

VLinEx. Una herramienta para comunicaciones multimedia en entornos colaborativos.

David M. Cortés-Polo, José Luis González-Sánchez, Javier Carmona-Murillo, Manuel Domínguez-Dorado,
Francisco J. Rodríguez Pérez
Departamento de Ingeniería de Sistemas Informáticos y Telemáticos. Universidad de Extremadura
Avenida de la Universidad S/N
10071 – Cáceres (Cáceres)
Teléfono: 927 257 431 Fax: 927 257 202
E-mail: dcorpol@unex.es, jlgs@unex.es, jcarmur@unex.es, mdomdor@unex.es, fjrodri@unex.es.

Abstract. *Nowadays multimedia communications are changing very fast; VoIP, TV over IP, Videoconference... For this reason, new protocols are being developed to afford a better transmission over Internet. Some other mature protocols as Multicasting were discarded by those new protocols even though the benefit provided by those mature protocols. In this way VLinEx, a collaborative application, try to approach the Multicast transmissions to the users, who don't have any knowledge of the Multicast communications or Mbone applications.*

1 Introducción

Desde las primeras fases de implantación de las redes se pensó la posibilidad de transmitir por estas no sólo texto sino también vídeo o audio. Los primeros experimentos se producen sobre la red *ARPANET* probando la transmisión de datos digitalizados. Los protocolos con los que se experimentan son *NVP (Network Voice Protocol)* [1] y *ST (Stream Protocol)* [2]. En 1996, la *ITU-T* desarrolló un conjunto de herramientas que pretendían ser la base de las comunicaciones multimedia sobre la red de redes, con la consecución del protocolo *RTP (RealTime Transfer Protocol)* [3]. En esta última década, la industria de las telecomunicaciones ha contemplado la aparición de varias tecnologías revolucionarias; la telefonía móvil, que ha modificado los conceptos de disponibilidad y aumentado la productividad; la expansión de Internet que ha multiplicado la información disponible a los usuarios y aumentado las posibilidades de comunicación; y por último, la aparición de las redes de banda ancha que han propiciado el acceso de los usuarios a nuevas formas de comunicación. Apoyándose en estas mejoras, han aparecido nuevas tecnologías para las transmisiones multimedia, las cuales están siendo utilizadas por la mayor parte de proveedores de servicios a través de Internet.

Sin embargo, existen técnicas ya maduras y que están siendo explotadas en un gran número de empresas, que por su alto coste o por sus necesidades se han quedado relegadas a un entorno más empresarial dejando de lado a los usuarios. Este es el caso de las comunicaciones colaborativas (*multicast*), una tecnología madura que bien por la complejidad de uso para el usuario final o bien por los requerimientos de los protocolos en los que se basa, no ha conseguido una mayor difusión exceptuando diversos campos de actuación.

En este documento propone una herramienta de comunicaciones multimedia en entornos colaborativos bajo GNU/Linux llamada VLinEx [4] y posteriormente se muestran los resultados obtenidos de esta propuesta al usarse en una red real de explotación.

2 Una propuesta de aplicación colaborativa: VLinEx

Las herramientas Mbone [5] tuvieron un gran auge a finales de los 90 debido sobre todo al potencial que podría sacarse de este tipo de comunicaciones. En contra se encontraron grandes problemas al intentar implantarse en redes ya en producción basadas en tráfico *best effort*. Es por esto que a partir del año 2000 se abandonó el uso de Mbone y por tanto se produjo un gran desfase entre esta tecnología y otras que aparecieron después. Todas las herramientas *Mbone*, implementan el protocolo *RTP con control mínimo* [6], es decir, transmisiones multimedia sin gran calidad y usando códecs antiguos. *Multicast* [7, 8] está pensado para trabajar sobre UDP y por lo tanto todos los *routers* deberían procesar este tipo de paquetes. El problema es que Internet está basado en tráfico *best effort* con lo que los *routers* actuales no procesan tráfico *multicast*. Para esta problemática se desarrollaron diferentes técnicas que permitían enviar la información desde una *isla multicast* a otra. La solución más extendida son los *túneles multicast*, programas encargados de crear nuevas interfaces de red conectando diferentes islas.

Nuestra propuesta se encamina hacia una aplicación que mantenga la funcionalidad de las herramientas *Mbone* existentes y además introduzca mejoras sustanciales que veremos a continuación. La estructura general se muestra en la figura 1.

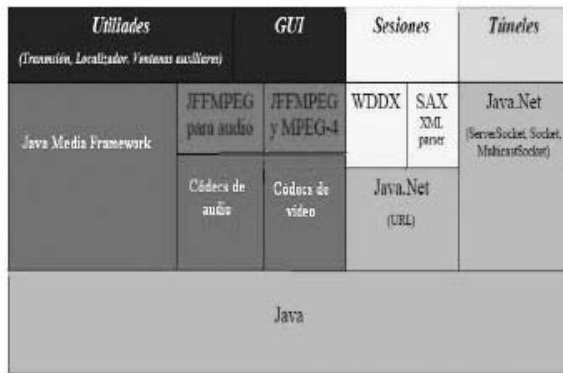


Figura 1: Estructura de la aplicación

La propuesta se basa en el uso de *Java* como lenguaje de programación, usando una *API* proporcionada por *SUN Microsystems* llamada *Java Media Framework (JMF)*, que proporciona las herramientas necesarias para el manejo de contenidos multimedia, así como para la transmisión de los mismos a través del protocolo *RTP*.

Como anteriormente se comentó, las aplicaciones *MBone* no se han actualizado desde el año 2000 y este *framework* es un claro ejemplo, por tanto, los códecs de audio y vídeo no incluyen los nuevos formatos aparecidos posteriormente y se hace necesario incluir a la aplicación un conjunto de códecs, más modernos, para que permitan reproducir los nuevos formatos aparecidos en estos años. Para ello usaremos *Jffmpeg*, un pack de códecs, adaptación de los códecs *ffmpeg* implementados en *C++* y que son usados por la mayoría de los reproductores actuales en *Linux*.

Para mantener la compatibilidad con los sistemas *Mbone* ya implantados, en la transmisión se usarán los códecs recogidos en el estándar *RTP con control mínimo*, es decir, la información de vídeo se transmitirá en el formato *H-263* [9, 10] y la información de audio se transmitirá en formato *γ-Law*.

Dado que cada vez más usuarios están familiarizados con el uso de reproductores de vídeo (ya que cada vez abunda más el elemento multimedia en la informática doméstica), es necesario que la aplicación tenga una apariencia sencilla, usando para ello un entorno estándar para que el aprendizaje de la aplicación sea lo más rápido posible, como se puede ver en la siguiente figura.



Figura 2: Interfaz de la aplicación

Una de las metas de esta propuesta, es la simplicidad de manejo, es por esto que la interconexión con las demás *islas multicast* no debe ser compleja como lo era anteriormente usando programas externos (*túneles multicast*). En este caso se ha ideado un sistema de sesiones que permita una administración de las mismas de manera sencilla y permitiendo que los usuarios se conecten a una sesión sin necesidad de tener conocimientos avanzados de sistemas operativos o de protocolos.

El túnel ha sido ideado partiendo de *TCP* sobre el que encapsularemos los datos de *RTP*. Han sido elegidas las transmisiones sobre *TCP* ya que en muchos casos, en las redes corporativas nos encontramos elementos como *firewalls*, *direccionamiento NAT*, etc, que hacen que el uso de *UDP* esté muy restringido mientras que *TCP*, aunque tiene desventajas [11], parece una solución más viable en estos tiempos.

En cuanto a la administración de sesiones, se ha optado por mantenerlas en un servidor central en el que se almacenará en una base de datos, toda la información referente a cada una de las sesiones y cada usuario, usando un navegador Web o la propia aplicación pueda acceder a esta información para el manejo de la sesión. La tecnología elegida para el servidor Web es *PHP* y *MySQL*, mientras que para el intercambio de datos con la aplicación se usó un protocolo de paso de mensajes llamado *WDDX* que es derivado de *XML*. De tal manera, que el servidor implementa un servicio de sesiones con el cual se podrá obtener la información necesaria para configurar la aplicación, crear túneles y así poder interconectar *islas multicast*. Aunque existen otras implementaciones ya estandarizadas para el sistema de sesiones como es el caso del *protocolo SIP* [12], se buscaba dotar de un mayor grado de simplicidad a la aplicación, de tal manera que el usuario no necesite conocimientos de este tipo de comunicaciones y como consecuencia de se ha optado por usar un sistema de sesiones propio más simplificado.

Esta aplicación al estar orientada hacia el software libre, está licenciada con *GPL* para que cualquiera pueda incluir nuevas funcionalidades o mejorar las presentes.

3 Pruebas a través de Internet

Este trabajo no tiene sólo una parte de desarrollo sino que además de la aplicación se ha estudiado el comportamiento de la misma en un entorno real como es una comunicación a través de Internet. Es decir, un entorno que puede generar problemas de manera aleatoria como congestiones, descarte de paquetes, caídas de routers, etc...

En este caso vamos a comprobar cómo se comporta el túnel anteriormente descrito en una comunicación a través de Internet. Se van a comunicar una isla *multicast* que se creó en la *Escuela Politécnica de Cáceres (EPCC)* y que constaba de dos PCs con Linux con *otra isla multicast* en la que había otros dos PCs también con Linux en un domicilio cualquiera.

Como se puede observar en la figura 3, la comunicación dentro de la *isla multicast de la EPCC* la comunicación es fluida y no hay problemas para la visualización del contenido multimedia, el cual es un archivo de vídeo y de audio codificado con *DIVX 5* y *MP3* y recodificado para la transmisión con los formatos anteriormente comentados.

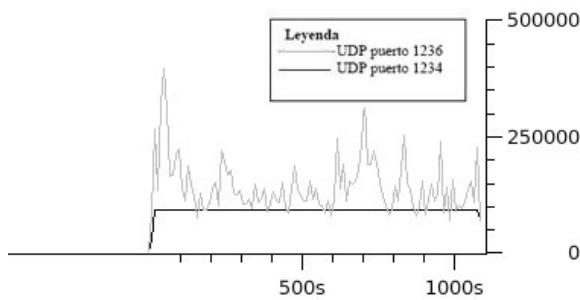


Figura 3: Canales de audio (UDP Puerto 1234) y vídeo (UDP 1236) para una comunicación dentro de una isla multicast

En la otra *isla multicast* de esta comunicación, se puede observar que la transmisión a través de Internet afecta negativamente al rendimiento de la comunicación. En la figura 4 se puede observar los dos canales del *túnel TCP*. Y en la figura 5, las retransmisiones de los paquetes generados por esta comunicación TCP.

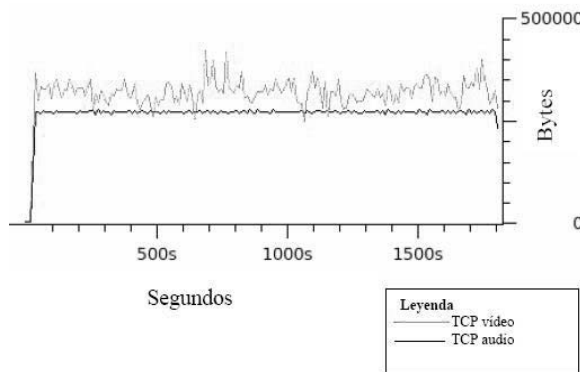


Figura 4: Ancho de banda usado por los dos túneles TCP

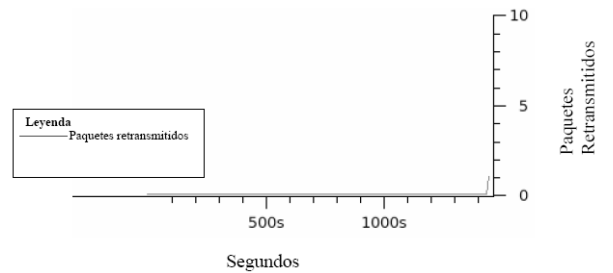


Figura 5: Retransmisión de los paquetes

Como vemos en la figura 4, tenemos dos túneles abiertos, el primero que nos encontramos (*TCP vídeo*), es la sesión de vídeo. El payload de vídeo es variable y depende del *frame-rate*. Es por esto que el tamaño del paquete depende de la codificación, el algoritmo codifica la información teniendo en cuenta los *frames* anteriores y posteriores y elimina la redundancia, con lo cual una secuencia rica en colores y en movimiento necesitará un paquete de datos mucho mayor que una secuencia oscura y estática, que se transmitirá por la red con un paquete menor. Como contrapunto vemos que la transmisión de audio es completamente plana. Esto es debido a que el *payload* es siempre el mismo y por esto se envía la misma cantidad de información en cada uno de los paquetes.

Al tratarse de una comunicación TCP, se asegura que todos los paquetes llegan a su destino, es por esto que con casi toda seguridad, se deberán hacer retransmisiones ya que al usar Internet como medio de transmisión se puede perder algún paquete por saturación de algún *router* intermedio. Como se puede observar en la figura 5, no se retransmite ningún paquete hasta casi el final de la prueba.

Una vez estudiada la transmisión a través del túnel, en esta propuesta, toda la información enviada por los dos canales del mismo, se transforma otra vez en tráfico *multicast* que se redistribuye por la segunda *isla*. El tráfico por la *segunda isla* reconstruido se comporta como muestra la figura 6.

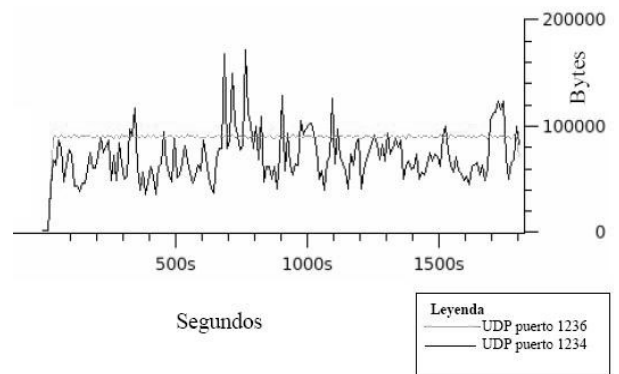


Figura 6: Tráfico UDP generado para audio (UDP Puerto 1234) y vídeo (UDP 1236)

Si comparamos la figura 3 con esta última, el ancho de banda utilizado por una y otra transmisión varía considerablemente. Esto es debido a los retardos incluidos a través de una red que no provea de QoS . Es por esto que se puede asegurar que la desincronización en la transmisión será grande con respecto a la reproducción en local. Para confirmar esta afirmación nos podemos fijar en el audio que se mantiene constante en unos 100 kbps. Con lo cual irá a una velocidad normal de reproducción mientras que el vídeo irá a una velocidad menor que la que se requeriría, debido a que en muchos momentos la gráfica del vídeo está por debajo de la del audio.

4 Conclusiones

Estamos en un momento dulce para la tecnología multimedia. Los grandes cambios que se avecinan en los próximos años hacen que sean todavía más emocionantes si caben las distintas investigaciones que pueden surgir a través de los diferentes campos de la telemática orientada al mundo multimedia.

En estos momentos la transmisión de contenidos multimedia a través de Internet es un hecho, no sólo mediante *streaming*, sino también mediante técnicas *multicast*. El punto conflictivo se produce al transmitir mediante multicast haciendo uso de redes compartidas como es el caso de Internet, que no puede ofrecer ninguna garantía de servicio como otros sistemas ya implantados, como por ejemplo la televisión ofrecida por los *ISP*. Es por esto que no se pueda transmitir con garantías ni con cierta calidad los contenidos multimedia y que se tenga que recurrir a herramientas como los túneles para las comunicaciones *multicast*.

Es por esto que al usar los túneles, siempre se introduce cierta redundancia de información (cabeceras TCP) y además se pierde la fluidez de una comunicación UDP debido a las retransmisiones que se producen al perder un paquete con TCP. Este es uno de los grandes motivos por los que esta tecnología solo queda reservada a ciertas redes de datos que permitan *multicast* nativo y a ciertos servicios que se puedan permitir el coste del mantenimiento de la línea.

En este trabajo no se ha podido describir con gran detalle todas las pruebas hechas a la herramienta en entornos reales, tanto locales como de área extensa. De tal manera que si se quiere profundizar en las pruebas y la implementación de la herramienta, la página del proyecto es <http://gitaca.unex.es/agila/agorared/>

Agradecimientos

Este trabajo está financiado, en parte, por la Junta de Extremadura (Consejería de Infraestructuras y

Desarrollo Tecnológico) por medio del proyecto AGILA2 (Expediente, PRIA060271)

Referencias

- [1] Danny Cohen, "SPECIFICATIONS FOR THE NETWORK VOICE PROTOCOL (NVP) and Appendix 1: The Definition of Tables-Set-#1 (for LPC), Appendix 2: Implementation Recommendations", Internet Engineering Task Force, RFC 741
- [2] Forgie, J., "ST - A Proposed Internet Stream Protocol", IEN 119, M.I.T. Lincoln Laboratory, 7 September 1979.
- [3] Schulzrinne H., Casner S., Frederick R., and Jacobson V., 2003, "RTP: A Transport Protocol for Real-Time Applications," Internet Engineering Task Force, Work in Progress (actualización RFC 1889).
- [4] Proyecto Agila 2, 2006, <http://gitaca.unex.es/agila/>
- [5] Kumar V., 1996, "Mbone. Interactive Multimedia on the Internet", New Riders Publishing.
- [6] H. Schulzrinne, "RTP Profile for audio and video conferences with minimal control", Internet Engineering Task Force, January 1996, RFC 1890
- [7] Almeroth, K.C., 2000 "The evolution of multicast: from the Mbone to interdomain multicast to Internet2 deployment", Network, IEEE, Volume 14, Issue 1, Jan.-Feb. 2000 Page(s):10 – 20.
- [8] Ganjam, A. and Zhang, H., 2005, Internet multicast video delivery, Proceedings of the IEEE, Volume 93, Issue 1, Jan 2005 Page(s):159 – 170
- [9] Ghanbari M., 2003, Standard Codecs: Image Compression to Advanced Video Coding 1ed . Ed. Institution of Electrical Engineers. Great Britain.
- [10] Zhu C., 1997, "RTP Payload Format for H.263 Video Streams" Internet Engineering Task Force, Work in Progress.
- [11] Allman M., Paxson V., and Stevens W., 1999, "TCP Congestion Control," Internet Engineering Task Force, RFC 2581.
- [12] Rosenberg J., Schulzrinne H., Camarillo G., Johnston A., Peterson J., Sparks R., Handley M., 2002, "SIP: Session Initiation Protocol", Internet Engineering Task Force, RFC 3261