

Uso de JADE-LEAP para recuperar automáticamente sesiones de vídeo-streaming en teléfonos móviles

Álvaro Suárez[‡], Francisco Espino[‡], Elsa Macías[‡]

[‡]Grupo de Arquitectura y Concurrencia

Departamento de Ingeniería Telemática – Universidad de Las Palmas de Gran Canaria

Campus Universitario de Tafira, 35017 – Las Palmas de Gran Canaria (Gran Canaria)

Teléfono: 928 45 12 39 Fax: 928 45 13 80

E-mail: {asuarez, emacias}@dit.ulpgc.es

[‡]Estudiante de doctorado - Universidad de Las Palmas de Gran Canaria

fjespino@gmail.com

Abstract — El streaming es una técnica que se utiliza para transmitir información por la red de tal forma que su emisión desde el servidor, su comunicación por la red y la descarga y procesamiento en el cliente se produce simultáneamente sin necesidad de guardarlo en la memoria del cliente. Esta técnica es muy efectiva para la transmisión de información multimedia (vídeo) y es muy adecuada para los teléfonos móviles porque no existe necesidad de almacenar el vídeo en su memoria que es limitada (incluso el almacenamiento de datos sigue estando limitado bastante en los teléfonos móviles de última generación). Aunque la velocidad de transmisión de las redes inalámbricas ha aumentado mucho, esta técnica sigue siendo muy efectiva porque oculta la latencia de la red. Sin embargo; las diferentes tecnologías de comunicación inalámbricas utilizadas por los teléfonos móviles actuales (principalmente *Wireless Fidelity (WiFi)*, *Bluetooth* y *3G*) son muy propensas a sufrir desconexiones físicas intermitentes haciendo que se pierda la sesión de vídeo-streaming cada vez que un teléfono móvil pierde la cobertura de red durante un cierto tiempo y lo obliga a reconectar manualmente, con la consiguiente pérdida de efectividad de la técnica de streaming (debido a que se pierde la efectividad de ocultación de la latencia de red). En este trabajo se muestra una solución de recuperación automática del flujo de vídeo perdido y la sesión creando un nuevo mecanismo basado en proxies implantados con agentes software de la plataforma *Java Agent DEvelopment framework-Lightweight Extensible Agent Platform (JADE-LEAP)*. Los resultados experimentales obtenidos son muy prometedores por su efectividad y relevancia para los distintos casos prácticos que hemos probado.

I. INTRODUCCIÓN

Actualmente los teléfonos móviles, debido a los enormes avances en microelectrónica y el aumento de calidad de las tecnologías de comunicación, han evolucionado hasta proporcionar al usuario una gran cantidad de prestaciones y servicios. Los avances en los servicios multimedia han producido, después de muchos años de investigación, una técnica optimizada de comunicación denominada *streaming* que básicamente intenta solapar los tiempos de: emisión de información del servidor, el tiempo de transmisión por la red y el tiempo de recepción y procesamiento de información en el cliente. Básicamente se trata de ocultar la latencia de la red al iniciar y desarrollar una sesión entre el cliente y el servidor de vídeo. Su uso es especialmente recomendado cuando la cantidad de información a enviar es muy elevada: por ejemplo para transmisión de vídeo consiguiendo que el tiempo de espera del cliente por tramas de vídeo consecutivas sea mínimo, ya que de otra manera, y debido al gran tamaño de este tipo de contenidos, sería muy elevado.

Los teléfonos móviles de última generación vienen provistos de tecnologías de comunicación de banda ancha, por ejemplo WiFi [1] y *High Speed Downlink Packet Access (HSDPA)* [2], que permiten el uso de la técnica de streaming para acceder a contenidos multimedia, porque se logra una velocidad de transmisión elevada que permite ocultar la latencia de la red. Con esta técnica es posible acceder a contenidos de *Televisión Digital* que proporcionan las cadenas tradicionales de televisión (como por ejemplo [3]) reforzando sus cuotas de audiencia poderosamente a un precio razonable (sobre todo si se utiliza la WiFi con cuotas de conexión de coste cero para el usuario cuando se encuentra en el área de cobertura de un *Punto de Acceso gratuito*). Por otro lado, la capacidad de memoria de estos teléfonos móviles puede llegar a alcanzar los 32 GB, con lo cual se puede proveer un buffer de memoria elevado y el despliegue de aplicaciones intensivas en datos de manera efectiva a un coste muy reducido. La ventaja de la técnica de vídeo streaming es que no requiere que el vídeo se almacene en el teléfono móvil dejando espacio para albergar memoria auxiliar y programas para tratar el vídeo de manera muy eficiente.

Es bien conocido que el comportamiento caótico espacio-temporal de los canales inalámbricos [4] puede producir desconexiones esporádicas temporales del teléfono móvil que provocan que la técnica de streaming pierda efectividad cuando se pierde la sesión debido a una de estas desconexiones (comenzar de nuevo la sesión provoca una pérdida de tiempo muy elevada no asumible por el usuario abandonando definitivamente la sesión con la consiguiente pérdida económica para la empresa multimedia). Este problema es tan importante que para el teléfono móvil *iphone* ya se ha provisto de una aplicación que informa al usuario de que podría entrar en una zona con riesgo de desconexiones [5]. Aunque los servidores

de vídeo streaming [6] incluyen un canal virtual de control no orientado a la conexión, ninguna orden de este canal suele informar de la pérdida de conexión explícita que pueda permitir controlar automáticamente la pérdida de sesión (la pérdida de conexión entre el cliente y servidor puede ser de naturalezas muy variadas y producirse a distintos niveles de la arquitectura de Red).

Normalmente se han utilizado proxies para adaptar la velocidad de transmisión, teniendo en cuenta el estado del canal inalámbrico [7], de recursos previamente almacenados [8], o de vídeo en vivo [9]. Estos proxies se han programado como agentes software de una plataforma propietaria [10] para resolver las desconexiones mientras se resuelven los procesos de roaming usando un esquema de buffering proactivo; sin embargo no se ha aplicado en la práctica sobre arquitecturas de teléfonos móviles comerciales. En este trabajo proponemos una arquitectura de proxies que soluciona las pérdidas intermitentes de conexión cuando se realiza streaming con teléfonos móviles comerciales como es el *Nokia N95* [11] (aunque es compatible con otras arquitecturas comerciales de similares prestaciones). Con ella se evita la pérdida de información al recibir un flujo de vídeo completo (recibiendo en diferido los datos que se hubieran perdido por efecto de la desconexión). Para ello se utiliza la plataforma de agentes JADE-LEAP [12] [13] (hasta donde alcanza nuestro conocimiento nunca antes se había usado esta plataforma para resolver este tipo de problemas prácticos), especialmente orientada a dispositivos móviles que incorpora una gran cantidad de los procesos necesarios. La razón más importante por la que se ha usado JADE-LEAP es que los propios agentes proporcionan mecanismos de buffering necesario para no provocar pérdidas de fotogramas y de la sesión de vídeo streaming, y además la solución se programa de manera muy elegante a alto nivel en el lenguaje Java (multiplataforma) sin necesidad de reprogramar ni el Cliente ni el Servidor (por tanto siendo aplicable a cualquier tipo de cliente y servidor comerciales). Los resultados experimentales invitan al optimismo.

En el apartado II se presentan los beneficios que proporciona el uso de la plataforma JADE-LEAP para programar los proxies, lo cual justifica su uso en el nuevo mecanismo que hemos diseñado para teléfonos móviles. En el apartado III se presenta el mecanismo propiamente dicho para los teléfonos móviles describiéndolo de tal manera que se entienda sus principales características. En el apartado IV se analizan los resultados experimentales que se han obtenido después de hacer una gran cantidad de pruebas en situaciones reales. Finalmente se presentan unas breves conclusiones y posibles líneas de trabajo futuro derivado de este trabajo.

II. BENEFICIOS DEL USO DE JADE-LEAP

Dentro de las características usuales de los agentes software (autonomía, aprendizaje, proactividad, escalabilidad, flexibilidad y fiabilidad) no se suele tener tan en cuenta que también proporcionan una abstracción de alto nivel que reduce la dificultad de diseño de las aplicaciones que los utilizan. Esto lo consiguen gracias a su naturaleza bien definida y método de desarrollo propio caracterizado por una alta modularidad, escalabilidad, flexibilidad y sencillez en las comunicaciones.

La modularidad es un aspecto inherente a los agentes ya que la lógica del proceso general se distribuye en los mismos. Además el sistema es fácilmente escalable, añadiendo o quitando agentes de forma sencilla ya que no tienen relación entre ellos más allá de los mensajes que se puedan enviar. La flexibilidad es otro factor muy beneficioso para una aplicación distribuida. La población de agentes puede distribuirse entre diferentes máquinas como sea preciso sin tener que reprogramar nada. En cuanto a las comunicaciones entre agentes éstas se reducen a un simple tráfico de mensajes con el que el programador se puede olvidar de todos los condicionantes adicionales a este proceso que estén a un nivel más bajo.

Entre las múltiples utilidades que se le pueden dar a los agentes software y que estén relacionadas con su uso en el streaming con móviles destacan la implementación de estructuras colaborativas que pueden ser de gran provecho para la monitorización de la sesión de streaming o el reparto de cómputo asociada a la misma. Esta última característica la planteamos como novedosa en [14] para programar computadores portátiles. En este artículo demostramos que también se puede programar sobre teléfonos móviles.

En estos momentos existen varias plataformas que permiten el despliegue de agentes software, tanto propietarias como de software libre. Los requisitos deseables para el presente trabajo son la utilización de una plataforma ya consolidada de software libre, con soporte para agentes que se ejecuten en dispositivos móviles, y que esté implantada en un lenguaje de amplia aceptación por parte de este tipo de aparatos. La plataforma que reúne todos estos requisitos y se ha utilizado es la JADE-LEAP.

JADE es una plataforma de software libre completamente implantada en Java. Se utiliza para el desarrollo de aplicaciones distribuidas basadas en agentes que cumplen con la especificación *Foundation for Intelligent Physical Agents (FIPA)* [15]. Es un proyecto ampliamente consolidado en el que están implicados Telecom Italia, Motorola, Whitestein Technologies AG, Profactor GMBH, y France Telecom R&D.

Para posibilitar la ejecución de JADE en dispositivos móviles se desarrolló un proyecto asociado denominado LEAP. Éste reemplaza algunas partes del núcleo de JADE pero no cambia su *Application Programming Interface (API)*. Se puede

desplegar en dispositivos compatibles con *Java 2 Standard Edition (J2SE)* a partir de la *Java Development Kit (JDK)* 1.4, con *Microsoft .NET Framework* a partir de su versión 1.1 y con *Java 2 Micro Edition (J2ME)* a partir de *Mobile Information Device Profile (MIDP)* 1.0 lo cual abre sobremanera el abanico de posibles usuarios.

Para mejorar la ejecución de JADE-LEAP sobre J2ME ésta se hace en modo *Split*. Esto quiere decir que el contenedor donde se despliega el agente se divide en un *FrontEnd*, en el dispositivo móvil, y un *BackEnd*, en un servidor remoto, permanentemente conectados. Con ello se consigue repartir la carga computacional. Otro beneficio es un uso optimizado de la conexión inalámbrica entre estos dos componentes.

JADE-LEAP implementa un mecanismo para gestionar las desconexiones del *FrontEnd* del agente residente en el dispositivo móvil con su *BackEnd*. Los mensajes que se envían al dispositivo móvil cuando éste está fuera de cobertura se guardan en un buffer del *BackEnd*, y se envían cuando se reconecta. El *FrontEnd* del dispositivo móvil se intenta reconectar automáticamente de forma periódica con el *BackEnd*.

III. DESCRIPCIÓN DEL MECANISMO

En [14] presentamos un mecanismo adaptado a *Computadores Portátiles (CP)*. Sin embargo, este mecanismo no es directamente migrable a teléfonos móviles por las razones expuestas en el apartado II. Por ello, es necesario adaptarlo identificando a cuatro entidades: el *Servidor*, el *Agente Proxy Servidor (APS)*, el *Agente Proxy Cliente (APC)* y el *Cliente* (Figura 1). El Servidor se conecta por cable con el APS y éste a su vez con el APC por medio de una *Wireless Local Area Network (WLAN)*. El APC está pensado para que se ejecute junto con el Cliente en el CP con conexión inalámbrica. En los enlaces Servidor-APS y APC-Cliente se transmiten mensajes usando *Real Time Streaming Protocol (RTSP)* [16] que utiliza a *Real Time Transport Protocol (RTP)* [17] para la comunicación del vídeo. Entre el APS y el APC, los cuales se despliegan dentro de la misma plataforma de JADE, se establece un tráfico *Foundation for Intelligent Physical Agents (FIPA) Agent Communication Language (ACL)* [18]. Las peticiones y respuestas RTSP se encapsulan en mensajes FIPA ACL de tipo INFORM, y los paquetes RTP en mensajes de tipo PROPOSE. Con ello se consigue simplificar la comunicación entre agentes evitando la declaración de una ontología, lo cual disminuiría el rendimiento de la plataforma JADE. Entre los pasos 1 al 4 se transmite una petición RTSP desde el Cliente al Servidor y entre los pasos 5 y 8 una respuesta RTSP desde el Servidor al Cliente. Durante la retransmisión de paquetes RTP (pasos 9 al 12) es posible que se produzcan desconexiones inalámbricas. En este caso los paquetes que llegan al APS y son encapsulados, y que se intentan enviar al APC, son tratados por el servicio *JADE Persistent Delivery*. Éste va guardando todos los mensajes FIPA ACL que no pueden ser entregados a su destinatario en un buffer (paso 14), hasta que el agente en cuestión vuelva a incorporarse a la plataforma después de su desconexión y se le puedan reenviar los mensajes (paso 15). Éste es un servicio adicional de la plataforma JADE que se debe arrancar y configurar manualmente, ya que no se encuentra activado por defecto. Con este mecanismo la desconexión a nivel físico del CP por problemas en la WLAN pasa inadvertida para el Servidor y para el Cliente, ya que los enlaces Servidor-APS y APC-Cliente se mantienen mientras tanto.

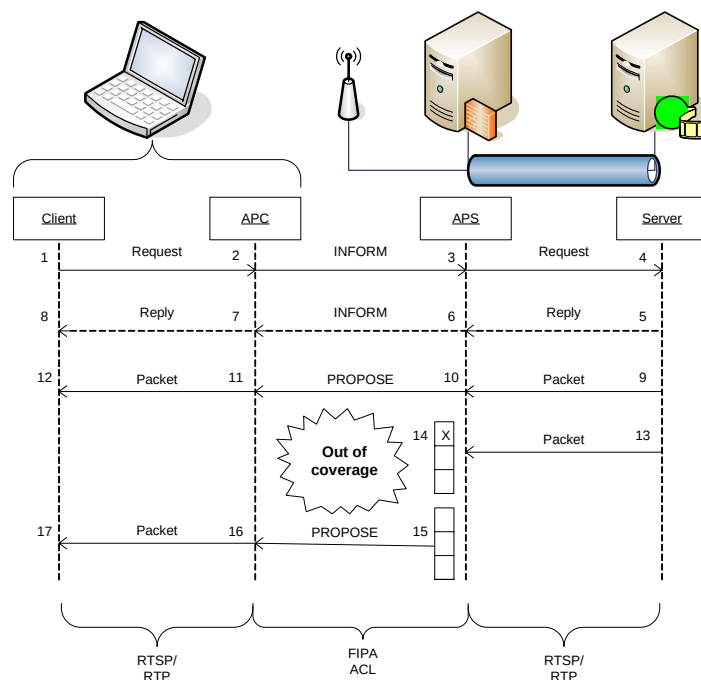


Fig. 1: Mecanismo para CPs.

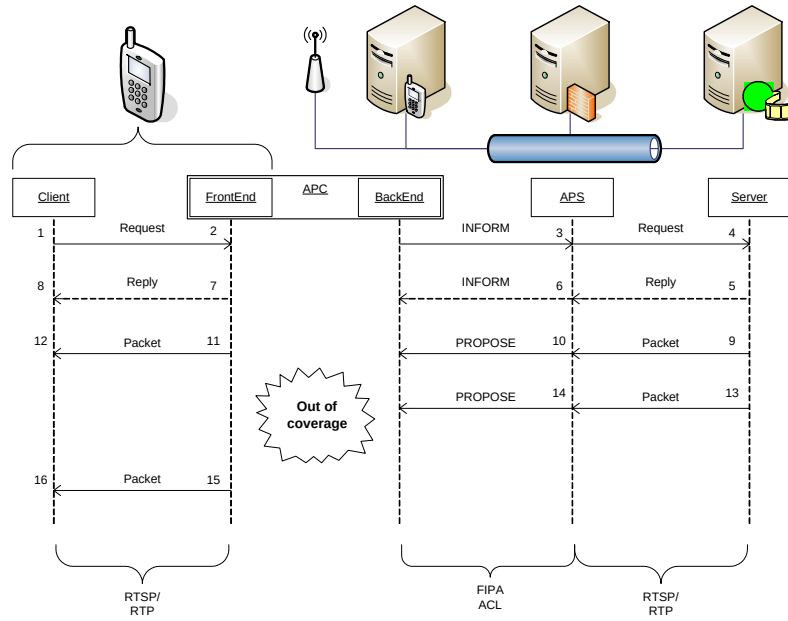


Fig. 2: Adaptación del mecanismo para teléfonos móviles.

Como se comentó, la migración directa a teléfonos móviles de este mecanismo no es posible debiendo usar para ello *Java 2 Micro Edition (J2ME)* y *JADE-LEAP*. Ahora el APC se debe dividir en dos partes, para ser ejecutado en el dispositivo móvil hay que utilizar el modo *Split*: el *FrontEnd* del APC se ejecuta en el móvil y el *BackEnd* en el Servidor; las dos partes se conectan por medio de la WLAN (las posibles desconexiones son gestionados de forma transparente por el APC, por lo que no se necesita ninguna prestación adicional como pasaba con el servicio *JADE Persistent Delivery* del sistema para CP). En la Figura 2 se muestra esta nueva arquitectura. Un aspecto importante se refiere a la implementación del mecanismo descrito en este artículo, la cual utiliza J2ME. Un total de 1.5 billones de dispositivos móviles [19] sobre una población estimada en 2.6 billones [20] pueden ejecutar programas escritos en este lenguaje. Esto da una idea de la compatibilidad posible del software desarrollado. Además, el uso de JADE-LEAP facilita la ampliación y mejora del sistema en futuras revisiones.

IV. EVALUACIÓN DE RESULTADOS EXPERIMENTALES

El Cliente y Servidor utilizados para las pruebas del sistema tratado en este artículo han sido creados especialmente para esa labor. El formato de vídeo que manejan es el *Motion Joint Photographic Expert Group (MJPEG)* [21]. Este formato se basa simplemente en una sucesión de imágenes JPEG por lo que su transmisión tiene una carga prácticamente constante. Esto nos permite observar el comportamiento del mecanismo en presencia de tráfico constante pero de manera conservativa al no optimizar la compresión. En cualquier caso, debido a que nuestro mecanismo únicamente almacena *frames* se comportaría eficientemente con formatos de compresión más potentes como *Moving Picture Experts Group (MPEG4)*.

A continuación se muestran unas gráficas comparativas del rendimiento del sistema sin y con proxies. Para la obtención de las muestras se ha utilizado una WLAN con tecnología *WiFi* con un punto de acceso de 54 Mbps que conecta un móvil Nokia N95, en el que se ejecutan el Cliente y el *FrontEnd* del APC, y un CP Pentium M 1.40 GHz con 1GB de RAM y con una tarjeta *WiFi* de 54 Mbps, en el que se ejecutan el *BackEnd* del APC, el APS y el Servidor de vídeo streaming. En la Figura 3 se puede observar una fotografía de este sistema en funcionamiento utilizando los proxies. En ella se puede observar en la zona inferior el móvil, y en su pantalla un fotograma del vídeo que el Cliente está reproduciendo. En el monitor del CP hay una ventana para el servidor en la zona superior derecha que indica el fotograma que se está enviando. En la parte central aparece la sucesión de mensajes FIPA ACL intercambiados entre el APC y el APS proporcionada por la utilidad *Sniffer* de la plataforma JADE-LEAP. La primera pareja de INFORM encapsula la petición y la respuesta positiva de la orden SETUP del RTSP. La segunda pareja de INFORM encapsula la orden PLAY del RTSP. Los PROPOSE que vienen a continuación encapsulan los paquetes RTP que contienen cada uno un fotograma, es decir, una imagen JPEG.



Fig. 3: Fotografía del móvil y el CP en tiempo de ejecución con los agentes proxies JADE-LEAP.

El vídeo utilizado tiene una resolución de 240×180 pixels y su frecuencia de muestreo se ha reducido a 2 *fps* para que el cómputo adicional que supone el registro de datos (marcas de tiempo de salida y llegada de paquetes RTP) no afecte al proceso. Los parámetros analizados son el retardo, o tiempo que transcurre entre el envío y la recepción de un paquete RTP, y el *jitter*, o diferencia de tiempo entre la llegada de un paquete y la recepción de su anterior.

En las figuras 4 y 5 se muestran, respectivamente, las gráficas de retardo y jitter del sistema manteniendo el teléfono móvil en una posición fija. El retardo cuando no se utilizan los proxies, aunque muy fluctuante, tiende a mantenerse alrededor de un valor constante, lo cual hace que el jitter también tenga las mismas características. Cuando sí se usan los proxies el sistema tiene un comportamiento similar aunque la principal diferencia se encuentra en la magnitud de las oscilaciones del retardo y del jitter, que en este caso son mucho más pequeñas. Esto además de cumplir con el requisito indispensable de no modificar el comportamiento del vídeo streaming, lo mejora, y hace que el flujo de datos tenga mayor constancia de presentación al usuario. Otra diferencia es la del valor de retardo promedio, unos 200 ms por encima del sistema sin proxies, tiempo éste utilizado para su establecimiento.

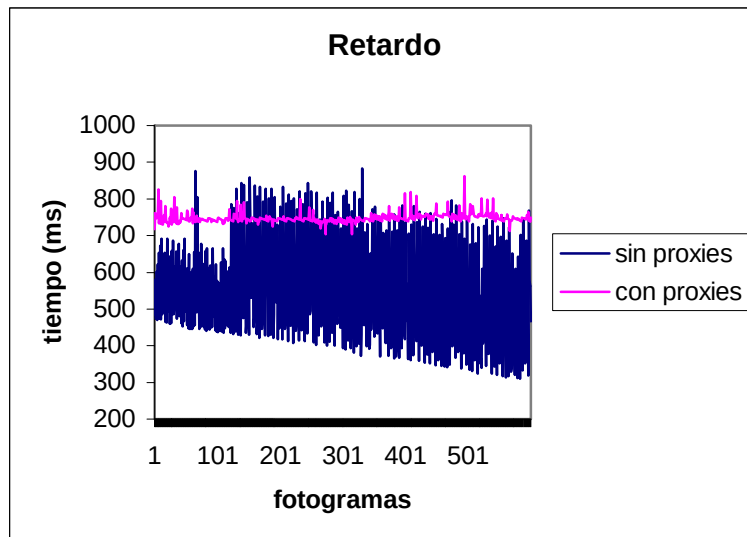


Fig. 4: Retardo del sistema desarrollado con y sin proxies.

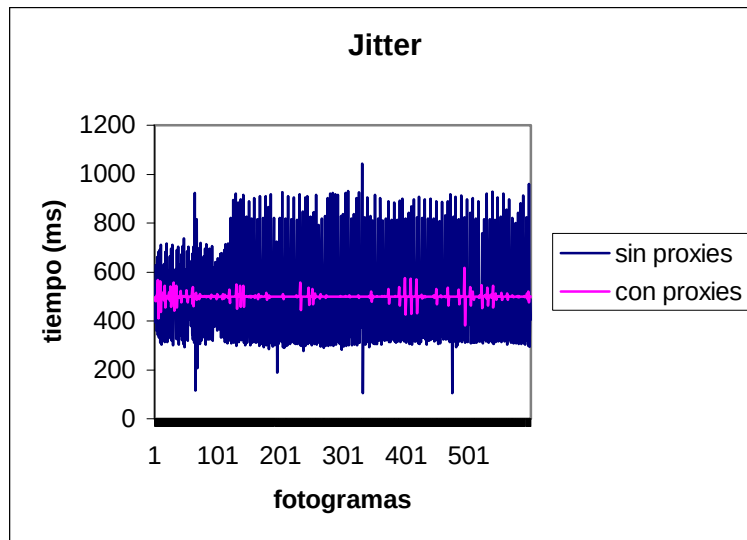


Fig. 5: Jitter del sistema desarrollado con y sin proxies.

Para observar el comportamiento del sistema desarrollado utilizando los agentes proxies JADE-LEAP en una gestión de una desconexión física del teléfono móvil se han incluido las figuras 6 y 7 que representan el retardo y el jitter, respectivamente. El momento de la pérdida de cobertura corresponde a la variación brusca en cada una de las gráficas, el escalón en el retardo y el pico en el jitter. Una vez recuperada la conexión los valores de retardo y de jitter tienden a un valor constante no modificándose el comportamiento del vídeo streaming.

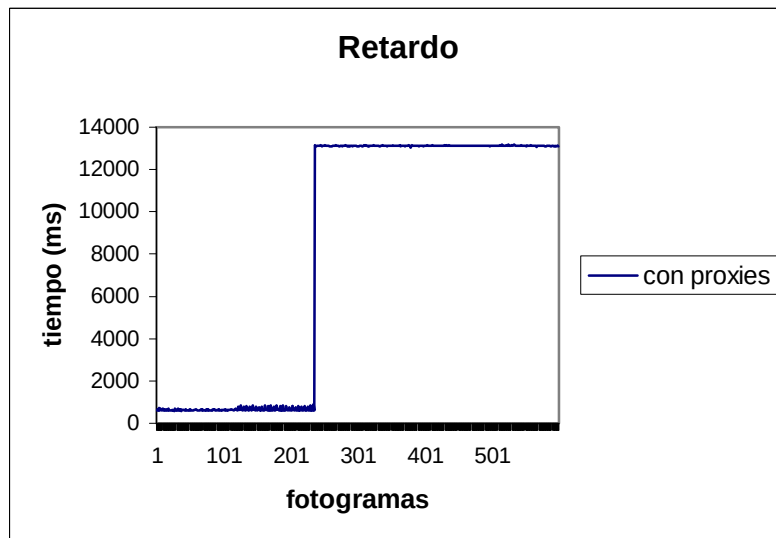


Fig. 6: Retardo del sistema desarrollado usando los agentes proxies JADE-LEAP y gestionando una desconexión física.

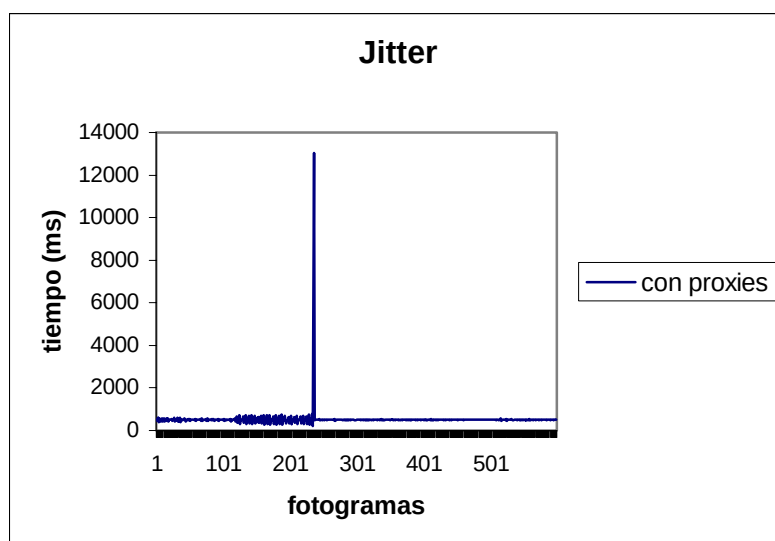


Fig. 7: Jitter del sistema desarrollado usando los agentes proxies JADE-LEAP y gestionando una desconexión física.

V. CONCLUSIONES

En este artículo se ha presentado un mecanismo basado en agentes software capaz de recuperar automáticamente sesiones de vídeo streaming perdidas por una desconexión física en teléfonos móviles usando la plataforma de agentes JADE-LEAP. Para verificar su funcionamiento se han realizado numerosas pruebas de desconexión, manteniendo el teléfono móvil fijo en una posición geográfica y también en itinerancia en interiores usando WiFi. En todos los casos el sistema recupera con facilidad y con baja latencia la sesión original (que comparado con el mismo sistema sin nuestro mecanismo produce una sensación objetiva de funcionamiento en el usuario muy elevada). Lo importante es que en ningún caso hubo que reprogramar el Cliente ni el Servidor que ya estaban programados para trabajos previos. Es de destacar que la arquitectura del CP es menos potente que la de los de última generación lo que da mayor relevancia a nuestros resultados.

Con este sistema de agentes proxies se mitiga el problema de las desconexiones intermitentes que sufren los móviles al perder la cobertura de la red inalámbrica a la que estén conectados. No se pierde información ya que la sesión no se cancela y todos los paquetes RTP se guardan hasta su llegada al cliente para su reproducción. Por lo tanto aumenta la fiabilidad del vídeo streaming.

Dentro de las futuras líneas de investigación y desarrollo originadas por el trabajo descrito en este documento están la de hacer colaborar a los agentes JADE-LEAP para la predicción del comportamiento de la conexión inalámbrica. Otra línea es la de ampliar el software para que pueda ser usado con clientes y servidores comerciales que comunican flujos de datos en MPEG4.

AGRADECIMIENTOS

Este trabajo ha sido subvencionado en parte por el Ministerio de Educación y Ciencia, CICYT y el Fondo Europeo de Desarrollo Regional (FEDER) bajo el proyecto de investigación TSI2005-07764-C02-01, la Consejería de Educación, Cultura y Deporte del Gobierno de Canarias y FEDER (PI042004/164), y por el Ministerio de Industria, Turismo y Comercio bajo el contrato PROFIT FIT-330210-2007-41.

REFERENCIAS

- [1] IEEE P802.11, "The Working Group for Wireless LANs", disponible on line: <http://grouper.ieee.org/groups/802/11/>.
- [2] Harri Holma, Antti Toskala (editores), "HSDPA/HSUPA for UMTS: High Speed Radio Access for Mobile Communications", Wiley, ISBN: 978-0-470-01884-2, 2006.
- [3] TVE, "TVE a la Carta", disponible on line: <http://www.rtve.es/alacarta/index.html>.
- [4] David Tse, Pramod Viswanath, "Fundamentals of Wireless Communication", Cambridge University Press, 2005.
- [5] Apple, "Iphone avisará de la falta de cobertura", disponible on line: <http://www.overpal.com/gadgets/19863-iphone-avisara-de-la-falta-de-cobertura.html>.
- [6] VideoLAN - Free Software and Open Source video streaming solution for every OS!. "VLC", disponible on line: <http://www.videolan.org>.
- [7] Bobby Vandalore, Wu-Chi Feng, Raj Jain, Sonia Fahmy, "A Survey of Application Layer Techniques for Adaptive Streaming of Multimedia", *OSU Technical Report*, OSU-CISRC-5/99-TR14, 1999, 19 pags (versión extendida es: Real-Time Imaging, Vol. 7, No. 3, 2001, p. 221-235).
- [8] Jacob Chakareski, Philip A. Chou, "RaDiO edge: Rate-distortion Optimized Proxy-Driven Streaming from the Network Edge", *IEEE/ACM Transactions on Networking*, pp. 1302 – 1312, diciembre 2006.
- [9] Lixin Gao, Zhi-Li Zhang, Don Towsley, "Proxy-Assisted Techniques for Delivering Continuous Multimedia Streams", *IEEE/ACM Transactions on Networking*, Vol. 11, No. 6, diciembre 2003, pp. 884-894.
- [10] Paolo Bellavista, Antonio Corradi, Carlo Giannelli, "Mobile Proxies for Proactive Buffering in Wireless Internet Multimedia Streaming", *Proceedings of the IEEE International Conference on Distributed Computing Systems Workshop (ICDCSW'05)*, junio 2005, pp. 297-304.
- [11] Nokia N95, disponible on line: http://www.nokia.es/link?cid=PLAIN_TEXT_815211.
- [12] F. Bellifemine, G. Caire, A. Poggi, G. Rimassa, "JADE, A White Paper". *Journal of Telecom Italia Lab*, Vol. 3, No. 3, pp. 6-19, septiembre 2003.
- [13] Giovanni Caire, Federico Pieri, "LEAP USER GUI", disponible on line: <http://jade.tilab.com/doc/LEAPUserGuide.pdf>.
- [15] FIPA, "The Foundation for Intelligent Physical Agents", disponible on line: <http://fipa.org/>.
- [14] A. Suárez, M. La Menza, E. Macías, and V. Sunderam, "Automatic resumption of streaming sessions over WiFi using JADE", *IAENG International Journal of Computer Science*, Vol. 33, pp. 92-100, ISSN 1819-656X, 2007.
- [16] RFC 2326 - RTSP: Real Time Streaming Protocol, 1998, disponible on line: <http://www.ietf.org/rfc/rfc2326.txt>
- [17] RFC 1889 - RTP: A Transport Protocol for Real-Time Applications, 1996, disponible on line: <http://www.faqs.org/rfcs/rfc1889.html>
- [18] FIPA ACL Message Structure Specification, "Foundation for Intelligent Physical Agents", 2002, Disponible on line: <http://www.fipa.org/specs/fipa00061/SC00061G.pdf>
- [19] Sun Microsystems, "Sun Announces Support For the Next Generation Mobile Java Platform Across Entire Mobile Product Line", disponible on line: <http://www.sun.com/aboutsun/pr/2007-03/sunflash.20070301.1.xml>.
- [20] Bill Roberts. "Cell Phone Market to Shift", disponible on line: <http://www.edn.com/article/CA6417983.html?partner=enews>.
- [21] Video Formats White Paper, "Understanding Video Formats", *BroadWare Technologies Inc*, disponible on line: http://www.axis.com/adp_cd/adp_cd8/companies/broadware/Video_Format-White_Paper-v3.pdf.