión (disponibilidad de ruta, número de saltos, pérdidas, retraso o cualquier otra métrica) asociada a los nodos alternativos y el destino actual. Es más, se consigue realizar una selección dinámica de ruta para cara trama. El MP puede usar múltiples MPP de forma que el esquema soporta multi-asociación. Adicionalmente, el uso de direcciones intercambiables también se utiliza para el envío de tramas tales como *Route Request* usadas para encontrar caminos por parte del protocolo de encaminamiento.

C. Combinación de la información de encaminamiento en los MPP. Los portales *mesh* (MPP) se conectan usualmente mediante redes cableadas de alta velocidad como Giga-Ethernet o FastGiga-Ethernet. Por ello los MPP disponen, al menos, de dos interfaces: una interfaz para la red cable y una segunda para la comunicación con los nodos de la red *mallada* inalámbrica. En nuestra propuesta ambos módulos se encuentran conectados de forma que la información de encaminamiento de ambas redes se combina.

Combinando ambas redes, los MPP pueden participar en la conexión entre dos MP que se encuentran dentro de la misma red *mesh* inalámbrica. A diferencia de otras propuestas previas, la colaboración entre MPP no está restringida al descubrimiento de rutas, sino que participan en el descubrimiento de rutas internas y en la retransmisión de las tramas usando tanto los enlaces inalámbricos como los cableados. Debido al funcionamiento integrador de redes que realizan los MPP al permitir usar la red cableada como parte de la red inalámbrica mallada, el número de saltos que las tramas deben llevar a cabo para atravesar la red (diámetro de la red) se ve considerablemente reducido. Es más, el uso de la red cableada en lugar de la red inalámbrica reduce las interferencias y la congestión del medio radio con lo que mejora el comportamiento de las comunicaciones en la propia red mallada.

Por otro lado, la combinación de ambas interfaces en los MPP también afecta al descubrimiento de los portales. Como se ha comentado previamente, los MPP son representados por uno de ellos quien es el encargado de iniciar la publicación de la lista del EAT.

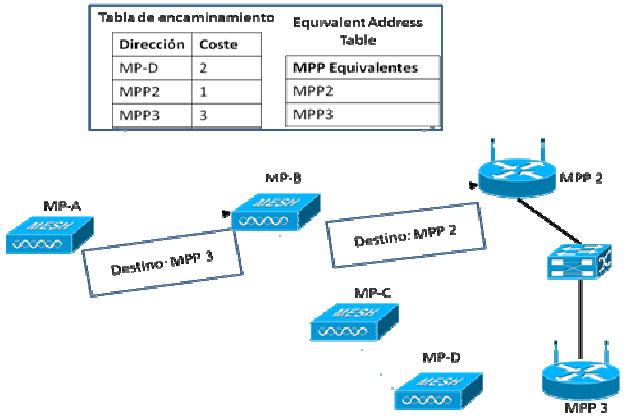


Fig. 1. Emulación *anycast* en la transmisión de un paquete. Una redirección tiene lugar en MP-B.

Este rol rota periódicamente entre los distintos MPP. Es el representante del grupo el que cada momento está encargado de iniciar la emisión con la lista que contiene todas las direcciones MAC de los MPP. Esta lista de direcciones se emite por ambas interfaces simultáneamente (cableada e inalámbrica), lo que permite una rápida difusión en la red de esta información. De una manera similar, permitimos que los mensajes de difusión y búsqueda de ruta (como son los RREP y RREQ) se propaguen también a través de la red cableada. De esta forma la comunicación entre dos nodos *mesh* cualesquiera de la red se puede beneficiar de la existencia de la red cableada.

IV. Evaluación

El rendimiento de esta propuesta se ha llevado a cabo mediante simulación sobre la herramienta de eventos discretos OMNeT++ [6]. El escenario es una representación esquemática del Campus de Teatinos de la Universidad de Málaga. El campus, y consecuentemente el área de simulación, tiene una superficie aproximada de 1245x630 m^2. A fin de evaluar nuestra propuesta hemos colocado el Gateway de acceso hacia el exterior en el centro geográfico del área y se ha configurado el nivel IP de todos los nodos para usar este Gateway como dirección por defecto. Así mismo, se han colocado otros 4 MPP adicionales en aquellos puntos donde se espera que el acceso a Internet tenga una mayor demanda como son las bibliotecas y cafeterías. Finalmente, se han distribuido un total de 77 MAP (nodos que permiten el acceso a la red *mesh*) formando una malla que cubre el resto del campus que permite un acceso inalámbrico desde cualquier punto del área de simulación. Todos los MPP están conectados entre sí con una red FastGiga-Ethernet. Además, se han añadido un total de 40 usuarios móviles que pueden acceder a la red a través de los MAP. Adicionalmente, hemos permitido que los usuarios móviles también puedan servir como estaciones retransmisoras de forma simular a la propuesta OLPC.

En lo relativo a la movilidad hemos considerado el modelo *Random WayPoint Mobility*], ampliamente conocido y usado. Para evitar el problema de velocidad media decreciente, la velocidad se ha escogido mediante una distribución uniforme entre 1 y 2 metros por segundo con un tiempo de pausa de 0 segundos. El tiempo de simulación es de 3000 segundos y de los 40 nodos 12 son generadores y receptores de tráfico. Se han realizado pruebas donde solo se ha generado tráfico hacia y desde el exterior y otras donde el 50% del tráfico generado por los nodos móviles tiene como destino otro MP mientras que el 50% restante tiene como destino Internet. El tráfico recibido desde Internet es igual, en todos los casos, al tráfico que tiene como destino Internet.

Las pruebas se han realizado con distintas cargas de tráfico ejecutando, para cada carga, un total de 5 simulaciones distintas con diferentes semillas.

Para nuestro trabajo hemos usado una versión simplificada del estándar 802.11s similar a la propuesta empleada para OLPC. Consecuentemente el protocolo de encaminamiento usado es AODV que es el mecanismo reactivo incluido en el protocolo HWMP [5]. Para nuestas simulaciones hemos usado el estándar MAC 802.11g fijando la velocidad de transmisión a 54 Mbit/s. Los parámetros de simulación se encuentran resumidos en la tabla 1.

El rendimiento de la red se cuantifica mediante dos métricas: tasa de paquetes entregados y retraso extremo a extremo. Este retraso se ha medido en el caso del tráfico

hacia Internet hasta el MPP. En nuestros experimentos evaluamos en rendimiento de usar *anycast* para el descubrimiento y transmisión de las tramas hacia el MPP. Para ello, se inyecta un tráfico desde y hacia el exterior de la red mallada. Los resultados se muestran en las Fig. 3 y 4. Los resultados se muestran con un intervalo de confianza del 95%.

Parámetro	Valor
Área de simulación	1245x630 m^2
Tiempo de simulación	3000 s
Número de experimentos para cada tráfico	5
Número de MPP	5
Número de MAP	77
Número de nodos móviles	40
Número de Fuentes	12
Periodo de anuncio de la EAT	100 s
Modelo de propagación	Doble Rayo
Área de cobertura	100 m
Conexión cable entre MPP	802.3ae

Tabla 1. Parámetros de simulación.

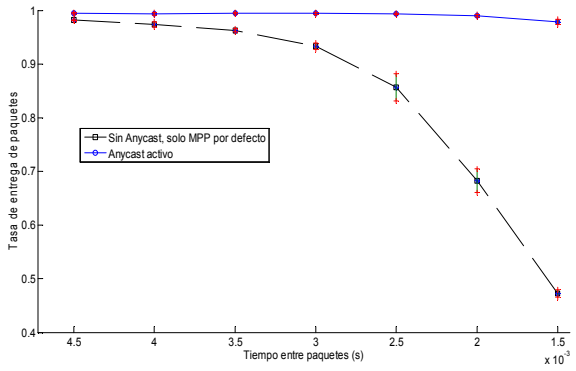


Fig. 3. Tasa de entrega de paquetes versus tiempo entre paquetes para tráfico de Internet.

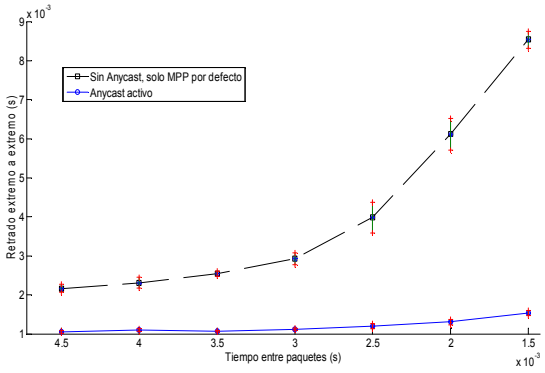


Fig. 4. Retraso extremo a extremo versus tiempo entre paquetes para tráfico de Internet.

Como se observa en las gráficas, la tasa de paquetes entregados mejora claramente con el esquema *anycast* usado. El uso de múltiples MPP junto con el esquema *anycast* usado mejora claramente en rendimiento de la red aunque todos los nodos tengan como destino por defecto el MPP situado en el centro del área de simulación.

V. Conclusiones

En este trabajo se presenta un mecanismo que emula el encaminamiento *anycast* sobre el nivel de enlace, aplicable al estándar 802.11s. Mediante este mecanismo es posible la selección dinámica de MPP (portales *Mesh*), así como la integración de los segmentos cableados que unen a estos dentro de la red inalámbrica mallada. Los resultados obtenidos por simulación de esta propuesta muestran una enorme mejora en el rendimiento, tanto a nivel de tasa de entrega de paquetes como a nivel de retraso extremo a extremo. Por otro lado, esta propuesta permite la coexistencia de nodos que la soporten y nodos sin soporte *anycast*, lo que facilita un despliegue gradual de la solución.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el ministerio de ciencia y tecnología a través del proyecto No. TEC2009-13763-C02-01.

Referencias

[1] Freifunk , International Project for free wireless networks and frequencies http://start.freifunk.net/

[2] 802.11s IEEE Draft 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 10: Mesh Networking

[3] Bahr, M.: Proposed Routing for IEEE 802.11s WLAN Mesh Networks. In: 2nd Annual International Wireless Internet Conference (WICON), ACM, Boston (2006)

[4] Varga, A.:OMNeT++ User Manual, http://www.omnetpp.org/

[5] Lakshmanan, S., Sivakumar, R., Sundaresan, K.: Multi-gateway association in wireless mesh networks, Ad Hoc Networks, vol. 7, Issue 3, pp. 622-637 (2009)

[6] Ashraf, U., Abdellatif, S., Juanole, G.: Gateway Selection in Backbone Wireless Mesh Networks. In: IEEE conference on Wireless Communications & Networking Conference (WCNC), Budapest (2009)

[7] Isabwe, G. M. N., Kim, K.-S.: A Novel Approach to WLAN Mesh Interworking with Multiple Mesh Portals. In: 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS), 641 – 646, Atlanta (2008).

[8] Sharif, K.; Cao; L., Wang, Y., Dahlberg, T.: A Hybrid Anycast Routing Protocol for Load Balancing in Heterogeneous Access Networks. In: 17th International Conference on Computer Communications and Networks (ICCCN), pp. 1-6, St. Tomas (2008).

[9] The One Laptop per Child project, http://laptop.org/en/

[10] Hu, Y., He., W, Yang, S., Zhou, Y.: Multi-Gateway Multi-Path Routing Protocol for 802.11s WMN. In: IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp. 308-315, Niagara Falls (2010)

[11] Ancillotti, E., Bruno, R., Conti, M.: Load-balanced routing and gateway selection in wireless mesh networks: Design, implementation and experimentation. In: IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), Montreal (2010)

[12] Song, W., Fang, X-M.: Design and Simulation of Fairness-aware Routing Algorithm in Wireless Mesh Networks, System Simulation, pp. 4320-4325 (2007)

[13] Papadaki, K., Friderikos, V.: Joint Routing and Gateway Selection in Wireless Mesh Networks. In: IEEE conference on Wireless Communications & Networking Conference (WCNC), pp. 2325, Las Vegas (2008)

Comparación de prestaciones de redes móviles 3G con EURANE

H. Barrientos González, M. Solera-Delgado, M. Toril-Genovés, F. Ruiz Vega, A. Durán Martínez
Departamento de Ingeniería de Comunicaciones
Universidad de Málaga
ETSI Telecomunicación. Campus Teatinos s/n. 29071 Málaga
msolera@ic.uma.es

***Resumen*- La simulación de redes de comunicaciones móviles es una de las técnicas más utilizadas para el estudio y el análisis de nuevas tecnologías radio debido a la dificultad para trabajar con escenarios reales. Entre los simuladores de red móvil de dominio público destaca EURANE, que amplía la funcionalidad del simulador de redes ns-2 dotándolo de las interfaces radio de UMTS y HSDPA. Sin embargo, EURANE no incluye ningún modelo para redes LTE. En este artículo se propone una primera aproximación para el estudio de estas tecnologías mediante el modelado de un sistema pre-LTE para EURANE. El desarrollo de este módulo se basa en la modificación de la arquitectura de la interfaz radio a partir de la creación de un nuevo nodo LTE (pre-eNB), pero que todavía mantiene las características de la capa física de HSDPA. A partir de esta herramienta se comparan las prestaciones de las tecnologías móviles 3G.**

***Palabras Clave*- UMTS, HSDPA, LTE, EURANE, ns-2**

I. INTRODUCCIÓN

Operadores e investigadores han dedicado gran parte de sus esfuerzos a mejorar el funcionamiento de las redes móviles de tercera generación (3G). Prueba de ello es el proyecto EURANE [1] *(Enhanced UMTS Radio Access Network Extensions for ns-2)*, desarrollado como una extensión del simulador de redes ns-2 [2] para analizar el comportamiento de redes 3G. Este proyecto se centra en el estudio del rendimiento extremo a extremo de las redes UMTS y HSDPA a partir del desarrollo de un simulador de red que modela la red de acceso radio de UMTS.

Desafortunadamente, EURANE no contempla modelos de simulación de tecnologías más avanzadas. En este trabajo, se plantea la ampliación de EURANE para considerar tecnologías radio más modernas basadas en los estándares del 3GPP *(3rd Generation Partnership Project)*. Para el análisis de la tecnología LTE, como primera aproximación, se ha partido de la herramienta EURANE para desarrollar un sistema pre-LTE. Para ello, se ha programado una extensión del simulador que evoluciona la arquitectura de la interfaz radio de HSDPA a LTE, pero manteniendo las características de la capa física de la primera. Con esta herramienta básica, se compara el rendimiento de las tecnologías 3G mediante simulación. Durante el análisis, se evalúan diferentes escenarios que consideran distintos modelos de movilidad, servicio y gestión de recursos radio.

El artículo se organiza de la siguiente manera. En primer lugar, se describe cómo EURANE modela UMTS y HSDPA, para en la siguiente sección explicar cómo se amplía el simulador para crear un sistema pre-LTE. A continuación, se describen los escenarios de simulación considerados y se presentan los resultados obtenidos. Por último, se resumen las conclusiones del trabajo.

II. MODELO DE SIMULACIÓN EN EURANE

EURANE es una extensión del simulador de red ns-2 que permite la simulación de UMTS R99 y HSDPA. Diseñado en el marco del proyecto SEACORN [3], su principal fin es servir de herramienta básica para la evaluación de posibles mejoras a la especificación original de UMTS.

A. Modelo UMTS

En la Fig. 1 se muestra el modelo de simulación de UMTS implementado en EURANE. Este modelo se construye sobre las funcionalidades de Aplicación, Transporte y Red ofrecidas por ns-2. Sin embargo, las funcionalidades de la capa PDCP *(Packet Data Convergence Protocol),* encargada principalmente de la compresión/descompresión de la cabecera IP, no están implementadas. En el modelo, la capa RLC desarrolla dos de los tres modos RLC definidos en UMTS: AM y UM. El modo TM no está implementado porque está orientado a servicios de conmutación de circuitos.

La simulación de entidades UM y AM considera la mayoría de las funciones definidas en el estándar [4]. Para el modo UM, se realizan las funciones de segmentación y reensamblado de paquetes, concatenación, relleno de bits, transferencia de datos de usuario y verificación de números de secuencia. Para el modo AM, se añaden la entrega ordenada de paquetes a capas superiores, detección de tramas duplicadas y control de flujo.

En cuanto a la capa MAC, se integran las funciones de conmutación de canales (en servicios portadores best-effort permite conmutar a los UEs desde el canal dedicado al canal común para que más usuarios compartan los recursos radio), identificación de los usuarios (creando, asignando y eliminando identificadores de UEs), selección del formato y el mapeado de los canales lógicos en canales de transporte y viceversa.

En la capa física, se implementa una versión simplificada de la trama UMTS y se define el modelo de propagación radio. Para UMTS el canal se modela con una probabilidad de error de paquete basada en los modelos de error que proporciona ns-2 (distribución uniforme, etc). En el simulador se han definidos solamente tres canales físicos que están mapeados uno a uno con los canales de transporte

ISBN: 978-84-694-5948-5

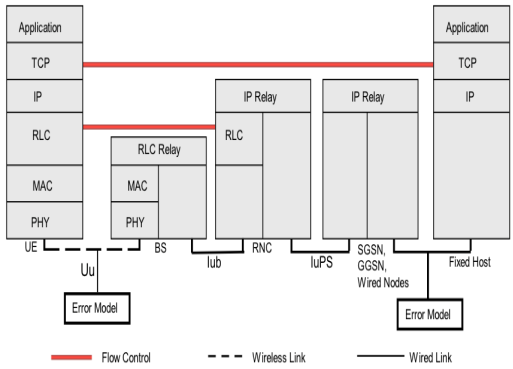


Fig. 1. Modelo UMTS en el EURANE

Forward Access Channel (FACH), *Random Access Channel (RACH)* y *Dedicated Transport Channel (DCH)*, respectivamente.

Además el diseño del simulador permite identificar interfaces y protocolos sobre los que se generan trazas de paquetes.

B. Modelo HSDPA

Para la simulación de HSDPA, se implementa una entidad MAC-hs, que soporta el nuevo canal de transporte y funciones propias de esta especificación. También aparece el nuevo canal físico HS-PDSCH, mapeado sobre el canal de transporte HS-DSCH que se crea entre BS y los UEs. Aunque el simulador en HSDPA no especifica un modelo para el plano de control, la planificación rápida de recursos y los mecanismos HARQ requieren retroalimentación desde los UEs. Es por ello que se requiere un modelo de canal más complejo que el utilizado en UMTS. El modelo de propagación radio básico incluido en el simulador para HSDPA corresponde al especificado en la norma del 3GPP. Se definen diferentes perfiles de potencia de banda ancha (vehicular, pedestre, interior, de visión directa, rural y urbano). Los perfiles de potencia permiten simular el efecto del canal cuando se transmiten señales de banda ancha, es decir, cuando el período de símbolo es comparable a los retardos de las diferentes componentes multicamino (lo que puede verse en el dominio de la frecuencia como señales cuyo ancho de banda supera al ancho de banda de coherencia del canal). Aquellas componentes multicamino cuyos retardos son indistinguibles entre sí (diferencias mucho menores que el período de símbolo) se suman en fase en cada eco recibido, dando lugar a interferencias constructivas y/o destructivas que pueden provocar desvanecimientos rápidos o instantáneos de las amplitudes de los distintos ecos del perfil. La norma tiene esto último en cuenta combinando los perfiles con los desvanecimientos rápidos para cada componente del perfil de potencia simulado. Los desvanecimientos rápidos, producto de la propagación multicamino que experimenta cada eco del perfil tal como se ha mencionado antes, se simulan mediante procesos de amplitud con distribución de Rayleigh y cuya forma espectral corresponde a las distintas desviaciones Doppler en frecuencia (función de la longitud de onda y la velocidad de desplazamiento de los móviles) de las componentes multicamino, debido a que cada una de ellas incide sobre la antena receptora con un ángulo de llegada diferente. A estos desvanecimientos rápidos se superponen variaciones lentas que reflejan fenómenos de obstrucción temporal por obstáculos de grandes dimensiones (ensombrecimiento). La magnitud de estos desvanecimientos lentos (expresada en dBs) se modela con una distribución lognormal y una cierta correlación espacial. Finalmente se calculan las pérdidas medianas de propagación siguiendo una ley de atenuación por distancia con modelo exponencial [5].

La MAC-hs incluye los tres algoritmos de scheluduling: Round Robin, Maximum C/I y Fair Channel Dependent Scheduling. En cada TTI *(Transmission Time Interval)*, la MAC-hs comprueba el valor de los CQI *(Channel Quality Idicator)* de cada usuario y dependiendo del algoritmo de planificación determina qué paquetes serán enviados en cada instante.

C. Modelo pre-LTE

El sistema pre-LTE descrito a continuación es una simplificación de LTE, que se concentra en los cambios de la estructura de red, dejando de lado los aspectos físicos de la transmisión. El modelo se basa en la creación de un nuevo nodo *pre-evolved NodeB (pre-eNB),* que incluya las funciones de HSDPA integradas actualmente en el RNC. Con ello, se consigue simplificar la estructura de red HSDPA, de la misma forma que en LTE.

En la nueva arquitectura desaparece la RNC, la interfaz Iub y el control de flujo entre la RNC y el Nodo B. Los paquetes IP llegan directamente al nodo pre-eNB. Se segmentan en unidades RLC SDUs *(Service Data Unit)* y se añade el código de control de error ARQ propio de RLC AM. A nivel MAC la MAC-hs se encarga del control de error HARQ. Esta entidad se conserva respecto a HSDPA. La arquitectura de red que se ha modelado es la que la tecnología LTE considera, en nuestro caso, las simplificaciones se han hecho a nivel de capa física y PDCP.

III. Implementación del sistema pre-LTE en EURANE

En su versión actual, EURANE simula UMTS R99 y HSDPA R5, pero no LTE. En esta sección se describen los pasos a seguir para ampliar EURANE a LTE. El sistema pre-LTE es una simplificación de LTE, en el que se modela los cambios de la estructura de red, eliminación de la RNC y incorporación del nodo pre-eNB, pero dejando de lado los aspectos físicos de la transmisión.

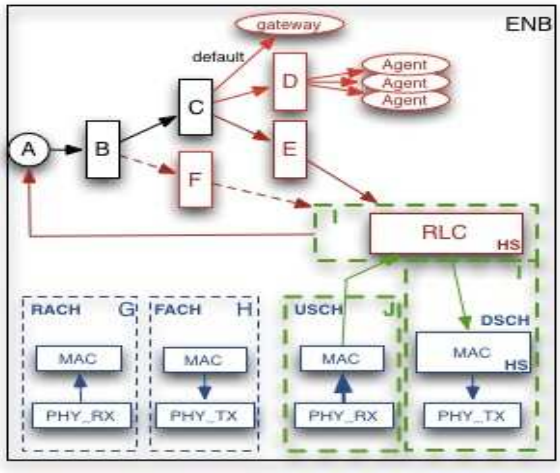


Fig. 2. Implementación del nodo pre-eNB

A. El Nodo pre-eNB

De forma general se puede decir que el nodo eNB es un nodo construido por la fusión de los nodos BS y RNC en HSDPA. (Aunque algunas de las funciones que recaían en el protocolo PDCP se hayan trasladado al aGW).

Se define el nuevo nodo pre-eNB, representado en la Fig. 2. Este nodo contiene la estructura básica de un nodo en ns-2, al que se le añaden todos los clasificadores ubicados en el RNC, además de las interfaces básicas que contienen la BS *(G y H)* y la interfaz para el acceso al canal compartido HS-DSCH *(Interfaz I)*.

Para poder conectar este nuevo nodo al núcleo de red, hay que habilitar en el nodo pre-eNB un enlace entre la red de acceso y el núcleo de red (*gateway)* (nótese que en HSDPA, el enlace de la red acceso a la red troncal está en la RNC). También se debe modificar los UEs para que reconozcan en su configuración la nueva estación base.

IV. METODOLOGÍA EXPERIMENTAL

El modelo de simulación de EURANE está limitado a una sola célula con solo una estación base, aunque sí incluye múltiples usuarios con múltiples conexiones activas simultáneas. En este modelo no se consideran traspasos.

El modelo de sistema UMTS/HSDPA está formado por el núcleo de red *(SGSN, GGSN)* y la red externa *(Node 1, Node 2),* ambos nodos ns-2, y la red de acceso *(RNC, BS y UE),* formada por nodos EURANE. Todos los nodos están conectados por un enlace de red fijo, excepto los UEs, que se conectan a través del enlace radio. Las distintas aplicaciones se conectan a los agentes de transporte, que estarán ubicados en el Node 2 de la red externa.

La diferencia de modelos entre UMTS y HSDPA radica en los canales de transporte utilizados para la transmisión de datos; mientras que en el primero se utilizan los canales comunes *(RACH y FACH),* en el segundo se usa el canal compartido *(HSDSCH)*. En el modelo del escenario pre-LTE sólo cambia la red de acceso en la que el nuevo nodo pre-eNB sustituye a la BS y el RNC.

duración de la simulación (s)	200s
número de nodos	5 nodos
tamaño de paquete TCP y UDP	500, 540 bytes
scheduler type	RR/MaxCI
HS, AMHS, UMHS, AM, UM buffers	500 PDU

Tabla 1. Valores de los parámetros durante las simulaciones.

La tabla 1 reproduce los parámetros de configuración del simulador, entre los que destacan los relacionados con los protocolos de transporte y enlace.

Como modelo de servicio, se simulan dos tipos de tráfico: transferencia de archivos y multimedia *(audio/video streaming)*. Los primeros se modelan mediante una aplicación FTP sobre TCP y está asociado al modo RLC AM. Los segundos se configuran como tráfico CBR *(Constant Bit Rate)* con una tasa de transferencia de 448Kbps sobre UDP y están asociados al modo RLC UM.

En cuanto al modelo de canal, que integra los modelos de propagación y movilidad de usuario, se consideran dos entornos. El primero que se corresponde con un canal ideal sin atenuación y sin propagación multicamino; y el segundo pedestre con peatones que circulan a 3 km/h.

El modelo de gestión de recursos radio se basa en dos algoritmos de *scheduling* Round Robin y Maximum C/I.

Para medir el rendimiento de la red se utiliza el caudal y el retardo. Los datos que se muestran son promedios a lo largo del tiempo de duración de las simulaciones. Se ha considerado que el inicio de la transmisión de cada usuario esta decalado con el resto 10s.

V. RESULTADOS DE RENDIMIENTO DE LOS SISTEMAS 3G

A. Pruebas UMTS

Durante las simulaciones, se consideran 5 terminales móviles compartiendo el canal descendente FACH. En el modo AM, los terminales compiten por el ancho de banda del canal compartido. Los servicios de datos están controlados por la capa de transporte, protocolo TCP, que ajusta los tráficos al ancho de banda disponible en el canal. La Fig. 3 presenta la tasa de transferencia de cada uno de los 5 UEs, junto a la tasa de transferencia total. Se observa que el ancho de banda se comparte de forma equitativa entre los usuarios y que en todo momento se consumen todos los recursos del canal.

Por el contrario, si se simula tráfico multimedia, el primer terminal que accede al canal ocupa todo el ancho de banda disponible. Esto ocurre por dos razones fundamentales: la primera es que el protocolo UDP no tiene control de flujo; por otro lado, la capa MAC implementada en el simulador UMTS no utiliza algoritmos de planificación de recursos, lo que hace que la capa de transporte se encargue de gestionar el ancho de banda. En este caso, el protocolo UDP carece de mecanismos para gestionar el ancho de banda entre los distintos usuarios, por ello el usuario que accede primero consume prácticamente todos los recursos. Este comportamiento también se observa si se mezclan flujos de ambos protocolos de transporte (UDP/UM y TCP/AM). En este caso, los servicios no orientados a conexión prevalecen sobre los servicios orientados a conexión.

B. Pruebas HSDPA

En la siguiente simulación se considera un escenario pedestre, cinco usuarios y tráfico de datos. Para calcular el impacto de las medidas de canal (CQIs) en los algoritmos de planificación, los terminales se ponen a diferentes distancias 300m, 500m, 700m del Nodo B con una movilidad reducida (3km/h).

En las tablas 2 y 3 se observa la tasa de transferencia con cada uno de los algoritmos de planificación. *MaxC/I* registra un mayor volumen de transferencia que *RR*, lo que se debe a que *Max C/I* optimiza al máximo los recursos del canal radioeléctrico. Los terminales que están más próximos a la celda experimentan velocidades mayores y retardos menores, mientras que los que se encuentran en el borde de la celda tendrán una peor calidad de transmisión.

En las tablas 2 y 3 también puede observarse el caudal y retardo para servicios multimedia en las columnas sombreadas en gris. Al no existir acuse de recibo en el modo UM, el volumen de datos transmitido aumenta significativamente en comparación con la simulación

anterior. Concretamente, con *RR* el caudal aumenta casi un 14 % y, con *Max C/I,* un 20 %.

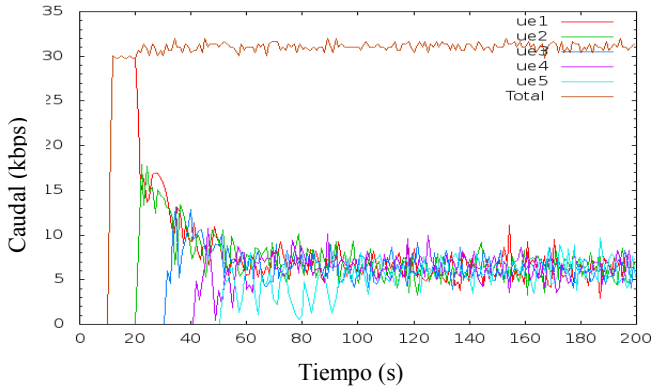


Fig. 3. Caudal 5-UE en el canal comun para 1CP/RLC AM en UMTS.

	Caudal (kbps)	Retardo (ms)	Caudal(kbps)	Retardo (ms)
ue1	332	160	382	379
ue2	224	256	262	876
ue3	322	168	364	494
ue4	203	322	239	974
ue5	153	424	162	1629
ue1-5	1234	266,0	1312	870,4

Tabla 2. Caudal y retardo para 5 usuarios en modo AM (blanco) y UM (gris) en un escenario pedestre con planificador RR en HSDPA.

	Caudal (kbps)	Retardo (ms)	Caudal (kbps)	Retardo (ms)
ue1	354	138	414	214
ue2	252	223	324	432
ue3	357	135	429	232
ue4	207	303	246	781
ue5	142	442	186	1283
ue1-5	1312	248,2	1599	588,4

Tabla 3. Caudal y retardo para 5 usuarios en modo AM (blanco) y UM (gris) en un escenario pedestre con planificador Max C/I en HSPDA.

	Caudal (kbps)	Retardo (ms)	Caudal (kbps)	Retardo (ms)
ue1	353	152	382	361
ue2	232	270	262	870
ue3	341	165	363	484
ue4	214	305	239	966
ue5	146	477	163	1609
ue1-5	1286	273,8	1409	858,0

Tabla 4. Caudal y retardo para 5 usuariosen modo AM (blanco) y UM (gris) en un escenario pedestre con planificador RR en sistema pre-LTE.

	Caudal (kbps)	Retardo (ms)	Caudal (kbps)	Retardo (ms)
ue1	378	127	415	199
ue2	268	210	327	418
ue3	379	127	429	218
ue4	218	281	246	768
ue5	149	431	187	1263
ue5-1	1392	235,2	1604	573,2

Tabla 6. Caudal y retardo para 5 usuarios en modo AM (blanco) y UM (gris) en un escenario pedestre, con planificador Max C/I en sistema pre-LTE.

C. *Pruebas pre-LTE*

Al simplificar la red de acceso, eliminando los nodos BS y RNC, y sustituyéndolos por un nodo pre-eNB, se consigue disminuir el retardo, como se aprecia en las tablas 4 y 5. Si se observan los caudales, se observa que, al igual que pasaba en HSDPA, el planificador Max C/I transmite más volumen de datos que el RR.

Si se comparan los sistemas HSDPA y pre-LTE se puede observar que para este último el caudal acumulado, es decir, la suma de todos los usuarios, es algo mayor que en HSDPA para todas las experiencias. Por ejemplo, en modo AM para un planificador Max C/I el caudal total en HSDPA es 1312 kbps y en pre-LTE 1392 kbps. De nuevo, matizar que este incremento se debe exclusivamente a la modificación de la topología de red y no de la tecnología radio, que se mantiene.

VI. Conclusiones

En este trabajo se ha descrito la extensión del simulador EURANE para considerar nuevas estructuras de red móvil. En esta primera etapa se ha desarrollado una estructura pre-LTE que contempla la arquitectura de LTE en la interfaz radio. A partir de este simulador, se han hecho pruebas para verificar su funcionamiento y compararlo con tecnologías móviles anteriores como HSDPA y UMTS. A partir de los resultados, se puede concluir que los canales compartidos en UMTS no están configurados para administrar tasas de tráfico altas y un número alto de usuarios porque tienen un ancho de banda limitado. Esto provoca que la lentitud de las transmisiones de los usuarios aumente con el número de usuarios conectados a él. En HSDPA, gracias al canal compartido HS-DSCH y la nueva entidad MAC-hs, que controla los distintos flujos de trabajo según el tipo o el estado del canal radioeléctrico, aumenta el volumen de tráfico de los distintos usuarios que usan este canal, obteniendo así tasas de transmisión mayores. Por último, se observa que, al simplificar la red de acceso con el nuevo nodo pre-eNB, aumentan las tasas de transmisión de los distintos tipos de tráfico y se reduce el retardo, dando un acercamiento a lo que sería una red LTE.

En el futuro se pretende incluir en el simulador las características físicas de LTE, como el modelado de canal selectivo en frecuencia y los algoritmos de planificación de recursos en tiempo y frecuencia. Posteriormente, se considerará la ampliación del simulador a LTE-Advanced con la inclusión de repetidores.

Agradecimientos

Este trabajo ha sido financiado por el proyecto TEC09-13413 del Ministerio de Ciencia e Innovación.

Referencias

[1] EURANE website. http://eurane.ti-wmc.nl/eurane/
[2] Kevin Fall, Kannan Varadhan. "*Network Simulator Manual*". http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf. ISI May, 2010
[3] SEACORN website. http://seacorn.cs.ucy.ac.cy/eumtssim/
[4] 3GPP. "*The Mobile Broadband Standard*" vía http://www.3gpp.org/
[5] EURANE User Guide (Release 1.3). Disponible vía http://www.tiwmc.nl/eurane/, 2006

Integración de modelos de información según los estándares de interoperabilidad en e-Salud UNE-EN ISO 13606 e ISO/IEEE11073

P. Muñoz[1], I. Martínez[1], A. Muñoz[2], P. del Valle[1], A. Aragüés[1], J. Escayola[1], J.D. Trigo[1], J. García[1]

[1]Instituto de Investigación en Ing. Aragón (I3A) - Univ. Zaragoza (UZ), c/ María de Luna, 1. 50018. Zaragoza
[2]Unidad Inv. Telemedicina e-Salud - Inst. Salud Carlos III (ISCIII), Av/Monforte de Lemos, 5. 28029. Madrid
[1]{pmg, imr, pdelvalle, aaragues, javier.escayola, jtrigo, jogarmo}@unizar.es, [2]{adolfo.munoz}@isciii.es

***Resumen*- Este artículo propone la integración de los modelos de información definidos en los estándares internacionales UNE-EN ISO 13606 (para intercambio interoperable de extractos de historia clínica electrónica) e ISO/IEEE 11073 (para comunicación interoperable de dispositivos médicos). Para ello, se presenta un estudio comparativo entre los modelos de información específicos de UNE-EN ISO 13606 e ISO/IEEE 11073. A partir de este estudio, se ha implementado una arquitectura multicapa soportada sobre tecnologías *Web Services* y dotNet, que incluye el nuevo conjunto armonizado de DATA_TYPEs ISO 21090 para el intercambio de información sanitaria, lo que constituye una propuesta integrada basada en estándares extremo a extremo.**

***Palabras Clave*- *Electronic Health Record* (EHR), interoperabilidad extremo a extremo, *HealthCare Information System* (HCIS), UNE-EN ISO 13606, ISO/IEEE 11073.**

I. Introducción

La necesidad de compartir información entre distintos sistemas de información sanitarios (*HealthCare Information Systems*, HCIS) es primordial a la hora de emitir un diagnóstico correcto y preciso. Dentro del complejo contexto de especificación de la información médica existen varios hándicaps para garantizar interoperabilidad en el intercambio de la historia clínica electrónica (*Electronic Healthcare Record*, EHR) del paciente [1]. Además, se está apostando unir los dos grandes ámbitos en el cuidado de la salud: el sanitario (centrado en el profesional en todos sus niveles de atención) y el extra-sanitario (entorno domiciliario, residencial o ubicuo que podrían incorporarse a la EHR). Por ello, es imprescindible establecer paralelismos entre los extremos de la comunicación: la interoperabilidad de dispositivos médicos (actualmente liderado por la familia de normas internacionales ISO/IEEE 11073, X73 [2]) y entre sistemas sanitarios (cuyo principal exponente es la norma internacional UNE-EN ISO 13606 [3]).

Este artículo propone la integración de los modelos de información definidos en ambos estándares internacionales. En la Sección II se detalla un estudio de ambas normas para identificar los valores a incluir en la EHR a partir de los datos obtenidos por los dispositivos médicos remotos. Además, se analiza la reciente evolución de DATA_TYPEs a la nueva recomendación ISO 21090 y se presenta la arquitectura multicapa implementada para permitir el intercambio de EHR generadas a partir de la adquisición remota de datos de telemonitorización. Las conclusiones se dan en la Sección III.

II. Estudio de compatibilidad entre los estándares internacionales ISO/EN13606 e ISO/IEEE11073

ISO/IEEE 11073 (X73) [2] es el estándar internacional para la transferencia interoperable de datos provenientes de dispositivos médicos. El modelo de información (*Domain Information Model*, DIM) de X73 permite definir las especificaciones de cualquier dispositivo médico mediante una estructura jerárquica compuesta por: *Virtual Medical Object* (VMO), *Medical Device System* y *Virtual Medical Device* (MDS y VMD), *Channel Object* y el resto de clases de medida y contenedores de información: *numeric*, *sample array*, *real-time sample array* (RT-SA), *enumeration*, *complex metric* y *persistent metric* (PM-*segment*).

UNE-EN ISO 13606 [3] es la norma internacional para transferencia interoperable de cualquier extracto de la EHR de un paciente. Se caracteriza por usar un Modelo de Referencia (definido en UNE-EN ISO 13606-1), que incluye elementos genéricos para la transmisión de la información clínica sustentando la interoperabilidad sintáctica y, complementado por el uso de un Modelo de Arquetipos (definido en UNE-EN ISO 13606-2), la interoperabilidad semántica. Este Modelo de Referencia estructura la información usando los siguientes bloques lógicos: EXTRACT, FOLDER, COMPOSITION, SECTION, ENTRY, CLUSTER y ELEMENT.

ISO/IEEE 11073 y UNE-EN ISO 13606 no presentan similitudes evidentes, aunque se pueden encontrar estructuras comunes a partir de un estudio comparativo de ambos modelos de información. Es posible hacer una primera aproximación del ISO/IEEE 11073 VMO al bloque lógico COMPOSITION de UNE-EN ISO 13606, ya que VMO es el objeto sobre el que se establecen relaciones con el resto de objetos numéricos, mientras que COMPOSITION es la estructura contenedor que acabará conteniendo los datos clínicos. Además, dentro de cada EHR hay diversos campos a ser cubiertos (e.g. el paciente al que pertenecen, el tipo de medida clínica, la fecha y hora, etc.) y algunos de esos datos han de ser proporcionados por el dispositivo médico. Por todo ello, es necesario hacer un estudio comparativo de los posibles valores que pueden tener las estructuras definidas en ambos modelos. En la Tabla I se muestra el detalle de cada bloque lógico definido en UNE-EN ISO 13606 distinguiendo entre sus atributos principales y por asociación (con *), indicando a qué DATA_TYPE corresponden, si son *mandatory* (MND) y si podrían estar vinculados o no a la norma X73.

ISBN: 978-84-694-5948-5

Se describen los principales bloques lógicos de UNE-EN ISO 13606, detallando sus relaciones con X73:

- **EHR_EXTRACT.** Es el contenedor de mayor orden jerárquico. Desde el punto de vista de su construcción y para un sistema de EHR determinado, se puede obtener casi todos sus campos de manera determinista salvo quién es el paciente al que pertenece la información. En lo que se refiere a la identificación del paciente dentro del estándar X73 existe un campo que indica a qué paciente/persona pertenecen los datos adquiridos por el dispositivo médico en la clase *PM-segment*. Sin embargo, esta información no es obvia para el resto de clases derivadas de la clase *metric* dentro de *channel object*. Hay una relación entre VMO y todas las clases que heredan de *metric*, pero no existe un campo específico. En esos casos sería necesaria la identificación de paciente por medios anexos a la pura comunicación X73.

El resto de clases contenedoras en un extracto UNE-EN ISO 13606 son heredadas de RECORD_COMPONENT, por lo tanto van a tener una serie de campos comunes a todas ellas. Dentro de RECORD_COMPONENT, destaca el atributo *sensibility* cuya función es establecer un determinado nivel de seguridad, de tal forma que ese registro sólo es accesible por un determinado profesional si las atribuciones que le otorga su rol superan dicho umbral. Del mismo modo, se pueden destacar los atributos *archetype_id* y *meaning,* para indicar si el registro está estructurado bajo un arquetipo o su significado es equiparable a algún concepto clínico.

- **FOLDER**. Es una clasificación opcional mediante la que un HCIS puede organizar cada COMPOSITION conforme a un criterio dado: todas las que correspondan a un episodio (e.g. un paciente que se nota débil y se hace un análisis, un paciente que va al médico o al especialista respiratorio, etc.), todas las que correspondan a la misma especialidad (e.g. psiquiatría, etc.), etc. El nivel de granularidad en esta clasificación puede incrementarse mediante el uso de FOLDERs dentro de FOLDERs.
- **COMPOSITION**. Por definición formal, recoge cualquier interacción médico-paciente y puede estar estructurada en SECTIONs para facilitar la lectura o navegación dentro de cada COMPOSITION. Este bloque incluye *session_time* que es el intervalo de tiempo en la que se adquieren los datos desde el manager (CE). La armonización de este campo con X73 podría efectuarse de distintas formas:
  - Estableciendo los umbrales temporales a través del VMO del CE y realizando los correspondientes cálculos de fecha y hora después de asociarse (IVL_TS.low) y justo antes de desasociarse (IVL_TS.high), según los estados definidos en la máquina de estados (*Finite State Machine*, FSM) de ISO/IEEE 11073-20601 [2]. Este tipo de planteamiento se da cuando la conectividad no presenta problema y se podría realizar un envío de datos médicos inmediatamente después de haberlos adquirido. X73 permite varias posibilidades para ese cálculo si dichos atributos están implementados: *date-and-time* (MDC_ATTR_TIME_ABS), *base-offset-time* (MDC_ATTR_ TIME_BO), *relative-time* (MDC_ATTR_ TIME_REL) o *HiRes-relative-time* (MDC_ATTR_TIME_ REL_HI_RES). En caso de que se produzca una notificación de ajuste temporal en la adquisición de las medidas, se deberá modificar también la marca temporal de inicio de adquisición de datos en el mismo sentido que la inferior para evitar inconsistencias.
  - A través de las marcas temporales de las diferentes medidas o anotaciones transmitidos durante la comunicación o conjunto de comunicaciones enviadas de forma conjunta si el CE presentara algún tipo de problema de conectividad, ya que en ese caso el CE mantendría dichas marcas para conservar la integridad de las medidas adquiridas. Este tipo de planteamiento podría estar más orientado a transmisiones *store-and-forward.* Las distintas posibilidades que permite X73 son: *absolute-time-stamp* (MDC_ATTR_TIME_STAMP_ABS), *base-offset-time-stamp* (MDC_ATTR_TIME_STAMP_BO), *relative-time-stamp* (MDC_ATTR_TIME_STAMP_REL) y *HiRes-time-stamp* (MDC_ATTR_TIME_STAMP_REL_HI_ RES). Consideración especial merece *PM-segment* porque permite obtener directamente el tiempo de comienzo de medida, con los atributos *Segment-Start-Abs-Time* (MDC_ATTR_TIME_ START_SEG) o *Segment-Start-BO-Time* (MDC_ATTR_ TIME_START_SEG_BO), y el tiempo de final de medida, con los atributos *Segment-End-Abs-Time* (MDC_ATTR_TIME_END_SEG) o *Segment-End-BO-Time* (MDC_ATTR_TIME_END_SEG_BO). Además, *Date-and-Time-Adjustment* (MDC_ATTR_TIME_ABS_ADJUST) permite la notificación de cambios en la fecha/hora.

  En este artículo se ha decidido considerar parte integrante de la misma COMPOSITION toda información médica que sea transmitida en una misma instancia de comunicación al servidor de EHR. Así, diversos elementos pertenecientes al sistema de EHR (*committal* y, opcionalmente, *composer*) serán los encargados de estructurar la información y grabarla adecuadamente en el HCIS. Otro hecho interesante es el atributo *other_participations*, a través del cual se podría identificar el CE que ha realizado esa instancia de comunicación de datos médicos, jugando un papel de colector de la información médica.
- **SECTION**. Permite estructurar la información dentro de una misma COMPOSITION para favorecer su lectura o reflejar el flujo de información dentro de un encuentro clínico. Igual que FOLDER, el grado de granularidad en esta división puede incrementarse utilizando SECTIONs dentro de otras SECTIONs. Como se ha comentado, la organización de la información es función de *composer* y, por tanto, queda fuera de las atribuciones del CE.
- **ENTRY**. Contiene toda la información relacionada con una medida/observación o batería de éstas y, representa la unidad mínima de significación clínica. Una ENTRY está compuesta de ITEMs (clase abstracta) que se hacen tangibles por medio de CLUSTERs y/o ELEMENTs. Para cada ENTRY sería interesante la determinación del atributo *meaning* (y por extensión, del atributo *name*) a través del cual se relaciona esa medida con un concepto del conocimiento. Sin embargo, algunos conceptos presentan singularidades que impiden que se pueda identificar directamente a partir del valor de la medida.

Tabla I. Comparativa de los modelos UNE-EN ISO 13606 e ISO/IEEE 11073
(II = *Instance Identifier*, TS = *Time Stamp*, CV = *Coded Value*, CS = *Coded Simple*, IVL = *Interval*, ED = *Encapsulated Data*).
[○ = sí, ● = no]

EHR_EXTRACT	**DATA_TYPE**	**MND**	**X73**
authorising_party	II	●	●
ehr_id	II	○	●
ehr_system	II	○	●
rm_id	String	○	●
subject_of_care	II	○	○
time_created	TS	○	●
**all_compositions*	Set<COMPOSITION>	●	●
**criteria*	Set<EXTRACT_CRITERIA>	●	●
**folders*	Set<FOLDER>	●	●
**demographic_extract*	Set<II>	●	●
FOLDER	**DATA_TYPE**	**MND**	**X73**
archetype_id	II	●	●
meaning	CV	●	●
name	TEXT	○	●
orig_parent_ref	II	●	●
policy_ids	Set<II>	●	●
rc_id	II	○	●
sensitivity	Integer	●	●
synthesised	Boolean	○	●
**links*	Set<LINK>	●	●
**feeder_audit*	AUDIT_INFO	●	●
**sub-folders*	Set<FOLDER>	●	●
**attestations*	Set<ATTESTATION_INFO>	●	●
**compositions*	Set<COMPOSITION>	●	●
COMPOSITION	**DATA_TYPE**	**MND**	**X73**
archetype_id	II	●	●
meaning	CV	●	●
name	TEXT	○	●
orig_parent_ref	II	●	●
policy_ids	Set<II>	●	●
rc_id	II	○	●
sensitivity	Integer	●	●
synthesised	Boolean	○	●
contribution_id	II	●	●
session_time	IVL<TS>	●	○
territory	CS	●	●
**links*	Set<LINK>	●	●
**feeder_audit*	AUDIT_INFO	●	●
**attestations*	Set<ATTESTATION_INFO>	●	●
**other_participants*	Set<FUNCTIONAL_ROLE>	●	○
**committal*	AUDIT_INFO	○	●
**content*	Set<CONTENT>	●	○
**composer*	FUNCTIONAL_ROLE	●	●
SECTION	**DATA_TYPE**	**MND**	**X73**
archetype_id	II	●	●
meaning	CV	●	●
name	TEXT	○	●
orig_parent_ref	II	●	●
policy_ids	Set<II>	●	●
rc_id	II	○	●
sensitivity	Integer	●	●
synthesised	Boolean	○	●
**links*	Set<LINK>	●	●
**feeder_audit*	AUDIT_INFO	●	●
**members*	Set<CONTENT>	●	●
ENTRY	**DATA_TYPE**	**MND**	**X73**
archetype_id	II	●	●
meaning	CV	●	○
name	TEXT	○	○
orig_parent_ref	II	●	●
policy_ids	Set<II>	●	●
rc_id	II	○	●
sensitivity	Integer	●	●
synthesised	Boolean	○	●
act_id	String	●	●
act_status	String	●	●
subject_of_info_category	CS	●	●
uncertainly_expressed	Boolean	○	●
**links*	Set<LINK>	●	●
**feeder_audit*	AUDIT_INFO	●	●
**items*	Set<ITEM>	●	○
**info_provider*	FUNCTIONAL_ROLE	●	○
**other_participants*	Set<FUNCTIONAL_ROLE>	●	●
**subject_of_information*	RELATED_PARTY	●	●

CLUSTER	**DATA_TYPE**	**MND**	**X73**
archetype_id	II	●	●
meaning	CV	●	○
name	TEXT	○	○
orig_parent_ref	II	●	●
policy_ids	Set<II>	●	●
rc_id	II	○	●
sensitivity	Integer	●	●
synthesised	Boolean	○	●
emphasis	CV	●	●
item_category	CS	●	○
obs_time	IVL<TS>	●	○
structure_type	CS	○	●
**links*	Set<LINK>	●	●
**feeder_audit*	AUDIT_INFO	●	●
**parts*	Set<ITEM>	●	○
ELEMENT	**DATA_TYPE**	**MND**	**X73**
archetype_id	II	●	●
meaning	CV	●	○
name	TEXT	○	○
orig_parent_ref	II	●	●
policy_ids	Set<II>	●	●
rc_id	II	○	●
sensitivity	Integer	●	●
synthesised	Boolean	○	●
emphasis	CV	●	●
item_category	CS	●	○
obs_time	IVL<TS>	●	○
value	DATA_VALUE	●	○
**links*	Set<LINK>	●	●
**feeder_audit*	AUDIT_INFO	●	●

AUDIT_INFO	**DATA_TYPE**	**MND**	**X73**
commiter	II	○	●
ehr_system	II	○	●
previous_version	II	●	●
reason_for_revision	CV	●	●
time_committed	TS	○	●
versión_set_id	II	●	●
versión_status	CS	●	●
ATTESTATION_INFO	**DATA_TYPE**	**MND**	**X73**
attested_view	ED	●	●
proof	ED	●	●
reason_for_attestation	CV	○	●
time	TS	○	●
**target*	Set<RECORD_COMPONENT>	○	●
**attester*	FUNCTIONAL_ROLE	○	●
FUNCTIONAL_ROLE	**DATA_TYPE**	**MND**	**X73**
funtion	CV	●	○
healthcare_facility	II	●	●
mode	CS	●	○
performer	II	○	○
service_setting	CV	●	○
RELATED_PARTY	**DATA_TYPE**	**MND**	**X73**
party	II	●	●
relationship	TEXT	○	●
LINK	**DATA_TYPE**	**MND**	**X73**
follow_link	Boolean	○	●
nature	CS	○	●
role	CV	●	●
**target*	Set<RECORD_COMPONENT>	○	●

Por lo tanto, será necesario procesado que permita el mapeo con alguna terminología médica como SNOMED-CT [4]. Es interesante indicar el dispositivo médico que ha proporcionado cada ENTRY mediante el atributo *info_provider* (de tipo *functional_role*). Este atributo se completaría con el atributo obligatorio *performer* gracias al *system-id* del MDS de X73 (MDC_ATTR_SYS_ID) y los atributos opcionales que se obtendrían de manera estática: *mode* (MOD01), *service_setting* (donde se codifique que la medida fue obtenida de manera remota) y *function* (donde se codifique que el rol ha sido la adquisición de la medida).

- **ITEM**. Aunque ITEM es una entidad abstracta, presenta algunos atributos de gran importancia en entornos de telemonitorización y que heredarán tanto CLUSTER como ELEMENT. Destacan *obs_time*, para recoger la fecha y hora exactas en la que los datos médicos fueron adquiridos (utilizando cualquiera de las marcas temporales mencionadas ya que el instante temporal en el que se adquiere una medida no tiene por qué ser el mismo que el instante en el que se ingresan los datos en el HCIS) e *ítem_category,* para diferenciar según los modelos de conocimiento médico lo que representa el núcleo de la medida y otro tipo de anotaciones como el protocolo seguido o la información de contexto.
- **CLUSTER**. Es una estructura que sirve para organizar información compleja. Dentro de un CLUSTER se pueden encontrar otros CLUSTERs y/o ELEMENTs. Como ocurría con ENTRY, su atributo *meaning* (o *name*) podría no obtenerse por asignación directa X73, aunque al comparar esta entidad con RT-SA el mapeo entre el atributo *type* (MDC_ATTR_ID_TYPE) y un determinado concepto clínico puede ser directo (e.g. curva pletismográfica).
- **ELEMENT**. Es la unidad contenedora más baja que almacena los DATA_VALUE. El atributo *meaning* (y *name*) se puede adquirir directamente a partir del atributo *type* (MDC_ATTR_ID_TYPE) de X73.

Para completar esta descripción, se listan el resto de bloques lógicos UNE-EN ISO 13606, indicando qué atributos incluyen y cuál es su vinculación (si la tienen) con X73:

- **AUDIT_INFO**. Representa información del momento (cuándo) y el responsable (quién) del envío de la información médica, tanto en la adquisición inicial del dato médico como en sus sucesivas versiones (si las hubieras).
- **ATTESTATION_INFO.** Da soporte a cualquier tipo de testimonio o prueba de que la información médica es auténtica (e.g. cuando se realiza una ecografía o prueba similar, en diversos países debe quedar constancia de qué imagen mostraba la pantalla cuando se hizo el diagnostico).
- **FUNCTIONAL_ROLE.** Documenta la participación de una tercera persona, dispositivo o componente *software* cuando se obtiene la información médica.
- **RELATED_PARTY.** Identifica la relación entre *subject_of_information* y *subject_of_care*.
- **LINK**. Sirve para definir la relación entre distintos RECORD_COMPONENTs, como relaciones causa/efecto.

A partir de las consideraciones anteriores y de la especificación de parámetros de ISO/EN13606-5 para el intercambio de extractos de EHR, se ha implementado una arquitectura multicapa basada en tecnologías *Web Services* y desarrollada en C#, incluyendo un *Internet Information Server* (IIS) de páginas web dinámicas ASP.Net. Para evaluar el diseño realizado y su integración con otros sistemas interoperables de HCE se han desarrollado una serie de pruebas centradas en la arquitectura multicapa y los posibles problemas derivados de la evolución de los nuevos DATA_TYPE, de TS14796 a ISO 21090 [5] [6]. En concreto, y a partir de una recopilación de las medidas de obligada implementación en las diferentes especializaciones publicadas a fecha de redacción, todos ellas asimilables a elementos Physical Quantity (PQ) y Coded Value (CD.CV) se realizaron diferentes pruebas de integración de estos DATA_TYPEs junto con otros como Instace Identifier (II) o Time Interval (IVL<TS>), necesarios para la correcta construcción del EHR_EXTRACT.

Los resultados de estas pruebas fueron completamente satisfactorios, comprobando la integridad de la solución propuesta, y constituyeron una conferencia invitada en "CEN/ISO EN 13606 *Invitational Workshop*" [7], el principal foro de desarrolladores de UNE-EN ISO 13606, donde se seleccionaron las experiencias más representativas de 12 países que han adoptado la norma en sus soluciones.

III. Conclusión

En este artículo se ha propuesto un estudio comparativo entre los estándares internacionales UNE-EN ISO 13606 e X73 para la integración de sus modelos de información. A partir de este estudio, se ha implementado una arquitectura multicapa soportada sobre tecnologías *Web Services* y dotNet, lo que constituye una propuesta integrada extremo a extremo. Esta arquitectura soporta el nuevo conjunto de DATA_TYPEs definidos por ISO 21090 y facilita el acceso a todo tipo de datos mediante la implementación de múltiples interfaces. Los resultados de validación garantizan la aplicabilidad del estudio propuesto.

Agradecimientos

Los autores quieren agradecer a Dipak Kalra (CEN/TC 251-WG1 Task Force 13606: EHRCom), Carolina Hernández y Francisco Ramos (Técnicas Competitivas S.A.) y Roberto Somolinos (Hospital Univ. Puerta de Hierro) por su asesoramiento técnico. Este trabajo ha sido parcialmente subvencionado por los proyectos TIN2008-00933/TSI del Ministerio de Ciencia e Innovación (MICINN) y Fondos Europeos para el Desarrollo Regional (FEDER), TSI-020100-2010-277 y TSI-020302-2009-7/Plan Avanza I+D del Ministerio de Industria, Turismo y Comercio, PI08-1148 del Fondo de Investigación Sanitaria (FIS) Plan Nacional de I+D+i y PI029/09 del Gobierno de Aragón

Referencias

[1] B. Blobel, "Advanced EHR architectures: promises or reality", *Methods Inf Med*, vol. 25, pp. 95-101, 2006.

[2] ISO/IEEE11073 Point-of-Care (X73-PoC). Health informatics. [Part 1. Medical Device Data Language (MDDL)] [Part 2. Medical Device Application Profiles (MDAP)] [Part 3. Transport and Physical Layers]. ISO/IEEE11073 - Personal Health Devices standard (X73-PHD). Health informatics. [P11073-00103. Technical report - Overview] [P11073-104xx. Device specializations] [P11073-20601. Application profile - Optimized exchange protocol]. [On line] IEEE Standards Association webpage: *http://standards.ieee.org/*. Last visit: 03/2011.

[3] ISO/EN13606. CEN/TC251 – ISO/TC215. Electronic Healthcare Record (EHR) Communication. Part 1: Reference Model, Part 2: Archetype Model, Part 3: Reference Archetypes, Part 4: Security and Part 5: Interface". [On line] *http://www.iso.org/iso/catalogue_detail.htm?csnumber=40784*. Last visit: 03/2011.

[4] SNOMED-CT. International Health Terminology Standards Development Organization (IHTSDO). [On line] *http://www.ihtsdo.org/*

[5] TS14796 DATA_TYPEs for Use in Health Care Data Interchange. [On line] *www.cen.eu - http://isotc.iso.org*. Last visit: 03/2011.

[6] ISO 21090. International Standard Health Informatics – Harmonized DATA_TYPEs for information exchange. [On line] *http://isotc.iso.org*. Last visit: 03/2011.

[7] CEN/ISO EN13606 invitational workshop. [On line] *http://pangea.upv.es/en13606/ index.php/en13606-invitational-workshop-madrid-june-2010*. Last visit: 03/2011.

Análisis de la Web Oculta en España

Manuel Álvarez, Fidel Cacheda, Rafael López-García, Víctor M. Prieto.
Departamento de Tecnologías de la Información y las Comunicaciones,
Universidade da Coruña
Facultade de Informática, Campus de Elviña, S/N, 15071, A Coruña (Spain).
mad@udc.es, fidel@udc.es, rafael.lopez@udc.es, victor.prieto@udc.es.

***Resumen*- Este artículo presenta un estudio sobre los sitios web de los dominios ".es" orientado a determinar el nivel de utilización de determinadas tecnologías que dificultan el recorrido de la Web a los sistemas de *crawling*. En particular, el estudio se centra en dos aspectos relacionados con la "Web Oculta": los *scripts* y los formularios. En base a los resultados obtenidos, se concluye que un *crawler* que pretenda obtener la mayor parte de documentos de la Web debe de tratar tecnologías tales como *scripts* o formularios para conseguirlo.**

***Palabras Clave*-Recuperación de Información, Web Oculta, Web española, formulario, *script*.**

I. INTRODUCCIÓN

Se conoce como "Web Oculta" o "Web Profunda" [1] a la parte de la Web que no está directamente enlazada. Existe un conjunto de tecnologías que los sistemas de *crawling* convencionales no son capaces de tratar y que constituyen los puntos de acceso a esos documentos denominados "ocultos". Por una parte se pueden considerar los formularios web como puntos de entrada a la Web Oculta del lado servidor. Por otra parte, para acceder a la Web Oculta del lado cliente es necesario tratar con tecnologías como lenguajes de *scripting* o Flash.

Para determinar el nivel de utilización de estas tecnologías que dificultan el acceso a los documentos por los sistemas de *crawling*, en 2009 Álvarez et al. [2] comenzaron un estudio sobre los dominios ".es". El estudio se dividió en dos fases: (1) diseñar, implementar y ejecutar un "*crawler*" que descargase la primera página de dichos dominios a partir de una lista actualizada de los mismos y generase estadísticas cuantitativas y (2) analizar el contenido de las páginas para determinar las tecnologías que utilizan.

Este artículo aborda la segunda fase del estudio. Para ello, parte de la arquitectura definida en [2] y la extiende dotando al sistema de un analizador del contenido de las páginas.

La estructura de este artículo es la siguiente: en la sección II se repasan los trabajos relacionados con la materia. La sección III explica la arquitectura del *crawler* usado en el experimento. La sección IV analiza los resultados del experimento y en la sección V se explican las conclusiones y trabajos futuros.

II. TRABAJOS RELACIONADOS

La mayor parte de los estudios sobre la Web se ocupan de la Web de Superficie, aunque existen algunos que se ocupan de la Web Oculta [3]. Existen sitios web que ofrecen estadísticas sobre el contenido de la Web indexado [4], sobre el número de servidores web [5] o sobre el contenido de las páginas [6][7]. Por otra parte, existen organizaciones encargadas de mantener los nombres de dominios y de realizar el recuento de las máquinas dadas de alta en ellos [8]. Algunas, como Red.es [9], que es la que mantiene los dominios españoles, también publican datos sobre la evolución del número de dominios en el tiempo. Otras, como Verisign [10], que administra los dominios ".com" y ".net" hacen informes más completos. Sin embargo, no existen informes públicos que analicen las páginas de los sitios Web españoles para determinar las tecnologías que utilizan.

También son varios los trabajos que ponen de manifiesto las diversas dificultades con las que los *crawlers* tienen que lidiar para acceder a algunos documentos. Un ejemplo lo constituyen los lenguajes de *scripting* siguiendo el estándar ECMAScript [11]. Sin embargo, según reflejan Weideman y Schwenke en [12] y Wu y Davison en [13], aunque sean tecnologías ampliamente usadas, los *crawlers* no suelen evaluarlos en busca de URLs.

III. ARQUITECTURA

A diferencia de los *crawlers* convencionales, se ha implementado uno que no sigue los enlaces de las páginas web, sino que parte de una lista completa de dominios para obtener el estado de cada uno y el contenido de su página principal. Con estos datos, un módulo de análisis de *crawling* se encarga de generar las estadísticas de la fase 1. Para la fase 2 se ha añadido un módulo de análisis del contenido de los documentos, el cual almacena los datos de cada una de las páginas en una base de datos para facilitar la generación de estadísticas. La Fig. 1 muestra la arquitectura del sistema.

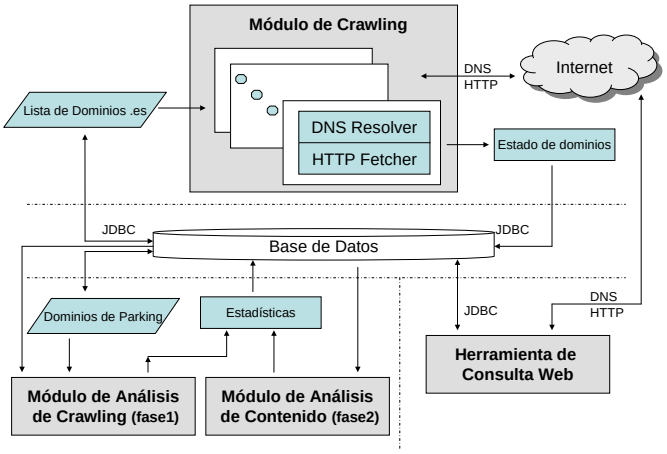


Fig. 1. Arquitectura del sistema de *crawling*.

El módulo de análisis de contenido emplea un analizador sintáctico XML que trata cada documento HTML como un recurso XHTML. Para ello usa el analizador CyberNeko

ISBN: 978-84-694-5948-5

HTML [14]. El analizador XML obtiene la metainformación de la página e identifica diferentes elementos y tecnologías que pueden ser utilizados para navegar o generar contenido, como lenguajes de *scripting* o formularios.

IV. ANÁLISIS DE RESULTADOS

Según el estudio de Álvarez et al. [2], la Web española presentaba 1.093.193 dominios en mayo de 2009. De ellos, solo 577.442 (52,82%) tenían un servidor web. En los siguientes apartados se comentan los resultados obtenidos para el análisis del contenido de la página principal de los dominios ".es" con servidor web. Los resultados se muestran en base al nivel de utilización de lenguajes de *scripting* (IV.A), formularios (IV.B) y otras tecnologías (IV.C).

A. Scripts

Los *scripts* constituyen la principal barrera de acceso a la Web Oculta de lado cliente. En el caso de la Web española, se han encontrado *scripts* en 266.737 dominios, un 46,2% de los que tenían un servidor que no devolvía error. En cuanto al uso de ficheros de *script* externos, se han contabilizado 542.322 invocaciones en 179.576 dominios (31,1%), cifra inferior a los 744.111 *scripts* internos que se encontraron en sus respectivas etiquetas <script> en 231.059 dominios (40%). Ambas cifras son también inferiores a las del uso de *scripts* en atributos HTML, con 2.266.881 ocurrencias en 147.617 dominios (25,6%) .

Según el último RFC de "*Scripting Media Types*" [15] los *scripts* deberían indicar el lenguaje en el que están escritos. Sin embargo, la Tabla 1, basada en los 1.286.419 *scripts* que no se encontraban en atributos HTML, muestra que esto no siempre se cumple. Las invocaciones a scripts externos suelen indicarse de una forma un tanto más rigurosa, probablemente porque se añaden mediante programas de diseño web. Cuando estos mecanismos de identificación de lenguaje fallan, se puede determinar a través de la meta-información de la página, de un análisis del código o de las extensiones de los ficheros. Esta última no es formal, pero ofrece buenos resultados.

La gran mayoría de los *scripts* encontrados en la Web española siguen el estándar ECMAScript [11]. Es más, casi todos están escritos en lenguaje JavaScript.

Scripts	Internos	Externos	Total
Con "type"	75,30%	89,90%	87,70%
Con "language"	14,65%	6,95%	5,20%

Tabla 1. Identificación del lenguaje de los *scripts*.

Para el caso de *scripts* en atributos HTML, se ha comprobado que 137.802 dominios (un 23,8%) hacen uso de eventos "onXXX" sobre diferentes etiquetas. Sin embargo, solo 488.236 scripts en atributos HTML (un 21,5%) incluyeron la etiqueta "javascript:". Una cantidad residual (426) presentaba la etiqueta <script> en el atributo HTML.

También se ha realizado un estudio para determinar qué etiquetas de HTML suelen contener más código *script*. En general, un 37,5% de los dominios contienen bloques <script> en algún lugar dentro del <body>, mientras que solo un 25,7% de los dominios contienen dichos bloques fuera del mismo.

También se ha estudiado la localización de las etiquetas <script> según la etiqueta que las contiene, incluyendo tanto las llamadas a ficheros externos como los bloques de *script* embebidos. Para garantizar la visibilidad de los *scripts*, se recomienda que los bloques se incluyan en la etiqueta <head> o al principio del <body>. Sin embargo, muchos han aparecido en otras etiquetas. Un ejemplo es el de la etiqueta , que debería usarse para crear bloques de marcado. Los primeros resultados se pueden ver en la Tabla 2:

Etiqueta HTML	Número de scripts	Dominios
<head>	530.522	200.713
	260.275	75.619
<body>	228.887	99.361
<td>	116.191	43.992
<p>	31.488	13.941

Tabla 2. Localización de etiquetas <script> en HTML

La Tabla 3, por su parte, muestra las etiquetas que contienen más *scripts* en sus atributos:

Etiqueta HTML	Número de scripts	Dominios
<a>	1.305.831	95.402
<img>	196.419	18.880
<td>	184.657	6.495
	140.825	12.519
<input>	111.013	36.531

Tabla 3. Localización de scripts de eventos en HTML

Como se puede comprobar, la mayoría de los eventos están localizados en enlaces, en atributos "onXXX". En muchas ocasiones esto se hace para generar dinámicamente la URL a la que apuntan. Sin embargo, se han detectado *scripts* en el atributo "action" de los formularios, etc.

También se ha estudiado el evento "onLoad" de la etiqueta <body>, apareciendo en 47.064 dominios (un 8,2%).

Respecto al número de *scripts* empleados por página, tanto en general como separando las invocaciones a ficheros externos y los *scripts* embebidos (contando los que se alojan en atributos de la etiqueta HTML), se obtiene una distribución de ley de potencia con una cola larga. Esto quiere decir que la mayor parte de las páginas no invocan ningún *script*, un gran número invocan pocos *scripts* y solo unas pocas invocan un gran número de ellos. La Fig. 2 muestra las distribuciones del total de *scripts* y de los externos con escala logarítmica en el eje Y. Como el número de *scripts* embebidos es muy superior al de externos, su distribución es muy similar a la del total, por lo que no se muestra.

Existen algunos sitios que utilizan más de 300 *scripts*, pero no son muchos, por lo que gran parte de la cola de la distribución es despreciable. También se han detectado un buen número de páginas que invocan repetidamente al mismo fichero de *script*. El uso de los mismos ficheros de *script* en varios dominios también ha resultado una buena forma de encontrar dominios con el mismo contenido.

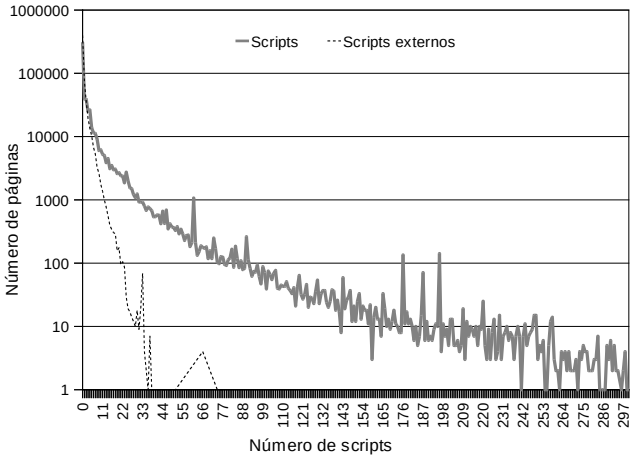


Fig. 2. Uso de ECMAScript en la Web española

Por otro lado, se ha realizado un estudio orientado a recopilar las librerías de *script* más populares. Así, se ha concluido que existe un grupo de 63 ficheros cuyos nombres aparecen más de 1.000 veces y que todos ellos siguen el estándar ECMAScript. La Tabla 4 muestra los primeros *scripts* de dicho grupo, con su número de ocurrencias y el número de dominios en los que se han detectado:

Nombre del fichero	Invocaciones	Dominios
show_ads.js	36.248	18.443
urchin.js	33.898	32.873
AC_RunActiveContent.js	29.746	28.526
swfobject.js	19.918	18.887
prototype.js	9.029	8.837
mootools.js	8.483	8.032
jquery.js	8.207	7.851
caption.js	5.947	5.916
scriptaculous.js	5.172	5.028
funciones.js	4.578	4.419

Tabla 4. Ficheros de script que se invocaron más de 1000 veces.

También se ha tratado de averiguar la función de las librerías más populares y de contar el número de dominios que hacían uso de las mismas. La Tabla 5 muestra el número de dominios y de invocaciones que hacen uso de ellas, agrupándolas según las funcionalidades para las que fueron diseñados los *scripts*.

Funcionalidad	Dominios	Invocaciones
Gestión de Flash y contenido activo	51.895	59.713
Recuento de visitas y generación de estadísticas	39.354	41.777
Dinamización del contenido con AJAX	28.819	41.767
Renderización del contenido/tratamiento de imágenes	22.185	24.598
Generación de menús	4.714	5.198
Tratamiento y validación de datos	4.376	6.586

Tabla 5. Dominios que emplean scripts con funcionalidades comunes.

Se ha llegado a la conclusión de que aunque hay muchas librerías en la red, casi todas se pueden clasificar en un número reducido de funcionalidades (generación de estadísticas, dinamización con AJAX, gestión de Flash, tratamiento de imágenes, etc.).

El uso de otros lenguajes de *script* es testimonial. Por ejemplo, tres de los cuatro *scritps* marcados como lenguaje TCL eran en realidad JavaScript marcado erróneamente. Por su parte, solo se han encontrado 1.769 llamadas a código VBScript, de las cuales únicamente 268 refieren a ficheros externos. De este último grupo, 246 contienen código relacionado con Flash (descarga del complemento, etc.).

Por último, se han estudiado informalmente los ficheros cuyo lenguaje no se ha podido detectar automáticamente. El resultado es que la mayoría de ellos son JavaScript, aunque puede ser generado dinámicamente (e.g.: aplicaciones CGI cuyo valor de retorno es el *script* que se ejecutará).

B. Formularios

Los formularios proporcionan el punto de entrada a la Web Oculta del lado servidor, por lo que también ha sido necesario estudiarlos en detalle. Se han encontrado 188.712 formularios en 124.865 dominios (21,6%). De ellos, 122.417 (un 64,9%) hacen su petición por POST y 48.443 (un 25,7%) hacen su petición por GET. Del resto, 17.779 (un 14,2%) asumen el valor por defecto (GET). Finalmente, 73 formularios establecen un valor no válido.

También se ha estudiado la distribución del número de formularios por dominio, obteniendo los datos mostrados en la Fig. 3:

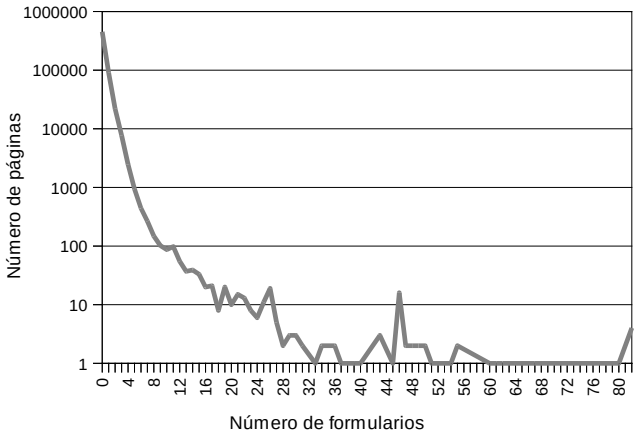


Fig. 3. Uso formularios en la Web española

La Tabla 6 muestra el uso de campos *password* en formularios. El porcentaje es relativo al número de dominios con formularios. El uso de este tipo de campos se asocia a funciones de autenticación, registro o cambio de contraseña.

Campos *password*	Formularios	Dominios	%
1 (autenticación)	26.918	25.832	20,7%
2 (registro)	251	239	0,2%
3 (cambio de clave)	33	33	<0,1%

Tabla 6. Formularios con campos de contraseña.

Se ha encontrado que 5.346 formularios en 4.861 dominios (3.9%) contienen un campo de texto y un botón. Se asume que su función es la de realizar búsquedas sencillas. En cuanto a las búsquedas e introducciones de datos más complejas, se ha hallado que 34.006 formularios en 31.288 dominios (25%) estaban conformados únicamente por

campos de texto y botones y que 126.095 formularios en 91.235 dominios (73%) contenían al menos dos elementos de las siguientes categorías: <input>, <select> y <textarea>.

En cuanto al uso de *scripts* en formularios, 743 dominios contienen formularios que incluyen la etiqueta "javascript:" en el atributo "action" y 735 dominios contienen formularios que incluyen los caracteres "()" en el "action", lo cual puede ser característico de una llamada a una función en un *script*.

También se han detectado 35.554 componentes típicos de formularios fuera de los mismos en 11.033 dominios (1,9%).

Por otro lado, existen 5.082 formularios en 3.742 dominios (0,6%) que no contienen componentes incluidos en la especificación de HTML 4.

C. Otras tecnologías

Las etiquetas <meta> pueden contener información de interés para los *crawlers* (exclusión de robots, redirecciones, *cookies*, etc.). La Tabla 7 muestra algunas funcionalidades para las que se han usado:

Función	**Dominios**	**%**
Refresco o redirección	22.064	3,8%
Refresco (sin URL para redirección)	1.346	0,2%
Estándar de exclusión de robots [16]	128.288	22,2%
Indicar las palabras clave	246.752	42,8%
Envío de *cookies*	26	<0,1%

Tabla 7. Uso de etiquetas <meta>.

Sin embargo, muchos buscadores no tienen en cuenta las palabras clave porque, tal y como se explica en [17], se pueden emplear para hacer *boosting*, es decir, para aumentar el *ranking* de la página y del dominio de forma injusta.

De los dominios que usan redirección, en 15.416 de los casos (2,6% del total de dominios), la página no incluía ningún enlace. En 15.039 casos (también un 2,6%) la página no incluía ni enlaces ni etiquetas <object>, lo cual descarta sitios 100% Flash.

Las aplicaciones Flash son otra dificultad a las que se han de enfrentar los *crawlers*. Se han encontrado etiquetas <object> en 89.911 dominios, lo que representa un 15,6% del total de la Web española. De ellos, 25.060 (un 4,3%) no presentaban ningún enlace mediante etiquetas <a>. En la mayor parte de esos casos, se trata de sitios web 100% Flash.

Por último, Se ha detectado que 257.084 dominios (un 44,6%) contienen elementos <link>. Estos elementos se suelen usar para hacer referencia a hojas de estilos, aunque también podrían referir a otro tipo de recursos (e.g.: *scripts*).

V. Conclusiones y trabajo futuro

Este artículo muestra los principales resultados de un análisis realizado sobre los sitios web de los dominios ".es" a fecha de 2009. En particular un 15,6% de los dominios presentan etiquetas <object> en la primera página, un 21,6% presentan formularios y un 46,2% contienen *scripts*.

La gran mayoría de las páginas emplean JavaScript como lenguaje de *scripting* y la mayoría de librerías de *script* responden a un conjunto reducido de propósitos: estadísticas, dinamización con AJAX, gestión de Flash, creación de menús, renderización de contenido, tratamiento de imágenes y validación de datos.

Por tanto, se puede concluir que una gran parte de los sitios web de los dominios ".es" hacen uso de tecnologías denominadas de Web Oculta, principalmente lenguajes de *scripting*. Por este motivo, están justificados los esfuerzos encaminados en dotar a los sistemas de *crawling* de mecanismos capaces de tratar con estas tecnologías para llegar al mayor número de documentos. El esfuerzo por interpretar dichos lenguajes debe estar dirigido a ECMAScript y en particular a JavaScript. Interpretar otros lenguajes como TCL o VBScript puede requerir un esfuerzo demasiado grande para un resultado poco significativo.

Como trabajo futuro se propone la realización de nuevos *crawlings* de los dominios ".es" para completar el estudio con una evolución de la Web española en los términos tratados en este artículo, para poder determinar de forma más precisa las tecnologías que deben de ser tratadas por los *crawlers*. Este estudio también puede ser interesante para analizar la frecuencia de refresco con la que los *crawlers* deberían recorrer ciertos sitios.

Agradecimientos

Este trabajo de investigación ha sido financiado por el Ministerio de Educación y Ciencia de España y los fondos FEDER de la Unión Europea (Proyecto TIN2009-14203).

El listado de dominios ".es" a partir del cual se ha realizado el *crawling* ha sido proporcionado por la Entidad Pública Empresarial Red.es.

Referencias

[1] M. Bergman. "The Deep Web. Surfacing Hidden Value," *Technical report, BrightPlanet LLC*. December 2000.
[2] M. Álvarez, F. Cacheda and A. Pan. "Análisis Macroscópico de los Dominios .es," *VIII Jornadas de Ingeniería Telemática (JITEL)*. 2009.
[3] K. C.-C. Chang, B. He, M. Patel, C. Li, and Z. Zhang. "Structured Databases on the Web: Observations and Implications," *SIGMOD Record*, vol. 33, no. 3, 2004.
[4] The size of the World Wide Web. http://www.worldwideWebsize.com/
[5] Netcraft. "March 2011 Web Servers Survey": http://news.netcraft.com/archives/category/web-server-survey/
[6] BuiltWith Technology Usage Statistics: http://trends.builtwith.com/
[7] Google - Web Authoring Statistics: http://code.google.com/intl/es-MX/webstats/index.html
[8] Internet Systems Consortium. "The ISC Domain Survey": http://www.isc.org/solutions/survey, 2011.
[9] Entidad Pública Empresarial Red.es: http://www.red.es
[10] VeriSign. "IPS Statistics - Internet-Profiling-Service": http://www.nic.at/en/uebernic/statistics/ips_statistics_informations/
[11] Standard ECMA-262: ECMAScript Language Specification: http://www.ecma-international.org/publications/standards/Ecma-262.htm
[12] M. Weideman and F. Schwenke. "The influence that JavaScript™ has on the visibility of a Website to search engines - a pilot study," *Information Research*, vol. 11, no. 4, July 2006.
[13] B. Wu and B.D. Davison. "Cloaking and Redirection: A Preliminary Study," *Proceedings of First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb '05)*. 2005.
[14] CyberNeko HTML: http://sourceforge.net/projects/nekohtml/
[15] Scripting Media Types: http://www.rfc-editor.org/rfc/rfc4329.txt
[16] M. Koster "A Standard for Robot Exclusion," *Published online*. http://www.robotstxt.org/wc/norobots.htm, 1994.
[17] Z. Gyöngyi and H. Garcia-Molina. "Web Spam Taxonomy," *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb '05)*. 2005.

Caracterización de Servicios en Redes *Ad-Hoc* Inalámbricas mediante Métricas *Cross-Layer*

Leovigildo Sánchez-Casado, Gabriel Maciá-Fernández, Pedro García-Teodoro
CITIC - Departamento de Teoría de la Señal, Telemática y Comunicaciones,
E.T.S. de Ingeniería Informática y de Telecomunicación, Universidad de Granada
C/Periodista Daniel Saucedo Aranda s/n E-18071
{sancale, gmacia, pgteodor}@ugr.es

***Resumen*- Las WANET son redes inalámbricas constituidas sin ningún tipo de infraestructura preexistente y con una serie de peculiaridades, como son los limitados recursos disponibles y la actuación de cada nodo como *router*. Partiendo de la premisa de que las arquitecturas de red derivadas de modelos por capas independientes no resultan adecuadas en redes inalámbricas, el propósito del presente trabajo es el estudio y evaluación de métricas *cross-layer* que permitan caracterizar adecuadamente distintos servicios, con objeto de modelar el comportamiento del entorno y, a partir de ello, determinar desviaciones en el mismo. Los resultados apuntan que esta línea de investigación resulta prometedora, poniendo de manifiesto que el uso de medidas multi-capa proporciona información de gran utilidad para la caracterización de un sistema.**

***Palabras Clave*- WANET; caracterización; métricas *cross-layer*; servicios.**

I. INTRODUCCIÓN

El desarrollo experimentado por las TIC en las últimas décadas ha convertido a estas tecnologías en parte imprescindible de nuestra vida cotidiana. De entre las distintas posibilidades que brindan las TIC, cada día cobran mayor interés las redes inalámbricas (*"wireless"*) [1]; en concreto, las redes *ad-hoc* inalámbricas o WANET.

Las WANET (*Wireless Ad-Hoc Networks*) son un tipo de redes constituidas por dispositivos auto-configurables, geográficamente distribuidos en un área dada y capaces de unirse y abandonar la red dinámicamente. Dichas redes permiten la comunicación entre los nodos sin necesidad de una infraestructura fija o administración centralizada, usando para ello una estrategia multi-salto. Estas características hacen de estas redes un candidato óptimo y especialmente útil en campos como entornos medioambientales o militares, gestión de catástrofes, organización de conferencias, etc. Sin embargo, existen una serie de particularidades a tener en cuenta, referidas a los limitados recursos disponibles: ancho de banda, tiempo de vida de la batería, canal fácilmente accesible, prestaciones de cálculo y procesamiento, etc.

Por ello, el desarrollo de nuevas técnicas *cross-layer* está centrando la atención de numerosas investigaciones. En este contexto, el objetivo del presente trabajo consiste en la obtención de modelos de comportamiento *cross-layer* para distintos servicios. Con este fin, se establecerá un conjunto de métricas que tratan de recoger el comportamiento global del sistema, posibilitando una mejor monitorización del mismo, lo que permitiría actuaciones futuras de interés como, por ejemplo, la posible determinación de anomalías. Como primera aproximación, el presente trabajo aborda la caracterización de servicios de red en distintos puntos del entorno, en base a la utilización de las mencionadas métricas.

El resto del artículo se organiza de la siguiente forma. La Sección II proporciona un análisis del estado del arte, describiéndose las principales métricas a considerar en la Sección III. La Sección IV detalla los escenarios de estudio, así como el entorno de experimentación. La Sección V muestra los resultados experimentales. Finalmente la Sección VI expone las conclusiones y líneas de trabajo futuro.

II. TRABAJOS PREVIOS EN *CROSS-LAYERING* Y CARACTERIZACIÓN DE SISTEMAS

Las arquitecturas derivadas de modelos basados en capas (independientes) como OSI [2], son particularmente prácticas en redes cableadas, pues cada capa se encarga de tareas que únicamente afectan a la capa en cuestión y a sus interfaces con las adyacentes, lográndose de este modo una gran modularidad [3]. Sin embargo, dichas arquitecturas no resultan adecuadas para su aplicación en redes inalámbricas, dadas las características propias de estos entornos [4]. Dichas propiedades, como las interferencias/colisiones o el acceso múltiple al canal, dan lugar a interdependencias que no se presentan en las redes tradicionales. De esta forma, el empleo de técnicas *cross-layer* permite determinar de forma más precisa la interacción entre las distintas capas, aunque generalmente a costa de mayor complejidad y sobrecarga.

La citada ventaja y su aplicabilidad en numerosos ámbitos, como la mejora del control de congestión [5], el soporte para QoS [6], la reducción del gasto de energía [7] o la minimización de la latencia [8], han centrado el interés de numerosos trabajos en el estudio de distintas opciones *cross-layer*, así como en la caracterización de los servicios en función del patrón de tráfico observable. Otro de los focos de investigación es su aplicación en nuevos protocolos de enrutamiento que den soporte a una seguridad mejorada en entornos inalámbricos [9].

ISBN: 978-84-694-5948-5

Otros trabajos persiguen el despliegue de nuevos sistemas de detección de intrusiones o IDS (*Intrusion Detection Systems*). Utilizando una combinación de distintas métricas, se aumenta la tasa de detección, manteniendo estables la tasa de falsos positivos y la de falsos negativos. Thamilarasu [10] propone un IDS con detección individual en cada capa, correlando las salidas con el fin de obtener una decisión final más precisa. CRADS [11] emplea medidas de las capas MAC y de red para obtener una métrica que le permite un mejor proceso de detección. En [12] se demuestra la eficacia de las arquitecturas multi-capa en la detección de ataques en redes *ad-hoc* móviles, más allá del algoritmo de detección usado.

En este marco, nuestro objetivo central a largo plazo es la caracterización multi-capa de una red inalámbrica con objeto de determinar posibles desviaciones en su comportamiento. Y ello orientado a mejorar la seguridad, mediante la adopción de mecanismos de detección y/o reacción más robustos. Planteado este trabajo como una primera fase, el propósito será el estudio y evaluación de métricas *cross-layer* para la caracterización de servicios. La viabilidad de dichas métricas permitirá su extensión posterior al campo de la seguridad.

III. MÉTRICAS *CROSS-LAYER*

Una métrica *cross-layer* es una medida que recoge información de más de una capa de red, bien de los nodos, bien de los enlaces que los interconectan. En este trabajo se ha realizado un estudio de diversas medidas que permiten identificar y representar de forma clara distintas características propias de las redes WANET. Las distintas métricas son clasificadas en función de si son representativas del tráfico agregado (nodos intermedios), o representan a los servicios finales (nodos origen y/o destino),

A. Medidas salto-a-salto

Son un conjunto de métricas que, relativas a los nodos intermedios, permiten modelar y extraer tanto características propias del nodo, como del tráfico agregado que éste soporta:

- *Potencia recibida*: indica la potencia de la señal recibida, en vatios, observada en la antena del receptor. Se calcula para cada paquete correctamente recibido, y su valor se suaviza promediando durante una ventana temporal.

  Variaciones de potencia pueden deberse a cambios en las rutas, obstáculos o nodos "egoístas". Suele facilitarse el valor RSSI (*Received Signal Strength Indicator*) [13].

- *Longitud de la cola*: se define como el número de paquetes a la espera de ser enviados de la capa LLC a la capa MAC en un momento determinado.

  Ante distintas eventualidades, la tasa de generación de la aplicación podría ser demasiado elevada, dando lugar al desbordamiento de la cola y a la pérdida de paquetes.

- *Ventana de contención*: se define como el tamaño actual, en *slots*, de la ventana de contención de la capa MAC en un instante dado. Varía en el rango $[CW_{min}–CW_{max}]$, duplicándose con cada intento de transmisión sin éxito y reiniciándose al valor mínimo con cada entrega correcta.

  De este modo, un valor alto revela la presencia de colisiones, lo cual puede apuntar, por ejemplo, a la existencia de un servicio con alta tasa de datos o a picos de tráfico generados por aplicaciones de tasa variable.

- *Número de colisiones*: indica el número de colisiones provocadas por el acceso simultáneo al medio por parte de dos o más nodos.

  Esta medida es de especial importancia dado que puede afectar a muchas otras, evidenciándose de este modo la naturaleza *cross-layer* de estos entornos.

- *Tamaño de la tabla de encaminamiento*: representa el número de entradas en la tabla de encaminamiento del nodo en cuestión.

  Un cambio en el número de entradas puede ocasionarse, por ejemplo, por colisiones reiteradas que den lugar a cambios en las rutas.

- *Tasa de Transmisión*: muestra la cantidad de datos, en *bits*, que pueden ser enviados por un nodo durante un intervalo temporal determinado. Se estudiará en capa MAC y en capas superiores.

  Una reducción drástica de la tasa de transmisión podría deberse, por ejemplo, a un ataque *blackhole*.

- *Número de flujos*: proporciona el número de flujos que se encuentran activos en el nodo durante un determinado intervalo temporal.

  Por ejemplo, un nuevo flujo junto con un incremento a ráfagas en la tasa de transmisión es indicativo de la existencia de un nuevo servicio de tasa variable.

B. Medidas extremo-a-extremo

En este apartado se engloban aquellas medidas cuya extracción se realiza en los extremos de la comunicación

- *Latencia*: suma de los distintos retrasos (tiempos de cola, procesamiento, transmisión y propagación) que sufre un paquete durante su transmisión entre origen y destino.

  Variaciones en la latencia pueden indicar la presencia de retransmisiones/colisiones, que pueden ser ocasionadas por un aumento del tráfico agregado de la red.

- *Rendimiento*: es una medida de la tasa de datos entregados al receptor. Se determina como el número de *bits* correctamente recibidos por unidad de tiempo.

  Es una de las principales medidas para caracterizar un servicio en el destino, al ser fuertemente dependiente del patrón de generación de tráfico seguido por la aplicación.

- *Tasa de entrega de paquetes*: porcentaje de paquetes que llegan correctamente al destino respecto del total de los enviados. También puede obtenerse su complementario, denominado *tasa de pérdida de paquetes*.

  Las pérdidas pueden deberse a fallos en las rutas, desbordamiento de las colas o múltiples colisiones.

- *Longitud de la ruta*: el número de enlaces existentes en el camino de comunicación entre nodo origen y destino.

  Debe ser tenida en cuenta, pues si varía, se podría estar ante la presencia de congestión en algún nodo intermedio, ocasionada por nuevos servicios agregados.

Las citadas medidas pueden dar lugar a problemas de sub-estimación o sobre-estimación, ya sea por demasiado genéricas, poco precisas o incapaces de detectar determinadas condiciones. Sin embargo, la extracción de métricas *cross-layer* podrá modelar con precisión posibles comportamientos de la red.

IV. Entorno de Experimentación

En esta sección se presenta una descripción del entorno de experimentación utilizado para evaluar la capacidad de caracterización de las métricas descritas, así como los parámetros de configuración y los escenarios considerados.

Para la experimentación se ha usado Network Simulator 2 (NS-2) [14], uno de los simuladores de redes más usados hoy día por la comunidad académica e investigadora. Los nodos forman una red *ad-hoc* operando bajo el módulo IEEE 802.11 [15], cuyos parámetros de configuración (Tabla 1) simulan una red 802.11b. En la topología planteada se han distribuido aleatoriamente 20 nodos en un área de 900x700 metros, como se ve en la Fig. 1. Las simulaciones se han realizado durante un período de 600 segundos, muestreando las medidas a intervalos de 2 segundos.

Parámetro	Valor	Parámetro	Valor
Modelo Radio	*TwoRayGround*	**Tipo MAC**	*802_11*
Canal	*WirelessChannel*	**├CW$_{min/max}$**	31/1023 *slots*
Antena	*OmniAntenna*	**├Tiempo *slot***	20 µs
├Ganancia Tx/Rx	1	**├SIFS**	10 µs
└Altura	1.5 m	**├Tasa Datos**	11 Mb
Interfaz de Red	*WirelessPhy*	**├Tasa Básica**	2 Mb
├Umbral Captura	10 dB	**├Tasa PLCP**	1 Mb
├Umbral Portadora	1.5e^{-11} W ≈ 550 m	**├Preámbulo**	144 bits
├Umbral Rx	3.6e^{-10} W ≈ 250 m	**├Cab. PLCP**	48 bits
├Potencia Tx	0.2818 W ≈ 250 m	**└Umbral RTS**	0 bytes
├Frecuencia	914 MHz	**Tipo Cola**	*PriQueue*
└Factor Pérdidas	1	**└Tamaño**	50

Tabla 1. Parámetros de configuración utilizados en NS-2.

En el modelo se ha seleccionado de forma aleatoria un nodo como fuente del tráfico y otro como sumidero. El objetivo será comprobar si se puede extraer información de las diferentes capas que permita identificar la categoría del flujo de referencia. En concreto, se experimentará con dos servicios a distintas tasas, mostrados en la Tabla 2: tráfico CBR (*Constant Bit-Rate*) y VBR (*Variable Bit-Rate*). La tasa media de transmisión en ambas aplicaciones es la misma, con el fin de simular condiciones semejantes.

	CBR	***VBR***
Tamaño paquete	512 *bytes*	512 *bytes*
Tasa de Tx (T_{ON})	21 / 44.8 / 76 Kbps	30 / 64 / 95 Kbps
Distribución de llegada	constante	exponencial
Tiempo de ON/OFF	-	0.7 / 0.3 s

Tabla 2. Parámetros de las fuentes de tráfico.

Como protocolo de enrutamiento se ha utilizado AODV, mostrándose los valores de sus parámetros en la Tabla 3.

Parámetro	Valor	Parámetro	Valor
Tiempo vida ruta	10 s	**# Retransm. RREQ**	3
Tiempo vida ruta inversa	6 s	**Tiempo espera RREP**	1 s
Tiempo límite RREQ	10 s	**Detección capa enlace**	si

Tabla 3. Parámetros de AODV.

Para simular los escenarios, se añadirá tráfico de fondo, variando la carga de la red y el tipo de tráfico agregado. En concreto, se han estudiado 4 posibles escenarios, repitiendo cada simulación y modificando el tipo de fuente de referencia (CBR o VBR), con el fin de comprobar si es posible la clasificación de dicho flujo en su categoría correspondiente. La configuración es la siguiente:

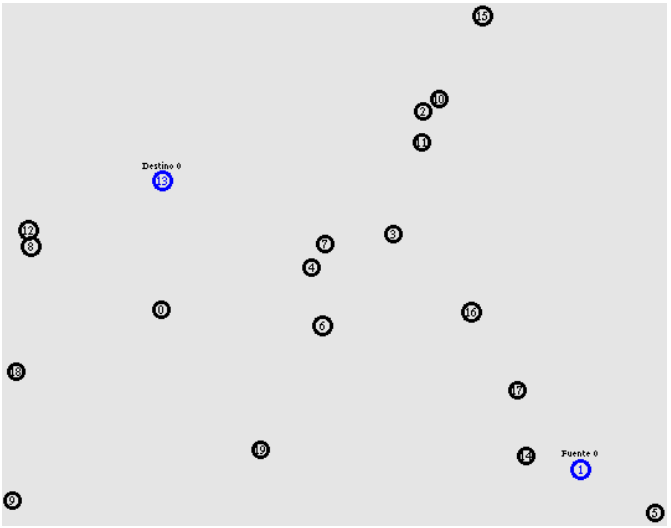


Fig. 1. Topología utilizada en la experimentación.

A) *Escenario sin tráfico agregado:* representa el caso básico: se transmite únicamente el flujo generado por la fuente, que puede ser tanto CBR como VBR.
B) *Escenario con 9 flujos CBR y VBR*: se introducen flujos agregados en la red, distribuyéndose de forma proporcionada los dos servicios generadores.
C) *Escenario con 9 flujos CBR: en e*ste caso todos los flujos agregados pertenecen a una aplicación de tipo CBR, obteniéndose un escenario poco equilibrado.
D) *Escenario con 9 flujos VBR:* análogamente al escenario previo, se introduce tráfico de un único servicio (VBR) con el fin de obtener un entorno no balanceado.

V. Resultados Experimentales

Toda vez que se ha realizado la simulación y se han extraído las diferentes medidas, el siguiente paso es comprobar si existe información en las métricas que permita la diferenciación de los servicios considerados.

Para ello se ha utilizado Weka [16], una plataforma *software*, escrita en Java y de libre distribución, que permite realizar operaciones de *data mining* y *machine learning*. Permite aplicar, sobre un conjunto de datos (o *atributos*), diferentes algoritmos de clasificación, regresión, asociación o *clustering*. En el caso de estudio, los datos son los valores de las métricas para distintas muestras temporales (o instancias).

Como *baseline* para la clasificación se ha tomado un algoritmo de aprendizaje ampliamente usado: el clasificador *Naïve Bayes* [17], un modelo generativo que hace uso de la regla de Bayes para estimar distribuciones de probabilidad conjuntas sobre parejas de etiquetas y observaciones. La característica principal de dicho modelo es que asume la independencia condicional de los atributos, lo que reduce drásticamente el número de parámetros a ser estimados para la generación del modelo y, en consecuencia, el número de instancias necesarias para obtener resultados precisos.

La conclusión inmediata que puede deducirse de los resultados mostrados en la Tabla 4 es que, como se había supuesto, el análisis de métricas *cross-layer* proporciona información suficiente como para poder realizar la clasificación de los servicios. De hecho, empleando un

clasificador simple como *Naïve Bayes*, la tasa de instancias correctamente clasificadas supera, en promedio, el 80 %. Además, es de esperar que estas tasas de clasificación sean mejoradas utilizando otros clasificadores más eficientes.

Esc.	Tasa de Tx	Origen	Intermedios	Destino
A)	Baja	82.72 %	82.56 %	79.07 %
	Media	85.05 %	84.72 %	85.22 %
	Alta	85.54 %	85.38 %	83.55 %
B)	Baja	84.39 %	76.00 %	83.72 %
	Media	85.88 %	75.91 %	87.54 %
	Alta	80.56 %	62.13 %	65.12 %
C)	Baja	83.22 %	87.13 %	92.52 %
	Media	87.71 %	98.26 %	99.50 %
	Alta	82.56 %	71.20 %	75.42 %
D)	Baja	82.72 %	78.07 %	78.90 %
	Media	85.38 %	82.32 %	86.89 %
	Alta	83.72 %	57.72 %	70.93 %

Tabla 4. Tasa de clasificación con *Naïve Bayes* para los distintos escenarios.

Asimismo, se puede concluir que, por regla general, la clasificación se realiza con mayor precisión en el nodo origen. Este resultado es coherente, dado que la propagación del flujo introduce distintas alteraciones (colisiones, retransmisiones, encolamientos, pérdidas, etc.) provocando que éste llegue al destino *distorsionado*. Se puede deducir también que los nodos intermedios no son, en principio, el mejor punto de la red en el que caracterizar el servicio. La agregación de diversos flujos y la falta de métricas extremo-a-extremo dificultan esta caracterización. A pesar de todo, se alcanza una tasa de clasificación del 70-80 % sobre tráfico agregado, lo que resulta, a todas luces, prometedor.

Por último se debe indicar que la tasa de clasificación es superior en escenarios de tráfico no equilibrado. Es decir, se obtiene una mejor discriminación cuando la proporción de instancias que se diferencian de la mayoría es reducida. Esta conclusión puede mostrarse de gran importancia, en tanto que el propósito final es detectar anomalías en el funcionamiento de la red y es de suponer que éstas representen, respecto del conjunto global del tráfico, una proporción minoritaria.

Como se presuponía en un principio, esta línea de investigación se ha demostrado prometedora, manifestándose que el uso de métricas multi-capa proporciona información adicional de gran utilidad para la caracterización de servicios.

VI. Conclusiones y Trabajo Futuro

A lo largo del presente trabajo se ha presentado un conjunto de métricas de naturaleza *cross-layer* que permiten caracterizar servicios en entornos *ad-hoc* inalámbricos. Su uso en el modelado del comportamiento del sistema permitirá la determinación de desviaciones en el mismo, lo que se prevé de gran relevancia. Tal y como se ha mostrado en la sección previa, los resultados experimentales son esperanzadores. Sin embargo, este primer estudio realizado adolece de algunas limitaciones que serán tenidas en consideración en el futuro:

- La inclusión de modelos de movilidad. Varias de las métricas empleadas pueden proporcionar información adicional en entornos con movilidad.
- La introducción de modelos energéticos. Asimismo, establecer una serie de métricas relacionadas, como pueda ser la tasa de consumo o EDR (*Energy Drain Rate*) [7].
- Estudio de métricas más precisas. Cabe citar la *utilización de la capa MAC*, definida como la fracción de tiempo en la cual el nodo se encuentra disponible para transmitir.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el MICINN mediante el proyecto TEC2008-06663-C03-02.

Referencias

[1] T.S. Rappaport, A. Annamalai, R.M. Buehrer and W.H. Tranter, "Wireless communications: Past events and a future perspective", *IEEE Communications Magazine*, vol. 40, pp. 148-161, May 2002.

[2] H. Zimmermann, "OSI Reference Model – The ISO Model of Architecture for Open Systems Interconnection", *IEEE Transactions on Communications*, vol. 28, no. 4, pp. 425-432 April 1980.

[3] D. Chafekar, "Capacity Characterization of Multi-Hop Wireless Networks- A Cross Layer Approach", *Ph.D. Dissertation*, State University of Virginia, March 2009.

[4] C. Barrett, A. Marathe, M. Marathe and D. Martin, "Characterizing the interaction between routing and MAC protocols in ad-hoc networks", *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC'02)*, pp 92-103, 2002.

[5] L. Chen, S. Low, M. Chiang and J. Doyle, "Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks", *Proceedings of IEEE International Conference on Computer Communications (INFOCOM'06)*, pp. 1-13, April 2006.

[6] S. Sridhar, R. Baskaran, "A Survey on QoS Based Routing Protocols for MANET", *International Journal of Computer Applications*, vol. 8, no.3, Oct. 2010.

[7] K. Ghada, J. Li, Y. Ji and G. Wang, "Cross-layer Approach for Energy Efficient Routing in WANETs", *IEEE 6th International Conference on Mobile Adhoc and Sensor Systems (MASS'09)*, pp. 393-402, Oct. 2009.

[8] D. Chafekar, V.S. Anil Kumar, M. Marathe, S. Parthasarathy and A. Srinivasan, "Cross-layer latency minimization in wireless networks with SINR constraints", *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC'07)*, pp. 110-119, Sept. 2007.

[9] Poonam, K. Garg and M. Misra, "Trust Based Security in MANET Routing Protocols: A Survey", *Proceedings of the 1st Amrita ACM-W Celebration on Women in Computing*, pp. 1-7, Sep. 2010.

[10] G. Thamilarasu, A. Balasubramanian, S. Mishra and R. Sridhar, "A cross-layer based intrusion detection approach for wireless ad hoc networks", *IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS'05)*, pp. 854-861, Nov. 2005.

[11] J.F.C. Joseph, A. Das, B.C. Seet and B.S. Lee, "CRADS: Integrated Cross Layer Approach for Detecting Routing Attacks in MANETs", *Proceedings of the IEEE Wireless Communications and Networking Conference*, pp.1525-1530, 2008.

[12] J.F.C. Joseph, A. Das, B.C. Seet and B.S. Lee, "Cross Layer versus Single Layer Approaches for Intrusion Detection in MANETs", *15th IEEE International Conference on Networks (ICON'07)*, pp.194-199, Nov. 2007.

[13] V.C.M. Borges, M. Curado and E. Monteiro, "Cross-layer routing metrics for mesh networks: Current status and research directions", *Elsevier Journal on Computer Communications*, vol 34, no. 6, pp. 681-703, May 2011.

[14] S. McCanne and S. Floyd, "NS Network Simulator", http://www.isi.edu/nsnam/ns/

[15] ISO/IEC 8802-11; ANSI/IEEE Std 802.11, 1999 edition, "Information technology - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", *The Institute of Electrical and Electronics Engineers*, 1999.

[16] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, "The WEKA Data Mining Software: An Update", *SIGKDD Explorations*, vol. 11, no. 1, 2009.

[17] T. Mitchell, *Machine Learning, Chapter 1. Generative and Discriminative Classifiers: Naïve Bayes and Logistic Regression*, McGraw Hill, Draft of Jan. 2010.

Mejorando el rendimiento de las redes de acceso WiFi

Domingo Marrero Marrero, Elsa Mª Macías López, Álvaro Suárez-Sarmiento
Grupo de Arquitectura y Concurrencia (GAC)
Departamento de Ingeniería Telemática-Universidad de Las Palmas de Gran Canaria
Campus Universitario de Tafira. Edificios de Telecomunicación
35017 Las Palmas de Gran Canaria (Gran Canaria)
Email: {dmarrero,emacias,asuarez}@dit.ulpgc.es

***Resumen*- Las redes de acceso WiFi no garantizan plena conectividad ni Calidad de Servicio (cortes y retrasos en las comunicaciones) especialmente para servicios multimedia. Las mejoras propuestas a la tecnología WiFi y otros estándares no han tenido el éxito deseado en la práctica. Soluciones como regulación de tráfico, el balanceo de terminales entre los puntos de acceso y conocimiento proactivo de ubicación de terminales introducen mejoras parciales. Nosotros presentamos una solución teórica y diferentes pruebas prácticas en estas tres líneas. Usamos una solución software (modelo gestor-agente) a nivel de aplicación considerando el estado del canal. El agente en los terminales móviles regulan sus tráficos y el gestor recibe información de los agentes para indicar o aconsejar una mejor re-asociación a los puntos de acceso. Los resultados experimentales demuestran una mejora considerable de las condiciones de acceso utilizando estas mejoras.**

***Palabras Clave* - Calidad de Servicio, Colas, Gestor-Agente, Puntos de Acceso, RSSI, WiFi, Regulación de Tráfico, Servicios Multimedia.**

I. INTRODUCCIÓN

Actualmente, el acceso inalámbrico a internet esta mayoritariamente basado en la tecnología *Wireless Fidelity* (*WiFi*) [1]. En ésta, si más de un terminal utiliza el mismo canal solapado o compartido, se produce una degradación de la *Calidad de Servicio* (*Qos*, *Quality of Service*): reducido ancho de banda, interferencias radio y saturación del espectro radioeléctrico [2] y contemplando una única clase de tráfico: *best effort*. Para reducir estos problemas se aplican algunas mejoras: a) la norma 802.11e [3] [4] del *Institute of Electricals and Electronics Engineers* (*IEEE*) prioriza el acceso al canal y clasifica el tráfico, b) la IEEE 802.11n aprovecha las múltiples transmisiones y el mayor ancho de banda para aumentar la velocidad de transmisión, c) rediseño de la subcapa *Medium Access Control* (*MAC*) para priorizar tráfico: audio, vídeo, y *best-effort* [5].

Para mejorar la QoS se deben gestionar diferentes parámetros: velocidad de comunicación, throughput, latencia, jitter, tiempo invertido en *handover* entre Puntos de Acceso (*PAs*), etc. Además, se consideran otros aspectos como la re-asociación de terminales a PAs [6] eligiendo el mejor PA. Para ello se puede tener en cuenta conjuntamente la cantidad de terminales asociados a él y el mejor nivel de *Received Signal Strength Indicator* (*RSSI*) que percibe el terminal, minimización de interferencias y colisiones en el MAC, ubicación física de los PAs y los terminales, el tipo de tráfico que envían los terminales al canal y el que reciben de Internet, etc. Todos ellos pueden mejorar la experiencia del usuario y la calidad de la comunicación.

En trabajos previos hemos demostrado que, en la práctica, es posible mejorar las condiciones de conectividad en redes de acceso WiFi mediante la incorporación de diferentes funcionalidades. En este trabajo presentamos una formulación matemática y nuevos resultados experimentales para mejorar el acceso a Internet desde redes WiFi. Estas mejoras consisten en: a) una regulación distribuida del tráfico b) la re-asociación de terminales a PAs teniendo en cuenta parámetros de rendimiento y c) re-asociación basada en la técnica de localización usando un *mapa de cobertura* WiFi.

En el apartado II se presenta el modelo matemático para caracterizar los distintos mecanismos de mejora de la QoS. En el apartado III exponemos los diferentes mecanismos y una solución algorítmica integrada para implementar los mecanismos anteriores. En el apartado IV presentamos brevemente los resultados experimentales. Finalmente, en el apartado V, exponemos algunas conclusiones y líneas de trabajo futuro.

II. MODELADO MATEMÁTICO DE CONTROL DE TRÁFICO

Nuestro modelo se usa para el control de tráfico (por flujos vinculados a conexiones de los terminales en los PAs) para obtener una regulación y distribución eficiente entre todos los PAs disponibles. Consideramos que el tráfico es el conjunto de paquetes entrantes o salientes a los terminales. Sea Tt_i el tráfico de un terminal y sea T_T el tráfico entrante o saliente por un determinado PA (al que se supone que existen *n* terminales asociados) entonces se cumple que:

$$T_T = \sum_{i=1}^{n} Tt_i \qquad (1)$$

El Tt_i está constituido por *m'* flujos de tráfico (F_k^i, *k=1..m'*, $m' \geq 1$) heterogéneos (vídeo, audio, *best effort,* etc.), en general. Esto es,

$$Tt_i = \sum_{k=1}^{m'} F_k^i \qquad (2)$$

Con lo cual, partiendo de la ecuación (1) y sustituyendo en ella la ecuación (2), tenemos:

$$T_T = \sum_{i=1}^{n} \sum_{k=1}^{m'} F_k^i \qquad (3)$$

Lo cual explica la contribución de tráfico de cada terminal y sus flujos en el PA al que se vinculan los diferentes terminales. Esta contribución se puede representar

ISBN: 978-84-694-5948-5

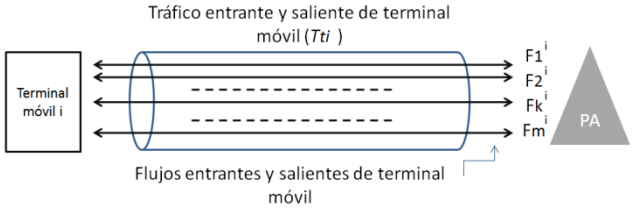


Fig. 1. Esquema de tráficos y flujos

con una matriz cuyas *n* columnas representan los Tt_i de los terminales y las *m* filas todos los flujos por cada terminal (*m* es la suma de todos los valores de *m'* de cada Tt_i). Sea C_T la capacidad teórica máxima de cada canal WiFi (Fig. 1) y sea $\alpha_k^{\,i}$ un coeficiente de reducción del tráfico vinculado al $F_k^{\,i}$, la regulación se puede expresar como:

$$F_k^i = \propto_k^i \; C_T \qquad (4)$$

Lo cual indica que cada flujo solo podrá hacer uso de una proporción $\propto_k^i$ de la capacidad total máxima del canal. Con lo cual, sustituyendo F_k^i en (3) y factorizando:

$$T_T = C_T \sum_{i=1}^{n} \sum_{k=1}^{m'} \propto_k^i \qquad (5)$$

Siendo:

$\propto_k^i = 0$	el valor para un F_k^i no permitido.
$0 < \propto_k^i < 1$	el valor para un F_k^i regulado. Los valores próximos a *1* se asignan para flujos de vídeo/audio y bajos para los *best-effort*.
$\propto_k^i = 1$	el valor para un F_k^i no regulado.

Esta regulación se puede representar como una *matriz de regulación de tráfico* cuyas columnas representan tráficos de los diferentes terminales (Tt_i) y las *m* filas representan a todos los F_k^i . La ampliación para contener a *n* PAs (diferentes tráficos T_{Ti}) es inmediata, con lo cual nos queda la representación de la Fig. 2. Cada T_{Ti} se comunica por el canal WiFi especificado por el PA al que esté asociado y compite por el uso del ancho de banda disponible en ese canal con el resto de flujos de otros tráficos de otros terminales en ese mismo canal.

Operando de forma distribuida entre el gestor y los agentes sobre esta matriz de tráfico, se puede alcanzar una mejor distribución del uso de cada canal por los distintos terminales. Para ello es necesario que el gestor averigüe el estado del mismo de forma regular.

III. MECANISMOS DE CONTROL: SOLUCIÓN INTEGRADA

A continuación presentamos las tres técnicas integradas en nuestro método. Para cada una, presentamos el trabajo relacionado, un modelo formal para su aplicación y algunas claves de integración.

$$\begin{pmatrix} \propto_1^1 & \propto_1^2 & \dots & \propto_1^n \\ & \propto_2^2 & & \dots \\ \dots & & \dots & \propto_{m-1}^n \\ \propto_m^1 & \propto_m^2 & \dots & \propto_m^n \end{pmatrix}$$

Fig. 2. Matriz de tráfico

III.1 Regulación de tráfico en el terminal

En la mayor parte de los PAs, un terminal inalámbrico debe pasar una política de control de admisión [7] para acceder a Internet, siendo éste el que controla el acceso y prioriza los flujos como en [8]. Esta regulación, pensamos que podría ser aplicada en el propio terminal con resultados que pueden ser más eficientes. Para ello, a través de un portal cautivo, el usuario puede registrar la clase de tráfico que pretende utilizar, o dinámicamente el PA puede determinar la cantidad de tráfico a ser tratado por cada terminal globalmente y la cantidad de paquetes a ser enviados para cada tipo de tráfico. Toda esta información se envía a cada terminal para que éste regule su tráfico (flujos). El PA conoce la matriz de regulación y envía una porción de filas determinada a cada terminal para que aplique los valores de $\propto_k^i$.

En la Fig. 3, se muestra que el terminal 1 tiene toda la capacidad máxima del canal disponible para acceder al *PA1* y no debería tener restricciones, salvo para priorizar un flujo frente a otros, si fuese el caso. El tráfico entrante o saliente del terminal 1 en el canal 1, para un solo servicio sería:

$$Tt_1 = F_1^1 \qquad (6)$$

Por tanto, teniendo en cuenta el tipo de tráfico, número de flujos y número de terminales, podría especificarse una regulación o distribución del ancho de banda acorde a las condiciones del canal y, adicionalmente, al estado de la red cableada; no en vano confluyen en ella el tráfico de todos los PAs de la misma subred o sistema de distribución. En el canal 2 podría priorizarse un flujo o tráfico frente a otro y no competir en igualdad de condiciones.

III.2 Re-asociación de terminales al mejor PA

El proceso de re-asociación en una red WiFi significa que un terminal se desasocia de un PA y se asocia a otro PA diferente. Nosotros consideramos un caso especial en el que un terminal se encuentre asociado a cierto PA y sea forzado a re-asociarse a otro PA (este proceso lo hemos denominado *roaming estático*) no sólo considerando el nivel de RSSI sino la carga de los PAs y otros parámetros. En la Fig. 4 se muestra un posible escenario. El canal controlado por el *PA2* estaría compartido por dos terminales. Por el contrario, el canal del *PA1* está vacío. En teoría y en igualdad de características de todos los PAs, el terminal 1 tendría mejores prestaciones si estuviera asociado al *PA1*. Si el terminal 1 está asociado al *PA2*, entonces el *PA2* activaría el *roaming*

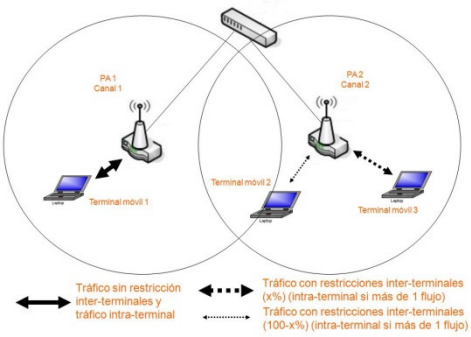


Fig. 3. Regulación inter-terminal e intra-terminal

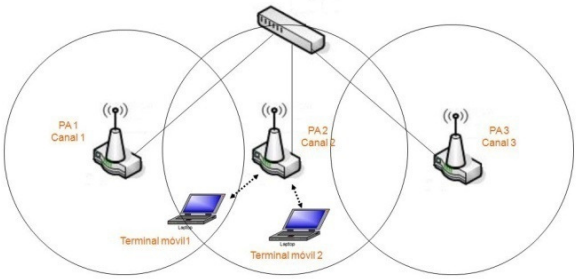


Fig. 4. Reasociación guiada desde los PAs

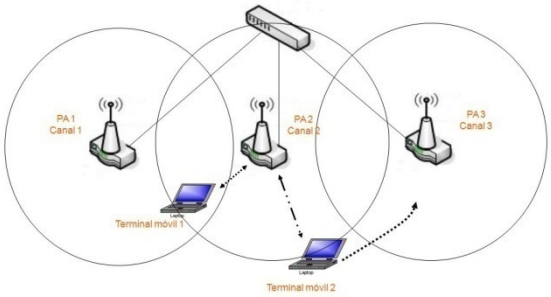


Fig. 5. Re-asociación según localización desde los PAs

estático del terminal 1 al *PA1* y no el terminal, como se hace, por defecto, cuando decae el nivel mínimo de RSSI.

III.2 Re-asociación de terminales a PA usando localización de terminales

La localización de los terminales se realiza teniendo en cuenta los niveles de RSSI. Para ello, consideramos que los PAs son servidores de localización e incorporan un mapa de cobertura (previamente creado con valores de RSSI para cada zona y PA). Estos servidores determinan la posición estimada de cada terminal en función de los datos que los propios terminales le proporcionen en cada momento. Con estos resultados, los terminales pueden ser guiados hacia una mejor ubicación. En la Fig. 5 se muestra que el terminal 2 puede detectar sólo el *PA2*, y quizás con bajo RSSI. Si recibe información de su ubicación o una indicación de una mejor posición desde el servidor de mapa de cobertura, podría ser capaz de descubrir la presencia también del *PA3*.

III.3 Una solución algorítmica integrada

Una solución matemática exacta para el problema de optimización global identificado es muy complejo o quizás imposible. Una solución heurística algorítmica se puede especificar programando el comportamiento concurrente de los agentes. La principal ventaja para alcanzar una rápida solución es analizar el estado del canal y el comportamiento de los agentes pueda ser modificado para adecuarse a la variabilidad de los parámetros de control. Además, es relativamente fácil incluir nuevos parámetros de control. Esta es la razón por la que nosotros consideramos una arquitectura gestor-agente para especificar una solución algorítmica.

Los gestores fueron programados en varios PCs funcionando como encaminadores inalámbricos: *Linux Wireless Router* (*LWR*) y la regulación de tráfico aplicada en un agente (*Agent*) en cada terminal. Para la política de control de admisión, es necesario que el usuario indique la clase de servicio y los requisitos de ancho de banda del servicio (F_k^i) o detectado dinámicamente. Con esta información, el gestor podría ser capaz de estimar cómo este tráfico afectaría a otros flujos existentes o entrantes (F_w^i, $k \neq w$). Debido a que la regulación se aplica en los terminales, el gestor comunica a cada agente qué regulación debe aplicar en cada caso (filas de determinada columna de matriz con $\propto_k^i$). Esta distribución es dependiente del número de sesiones, estado del canal y clase de servicio. Por último, es necesario contar con un *sniffer* para detectar la alteración de las condiciones asignadas.

Periódicamente, los gestores difunden anuncios con la identificación de terminales asociados a cada uno de ellos (y clases de tráfico o flujo (F_k^i)), el estado del canal WiFi, características de los terminales, etc. Simultáneamente los terminales detectan (usando RSSI) los diferentes PAs. Por otro lado, los PAs realizan una negociación acerca del mejor mapeo de los *m'* F_k^i en el *PA'Set* (donde $m' \leq m$ y el $PA'Set \subseteq PASet$). Cada agente informa periódicamente al gestor acerca de los PAs que detecta y estos, determinan de forma aproximada la localización del terminal.

IV. Nuevos Resultados Experimentales

Para testear y aplicar nuestra herramienta hemos definido una plataforma de testeo [9] [10] basada en LWRs y varios terminales (PC portátiles *Linux*).

IV.1 Resultados experimentales para regulación de tráfico en los terminales

Con nuestro sistema de regulación de tráfico en el terminal guiado desde el gestor, hemos obtenido unos resultados adecuados asignando un alto coeficiente $\propto_k^i$ para flujos multimedia y bajos para otros flujos. En todos los casos de test realizados, la QoS observada para flujos multimedia se mejoró considerablemente cuando se limitaron otros flujos respecto a cuando no se aplicaron regulaciones. Los agentes aplican la regulación con funciones de control de tráfico desarrolladas para el kernel de Linux (*iproute* y *Traffic Control* (*TC*)).

Como herramienta generadora de tráfico y medida, utilizamos *iperf*, que permite medir las prestaciones de las redes (*throughput*). Dado que inyecta una gran cantidad de tráfico, crea muchos problemas al resto de aplicaciones (retardos, pérdidas y cortes). Para analizar la experiencia visual del usuario, usamos tráfico multimedia *Real Time Protocol* (*RTP*)/*Real Time Streaming Protocol* (*RTSP*) mediante la aplicación *VídeoLAN Client* (*VLC*) y usando *mplayer* desde un servidor web. La calidad de la señal multimedia cuando *iperf* inyectaba tráfico era muy mala (largos períodos de inactividad, pérdida de paquetes y pixelaciones). Por el contrario estos efectos se limitaron cuando a *iperf* se le aplicó regulación mediante esta funcionalidad aplicada por el agente correspondiente.

Complementariamente a las pruebas anteriores, mediante una aplicación independiente también desarrollada, denominada *inyecttraffic*, pudimos generar tráfico masivo en el canal (denominado *tráfico interferente*) para simular el efecto de la saturación del canal y, además variar la latencia y el tamaño de las tramas insertadas en el canal para testear su efecto sobre otras comunicaciones, con y sin regulación. En la figura 6 se muestra la capacidad teórica disponible en *Mbps* (C_T) para nuestro escenario y los instantes de tiempo en

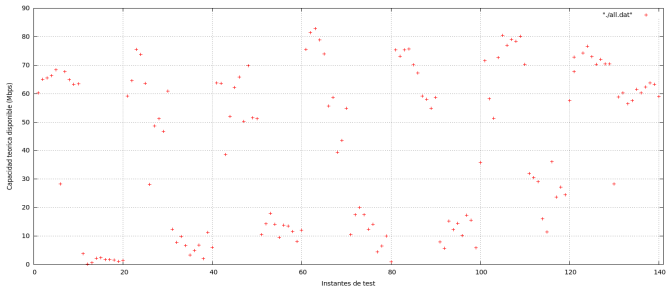


Fig. 6. Efecto de trafico interferente sobre capacidad de canal 802.11n

los que usó de forma simultánea con la herramienta *iperf*. Los resultados experimentales por intervalos de tiempo fueron:

1. ***Instantes de test de 1 .. 9***. Solamente tráfico *iperf*. Se obtienen valores en torno a los 65 Mbps.
2. ***Instantes de test de 10 .. 19***. Se inyecta tráfico interferente: tramas de 1.500 Bytes con la mínima latencia entre tramas. Se aprecia que este tráfico interferente permite solo unos 2 Mbps.
3. ***Instantes de test de 20 .. 29***. Se detiene el tráfico interferente y la velocidad retorna a los iniciales.

A la vista de los resultados, se observa una degradación importante en la disponibilidad del canal WiFi. A continuación se analiza el comportamiento de la regulación del tráfico interferente limitándolo con distintos valores de los coeficientes $\propto_k^i$ (nótese que tenemos un solo flujo interferente a regular (*injecttraffic*) y otro sin regulación (*iperf*)).

4. ***Instantes de test de 30 .. 140***. Se aplica diferente regulación al tráfico interferente: 20 Mbps entre 30 y 39, 18 Mbps entre 50 y 59, 15 Mbps entre 70 y 79, 10 Mbps entre 90 y 99, 5 Mbps entre 110 y 119 y 1 Mbps entre 130 y 140. En los otros intervalos (40 y 49, 60 y 69) no se regula el tráfico interferente. Nótese que solamente se consiguen capacidades elevadas, por encima de 50 Mbps, cuando el tráfico interferente está limitado a 1 Mbps frente a los 75 Mbps sin tráfico interferente. Estos datos corroboran la necesidad de aplicar regulación a ciertos flujos en beneficio de los dependientes del tiempo para garantizar una mayor disponibilidad del canal.

IV.2 Test experimentales para re-asociación de terminales

Se han realizado múltiples pruebas con varios LWR y varios terminales asociados. Bajo ciertas condiciones de servicios y estado de los LWR, el terminal seleccionado por el LWR por contar con las condiciones idóneas, fue forzado desde el gestor a cambiar de LWR. Se definió una tabla de clasificación de flujos para la toma de decisiones por los LWR. Por último, la localización de terminales se incorporó a nuestro escenario de pruebas (se cuenta con pocos Pas) y tras múltiples pruebas realizadas destacamos: la covarianza de los diferentes valores de RSSI y pocas referencias dado el reducido número de PAs. De acuerdo con la variabilidad de valores de RSSI, no sólo dependientes de múltiples factores (terminales, flujos, etc.), sino de la interfaz del terminal que rastrea los canales, la localización en interiores mediante *fingerprinting* basado en los valores de RSSI es muy poco precisa para nuestro escenario, dado el reducido número y la ubicación de los PAs disponibles. Aún así, su aplicabilidad para nuestro objetivo fue contrastada en muchos de los casos.

V. CONCLUSIONES Y LÍNEAS FUTURAS DE INVESTIGACIÓN

En redes Wifi, los requisitos de QoS: velocidad, ancho de banda, latencia, jitter y *throughput* no están garantizados cuando en el canal compiten varios usuarios concurrentemente usando flujos de tráfico heterogéneos. Para evitar o mejorar estos problemas nosotros hemos trabajado integrando varios mecanismos. Para ello realizamos una clasificación de servicios y aplicamos diferentes estrategias de priorización o regulación. Manejamos una re-asociación más eficiente de los terminales a los PAs disponibles, no sólo considerando los niveles de RSSI, sino basándonos en el estado de los canales y la ubicación de dichos terminales. Hemos implementado una herramienta combinada que integra estos tres mecanismos y pensamos que podría formar parte de soluciones como DD-WRT, u otras similares, o como parte de otros sistemas empotrados actuando como PA/Encaminador. Los resultados experimentales nos demuestran la eficacia de llevar a cabo esta solución integrada.

Varias líneas de trabajo incluyen el estudio del efecto del tamaño del paquete en el balanceo de la carga de los PAs, la mejora del algoritmo distribuido basado en la matriz de regulación de tráfico y la agrupación de flujos por características.

REFERENCIAS

[1] IEEE 802.11g (2003), 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 4: Further Higher-Speed Physical Layer Extension in the 2.4 GHz Band

[2] Jangeun, J., Peddabachagari, P., Sichitiu, M. (2003). "Theoretical Maximum Throughput of IEEE 802.11 and its Applications", *IEEE International Symposium on Network Computing and Applications (NCA)*, Pages: 249-256.

[3] Xiao, Y. (2004). "IEEE: 802.11e: QoS Provisioning at the MAC layer". IEEE Wireless Communications.

[4] Qiang, N., (2005, July). "Performance analysis and enhancements for IEEE 802.11e wireless networks"IEEE Communication, ISSN: 0890-8044, Volume: 19, Issue: 4, Pages: 21 – 27.

[5] Vinnakote, S., Naresh, S.V.S., Pasupuleti, S. (2006, August). "New-MAC protocol for enhancement of QoS performance in wireless LAN". Wireless and Optical Communications Networks, 2006 IFIP International Conference on. ISBN: 1-4244-0340-5.

[6] Rodrigues, J.C., Fraiha, S., Araujo, J., Gomes, H., Frances, C., Cavalcante, G. (2009, November). "Empirical study of the QoS parameters behavior of a VoIP application in wi-fi networks". Microwave and Optoelectronics Conference (IMOC), 2009 SBMO/IEEE MTT-S Int. ISSN: 1679-4389. Issue 3-6 Nov. 2009. Pages: 257 - 261.

[7] Prihandoko, F., M. H. Habaebi and B. M. Ali, (2003). "Adaptive call admission control for QoS provisioning in multimedia wireless networks, Computer Communications, vol. 26, Pages: 1560-1569.

[8] Fang, Y., Zhang, Yi, (2002, March) "Call admission control schemes and performance analysis in wireless mobile networks", ISSN: 0018-9545 Volume: 51, Issue: 2, Pages 371 – 382.

[9] Marrero, D., Suárez, A., Macías, E.M. (2007, July). "Dynamic Traffic Regulation for WiFi Networks". World Congress on Engineering 2007 (WCE2007). ICWN'07. July 2-4 2007, ISBN978-988-98671-2-6. Londres.

[10] Marrero, D., Suárez, A., Macías, E.M. (2009, September). "Aplicación multifuncional para gestión del canal en redes IEEE 802.11."VIII Jornadas de Ingeniería Telemática. JITEL2009. pp 16-23. 15-17 September. ISBN: 978-84-96997-27-1, Cartagena (España).

Construyendo redes empleando recursos prestados de otros

Oriol Madriles, Xavier Hesselbach
Departamento de Ingeniería Telemática
Universidad Politécnica de Cataluña UPC
C/ Jordi Girona, 1-3, 08034 Barcelona
oriol.madriles@gmail.com, xavierh@entel.upc.edu

***Resumen*- La virtualización de red es considerada como una de las soluciones para resolver el problema de la osificación en la Internet actual, incapaz de superar los retos que requieren los nuevos servicios debido a la falta de coordinación entre los proveedores de servicio. Permitiendo que múltiples arquitecturas de red heterogéneas cohabiten sobre un mismo substrato físico, la virtualización de red provee flexibilidad y posibilita la mejora en la gestión de recursos. Esta ponencia presenta una arquitectura flexible e independiente de cualquier tecnología para la gestión de redes virtuales, por medio de la introducción de módulos específicos de control que permitan la gestión de la calidad de servicio y la ingeniería de tráfico entre diferentes clases de tráfico en cada red virtual.**

***Palabras Clave*- Virtualización de Red, Calidad de Servicio, Ingeniería de Tráfico, Autogestión de Recursos.**

I. INTRODUCCIÓN

La arquitectura de Internet después de 35 años de existencia presenta serias dificultades para superar los retos que se derivan de la demanda de nuevos servicios.

La falta de cooperación entre los distintos Proveedores de Servicio de Internet no permite cambios radicales en la arquitectura actual, y por lo tanto dificulta el testeo e implementación de nuevos protocolos e innovaciones propuestos por la comunidad de investigadores. Un ejemplo de ello es la migración de IPv4 a IPv6. Esta situación es la que se conoce por osificación.

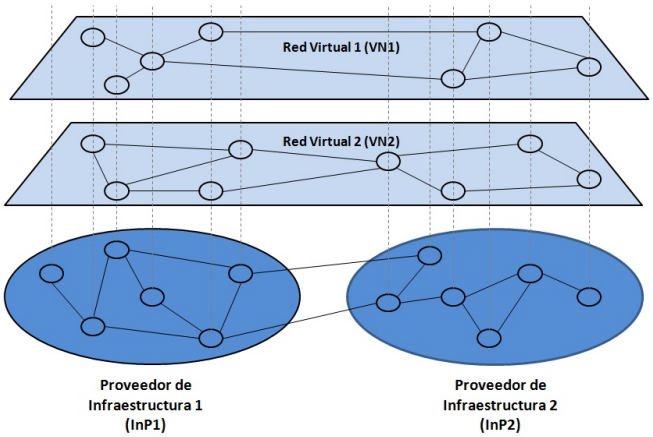


Fig. 1. Entorno de Virtualización de Red.

En el caso de la solución de virtualización de red usada en este documento y a diferencia de la Internet basada en IP actual, un entorno de virtualización de red es una colección de múltiples arquitecturas de red heterogéneas de distintos Proveedores de Internet. La mayor distinción entre los participantes en el modelo de virtualización de red con respecto al modelo tradicional donde sólo existen los Proveedores de Servicio de Internet es la existencia de dos tipos de roles distintos: los Proveedores de Infraestructura (InP) y los Proveedores de Servicio (SP) (Fig. 1).

El Proveedor de Infraestructura despliega y gestiona los recursos en la red física, ofreciéndolos a través de interfaces programables a los Proveedores de Servicio. La principal diferencia entre distintos Proveedores de Infraestructura reside en la calidad de los recursos proveídos, la libertad delegada a sus clientes y las herramientas ofrecidas para explotar esa libertad.

El Proveedor de Servicio toma prestados recursos de distintos InPs para crear y desplegar Redes Virtuales.

Los usuarios finales en el modelo de virtualización tienen una mayor elección debida a la existencia de varias Redes Virtuales de distintos Proveedores de Servicio. En este sentido los usuarios se pueden conectar a múltiples Redes Virtuales de diferentes Proveedores de Servicio según los recursos específicos demandados por cada servicio, tales como el ancho de banda, el retardo, el jitter o el consumo de energía entre otros.

Esta ponencia presenta una arquitectura que engloba los mecanismos y módulos necesarios para poder gestionar los recursos utilizados por las Redes Virtuales en un entorno de virtualización de red, y paralelamente proporcionar calidad de servicio e ingeniería de tráfico de forma independiente en cada una de dichas Redes Virtuales.

II. MAPEO DE REDES VIRTUALES

En un entorno de virtualización de red, la Red Virtual es la entidad básica. La Red Virtual se puede definir como el conjunto de Nodos Virtuales conectados a través de Enlaces Virtuales para formar una topología virtual la cual es un subconjunto de la topología física.

Los elementos virtuales (nodos, enlaces, dispositivos en general) son particiones del elemento físico real del cual proceden, manteniendo sus propiedades fundamentales y apareciendo como único y de uso exclusivo para sus usuarios.

El Nodo Virtual es un nodo dentro de la Red Virtual que usa parte de los recursos de un nodo físico. Diferentes Nodos Virtuales pueden pertenecer al mismo nodo físico.

ISBN: 978-84-694-5948-5

El enlace virtual es un enlace dentro de de la Red Virtual que usa parte de los recursos de un enlace físico. Diferentes Enlaces Virtuales pueden existir al mismo tiempo a lo largo del mismo enlace físico.

Cada Red Virtual es operada y administrada por un solo Proveedor de Servicio, y éste es libre de implementar servicios extremo a extremo utilizando su propia solución, incluyendo formatos de paquetes, protocolos de enrutamiento, mecanismos de envío, planos de control y la administración.

La necesidad de mapear eficientemente los recursos que demandan las Redes Virtuales a los recursos de la red física es un tema actual de especial interés e investigación debido a su gran número de aplicaciones. En [3] se estudian en profundidad los dos enfoques con los que el algoritmo responsable de asignar los recursos virtuales en la red física puede operar: estático y dinámico.

Los algoritmos estáticos asignan recursos según la disponibilidad de los mismos. Por el contrario los algoritmos dinámicos optimizan la asignación de recursos para que se adapten a los cambios de la red física, dado que a lo largo del tiempo se crean y eliminan Redes Virtuales.

En ambos casos, los recursos mapeados de la Red Virtual a la red física se mantienen constantes hasta que dichos recursos son modificados si se puede o son eliminados por el usuario final.

Partiendo del hecho que la arquitectura que se presenta en la siguiente sección debería ser lo más flexible posible, es deseable que soporte ambos algoritmos.

III. PROPUESTA DE LA ARQUITECTURA

La arquitectura presentada a continuación está basada en la implementación del prototipo E3MS [2] por lo que sigue un enfoque central donde la inteligencia del plano de gestión está concentrada en una única entidad que tiene conocimiento global de la red, y al mismo tiempo un enfoque compuesto por cuatro capas las cuales son Interfaz de Usuario (UI), Gestión de Red (NM), Elemento de Red (NE) y Red Física (PN) que permiten proveer la flexibilidad y modularidad deseada a la arquitectura (Fig. 2).

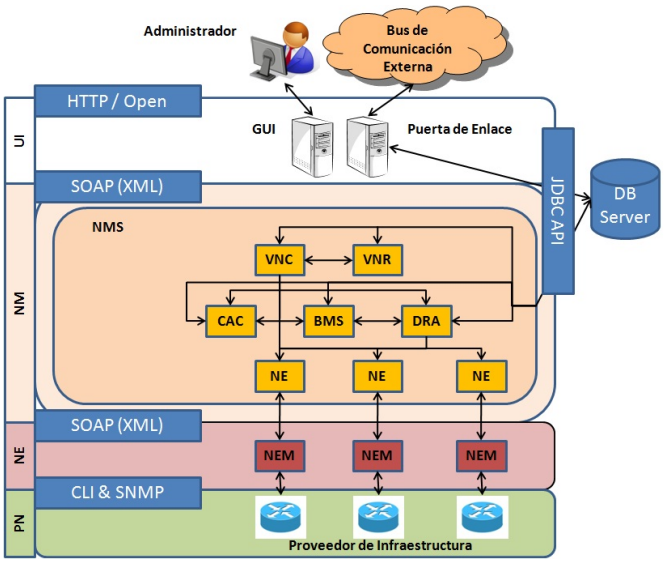


Fig. 2. Propuesta de la Arquitectura Siguiendo un Enfoque de Capas.

La capa UI está compuesta por una Interfaz Gráfica de Usuario (GUI) o entidades de puerta de enlace. GUI permite al usuario final configurar los distintos parámetros al crear una Red Virtual como los recursos a mapear a la red física y otros referentes a las diferentes clases de tráfico dentro de la Red Virtual que serán comentados más adelante. Cuando las peticiones provienen de un bus externo una entidad de puerta de enlace es usada para traducir los parámetros recibidos a operaciones concretas dentro de la arquitectura.

La capa NM está compuesta por el Sistema de Gestión de Red (NMS) el cual concentra la mayor parte de inteligencia de la arquitectura y actúa como el controlador de todo el sistema. Su principal función es gestionar las peticiones del usuario final y manejar los mecanismos necesarios para proveer una gestión autónoma de la calidad de servicio de las diferentes clases de tráfico. El NMS usa una imagen abstracta de la red donde se mapean los Nodos Virtuales, Enlaces Virtuales y recursos usados en la Red Virtual en la base de datos la cual es actualizada periódicamente por los Gestores de Elemento de Red (NEM) dentro de la capa NE.

Siguiendo el esquema planteado, por cada Red Virtual distinta existiría una instancia independiente del Sistema de Gestión de Red con sus respectivos módulos. Estos módulos son básicamente los siguientes: *Virtual Network Control* (VNC), *Virtual Network Reconfiguration* (VNR), *Call Admission Control* (CAC), *Background Monitoring System* (BMS) y *Dynamic Resource Allocation* (DRA). Todos ellos están especificados en la sección IV.

La capa NE está compuesta de uno o varios NEMs los cuales tienen como función configurar y obtener información de los nodos físicos con los cuales están asociados a través de CLI (*Command Line Interface*) o SNMP *(Simple Network Management Protocol*). Las principales funciones de la capa NE son: coleccionar y escribir datos en los nodos físicos, informar al NMS de eventos excepcionales, y monitorizar los recursos usados por los nodos físicos.

La comunicación entre las diferentes capas se puede implementar usando Servicios Web y siguiendo la *Arquitectura Orientada a Servicio* (SOA). Esta implementación hace que la arquitectura sea lo suficientemente abierta como para no depender de ninguna plataforma de proveedor ni tecnología en particular.

En la capa GUI el usuario final puede configurar las diferentes clases de tráfico dentro de la Red Virtual, así como sus prioridades. En la arquitectura que se presenta en este documento, se pretende que todas las clases de tráfico puedan hacer uso de los recursos sobrantes de las demás siguiendo un modelo de Restricción de Ancho de Banda (*Bandwidth Model Constraint*, BMC) que tenga en cuenta las prioridades de cada clase de tráfico. En este modelo el usuario final define cuales son las restricciones en términos del tanto por ciento mínimo y máximo respecto a la capacidad del enlace virtual que cada clase de tráfico puede hacer uso según su prioridad.

En la arquitectura propuesta se puede soportar cualquier tipo de BMC definido por el usuario final. A la vez y a través de la capa GUI se podrían por lo tanto configurar parámetros como clases de tráfico, prioridad de clase de tráfico, calidad de servicio de clase de tráfico, modelo de restricción de ancho de banda, listas de acceso, políticas de acceso y tipo de enrutamiento.

IV. MÓDULOS DEL SISTEMA DE GESTIÓN DE RED

Los módulos que a continuación se especifican para el sistema de gestión de las redes son: VNC, VNR, CAC, BMS y DRA.

El VNC es un módulo que se ejecuta cuando el usuario final quiere crear una nueva Red Virtual, su función principal es teniendo en cuenta la topología de la red física y los recursos disponibles no reservados por las Redes Virtuales existentes, determinar si existen recursos suficientes para asignarlos a la nueva Red Virtual. Tanto los datos de la topología de la red física como los recursos disponibles estarían ubicados en la base de datos actualizados de forma dinámica. Cuando se crea una Red Virtual, se le asigna un identificador, se activan los NEMs correspondientes, y tanto los NEMs asociados a los Nodos Virtuales como los recursos asignados a los Enlaces Virtuales son mapeados en la base de datos. Una vez creada la Red Virtual el módulo VNC intenta acceder a cada NEM perteneciente a la Red Virtual para definirle los parámetros correspondientes de la configuración inicial introducida por el usuario final. Cada NEM a través del protocolo CLI configura el nodo físico.

El módulo VNR se ejecuta sólo si el algoritmo usado para asignar los recursos demandados por las Redes Virtuales sobre la red física es dinámico. En caso de que el VNR se ejecute, éste reconfiguraría las Redes Virtuales previamente marcadas como estresadas. Una vez finalizada la reconfiguración, el módulo VNC se ejecutaría de nuevo para analizar si con el nuevo escenario, mejor balanceado, se podrían asignar los recursos demandados por la nueva Red Virtual.

El módulo CAC es el encargado de buscar la mejor ruta disponible según los requerimientos de calidad de servicio de cada clase de tráfico y su prioridad siempre y cuando cumpliendo el modelo de restricción de ancho de banda. El enrutamiento puede ser explícito por el usuario final o usando un protocolo de enrutamiento con las debidas extensiones para optimizar la ruta según los requerimientos de la calidad de servicio como podrían ser ancho de banda, retardo, jitter y consumo energético entre otros.

El módulo BMS tiene la función de monitorizar la información recibida por los NEMs de la Red Virtual y aplicar los cambios de configuración de los parámetros de calidad de servicio en los nodos físicos a través del módulo DRA. El BMS también es el encargado de supervisar que en cada Enlace Virtual se cumple el modelo de restricción de ancho de banda definido por el usuario final. En caso de comprobar que se genera un excedente de tráfico (entre una situación estable antigua y situación estable nueva) de las clases menos prioritarias, el módulo BMS pide al módulo CAC rutas alternativas para dichos tráficos. Todos los cambios que ejecuta el BMS a través del DRA no son inmediatos, para dar validez y minimizar los constantes cambios de configuración en los nodos físicos se usa un tiempo de histéresis después del cual si el cambio persiste se considera una situación nueva estable.

El módulo DRA tiene la función de recibir los parámetros de configuración de los módulos CAC y BMS y aplicarlos a través de los NEMs correspondientes asociados a los nodos de la red física. Para ello y debido que cada Red Virtual usa una instancia independiente del Sistema de Gestión de Red (NMS), se necesita garantizar que una única instancia haga los cambios de configuración necesarios de forma excluyente.

En la Fig. 3. se muestra el diagrama de flujo cuando se crea una Red Virtual. En primer lugar se detecta si el algoritmo usado para asignar los recursos demandados por las Redes Virtuales es estático o dinámico. Si es estático a través del módulo VNC se detecta si hay suficientes recursos en la red física para crear una nueva Red Virtual, si la respuesta es afirmativa se procede con la reserva de recursos y configuración de los parámetros introducidos por el usuario final, y si la respuesta es negativa se vuelve al inicio. Si se usa un algoritmo dinámico igual se ejecuta el módulo VNC pero en caso que no haya recursos disponibles se ejecuta el módulo VNR para reconfigurar las Redes Virtuales previamente marcadas como estresadas. Si una vez balanceadas las Redes Virtuales hay recursos disponibles suficientes se procede con la reserva de recursos y configuración de los parámetros introducidos por el usuario final.

En la Fig. 4. se muestra el diagrama de flujo cuando se modifica una Red Virtual en el sentido que se modifican los parámetros de configuración iniciales o simplemente se aplican cambios monitorizados por el módulo BMS. En este caso si el usuario final decide crear una nueva clase de tráfico con su prioridad y calidad de servicio después de haber creado satisfactoriamente la Red Virtual, el módulo CAC buscaría una ruta satisfactoria para el nuevo tráfico. Si la encuentra consultaría con el módulo BMS si satisface los requisitos del modelo de restricción de ancho de banda, si la respuesta es negativa el módulo CAC debería buscar una ruta alternativa y en caso de no encontrarla se descartaría el nuevo tráfico. En caso de encontrar una ruta ya sea en el primer intento o alternativa, el módulo CAC pasaría los parámetros de calidad de servicio al módulo DRA para que éste pudiera configurarlos a través de los NEMs en cada nodo físico por donde pasaría el Enlace Virtual. Si la nueva clase de tráfico es de prioridad elevada puede darse la situación que debido al modelo de restricción de ancho de banda usado se genere un tráfico excedente de otro tráfico existente pero de menor prioridad. En tal situación el módulo CAC debería buscar una ruta alternativa para el tráfico excedente que cumpliera de nuevo el modelo de restricción de ancho de banda y así de forma iterativa hasta que todo el tráfico excedente se pudiera reubicar manteniendo los parámetros de calidad de servicio o similares, o hasta que el tráfico excedente se descarte.

En la otra situación planteada donde el módulo BMS a través de la monitorización recibe alertas de los NEMs indicando que una clase de tráfico ha modificado los recursos usados estableciendo una nueva situación estable. En la situación que haya modificado el uso de ancho de banda haciendo un subutilización simplemente se pasan los nuevos parámetros a configurar al módulo DRA. En el caso que haya habido una sobreutilización el módulo BMS tiene que comprobar si se cumple el modelo de restricción de ancho de banda en la nueva situación, si la respuesta es afirmativa se mandan los nuevos parámetros a configurar al módulo DRA y hay que comprobar si durante el proceso se generó algún excedente de tráfico de menor prioridad. Si la respuesta es negativa se finaliza la modificación, pero si es positiva el módulo CAC debería buscar una ruta alternativa para el tráfico excedente que cumpliera de nuevo el modelo de restricción de ancho de banda de forma iterativa de nuevo.

V. ELEMENTOS DE EVALUACIÓN

Para una futura evaluación de la arquitectura, deben considerarse los elementos asociados al número de nodos y enlaces activos así como los niveles de estrés a los que están sometidos, incluyendo los siguientes:

- Número de nodos y enlaces físicos dentro del Proveedor de Infraestructura.
- Recursos originales (máximos) y disponibles de cada nodo físico dentro del Proveedor de Infraestructura (Recursos son el consumo de CPU, energético y de memoria).
- Recursos originales (máximos) y disponibles de cada enlace físico dentro del Proveedor de Infraestructura (Aquí recursos son el ancho de banda, el retardo y el jitter).
- Número de nodos y enlaces virtuales sobre cada nodo físico.
- Número de nodos y enlaces físicos frontera entre distintos Proveedores de Infraestructura.
- Parámetro de estrés de cada nodo y enlace físico dentro del Proveedor de Infraestructura (determina el número de instancias activas en relación a los recursos disponibles y el número de nodos virtuales que puede soportar).

Esta evaluación forma parte de los trabajos que actualmente se están desarrollando.

VI. CONCLUSIONES

En esta ponencia se ha presentado una arquitectura flexible y modular para la gestión de redes virtuales, que permite gestionar la calidad de servicio e ingeniería de tráfico entre diferentes clases de tráfico en cada red virtual. Para ello se ha introducido una arquitectura y se han definido los diversos elementos necesarios y su funcionalidad. Actualmente se están llevando a cabo las tareas de evaluación de los diversos elementos como del sistema global para estudiar su estabilidad y escalabilidad.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el MICINN del Gobierno Español, en el proyecto TIN2010-20136-C03.

REFERENCIAS

[1] N.M.M.K. Chowdhury and R.H. Boutaba, "Network Virtualization: State of the Art and Research Challenges," IEEE Communications Magazine, vol. 47 no. 7, 2009, pp. 20-26.

[2] X. Hesselbach, J.A. García-Espín, M. González, J. Gonzalo and S. Figuerola, "E3MS: A traffic engineering prototype for autoprovisioning services in IP/Diffserv/MPLS networks," JITEL 2010.

[3] J.F. Botero and X. Hesselbach, "Study, Evaluation and Contributions to New Algorithms for the Embedding Problem in a Network Virtualization Environment," Ph.D Thesis, Technical University of Catalunya, 2010.

[4] J.F. Botero and X. Hesselbach, "The Bottlenecked Virtual Network Problem in Bandwidth Allocation for Network Virtualization," IEEE Latin-American Conference on Communications, 2009.

[5] J.A. García-Espín and X. Hesselbach, "Squat-based Resource Management Strategy for Enabling Shared Infrastructures over Optical Networks," ICTON 2010.

[6] Y. Zhu and M. Ammar, "Algorithms for Assigning Substrate Network Resources to Virtual Network Components," In Proc. IEEE INFOCOM, 2006, pp. 2812-2823.

[7] E. Rosen, A. Viswanathan and R. Callon, "Multiprotocol Label Switching Architecture," RFC 3031, Enero 2001.

[8] S. Blake et al., "An Architecture for Differentiated Services," RFC 2475, Diciembre 1998.

[9] D. Awduche et al., "Requirements for Traffic Engineering Over MPLS," RFC 2702, Septiembre 1999.

[10] F. Le Fauncher et al., "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services," RFC 3270, Mayo 2002.

[11] F. Le Fauncher and W. Lai, "Requirements for Support of Differentiated Services-aware MPLS Traffic Engineering," RFC 3564, Julio 2003.

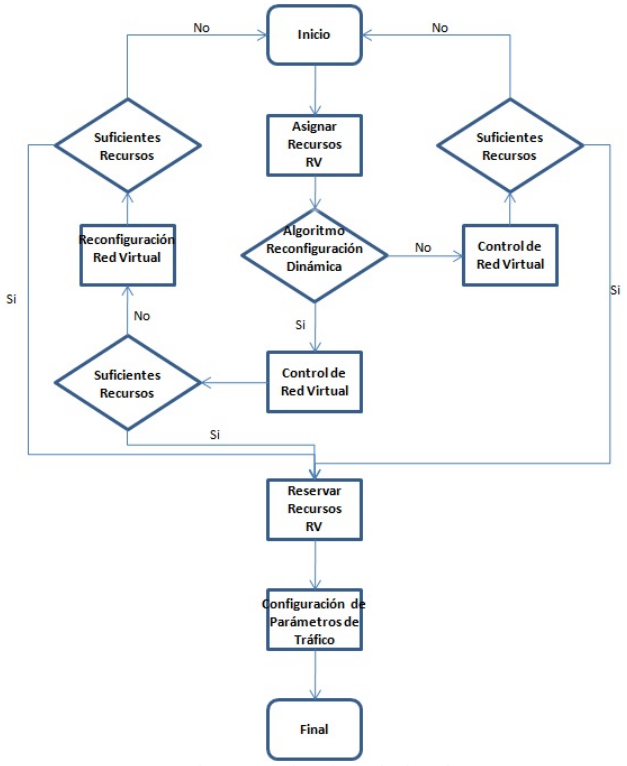


Fig.3. Creación de Red Virtual

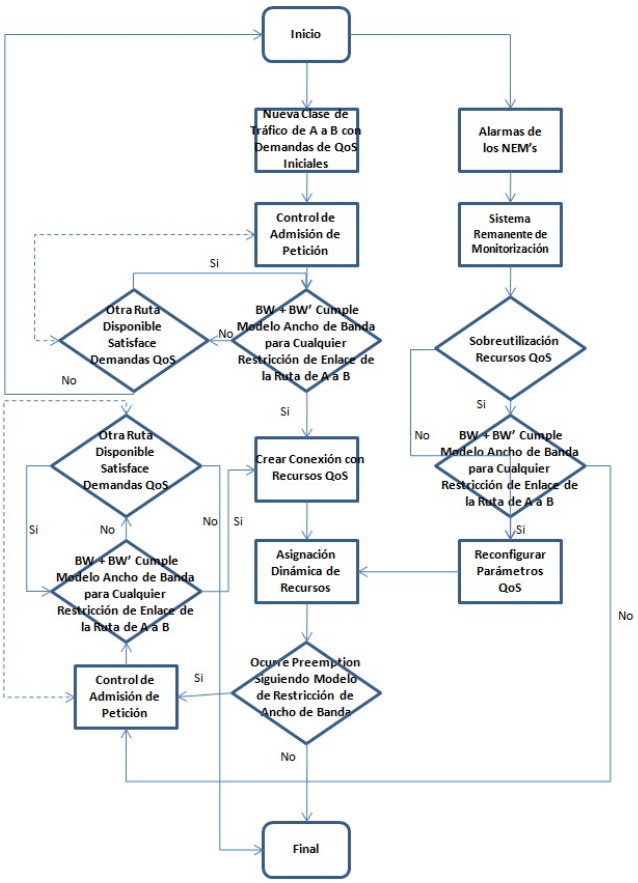


Fig. 4. Modificación de Red Virtual

Kraken, un prototipo de sistema de streaming P2P en directo basado en codificación SVC

Mateo Matachana López, David Melendi Palacio, Xabiel García Pañeda,
Sergio Cabrero Barros, Roberto García Fernández
Departamento de Informática,
Universidad de Oviedo
Escuela Politécnica de Ingeniería, 2ª planta, módulo 7
Campus Universitario de Viesques
33024 Gijón
matachana.mateo@gmail.com, melendi@uniovi.es, xabiel@uniovi.es,
cabrerosergio@uniovi.es, garciaroberto@uniovi.es

***Resumen*—La popularización y mejora tecnológica de las conexiones de banda ancha a Internet, han permitido el auge del consumo de multimedia a través de la red. Además, la gran diversidad de dispositivos, la heterogeneidad de velocidades de acceso y la ausencia de garantías sobre la calidad de transmisión, suponen un desafío que se complica aún más cuando los contenidos se distribuyen en la modalidad en directo. En este sentido, la codificación SVC (*Scalable Video Coding*) puede servir para maximizar la experiencia de los usuarios, mientras que la distribución P2P (Peer to Peer) puede ayudar a minimizar la complejidad técnica y los costes del despliegue que es necesario realizar a la hora de dar servicio a un gran número de clientes. En este artículo se presenta Kraken, un protocolo que nace con el propósito de facilitar la distribución de contenidos multimedia en directo, incorporando múltiples innovaciones sobre sus predecesores.**

***Palabras Clave*—p2p, multimedia, vídeo, streaming, SVC**

I. Introducción

La popularización de las conexiones de banda ancha a Internet y el incremento de velocidad de las mismas, han permitido el auge en los últimos años del consumo de multimedia a través de la red. Además, la gran diversidad de dispositivos con capacidad de acceso a la red, la heterogeneidad de velocidades de acceso y capacidades de cálculo de los terminales usados para la recepción, han planteado nuevos desafíos que implican la puesta en marcha de mecanismos para adaptar la emisión a cada receptor, de manera que el flujo de audio-vídeo recibido no exceda los límites computacionales disponibles en el dispositivo usado para recibir la transmisión.

Este escenario plantea un tremendo desafío para las emisiones en directo, donde un importante retraso en la recepción puede hacer perder gran valor a la información recibida. Además, dar servicio a un número cada vez mayor de espectadores supone un gran desafío técnico y la necesidad de enfrentar grandes costes a consecuencia de los despliegues de red necesarios para soportar el tráfico que puede generar la potencial audiencia.

Es por ello por lo que la transmisión de vídeo en directo haciendo uso de un enfoque P2P puede aportar grandes ventajas, permitiendo minimizar la complejidad técnica y los costes del despliegue de red que es necesario realizar a la hora de dar servicio a un gran número de clientes.

Asimismo, han de ser tenidas en cuenta las posibilidades de personalización del flujo multimedia que ofrece la codificación de vídeo multicapa. El hecho de emitir un único flujo y que este pueda adaptarse a las necesidades del receptor, sin duda simplifica en gran medida la transmisión del *stream* y la infraestructura requerida para realizar la codificación del flujo de vídeo.

Las soluciones existentes estudiadas presentan varios inconvenientes y problemas. La mayoría de estas, como por ejemplo Octoshape [1] y Sopcast [2], no soportan codificación de vídeo multicapa y en aquellas que sí lo hacen, como por ejemplo en la descrita por Y. Cui [3], el descartado de capas se hace a posteriori. Usando este planteamiento, la codificación de capa no serviría para adaptarse a la velocidad de la conexión a la red. Muchas de las soluciones estudiadas no soportan ningún mecanismo para personalizar el protocolo, por lo que este queda inmutable a lo largo del tiempo, no pudiendo ser ni extendido ni adaptado para incorporar necesidades futuras. Por otra parte, apenas existen soluciones planteadas de un modo atractivo para la industria, y es que casi ninguna incorpora métodos de gestión de la propiedad intelectual, integración con despliegues ya existentes, etc. Por ello se hace muy difícil plantear servicios de suscripción encima de soluciones previas.

El resto del artículo está estructurado de la siguiente manera: la sección II está dedicada a una pequeña introducción a la codificación de vídeo SVC. En la sección III se entrará a comentar en detalle el protocolo Kraken, mientras que en la sección IV se dedica a exponer los resultados experimentales obtenidos a partir de la implementación realizada del protocolo. Por último, en la sección V se esbozarán las conclusiones y posibles líneas de futuro de este trabajo.

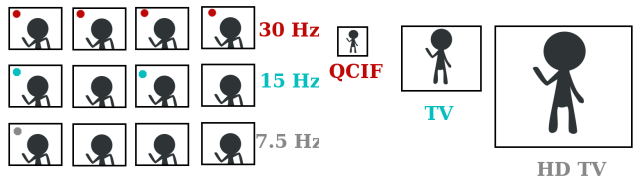
II. La codificación de vídeo H.264/SVC

Un *stream* de vídeo se considera escalable cuando partes de dicho *stream* pueden ser eliminadas de manera que el substream resultante forme otro *stream* válido para el decodificador y represente al *bitstream* original con una calidad menor.

Podemos diferenciar tres tipos básicos de escalabilidad, tal y como se ve en la figura 1.

La **escalabilidad temporal** consiste en que las capas de mejora provocan un aumento en la tasa de fotogramas por

ISBN: 978-84-694-5948-5



(a) Escalabilidad temporal

(b) Escalabilidad espacial

(c) Escalabilidad de calidad

Fig. 1: Diferentes tipos de escalabilidad

segundo que se pueden visualizar. La **escalabilidad espacial**, en cambio, implica poder aumentar el tamaño de la imagen decodificada con cada capa de mejora mientras que en la **escalabilidad de calidad** con cada capa se incrementa la calidad global de la imagen.

Aunque desde 1994 sucesivos estándares de vídeo han venido incorporando algún perfil de escalabilidad [4], la extensión SVC al estándar H.264 [5] supone un soplo de aire fresco en este área.

H.264/SVC nace con el objetivo de buscar un impacto mínimo sobre la complejidad de los códecs y sobre el *bit-stream* resultante. Por ello incorpora técnicas que reducen la pérdida de eficiencia de codificación así como posibilitan la transcodificación[1] desde y hacia H.264/AVC de manera relativamente sencilla ([6], [7]).

Según M.Schwarz y M. Wien [8], los estudios realizados muestran que H.264/SVC conlleva, aproximadamente, un incremento del 10% en la tasa de bits comparado con codificación mono-capa. Respecto al consumo de recursos computacionales, el incremento se cifra entre un 10 y un 50 %.

De cara a soportar dispositivos de escaso poder computacional – *smartphones*, *tablet pc's*, etc –, que requieren de un *chipset* para poder decodificar vídeo, H.264/SVC es especialmente interesante. Aunque la decodificación de vídeo multicapa exceda las características computacionales de dichos dispositivos, la capa base de esta codificación es un flujo H.264/AVC que puede ser descodificado sin problemas por dicho *chipset*

Otras ventajas que nos ofrece la codificación de vídeo multicapa son que el receptor puede adaptar el *stream* a su tamaño de pantalla, velocidad de red, etc. tan solo rechazando determinadas capas de vídeo. También nos permite amortiguar errores introducidos durante la transmisión de los datos.

III. El protocolo Kraken

Kraken es un protocolo de red para la transmisión de vídeo en directo que nace con unos objetivos bien definidos que se detallarán a continuación.

Uno de los objetivos más importantes de Kraken es añadir soporte para las últimas técnicas de codificación de vídeo escalable. Aunque se haya usado como formato de cabecera el estándar H.264/SVC, Kraken no queda limitado a ningún formato de vídeo en concreto.

A diferencia de otras soluciones previas estudiadas, las capas de vídeo a recibir deben poder elegirse antes de comenzar su recepción. De esta manera, no sólo se consigue ofrecer un vídeo a la máxima calidad que el terminal admite sino que se consigue una adaptación óptima a la velocidad de red disponible en la conexión.

Otro objetivo muy importante para Kraken es ser atractivo a la industria multimedia. Uno de los mayores recelos de esta es el poco control que tiene sobre el flujo multimedia una vez sale de sus equipos. Es por esto que Kraken incorpora una serie de mecanismos encaminados a que el emisor tenga un control efectivo sobre los receptores capaces de visualizar la emisión.

Asimismo en el diseño de Kraken se ha buscado usar como base protocolos que funcionen en situaciones de red adversas. Es por esto por lo que el protocolo se construye sobre HTTP. En la mayoría de configuraciones de red con filtrado de tráfico, como *hot spots* de aeropuertos, los únicos protocolos permitidos son DNS y HTTP.

Otra meta reseñable es la coexistencia con despliegues de *streaming* previos. Si bien las ventajas que aportan las técnicas de codificación de vídeo multicapa son muy interesantes, estas todavía no están lo suficientemente maduras como para ser puestas en producción. La escasez de códecs y los altos recursos computacionales requeridos por estos hacen que a día de hoy el uso de vídeo multicapa sea algo poco más que teórico.

Por esta razón el protocolo diseñado soporta también ser usado con las técnicas de codificación de vídeo tradicionales.

Asimismo, se desea que el protocolo sea suficientemente flexible como para adaptarse a las diferentes innovaciones que en un futuro puedan producirse.

Por ello en la especificación del protocolo se contemplan mecanismos para que en un futuro pueda ser expandido sin afectar a despliegues previos, dejando incluso la puerta abierta a que emisores puedan crear sus propias extensiones para añadir servicios de valor añadido como, por ejemplo, servicios de teletexto, información de la programación, etc.

El protocolo Kraken está ligeramente inspirado por BitTorrent. De este toma la manera de organizar la red, la filosofía de distribución del contenido a través del troceado para una mayor velocidad de transmisión y de que las respuestas del tracker, en formato SDP (*Session Description Protocol*), pueden ser usadas a modo de fichero Torrent, es decir, contienen suficientes metadatos como para poder localizar el *stream*. Inspirado en su diseño por BitTorrent, podemos distinguir en la red tres tipos de roles bien diferenciados, tal y como se puede ver en la figura 2:

- El **tracker** es el nodo encargado de gestionar el listado de todos los *peers* conectados a la red y que están compartiendo un mismo *stream*.
- El **servidor de vídeo** Si bien tradicionalmente en las redes P2P este rol es difuso, compartido por los diferentes *peers* unidos a la red, cuando hablamos de

[1] Transformar el formato de codificación de vídeo sin necesidad de pasar por un formato decodificado

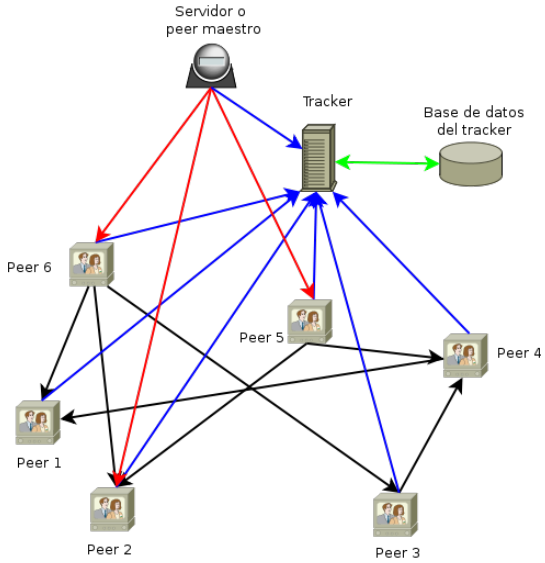


Fig. 2: Esquema de los componentes que conforman la red Kraken

streaming en directo podemos diferenciar un grupo de *peers* que actúan exclusivamente como servidor, a diferencia del resto, que actúan como cliente y servidor simultáneamente.
Los servidores de vídeo están en contacto directo con los elementos generadores de la imagen y el sonido y son los que inyectan la retransmisión a la red.

- Los **peers** son todos aquellos participantes que están recibiendo la transmisión. A diferencia del modelo tradicional cliente-servidor, donde estos nodos son netamente pasivos, en una arquitectura P2P colaboran activamente difundiendo la transmisión multimedia a otros *peers* también interesados en la misma.

A. *Un vistazo a la especificación del protocolo*

A la hora de establecer una nueva sesión con el protocolo Kraken podemos distinguir dos fases bien diferenciadas:

- **El anuncio al tracker** En esta fase, cada *peer* a partir de una URL se conecta al *tracker* para obtener una descripción del flujo multimedia así como un listado con los *peers* conectados al mismo y disponibles para empezar a descargar de ellos.
- **La descarga del *stream*** Una vez obtenida la lista de diferentes *peers* a los que conectarse, puede procederse a la descarga y visualización del flujo multimedia.

Para maximizar la velocidad de transmisión, el *stream* se divide en diferentes pedazos que permiten descargar el mismo flujo multimedia desde diferentes *peers*. Según S. Alstrup y T. Rauhe [9], el trocear un *stream* permite que, sin saturar los canales de subida de los *peers* a los que se está conectado, se maximice el uso del canal de bajada propio. En cambio, si solo se descarga de un único *peer* sin ningún tipo de troceo del *stream*, la velocidad de descarga quedará limitada a la velocidad de subida del *peer* al que se esté conectado.

Cada trozo en la red Kraken tiene un tamaño de 1024 bytes y podemos distinguir dos tipos de trozos: los *trozos base*[2] y los *trozos de redundancia*. Los primeros se obtienen extrayendo 1 KiB de los datos de vídeo listos para ser transmitidos, mientras que los segundos se obtienen como una combinación lineal de los *trozos base*. De esta manera, a partir de la recepción de n trozos diferentes, sin importar que sean base o de redundancia, se puede obtener el flujo de vídeo original.

El troceado de flujo se hace a nivel de capa, es decir, cada una de las capas de vídeo y canales de audio que componen la transmisión son troceados y transmitidos de manera independiente.

Por tanto un *stream* emitido con el protocolo Kraken queda determinado por tres parámetros básicos: el número n de trozos que se han de recibir para poder decodificar el flujo, el número i de trozos base y el número j de trozos de redundancia, de tal manera que $n = i$.

Toda esta información, además de detalles acerca de la codificación usada por el flujo multimedia e información de otros *peers* conectados, es devuelta en el anuncio que cada *peer* realiza al *tracker*.

En cada anuncio el *peer* envía al *tracker* información acerca de todas las capas y trozos que está recibiendo y, por tanto, en condición de retransmitir a otros *peers*. La información devuelta por el *tracker*, en formato SDP, es la siguiente:

- La versión del protocolo, el enlace del *tracker* y la descripción de la transmisión.
- En el caso de que se decidan aplicar técnicas de DRM[3] al flujo el *tracker* puede distribuir la información necesaria para poder decodificar el flujo. Para proteger esta información de usuarios no autorizados, se puede exigir autenticación HTTP a la hora de realizar la conexión con el *tracker*, así como el uso de HTTPS.
- El número de capas usado, la codificación de vídeo del flujo multimedia y la descripción y troceado de cada una de las capas de vídeo y audio.
- Los *peers* conectados y en situación de retransmitir cada una de las capas de audio y vídeo.

Este anuncio se ha de realizar de manera periódica.

Por otra parte, los *peers* se comunican entre ellos haciendo uso del protocolo HTTP. A través de este protocolo se descargan de cada *peer* las unidades mínimas de información o "paquetes Kraken". Cada trozo de cada capa se gestiona en una conexión HTTP diferente. Idealmente, esta conexión se renovará cada 100 paquetes transmitidos para amortiguar o evitar microcortes de red u otros problemas que aparecen con el establecimiento de conexiones de larga duración.

Todo paquete kraken lleva asociado un identificador dentro del rango $(0, \ldots, 255)$. En base a este identificador podemos diferenciar dos tipos de paquetes: los *paquetes de cabecera* (identificados por el valor 0) y los *paquetes de datos* (identificados por el valor 1). Los primeros tienen como objetivo intercambiar información fuera de banda entre *peers*. Por ejemplo, hay un paquete de cabecera definido para evitar el restablecimiento periódico de la conexión.

Los paquetes de datos contienen los datos multimedia transmitidos. Este tipo de paquete lleva asociado un número de secuencia que permite emparejar diferentes *streams* de trozos de una misma capa, además de un trozo de capa completo (1KiB de datos). Por otra parte, los identificadores

[2]No confundir con la capa base de codificación de vídeo multicapa

[3]*Digital Rights Management* o gestión digital de derechos

de paquetes en el rango $(2, \ldots, 32)$ quedan reservados para expansiones futuras del protocolo, mientras que los incluidos en el rango $(33, \ldots, 255)$ pueden ser usados libremente para hacer desarrollos propios a partir de Kraken.

IV. Resultados experimentales

Para comprobar cómo se comportaría Kraken en un despliegue de red a nivel práctico, se desarrolló una implementación de referencia. Esta implementación fue sometida a una batería de pruebas con el objetivo de ver la evolución de la presión sobre el emisor de vídeo a medida que el número de clientes se incrementa.

La batería de pruebas se ejecutó sobre la interfaz local de la una máquina. Se ejecutaron un total de 10 pruebas diferentes, variando en cada una de ellas el número de peers añadidos a la sesión. Los peers se introdujeron a la misma con un intervalo de 30 segundos y los anuncios al tracker se realizaban cada 25 segundos. Se capturo todo el tráfico transmitido en las diferentes sesiones, midiendo qué porcentaje del total era contribuido por cada uno de los *peers*. Los resultados se pueden ver en la figura 3.

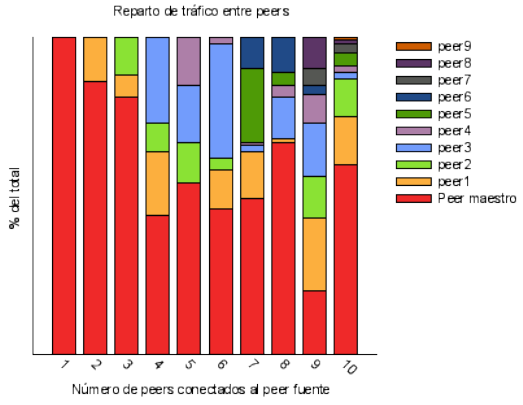


Fig. 3: Contribución de cada *peer* al tráfico total transmitido por la red

Una de las cosas que más llama la atención de dicha figura es la irregularidad de las contribuciones de tráfico para diferentes números de *peers*. Esto es debido a que el algoritmo de distribución de tráfico elegido para la implementación se basa en la selección aleatoria de los *peers* a los que conectarse y, por ello, en ocasiones el tráfico no se distribuye de manera igualitaria. Además, el hecho de que se seleccionen los *peers* de manera independiente para cada trozo de capa contribuye a reforzar las desigualdades.

Aún así, a pesar de las deficiencias del algoritmo, podemos comprobar que la presión de tráfico ejercida sobre la infraestructura de red del emisor se ve tremendamente reducida, en la mayoría de los casos por debajo del 50% de la presión que recibiría haciendo uso del modelo cliente-servidor tradicional.

Parte de la culpa de esta concentración de tráfico también la tiene la implementación del *tracker*. Con el algoritmo actual, una manera de mitigar los efectos producidos por elegir los *peers* sin ningún otro criterio que de la aleatoriedad es devolviendo una lista diferente de *peers* disponibles a cada *peer* que realice un anuncio.

Durante toda la prueba, los *peers* no cambiaron sus fuentes. De esta manera, si inicialmente se comienza a descargar de los *peers* 1 y 3, durante toda la prueba solo se descargará de esos *peers*. Como podemos ver en dicho gráfico, los primeros *peers* concentran mayoritariamente el tráfico en el *peer* maestro, efecto que se arrastra en el porcentaje global.

V. Conclusiones y líneas de futuro

En este artículo se presenta a Kraken, un protocolo que se perfila idóneo para hacer frente a todos los retos que la industria plantea de manera fiable, segura y manteniendo mínima la complejidad requerida para realizar la emisión. Además, debido a la versatilidad ofrecida a la hora de escoger la configuración de la codificación de vídeo, es posible integrar con relativa facilidad Kraken con soluciones ya desplegadas.

Sin embargo, hay varias líneas de futuro en las que Kraken puede ser mejorado. Como ya se ha mencionado, el protocolo debe de incorporar un algoritmo de distribución de tráfico mejorado, que dote de mayor inteligencia a la red y evite concentrar el tráfico en determinados *peers*.

Por otra parte, cabe estudiar la incorporación de metadatos dentro del protocolo para ofrecer servicios de valor añadido como pueden ser teletexto, comentarios de los espectadores, etc.

También es interesante explorar la posibilidad de usar Kraken en modo mixto, es decir, con algunos de los *peers* funcionando en modo cliente-servidor y otros en P2P. Esto permitiría, por ejemplo, la existencia de *peers* de capacidades reducidas implementados en lenguajes como Flash y que se ejecuten dentro del navegador con las elevadas políticas de seguridad que estos imponen y otros *peers* que hagan uso de todas las características de Kraken y contribuyan con su canal de subida a la red.

En definitiva, Kraken busca aunar las tendencias actuales en difusión de vídeo en un protocolo sencillo, que no requiera de demasiados recursos y que pueda ser usado en situaciones de red desfavorables o en equipos con poca potencia de cálculo sin por ello renunciar a una gran versatilidad y funcionalidad.

Referencias

[1] "Octoshape." [Online]. Available: http://www.octoshape.com
[2] "Sopcast." [Online]. Available: http://www.sopcast.org/
[3] Y. Cui and K. Nahrstedt, "Layered peer-to-peer streaming," in *Proceedings of the 13th international workshop on Network and operating systems support for digital audio and video*. ACM, 2003, pp. 162–171.
[4] J. Ohm and M. van der Schaar, "Scalable Video Coding," in *Tutorial material, Int. Conf. Image Processing ICIP*, vol. 2007, 2007.
[5] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H. 264/AVC standard," *To appear in IEEE Transactions on Circuits and Systems for Video Technology*, p. 1, 2007.
[6] A. Segall and J. Zhao, "Bit stream rewriting for svc-to-avc conversion," in *15th IEEE International Conference on Image Processing, 2008. ICIP 2008*, 2008, pp. 2776–2779.
[7] De Cock, J. and Notebaert, S., "Efficient conversion of single-layer H. 264/AVC video streams to multiple-quality-layer SVC streams," in *8th FirW PhD Symposium*, 2008.
[8] Schwarz, H. and Wien, M., "The Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Signal Processing Magazine*, 2008.
[9] S. Alstrup and T. Rauhe, "Introducing Octoshape – a new technology for streaming over the Internet," *EBU Technical Review*, vol. 303, 2005.

INTEGRIS: Seguridad en la integración de nuevas tecnologías sobre Smart Grids

David González-Tarragó, Agustín Zaballos, Guiomar Corral
Departamento de Informática, Enginyeria La Salle
Universitat Ramon Llull
c/ Quatre Camins, 2, 08022 - Barcelona
{dgonzalez, zaballos, guiomar}@salle.url.edu

***Resumen*- Las Smart Grids, o redes eléctricas inteligentes, son redes de distribución de energía que contienen elementos de control y gestión propios. Estas redes son capaces de gestionar la energía de forma eficiente a través de protocolos de actuación predeterminados, lo que permite una mejora del servicio en términos generales. El objetivo del proyecto INTEGRIS es integrar las nuevas tecnologías a la red actual, mejorando el control y aportando nuevas funcionalidades a la misma. Este proceso se lleva a cabo mediante dispositivos integradores llamados I-Dev (INTEGRIS Devices), encargados del proceso. En este artículo se introduce la problemática de seguridad que se deriva de tal integración y se detallan los principales mecanismos para hacerle frente.**

Palabras Clave: Smart Grids, Integración, Seguridad.

I. INTRODUCCIÓN

Hoy en día gestionar las redes eléctricas de forma eficiente y práctica se ha vuelto esencial en la industria. Disponer de conexiones remotas con los dispositivos de gestión de la infraestructura de distribución eléctrica (contadores, actuadores, medidores,…) es una imperiosa necesidad ya que reduce enormemente los costes de mantenimiento al mismo tiempo que aumenta el control sobre la propia red.

Por otro lado el avance de la tecnología ofrece la oportunidad de añadir nuevos servicios, lo que es algo de enorme interés para las compañías, teniendo en cuenta el mercado competitivo en el que se mueven. Sin embargo la introducción de nuevos sistemas es difícil, debido a las diferencias que hay entre las propias redes así como a su constante expansión.

El proyecto INTEGRIS [1] tiene como objetivo la integración de estas nuevas tecnologías (PLC, Wireless, RFID y Fibra óptica) en el marco de las Smart Grids actuales [2]. Para ello se vale de la infraestructura ya existente y de los sistemas que ya están en explotación. Para ello, persigue diseñar y desarrollar una infraestructura de telecomunicaciones con el mismo alcance que la red de distribución eléctrica que permita utilizar información en tiempo real. Uno de los puntos clave es desarrollar una infraestructura de bajo coste de despliegue que use como medio de transmisión el cable eléctrico ya existente, con tecnologías de telecomunicaciones típicas de las redes de sensores inalámbricos. Se trata en esencia de un proyecto de mejora de la red actual, aportando nuevas funcionalidades y nuevas aplicaciones que son necesarias para mejorar la calidad y la respuesta de los servicios (véase esquema en la Figura 1).

No obstante, esta integración debe realizarse de forma segura pues la seguridad es un punto clave. Proteger los datos confidenciales así como el acceso a los actuadores remotos no es solo una práctica necesaria sino que obedece a una serie de normativas impuestas por la propia industria. Al mezclar distintos sistemas se pueden producir nuevas brechas de seguridad, por lo que hacer una integración de forma segura es básico.

En este artículo pues, se analiza la problemática de la seguridad inherente a la integración de nuevas tecnologías en las Smart Grids actuales y se expone el trabajo realizado hasta el momento para hacerle frente. Se introduce la utilización de dispositivos integradores llamados I-Dev (INTEGRIS Devices), encargados de manejar la conectividad entre los distintos elementos de la red de comunicaciones así como las nuevas funcionalidades que se quieren aportar. Del mismo modo también se hace hincapié en los aspectos de seguridad más relevantes de la introducción de nuevas tecnologías en redes periféricas. Todo esto se lleva a cabo mediante una explicación paso a paso, siguiendo el plan de trabajo que se está llevado a cabo en la realización del proyecto.

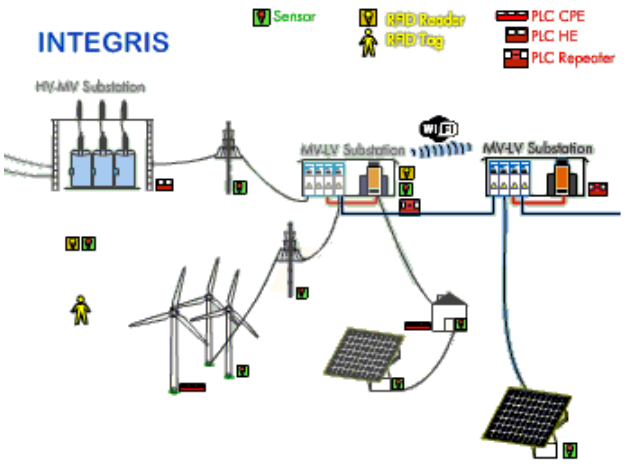


Fig. 1. Esquema del diseño del proyecto INTEGRIS

En el punto II se describen las Smart Grids actuales, en el punto III se describe el problema de la preservación de la compatibilidad, en el punto IV se hace un breve análisis sobre la seguridad existente que nos podemos encontrar hoy

en las Smart Grids, en el punto V se plantea una primera aproximación a la solución, en el punto VI se estudia la creación de túneles P2P, en el punto VII se trata la identificación de los dispositivos, en el punto VIII se enfoca la problemática de seguridad en las redes periféricas, en el punto IX se describe la integración completa, en el punto X se comenta el estado actual de desarrollo del proyecto y finalmente en el punto XI se exponen las conclusiones. Por último hay un apartado más con los agradecimientos.

II. Las Smart Grids Actuales

Las Smart Grids actuales se componen en su mayoría de sistemas basados en el estándar IEC 60870, los cuales suelen gestionar zonas delimitadas de la red eléctrica. Si bien existen estándares más extensibles que se están empezando a implantar paulatinamente (como el IEC 61850), lo cierto es que hoy en día, el control sobre la red sigue llevándose a cabo con dichos mecanismos.

A pesar de que se trata de sistemas que en su mayoría ofrecen un control bastante granular sobre la red, carecen de integración con otros sistemas que puedan aportar mayor control o funcionalidad [3]. Este sería el caso por ejemplo de la integración de redes de sensores distribuidas (WSN), o el uso de sistemas RFID para el manejo de sistemas de acceso. En esencia las Smart Grids se organizan siguiendo un modelo conceptual de referencia bajo el que se estructuran todas las tecnologías y procesos relacionados con la generación, transmisión, distribución y consumo de energía, así como los negocios relacionados con su comercialización y operación [4]. En la siguiente imagen se esquematiza tal organización.

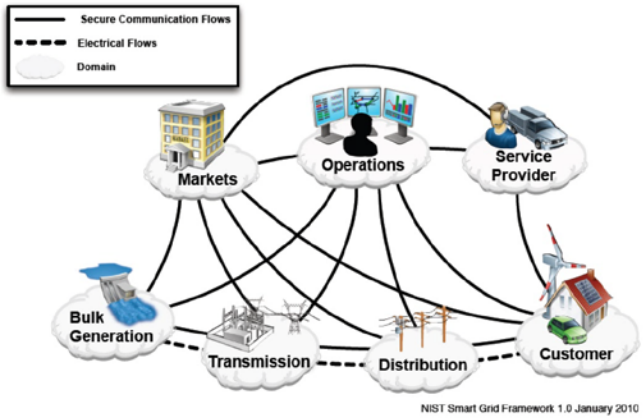


Fig. 2. Organización de una Smart Grid

En los últimos años se han estado estudiando formas de actualizar las Smart Grids de forma gradual para poder afrontar los nuevos requisitos del mercado y poder aportar nuevas aplicaciones de control. Sin embargo las Smart Grids resultan difícilmente escalables debido a su diseño de base, siendo necesario un sistema de integración externo el que lleve a cabo tal tarea.

III. El Problema de la Compatibilidad

El primer problema que nos encontramos a la hora de llevar a cabo la integración de nuevas tecnologías en un sistema antiguo, es la compatibilidad. Obviamente queremos ofrecer nuevos servicios y nuevas características pero también queremos mantener las existentes. Reconstruir toda la infraestructura desde cero, seria extremadamente costoso y muy poco eficiente ya que al cabo de unos años nos encontraríamos muy posiblemente con el mismo problema.

Se trata de actualizar la red y de mejorarla, pero conservando la operatividad con los sistemas ya en funcionamiento. Esto crea un situación un tanto delicada pues los sistemas en funcionamiento no deben verse afectados por la integración de los nuevos y viceversa. En otras palabras, debe existir una compatibilidad entre tecnologías que haga posible la convivencia.

Sin embargo esta compatibilidad puede crear un problema de seguridad inherente a este tipo de situación [5]. Las nuevas tecnologías implantadas seguramente serán capaces de asumir el funcionamiento de las antiguas, pero no al revés. Muy posiblemente el sistema en funcionamiento no podrá asumir una actualización que haga posible el uso de los nuevos protocolos y sistemas de seguridad de las nuevas tecnologías, por lo que habrá que llegar a un compromiso entre ambos bandos. Este problema ya se ha detectado a pequeña escala en las redes inalámbricas de ámbito doméstico [6], y a pesar de que se han propuesto soluciones, ninguna parece lo bastante satisfactoria.

Así pues llegamos a un punto en el que hay que plantearse un análisis exhaustivo de la problemática de seguridad y encontrar una solución adaptativa que se ajuste al contexto.

IV. Análisis de la Seguridad Existente

El primer paso para realizar la integración de cualquier nueva tecnología en una red Smart Grid, es analizar el sistema actual. Se trata de saber qué puntos son vulnerables y qué mecanismos de seguridad están en funcionamiento para poder tener una visión global del estado de la seguridad del sistema [7].

La experiencia nos demuestra que cuanto más antiguo es un sistema, más vulnerable se vuelve, pues la evolución de la tecnología hace que la seguridad del mismo decaiga progresivamente. Aquí tendríamos buenos ejemplos como los protocolos antiguos basados en el cifrado DES [8]. En su momento se consideraron seguros pero a medida que evolucionó la tecnología acabaron sucumbiendo y resultaron inseguros. Por ello, deberemos tener en cuenta la antigüedad de los sistemas y las brechas de seguridad existentes para hacerles frente a la hora de integrar los nuevos mecanismos de seguridad.

En este aspecto es importante definir correctamente que propiedades tiene la red sobre la cual se van a integrar los dispositivos. Es probable que nos encontremos en entornos muy inseguros donde la utilización de la propia red ya suponga problemas de seguridad inherentes. Debemos considerar también problemáticas de seguridad en términos de conectividad entre extremos lejanos de la red que puedan padecer cortes de conexión así como problemas de fiabilidad.

V. Una Cuestión de Principios

Revertir los nuevos sistemas de seguridad al mínimo común de lo que soportan los que ya están en funcionamiento, sería tirar por los suelos cualquier oportunidad de poder ofrecer mayor seguridad al sistema. Si, por ejemplo, las nuevas tecnologías soportan el sistema de

cifrado AES-128 pero los dispositivos existentes no admiten mas allá de DES, no sería correcto operar todo con DES. Eso solo haría que añadir más inseguridad al sistema, pues la creación de más enlaces operando con DES no haría más que dar nuevas oportunidades a más ataques.

Por ello hay que plantearse el problema de la seguridad desde el principio de comunicación punto a punto. En otras palabras, debemos optar siempre por los métodos más seguros posibles dado un enlace cualquiera entre dos dispositivos. Si dos dispositivos tienen que conectarse lo tendrán que hacer eligiendo el mecanismo de seguridad más seguro que ambos sean capaces de soportar. Así pues, un dispositivo podría tener un enlace de baja seguridad hacia un destino, pero al mismo tiempo un enlace de alta seguridad hacia otro. De esta manera nos aseguraríamos que estamos utilizando la mejor opción disponible tecnológicamente en cada caso.

A pesar de que este diseño ofrece la máxima seguridad disponible para cada enlace, crea una problemática a la hora de manejar el tráfico pues es necesario algún mecanismo lo suficientemente flexible como para permitir la creación dinámica de dichas conexiones punto a punto. Las soluciones que se barajan son las siguientes:

- Túneles de datos
- VPNs (redes privadas virtuales)
- Conexiones cliente-servidor

VI. Túneles Seguros Para Entornos Inseguros

La creación de túneles de datos encriptados punto a punto se propone como la mejor opción para poder hacer frente a la mezcla de sistemas de seguridad existente. De esta forma se evita la problemática de la agrupación de equipos en redes virtuales al mismo tiempo que se reduce el impacto de usar servidores distribuidos para las conexiones cliente-servidor.

No obstante hay que analizar qué tipo de túneles crear y cómo. Optar por túneles de comunicación a nivel de red como es el caso de IPsec [9], es una de las mejores opciones en entornos mixtos. IPsec es lo suficientemente flexible como para proporcionar distintos mecanismos de cifrado e integridad de datos según convenga. Utilizar otro sistema para crear túneles como TLS podría ser una opción viable, pero asegurar más arriba en la capa de transporte, puede en algunos casos, no ofrecer la misma seguridad. De igual modo crear los túneles a nivel de enlace usando MACSec o 802.1x 2010 [10], podría crear problemas de conectividad pues no todos los equipos podrían soportarlo.

De este modo, mediante el uso de IPsec, la integración de nuevos dispositivos se podría realizar de forma segura. En este contexto, los nuevos dispositivos crearían túneles IPsec entre ellos mediante los cuales podrían comunicarse. No tendría mayor importancia que los datos fluyeran a través de enlaces inseguros mientras viajaran de un punto a otro pues el uso de IPsec protegería a los mismos.

En la siguiente imagen podemos ver un esquema de cómo se realizaría tal integración. En este caso, los I-Dev crean enlaces del tipo A (potencialmente inseguros) contra los dispositivos antiguos. Al mismo tiempo los dispositivos antiguos también crean enlaces de ese tipo entre ellos. Sin embargo, a la hora de comunicarse los I-Devs entre ellos, crean un tercer enlace punto a punto del tipo B, utilizando IPsec. Eso les permite comunicarse de forma segura entre ellos, pero al mismo tiempo no perder la comunicación con los antiguos dispositivos.

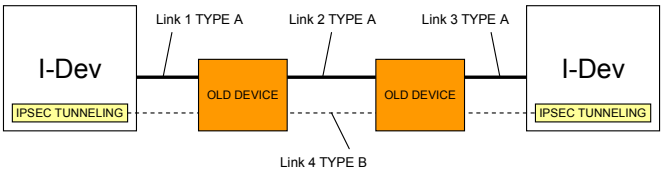


Fig. 3. Representación de los enlaces de comunicación integrados

Del mismo modo se puede asegurar el tráfico directo de Internet, ya que IPsec será igualmente válido para llevar a cabo tal objetivo. Dado que en la infraestructura es necesario un servidor de gestión de red o NMS (Network Management Server) se pueden crear conexiones punto a punto hacia cada uno de los I-Devs, de tal manera que se asegurarían las conexiones.

VII. Certificando la Identidad

Este contexto de seguridad y compatibilidad presenta, sin embargo, un problema de identidad evidente. El hecho de que el tráfico vaya a transcurrir sobre una red potencialmente insegura significa que nos podemos ver con ataques de suplantación de identidad de dispositivos.

Para subsanar este tipo de problema, se propone un sistema basado en certificados digitales que no sólo permita la identificación de los dispositivos sino que además establezca un mecanismo de cifrado de PKI (Public Key Infraestructure). De esta manera podremos identificar inequívocamente a los dispositivos y tener la certeza de que el túnel que se está creando es sólo contra aquellos dispositivos que están registrados.

Del mismo modo nos permitirá tener un mayor control sobre la red, al ser capaces de crear listas de revocación mediante las cuales podremos manejar aquellos certificados que hayan sido comprometidos o que no queramos que estén en uso dentro de la red. La siguiente imagen muestra un esquema simplificado del resultado final del contexto de seguridad presente en las comunicaciones de los I-Dev.

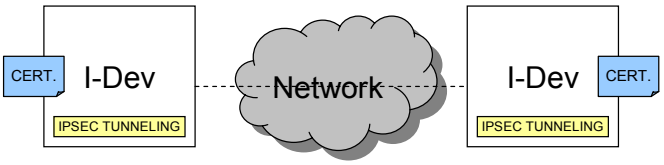


Fig. 4. Representación del contexto de seguridad resultante entre I-Devs

VIII. Seguridad al Integrar Redes Periféricas

Una vez tratado el problema de seguridad entre los nuevos dispositivos y los ya existentes, se nos plantea uno nuevo que hace referencia a las redes periféricas. Cabe recordar que el objetivo final del proyecto INTEGRIS es aportar nuevas funcionalidades añadidas a la red. Si bien es cierto que muchas funcionalidades nuevas podrán ser soportadas directamente por el propio I-Dev, la realidad es que en la mayoría de casos, se van a necesitar redes periféricas como WSN o RFID para poder aportar nuevos servicios.

En este caso los problemas de seguridad no se derivarán de la antigüedad de las redes (pues serán redes nuevas), sino de la escasa capacidad de cálculo y control sobre las mismas. Para solventarlos de forma correcta se plantea como mejor opción, efectuar un filtrado y control del tráfico procedente de este tipo de redes. También resultará muy importante realizar un análisis preliminar antes de la implantación para así escoger el estándar mas adecuado.

IX. Integración Completa y Segura

Como se puede ver, la seguridad va a estar presente a todos los niveles. Por un lado tenemos la conectividad entre I-Devs a través de redes existentes o agregadas, por otro las redes periféricas donde también incluiríamos los dispositivos de terceros, y por último la conexión directa a Internet. La siguiente imagen muestra un esquema.

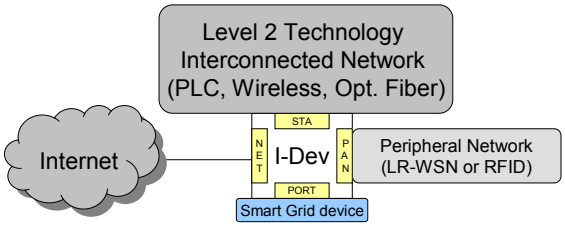


Fig. 5. Esquema simplificado de las redes que pueden ser potencialmente conectadas a los I-Dev

En los anteriores puntos hemos visto cómo asegurar el tráfico en cada uno de estos escenarios, bien sea mediante comunicaciones punto a punto en el caso de las redes convencionales o mediante técnicas de análisis y subsanación de vulnerabilidades para el caso de las redes periféricas.

Lo que se busca con todo esto es hacer una integración completa de forma segura, tal que todos los sistemas puedan operar entre ellos con la máxima seguridad posible para cada caso.

X. Estado Actual de Desarrollo de Integris

Actualmente se está llevando a cabo un proceso de pruebas sobre un entorno controlado. Se están desarrollando los procesos de gestión y control que operarán sobre los nuevos dispositivos así como las metodologías de funcionamiento pertinentes.

Por ahora y por lo que respecta la parte de seguridad, se han probado los mecanismos de control de redes periféricas así como los que se conectan varios I-Devs entre ellos mediante conexiones IPsec. También se ha probado la conectividad desde Internet desde y hacia las redes periféricas así como hacia otros I-Devs pertenecientes a la red. Las pruebas hasta el momento no han delatado problemas de seguridad relevantes más allá de los inherentes a las redes convencionales sobre las cuales operaba el escenario, pero en ningún caso se han visto comprometidos los datos de los tráficos entre I-Devs.

El estado del desarrollo por ahora se encuentra en su fase inicial pero el análisis preliminar está casi completado y los primeros tests se están ejecutando conforme lo especificado en el calendario del proyecto. En breve se iniciarán más pruebas, incluyendo un mayor número de módulos así como la operación simultánea de varios subsistemas.

XI. Conclusiones

Como hemos podido observar, la seguridad en las Smart Grids es un problema a tener muy presente y más aun cuando estamos mezclando distintas tecnologías y sistemas de seguridad. La solución que se propone en este artículo solventa los problemas analizados y expone el proceso a llevar a cabo.

Sin embargo hay que tener en cuenta que esta problemática está siendo estudiada cada vez con más atención por parte de más entidades, pues el crecimiento de la red implica un mayor control de la misma y formas de escalarla más seguras. Las pruebas llevadas a cabo hasta ahora demuestran que el planteamiento descrito en este artículo es factible y a priori puede considerarse práctico. Queda mucho por hacer aún, pero los primeros pasos ya se han dado en esa dirección.

Agradecimientos

La investigación de este trabajo ha recibido la financiación del *European Union European Atomic Energy Community Seventh Framework Programme* (FP7/2007-2013 FP7/2007-2011) mediante la subvención n° 247938.

Referencias

[1] INTEGRIS (Intelligent Electrical Grid Sensor Communications) http://fp7integris.eu/

[2] C Shuyong, S Shufang, LI Lanxin “Survey on smart grid technology” Power System Technology, 2009

[3] H Farhangi “The path of the smart grid” Power and Energy Magazine, IEEE, 2010

[4] NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 1.0 2010

[5] RE Brown “Security and Privacy Challenges in the Smart Grid” Power and Energy Society General Meeting, 2008

[6] D. Gonzalez “Home wireless security and privacy. A practical protocol mixing”, IARIA-AICT, 2010

[7] H Khurana, M Hadley, N Lu Smart-grid security issues IEEE Security & Privacy 2009

[8] Biham, Eli and Shamir, Adi "Differential Cryptanalysis of DES-like Cryptosystems". Journal of Cryptology, 1991

[9] RFC 2401 “Security Architecture for IP” 1998

[10] 802.1x-2010 http://www.ieee802.org/1/pages/802.1x-2010.html

Un algoritmo para el diseño de la topología de redes de comunicación con múltiples anillos

J. Silió, L. Rodríguez de Lope, K. Hackbarth
Departamento de Ingeniería de Comunicación, Grupo de Ing. Telemática
Universidad de Cantabria
Plaza de la Ciencia, 39005 Santander
jennifer.silio@alumnos.unican.es, laura@tlmat.unican.es, klaus@tlmat.unican.es

Abstract- **This paper provides two algorithms for the design of communication networks covering the determination of network hierarchy under a given number of network levels, the assignment of the lower level nodes to the higher ones and the design of the topology for the physical links. The objective is to minimize the total length of the network under the condition that the network is bi-connected; hence there are always two independent paths between each pair of nodes without any physical link in common. The topology design considers hierarchical ring topologies with the site constrains that the number of nodes in each ring is limited under a given maximal value. The paper shows the results of a study over an estimated set of 1250 Metro-Points of Presence (MPoP) from Spain for an hypothetical future Next Generation Network and indicates the application of the algorithms into two network planning tool, one for hybrid 2G/3G mobile networks and the other for the design of future NGN broadband networks and its practical use for techno- economical studies in the field of telecom regulation.**

Key words:-NGN planning, network design, network topology, algorithm

I. INTRODUCCIÓN

El diseño de redes tiene una fuerte importancia para estudios estratégicos sobre el desarrollo de futuras redes de comunicación. Los puntos clave son la alta integración del tráfico de todos los servicios, tanto de los servicios tradicionales como la voz y datos de baja velocidad, como también los servicios multimedia y de banda ancha incluyendo servicios de multicast y broadcast. Este proceso de integración se realiza en las redes fijas bajo el concepto de redes de la siguiente generación (Next Generation Networks NGN) pero también en las redes móviles de tercera generación, especialmente bajo HSPA.

Una parte muy importante en el diseño de redes, es determinar la infraestructura necesaria para las capacidades requeridas por la demanda de tráfico de los diferentes servicios. Esta infraestructura se compone la capa lógica de ubicaciones (nodos) con equipos de conmutación (capa 2 del modelo de OSI) o enrutadores (capa 3 del modelo de OSI) con interfaces (linecards) que los conectan y constituyen enlaces lógicos. La capa física integra capacidades de los enlaces lógicos en forma de enlaces físicos realizados por sistemas de transmisión (capa 1) y finalmente de los cables, tubos y zanjas (capa 0). Éstos últimos tienen unos altos costes fijos independientes del flujo de tráfico que se encamine, véase [1].

En el diseño de esta parte se debe calcular una topología que conecte todas las ubicaciones (nodos) de la red y minimice la longitud. Por razones de disponibilidad en el caso de una avería en un cable, la topología debe ser al menos biconexa, por este motivo la topología elegida es de tipo anillo.

Tradicionalmente las grandes redes nacionales se dividen en varios niveles y los nodos de un nivel inferior se asignan a un nodo superior y en algunos casos incluso a dos por razones de disponibilidad en caso de averías en un nodo. En caso de una red nacional con más de dos niveles, el proceso se repite hasta que en el nivel más alto quede un número limitado de nodos que a su vez se puedan conectar en forma de un anillo superior o de una red ligeramente mallada en forma de múltiples anillos.

Por razones técnicas y de disponibilidad ocurre que el número de nodos inferiores que se conecten en el mismo anillo está limitado. Entonces, resulta que en un nodo superior se conectan los nodos inferiores formando varios anillos, en la práctica típicamente desde dos hasta cuatro, resultando una forma de tipo trébol. Esto se aplica también en zonas montañosas donde la distancia directa entre dos nodos superiores es similar a las suma de sus distancias hacia el nodo superior.

En esta contribución se exponen unos algoritmos para el diseño de una red de comunicación en los que se trata de determinar la jerarquía de la red y el cálculo de los anillos que conectan los nodos inferiores a su nodo superior, teniendo en cuenta un número máximo de nodos que se pueden situar en un único anillo. En la siguiente sección se expone este problema con más detalles y se indica su lugar en la cadena de pasos para el diseño de una red. La tercera sección proporciona una descripción de los algoritmos y resultados basándose en un ejemplo práctico. La cuarta sección indica su aplicación en dos herramientas para el diseño y dimensionado de redes de comunicación y la última sección resume los resultados e indica futuras extensiones.

II. MODELO DE LA RED

En este caso los parámetros para el diseño de la red de telecomunicaciones son: el número de niveles en que la red se divide y el número máximo de nodos que se pueden ubicar en un anillo. El problema es determinar a partir de un conjunto de nodos, cuáles son superiores e inferiores para

ISBN: 978-84-694-5948-5

asignar cada nodo inferior a un nodo superior formando conjuntos de nodos llamados clusters. Este problema se denomina CLASIG (clasificación y asignación). En la selección de los nodos superiores se da preferencia a los nodos con más peso de tráfico para minimizar el transporte de tráfico entre los niveles. Además debe asegurarse una distribución equilibrada de los nodos superiores en el espacio, para minimizar la longitud de los enlaces lógicos que conectan cada nodo inferior con su nodo superior.

Sea N el número total de nodos, N_s el número de nodos superiores, p_i el peso del nodo, n_i el nodo superior, δ_i una variable binaria que indica si un nodo es superior y d_{min} el valor mínimo de la distancia geográfica entre los nodos superiores entonces la selección de los nodos superiores se modela como:

$$\max \sum_{i=1}^{N_s} \delta_i p_i \text{ con } \sum_{i=1}^{N} \delta_i = N_s \quad (1)$$

Una vez formados los nodos superiores, se asigna cada nodo inferior al nodo superior geográficamente más cercano, con lo cual se determina el conjunto de los nodos correspondiente a cada cluster C_k k=1…N_s con los nodos inferiores n_j j=1… I_k (I_k =nodos inferiores de cada cluster) y su nodo superior n_i.

Para cada cluster C_k se calculan los anillos que conectan los nodos inferiores con su nodo superior de forma que el número de nodos inferiores en un anillo sea menor o igual que un número máximo dado (nmax). Entonces el número de anillos Nan_k en un cluster C_k con el número máximo de nodos inferiores nmax en cada sub-cluster se calcula como:

$$Nan_k = \left\lceil \frac{[C_k]-1}{n\max} \right\rceil \quad (2)$$

El correspondiente modelo debe minimizar en cada cluster la longitud de los enlaces que forman los anillos. La variable d_{ij} indica la distancia entre cada enlace del anillo.

$$\min \sum_{i=1}^{C_k} \left(\sum_{j=1}^{I_K} d_{ij} \right) \quad (3)$$

III. ALGORITMOS PARA EL DISEÑO DE LA RED

En esta sección se exponen los dos algoritmos para el diseño de la jerarquía (CLASIG) y el diseño de los anillos (TREBOL) correspondiente a los modelos descritos en las ecuaciones (1) y (3).

A. El algoritmo CLASIG

El algoritmo CLASIG se basa en el principio "Depth First Search, DFS" que calcula entre todas las posibles combinaciones, la secuencia que proporciona en cada paso una mejora máxima en la optimización del modelo descrito en la ecuación (1), véase [2]. El comportamiento del algoritmo se basa en dos parámetros, el número de nodos superiores (N_s) y la distancia mínima entre los nodos superiores (d_{min}). Se compone de tres pasos:

Paso i) Inicio: se ordena la lista de N nodos según su peso en orden decreciente. Como resultado de dicha ordenación la carga de tráfico de un nodo es mayor o igual que la del siguiente de manera que el sumatorio de los tráficos de todos los N_s primeros nodos superiores, proporcione un límite superior de tráfico de la lista de nodos al que se denomina w_u:

$$w_u = \sum_{i=1}^{N_s} p_i \quad (4)$$

Como cada nodo tiene asignado el parámetro δ_i, el sumatorio de todos los parámetros δ_i debe ser igual al número de nodos superiores requerido:

$$\delta_i \in [0,1] \ \sum \delta_i = N_s \text{ con } N_s \leq N \quad (5)$$

Se calcula la matriz de distancias entre todos los nodos.

Paso ii) Bucle: se selecciona el primer nodo n_1 (se asume que es el superior), y se pone δ_i=1 y f=1 (contador de nodos superiores). Mientras que el contador de nodos superiores sea menor el parámetro N_s, se selecciona el siguiente nodo n_j y se calcula la distancia entre ellos d_{ij} y se comprueba que se cumple el criterio de la distancia mínima para todos los nodos recorridos previamente que sean superiores: $d_{ij} \geq d_{\min} \forall \delta_i$=1 y se pone $\delta_j = 1$.

Dado que este algoritmo ya ha sido implementado y probada su eficacia en anteriores ocasiones, se optimiza la solución obtenida incluyendo otro parámetro de entrada, la profundidad. Este nuevo parámetro realiza el proceso de clasificación de nodos tantas veces como indique su valor. Hace referencia al nodo por el que se comienza a aplicar el algoritmo DFS. Aplicando este algoritmo para varios niveles de profundidad, se obtienen tantas soluciones diferentes como niveles de profundidad, pudiendo conseguir de entre todas ellas la solución óptima, ver (1). Además se hace una mejora en el bucle, denominando IDFS(Improved Depth First Search). El funcionamiento es el siguiente: se compara el nodo actual (el valor que indique el parámetro de la profundidad) con todos los nodos anteriores que sean superiores (recorrido hacia delante) y con todos los posteriores que sean superiores (recorrido hacia atrás). Este algoritmo se denomina en la literatura "DFS with backtracking", véase [2]. Tras obtener una solución (vector de tamaño N de nodos superiores e inferiores) para cada valor de profundidad, de todas ellas se selecciona la óptima, que es la que tenga el mayor límite de tráfico inferior (1).

Paso iii)Asignación: se asigna cada nodo inferior a su superior más cercano geográficamente formando de esta manera los clusters.

Se ha estudiado el comportamiento del algoritmo con un ejemplo generado de datos de España basado en los "Post Area Codes, PAC" que se toman como aproximación para determinar los distritos en la planificación celular de una red móvil, véase [3]. Se pueden también interpretar como un conjunto donde se seleccionan los nodos básicos denominado Metro-PoP, que representan los puntos donde se termina la red de acceso, NGA, en las redes fijas tipo NGN, véase [4]. Se han seleccionado dentro de los 10000 PAC un subconjunto de 1250 nodos, seleccionando los nodos con más peso de cada comunidad autónoma y se ha generado un ejemplo con dos niveles. Las siguientes tablas indican los resultados del algoritmo CLASIG mejorado bajo diferentes parámetros sobre N_s y dmin. Se ha comprobado que las soluciones obtenidas con el CLASIG mejorado son mejores

respecto a las obtenidas con el CLASIG, por ello es el algoritmo utilizado en este caso.

En la Tabla 1 se muestran los resultados correspondientes a la suma de las longitudes de los niveles 0-1,1-2 y se ha añadido al total la longitud de la red que forma el nivel superior 2-2.

dmin / N_s	125,9	85,6	63,4
0	105468	104443	114804
30	66648	59910	70236
40	65467	58334	62326

Tabla 1. Longitud total de los enlaces lógicos para los dos niveles, (0-1,1-2,2-2)

Se puede apreciar que el valor óptimo de nodos superiores se consigue con la mayor distancia entre nodos superiores ya que así los nodos se distribuyen de la forma más equilibrada posible. Esto además ocurre cuando el número de nodos superiores en los diferentes niveles es grande ya que de esta forma el número de enlaces de capa lógica totales es menor, reduciendo así la longitud total.

B. El algoritmo TREBOL

El algoritmo TREBOL se aplica a cada uno de los cluster C_k k=1…N_s. El algoritmo se basa en el concepto de nodos vecinos que se define como:

Sean n_i y n_j dos nodos inferiores en un cluster C_k y sus coordenadas polares $(\rho_i, \alpha_i);(\rho_j, \alpha_j)$ en relación a su nodo superior. Entonces n_i es vecino de n_j si no existe otro nodo inferior, entre el ángulo menor formado entre n_i, n_j. La pareja vecina máxima es la cual, cuya distancia entre ellos tenga el valor máximo entre todas las parejas vecinas. En la Fig. 4 los nodos n_2, n_8 son la pareja vecina máxima. El algoritmo que se aplica a cada cluster C_k se basa en los siguientes pasos:

Paso i) Se define *m* como el contador de subcluster, el cual se inicia a 1 (*m*=1). Se calcula el número de anillos Nan_k ; si Nan_k =1 se pone $C_{mk} = C_k$ saltar al paso vii).

Paso ii) Si $Nan_k \neq 1$ poner $C_{mk} = C_k$.

Paso iii) Si el número de anillos es mayor que uno, se busca la pareja máxima entre todos los nodos inferiores que constituyen el primer nodo en los dos sub-clusters C_{mk}, $C_{(m+1)k}$ y se eliminan del cluster principal C_k.

Paso iv) Se busca para cada sub-cluster los (nmax-1) nodos vecinos, en la dirección contraria al sentido de las agujas del reloj para el nodo de la izquierda de la pareja de vecinos iniciales y en el sentido de las agujas del reloj para el nodo derecho; se eliminan los nodos del cluster principal.

Paso v) Se incrementa el contador de subclusters en dos ya que se han formado dos subclusters, m=m+2.

Paso vi) Si $[C_k] >0$ repetir desde el paso i).

Paso vii) Añadir a cada sub-cluster el correspondiente nodo superior y calcular el anillo de longitud mínima.

Con el ejemplo de la Fig. 3 se indica el trabajo del algoritmo suponiendo que nmax=4. En el primer paso se determina que se requieren tres anillos. En paso iii) se determinan el n_2 y n_8 como pareja vecina máxima y en el iv) se busca desde el nodo n_2 los vecinos en el contra-sentido de las agujas del reloj hasta haber llegado al nodo n_6. De igual forma se buscan desde el n_8 los vecinos en el sentido de las agujas del reloj hasta llegar a n_{10}. En el paso vi) se vuelve a i) ya que $[C_k] >0$; del paso i) se salta al vii)formando el tercer sub-cluster con nodos inferiores n_4, n_9. En el paso vii) se añade el nodo superior y se calculan los correspondientes anillos por lo que termina el algoritmo con el que se han formado los sub-clusters: $C_{1k} = \{n_0, n_2, n_1, n_3, n_6\}$, $C_{2k} = \{n_0, n_8, n_5, n_7, n_{10}\}$ y $C_{3k} = \{n_0, n_4, n_9\}$.

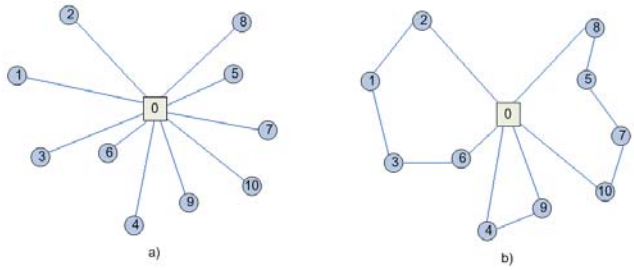


Fig. 3. Ejemplo de un cluster a) enlaces lógicos, b) enlaces de los anillos físicos

El cálculo de la longitud del anillo mínimo constituye el bien conocido problema del viajante comercial (Travelling Salesman Problem TSP) y para cuyo cálculo se han desarrollado varios algoritmos que se clasifican en algoritmos heurísticas y algoritmos exactos. Como el problema es "NP complete", que significa que el tiempo de cálculo puede ser largo en algunos casos, se aplica en este trabajo un algoritmo heurístico, véase [5] que se ha demostrado que calcula en la mayoría de los problemas reales la solución óptima, véase [6].

Para estimar la influencia de los dos parámetros sobre la red, en sus dos niveles se han sumado las longitudes de cada nivel y se ha añadido la longitud del anillo que conecta los nodos del nivel 2. El resultado se expone en la Tabla 2:

nmax/Ns	1:10 (125:9)	1:15 (85:6)	1:20 (63:4)
5	35358	44413	50080
6	33534	41028	45152
7	30912	37152	41095
8	30498	35205	38334
9	29312	35397	37085
10	28810	33248	35122

Tabla 2. Longitud de la totalidad de los enlaces físicos sobre los anillos entre los niveles 0-1, 1-2 y 2-2.

Se mejoran los resultados al aumentar el número de nodos por anillo, ya que se reduce el número de anillos totales creados minimizando la longitud total de los anillos de capa física. Aquí la longitud en el tercer caso (1:20) es 0,75 veces mayor que en el caso (1:10) y 0,85 veces mayor que en el segundo caso (1:15). La razón es la misma explicada en los niveles 0-1, y 1-2. La Fig. 4 muestra un ejemplo de los anillos formados entre el nivel 0-1.

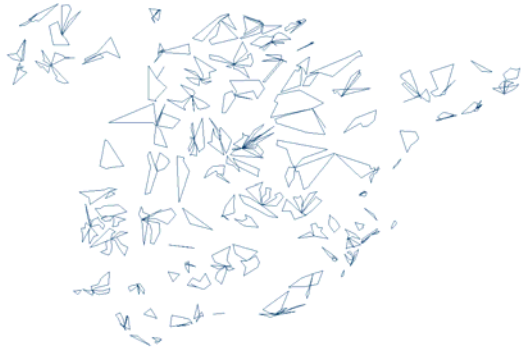


Fig. 4. Anillos entre los niveles 0-1 con 85 nodos superiores separados 40 Km y 7 nodos por anillo.

C. El algoritmo TREBOL modificado

En la parte B se ha considerado que las distancias entre los nodos se aproximan en base a una línea directa. Para mejorar el resultado se puede determinar la matriz de distancias en base a las distancias por carreteras, mediante un sistema GIS (Geographic Information System). En este caso puede pasar, que entre dos nodos vecinos no haya carretera directa o la carretera directa sea más larga que la conexión pasando por el nodo superior. En este caso los dos vecinos se asocian a diferentes clusters independientes del número máximo de nodos inferiores en un anillo. El paso iv) del algoritmo TREBOL se debe modificar de forma que, cuando se añada el siguiente vecino a un sub-cluster se debe controlar si la distancia entre ellos es menor que la suma de distancias en el camino sobre el nodo superior. En el último caso se termina el cluster y el vecino constituye el primer nodo de un nuevo sub-cluster.

IV. APLICACIONES

Los algoritmos presentados en la sección 3 se integran dentro de dos herramientas para la planificación y dimensionado de redes:

-2G/3G Connect, herramienta para la realización de estudios tecno-económicos de redes móviles con tecnología GSM, UMTS y HSPA.

-Taroca-NGN, herramienta para la realización de estudios tecno-económicos el estudio de redes de próxima generación.

Estas herramientas permiten realizar, a partir de la demanda de servicio de los usuarios, un despliegue de red aplicando los algoritmos CLASIG y TREBOL para establecer su jerarquía y estructura de anillos, y posteriormente encaminando el tráfico sobre la red establecida. Esto permite determinar las capacidades de los enlaces de la red y dimensionar el equipamiento necesario en los nodos para así evaluar su coste. Dichas herramientas proporcionan un instrumento para la realización tanto de estudios tecno-económicos como estratégicos por parte de los operadores de red, ya que permiten estimar la tendencia a medio y largo plazo de las inversiones, en función de la evolución de los servicios y el tráfico. También tienen una amplia aplicación en la regulación de las telecomunicaciones tanto a nivel nacional como internacional, en el establecimiento de límites en las tarifas de los servicios ofrecidos en función de diferentes parámetros, como puede ser la calidad de servicio, la disponibilidad, etc. Actualmente, ambas herramientas se están utilizando en estudios regulatorios en Austria, para el regulador RTR, véase [7]. Dada la geografía de este país alpino, el algoritmo TREBOL modificado permite establecer fronteras entre ubicaciones cercanas a vista de pájaro aunque separadas por cordilleras.

Por otra parte, está previsto incorporar la herramienta 2G/3G Connect al curso de Máster Planificación en Redes Móviles impartido por un consorcio de cinco universidades españolas, ver [8] donde aparece un ejemplo de España en el cual se utilizan los PAC ya mencionados.

V. CONCLUSIONES Y FUTUROS TRABAJOS

Se ha implementado el algoritmo CLASIG mejorado que logra distribuir los nodos de manera equilibrada ya que reduce la longitud de los enlaces de capa lógica. Además el algoritmo TREBOL calcula anillos de longitud mínima por lo que se consigue reducir la longitud total de los anillos de capa física.

El estudio en este documento se ha limitado al caso en que cada nodo inferior se asigna a un nodo superior. Futuros estudios deben tratar el caso de doble asignación o en caso de que existan varios niveles, las correspondientes combinaciones, por ejemplo asignación simple desde el nivel 0 hasta 1 y doble en el nivel 1-2.

En este documento se consideran como costes determinantes la longitud de la infraestructura de la red. En realidad, los costes se determinan con tres factores: la longitud, el ancho de banda requerido en los nodos, enlaces lógicos y físicos y el producto sobre el ancho de banda en los enlaces por su longitud. Actualmente se consideran los tres valores en las herramientas mencionadas con variaciones de los datos de entrada, como es el número de niveles y el número de nodos por nivel creando diferentes escenarios de la red y comparando los costes. Futuros estudios en el diseño de la jerarquía y el diseño de los anillos pueden considerar directamente criterios mixtos para acercarse directamente a una solución óptima sin los laboriosos estudios de un gran número de escenarios.

REFERENCIAS

[1]A.E García, K.D. Hackbarth, Next Generation IP Network Planning, WSEAS Transaction on Communications, 2006.

[2]T.H Cormen ,C.E Leiserson, R.L Rivest, C.Stein. Introduction to Algorithms, Second Edition. MIT Press and McGraw-Hill, 2001.

[3] J.A Portilla, Novel Heuristics for Cell Radius Determination in WCDMA Systems and Their Application to Strategic Planning Studies, Journal on Wireless Communications and Networking , 2009.

[4] K.D Hackbarth, L.Rodriquez, Kulenkampf, Cost Models for Bitstream Access Service, in M. Pagani, Encyclopedia of Multimedia Technology and Networking, Information Science reference, 2º ed. 2009.

[5] S.Lin, B.W. Kernighan, An efficient heuristic algorithm for the travelling salesman problem, Operation Research Vol 21 Nº 2, 1973.

[6] K. Domschke, Logisitc, Rundreisen und Touren, Ed. Oldenburg Munich, 1982.

[7] RTR, Bottom-up Cost Model for Austrian Mobile Networks http://www.rtr.at/de/tk/Praes12102010/WIK_Praesentation_Kostenrechnungsmodell_Mobilfunknetz.pdf

[8] K.D. Hackbarth, F. Gutiérrez, Asignatura "Planificación y Dimensionado de Redes Móviles", Máster y Doctorado en Tecnologías de la Información y Comunicaciones en Redes Móviles. http://www.ticrm.es/vigente/index.php