

# Escenario de Pruebas para el Soporte de Videostreaming Usando Herramientas Libres

Franco Arturo Urbano / Ingeniero en Electrónica y Telecomunicaciones / Magíster en Ingeniería Área Telemática / Docente Fundación Universitaria de Popayán  
frurbano5@gmail.com

Wilmar Yesid Campo / Ingeniero en Electrónica y Telecomunicaciones / Magíster en Ingeniería Área Telemática / Ph.D en Telemática Universidad del Cauca  
wilcampo@unicauca.edu.co

Gabriel E. Chanchí / Ingeniero en Electrónica y Telecomunicaciones / Magíster en Ingeniería Telemática  
Candidato a Ph.D en Telemática  
Universidad del Cauca  
gabrielc@unicauca.edu.co

**Resumen:** Una de las tecnologías que hoy en día tiene repercusiones importantes en ámbitos como la educación, la salud y el contexto empresarial es el videostreaming, lo anterior teniendo en cuenta las facilidades que este brinda para el acceso a contenidos multimedia de manera remota, en vivo o en diferido, permitiendo la comunicación de manera independiente a la ubicación. Es importante destacar que estas temáticas son un nicho fuerte de investigación debido a: los diferentes protocolos tanto a nivel de aplicación (propietarios RTMP y estándar RTSP) como de transporte, los diferentes codificadores, los diferentes tipos de redes (inalámbricas, cableadas, por cable), los diferentes sistemas operativos y el desarrollo de aplicaciones, derivando de esto una gran cantidad de tecnologías y protocolos que permiten su soporte. En este artículo se presenta una revisión de las tecnologías más difundidas para la implementación de este servicio y se propone un escenario de pruebas para el soporte de videostreaming haciendo uso de herramientas libres sobre diferentes sistemas operativos. El escenario de pruebas propuesto pretende servir de referencia o recomendación para diversos entornos de aplicación que requieran el montaje e implementación del servicio de videostreaming.

**Palabras clave:** Escenario de pruebas, videostreaming, herramientas libres, RTSP

## Introducción

Internet ha permitido la descarga de archivos desde las tempranas etapas de su desarrollo, buscando inicialmente ser una red para compartir información entre personas distantes geográficamente y acceder a archivos que no se habrían podido consultar sin la existencia de esta red; dando especial importancia al acceso a la información, así se tuviera que esperar mucho tiempo mientras esta se lograba descargar. Sin embargo Internet fue evolucionando, haciendo posible el acceso a archivos de audio y video del orden de los Megabytes y de buena calidad, en tiempos de descarga aceptables.

No obstante, hasta hace unos años las tecnologías de apoyo a la internet en cuanto a la descarga de archivos de audio y video requerían que la máquina cliente descargara completamente los archivos antes de que el usuario pudiera ver y escuchar; lo cual traía inconvenientes tales

como: tiempos de transferencia bastante largos y la dificultad de visualización en tiempo real. Este panorama se complicaba aún más al considerar los anchos de banda de la red que hasta hace unos años en países en vía de desarrollo, se medían en Kilobytes.

En este contexto es donde surge una tecnología llamada *streaming*, en la cual continuamente se envían peticiones de datos de video (videostreaming) o de audio al servidor, y como respuesta éste envía flujos o stream, los cuales no se espera a que lleguen todos para poder ver las imágenes u oír el sonido en el lado del cliente, sino que se va viendo el vídeo y/o escuchando el sonido conforme van llegando los flujos que componen todo el archivo solicitado.

Esta forma de trabajo sobre la red mejora notablemente los tiempos de espera y además permite manipular archivos multimedia en directo o en diferido [1].

Entre las ventajas del videostreaming están sus bajos requerimientos, puesto que con un servidor convencional y una conexión de al menos 512Kb es suficiente, en cuanto a los cortafuegos o firewalls de los clientes estos no supondrán problema alguno para la transmisión. Por otra parte, el videostreaming pensado como herramienta no solo se puede utilizar por un cliente para recibir un archivo multimedia sino que es una herramienta idónea para ser utilizada en el ámbito educativo, empresarial o de la administración pues permite transmitir o retransmitir conferencias, charlas, eventos, programas, seminarios, teleeducación, entrevistas, entre otros. Las redes sociales en Internet son hasta la fecha el fenómeno social más difundido por la rapidez con la que han multiplicado sus usuarios en muy poco espacio de tiempo y parte de ese éxito se lo deben al videostreaming[2].

A nivel técnico, existen protocolos diseñados y estandarizados para la comunicación entre los clientes y los servidores de streaming. Los primeros protocolos de streaming fueron desarrollados por gigantes multinacionales tales como Microsoft, Real o Apple quienes vieron enormes posibilidades de ofrecer video en tiempo real. A continuación se presentaran dos protocolos de uso reciente: RTSP y RTMP.

*RTSP (Real Time Streaming Protocol)* [3] es un protocolo de flujo de datos en tiempo real no orientado a conexión que se utiliza para definir cómo se hará el envío de información entre el cliente y el servidor, permitiendo iniciar y controlar el envío de contenidos multimedia, bien previamente almacenados o bien en directo. Este protocolo trabaja a nivel de aplicación y controla que la entrega de datos se realice correctamente. RTSP define diferentes tipos de conexión y diferentes conjuntos de requisitos, para intentar conseguir siempre un envío de flujo de datos a través de redes IP lo más eficiente posible. RTSP es independiente del protocolo de transporte y puede funcionar tanto sobre UDP o TCP, no obstante, en la mayoría de casos se utiliza TCP para el control del reproductor y UDP para la transmisión de datos con RTP [4]. Durante una sesión, un cliente puede abrir y cerrar conexiones fiables de transporte con el servidor mediante peticiones RTSP [4].

*RTMP(Real-Time Messaging Protocol)* [5] es un protocolo desarrollado por Macromedia, ahora Adobe, para transmisiones de alto desempeño de audio y video entre tecnologías de la plataforma Adobe Flash, incluyendo Adobe Flash Player y Adobe AIR. El protocolo ha sido liberado como una especificación abierta para crear

productos y tecnología que permita la entrega de audio, video y datos en formatos SWF, FLV y F4V compatibles con Adobe Flash Player. Este protocolo utiliza TCP a nivel de la capa de transporte y soporta el flujo multimedia codificado en formato FLV (Flash Video) ofrecida mediante el servidor FMS (Flash Media Server de Adobe System). 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 [6]. Este protocolo es posible configurarlo para que trabaje sobre el protocolo UDP. En la actualidad los servidores de streaming que implementan el protocolo RTMP son FMS, Onlinelib Vídeo VCS Communication Server (incluyendo un soporte para el iPhone), Wowza Media Server, WebORB Integración (libre y disponible para .NET Java y ColdFusion) y Red5.

A partir de los conceptos y protocolos básicos de streaming presentados anteriormente, en el presente trabajo se pretende proporcionar un conjunto de herramientas libres y tecnologías dentro de un escenario de pruebas, que permita la transmisión y recepción de contenidos multimedia en diferentes tipos de clientes (computador, celular, tablet) y sobre varios sistemas operativos, haciendo uso del protocolo de streaming RTSP. La escogencia de este protocolo tuvo en cuenta la necesidad de contar con herramientas libres para el estudio y evaluación en entornos académicos, de manera que el escenario de pruebas fue configurado considerando las herramientas más adecuadas y con mayor difusión en el mundo del software libre. Este trabajo pretende servir de referencia para proyectos que requieran la implementación de servicios basados en videostreaming sobre diferentes entornos de aplicación, como el sector salud y el sector educación.

Este artículo está organizado de la siguiente forma: en la sección 2 se presentan los conceptos y tecnologías tenidas en cuenta para la formulación del escenario de pruebas; en la sección 3 se describe cada uno de los módulos funcionales que componen el escenario de pruebas, así como las características tecnológicas de los clientes y servidores utilizados; en la sección 4 se presentan las pruebas de consumo de memoria realizadas al módulo servidor, tras la conexión simultanea de múltiples clientes; finalmente en la sección 5 se presentan las conclusiones obtenidas con este trabajo, así como los posibles trabajos futuros que se derivan del mismo.

## Marco Conceptual

Un video se puede entender como una serie de imágenes que al ser mostradas una detrás de otra, dan una sensación de movimiento, a su vez, una imagen se puede representar como una matriz de puntos (píxeles) cada uno con su correspondiente color.

En [7] se puede encontrar una descripción de las propiedades de los videos. Otro concepto fundamental es el de Códec, el cual es un algoritmo tanto de compresión como de codificación del video, su finalidad es comprimir, codificar y cifrar la información con el fin de poderla almacenar en el disco o transmitirla por la red. Para la implementación de aplicaciones cliente, un códec se encuentra en una librería externa, y la aplicación simplemente la llama cuando detecta que un video está en ese formato determinado [8].

Dentro de los formatos de video más utilizados se tienen los siguientes: MPEG-1 primer conjunto de estándares y formatos de compresión de video y audio diseñados por MPEG (Moving Picture Experts Group). MPEG-2 versión de MPEG de mayor calidad, el cual permite codificar con interlineado. MPEG-4, es un estándar de la ISO/IEC, versión mejorada del MPEG-2, soporta más códec de audio y video que sus predecesores [9]. H.264 es técnicamente idéntico al códec MPEG-4 parte 10 su objetivo es conseguir mayor calidad a menor bitrate que los formatos anteriores [10] y 3GP, versión reducida del contenedor MPEG-4 parte 14, diseñado para reducir al máximo los requisitos de almacenamiento y ancho de banda [11].

Los requisitos fundamentales para escoger un servidor de streaming son: uso de software abierto y gratuito, servicios de valor agregado (listas de reproducción, adaptación de tasa, multicast), flexibilidad para el desarrollo (mayores alternativas para la implementación del proyecto), y los menores tiempos de respuesta a las operaciones realizadas por el usuario del sistema (reproducir, parar, avanzar). Dentro de las alternativas como servidores se tienen las siguientes:

- Darwin Streaming Server: DSS, este servidor permite transmitir videos a través de internet, redes LAN y WLAN, mediante el uso de los protocolos RTP, RTSP y SDP (Session Description Protocol). Se trata de una versión de código abierto del Quicktime Streaming Server, es capaz de trabajar como servidor de archivos MP3, MP4 y 3GP hinted. No soporta adaptación de tasa en tiempo real aunque se anuncie de manera experimental mediante el intercambio

de cabeceras del mismo tipo. Su principal desventaja es las estadísticas poco fiables (CPU, throughput) [12].

- Catrastreaming (Open Streaming Server), proyecto Open Source publicado bajo General Public License, cuyas principales características son: servidor de código abierto, soporte de los protocolos: RTP, RTSP, SDP, no soporta adaptación de tasa en tiempo real, no tiene GUI aunque permite conexión por servidor http y páginas XML, debe hacerse una petición http para cada consulta, no cuenta con soporte y las estadísticas poco fiables [13].

- Helix DNA Server, es una versión gratuita de Helix Universal Server disponible en su página de proyecto. Se distribuye bajo General Public. Sus principales características: servidor de código abierto, soporta RTP, RTSP, SDP. Este servidor no soporta archivos 3gp [14].

- VLC, es un reproductor multimedia de código abierto, multiplataforma que permite la reproducción y distribución de contenidos bajo demanda utilizando los protocolos RTP/RTCP y RTSP. Si el cliente realiza una petición rtsp://ip:puerto/recurso accederá al archivo indicado. Pese a que VLC es una gran herramienta para la transcodificación o transmisión multicast no demuestra ser robusto ni cómodo para la transmisión de video bajo demanda. Dado que no es una opción interesante para el servidor principal, sí puede ser utilizado como transcodificador o incluso como servidor alternativo [15], siendo usado principalmente como cliente.

- Helix Universal Server, es la versión comercial de Helix DNA, permite adaptación de tasa en el servidor. Además provee todo un paquete de software que se puede utilizar, entre otras cosas, para realizar la codificación del material. No se ha considerado al no tratarse de un sistema gratuito [14].

- Quicktime Streaming Server (QTSS) es la versión comercial de Darwin Streaming Server. Contempla todas las características propias de este tipo de servidores pero también se trata de una versión no gratuita. De hecho solo se distribuye con una versión server del sistema operativo Mac OS X [12].

- LIVE555, es un servidor RTSP de fuente abierta que utiliza los protocolos RTSP, RTP y SDP para el streaming multimedia, siendo compatible con reproductores multimedia como VLC y QuickTime. LIVE555 es una aplicación de fuente abierta, su código fuente

está disponible y puede ser modificado para cumplir con requerimientos específicos. LIVE555 puede generar varios tipos de flujos de archivos multimedia como son: TransportStream de MPEG (.ts), WebM o Matroska (.webm o .mkv), MPEG-1, MPEG-2 (.mpeg), MPEG-4 (.m4e), H.264 (.264), DV (.dv), MP3 (.mp3), WAV (.wav), AMR (.amr), AAC (.aac). Estos flujos pueden ser recibidos y/o reproducidos por cualquier cliente multimedia RTSP/RTP que obedezca a los estándares, entre ellos se incluyen: VLC Media Player, QuickTime Player, Amino Set-Top Boxes y OpenRTSP [16]. De acuerdo a lo estudiado este servidor cumple con los requisitos especificados.

- Como clientes para el servidor LIVE555 además del VLC, se tiene el OpenRTSP el cual es un programa de línea de comandos que puede ser utilizado para abrir, generar, recibir y opcionalmente grabar flujos multimedia que son especificados por un URL RTSP, por ejemplo, una URL que comience con rtsp:// [17][15].
- Finalmente, como herramienta de codificación se tiene FFmpeg, la cual es un sistema multimedia de fuente abierta que permite decodificar, codificar, transcodificar, multiplexar, demultiplexar, transmitir, filtrar y reproducir archivos de audio y video. El proyecto FFmpeg intenta proveer la mejor solución técnica posible para desarrolladores de aplicaciones y usuarios finales [17].

Para el presente trabajo se han escogido las herramientas libres: LIVE555, VLC, OpenRTSP y FFmpeg, las cuales cuentan con cierto grado de madurez y desarrollo en el trabajo con contenidos multimedia, específicamente en lo que se refiere a: codificación, transmisión y recepción en múltiples plataformas.

### Escenario de Pruebas

En esta sección se presenta el diagrama modular y de implementación extremo a extremo de los diferentes componentes del escenario de pruebas, en el cual se destacan dos módulos fundamentales: el servidor de streaming RTSP y el cliente para el consumo de los contenidos multimedia que hace uso de este protocolo. La comunicación entre el cliente y el servidor se realiza a través de una red inalámbrica, teniendo en cuenta las características de movilidad de los clientes evaluados, ver Figuras 1 y 2.

### Módulo Servidor

Este módulo está formado por el servidor de streaming live555 y los contenidos multimedia empaquetados en el contenedor mpg, ver Figura 3. El servidor de streaming escucha las peticiones RTSP por el puerto 8554 y soporta los siguientes contenedores multimedia: .264, .aac, .ac3, .amr, .dv, .m4e, .mkv, .mp3, .mpg, .ts, vob, .wav, .webm. Los contenidos multimedia utilizados fueron codificados

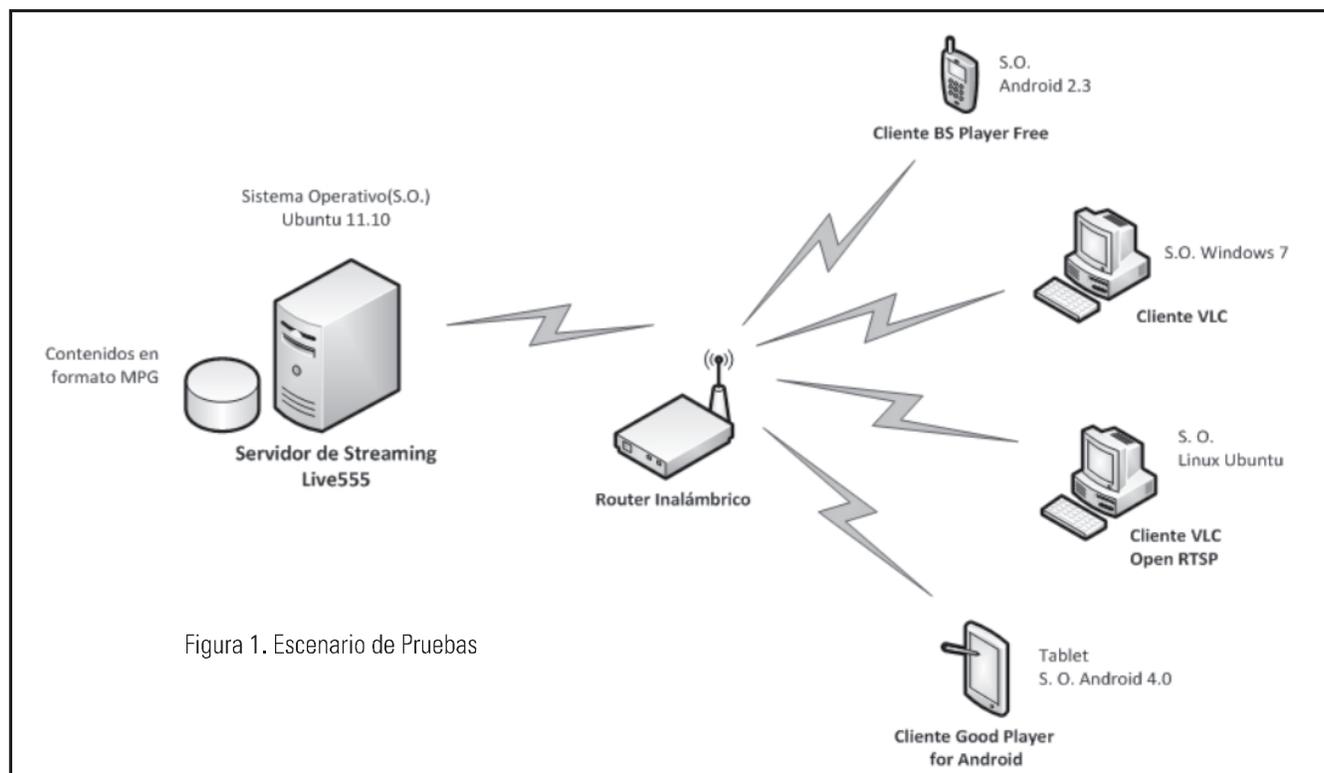
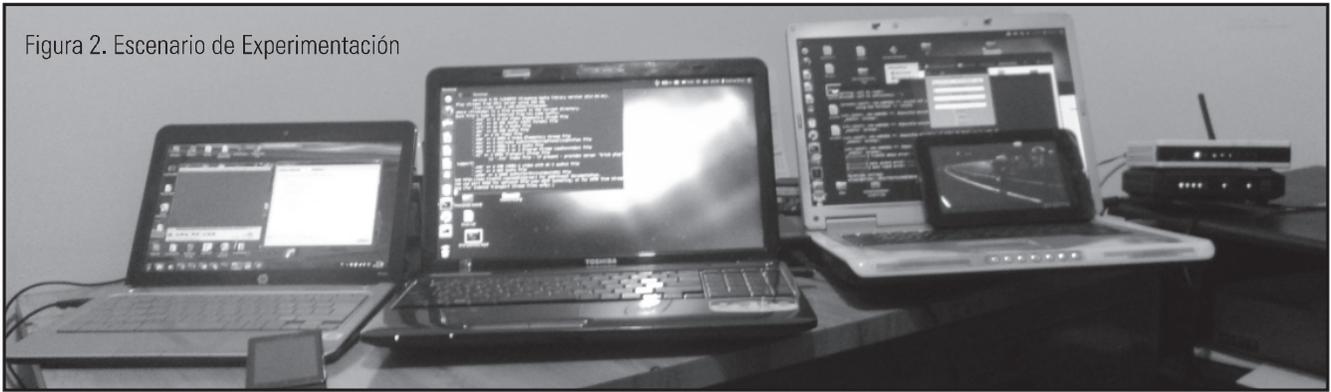


Figura 1. Escenario de Pruebas

Figura 2. Escenario de Experimentación



mediante el códec MPEG-1 (mpga para audio y mpgv para video) usando la herramienta de codificación libre ffmpeg. Este módulo fue implementado sobre un computador AMD Quad Core con un sistema operativo Ubuntu Linux 11.10.

### Módulo Cliente

Este módulo está formado por 3 clientes móviles diferentes: una tablet con sistema operativo Android 4.0, un celular con sistema operativo Android 2.3 y un computador portátil con un sistema operativo Ubuntu Linux 11.10. En la tablet se hace uso del reproductor multimedia gratuito: Good Player For Android [18], el cual soporta el contenedor .mpg y el protocolo de streaming RTSP; en el celular se hizo uso del reproductor multimedia gratuito: BS Player Free [19], el cual soporta el contenedor mpg y el protocolo RTSP.

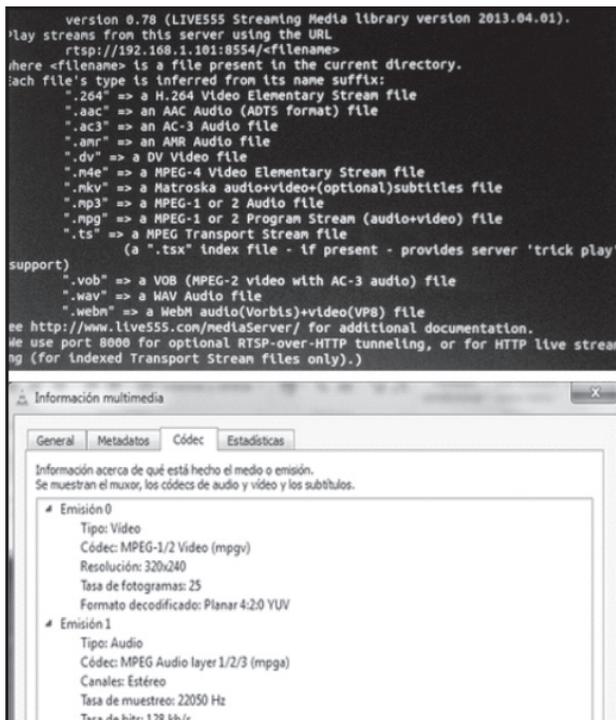


Figura 3. Servidor Streaming y Contenidos Difundidos

Finalmente en el computador portátil se hizo uso de 2 herramientas libres ampliamente difundidas en el campo multimedia: VLC y OpenRTSP. La diferencia básica entre estos 2 clientes es que OpenRTSP es un módulo software que no cuenta con interfaz gráfica y ha sido desarrollado con el objetivo de evaluar la conexión y el consumo de contenidos multimedia mediante el protocolo RTSP, ver Figura 4.



Figura 4. Clientes de Streaming

### Pruebas de uso de memoria

En la Figura 5 se muestran los resultados de las pruebas de consumo de memoria RAM y porcentaje de consumo de CPU, realizadas sobre el servidor live555. Estas pruebas fueron hechas al someter al servidor de streaming a

múltiples conexiones simultáneas, las cuales se generaron al lanzar diferente número de instancias de los clientes libres: OpenRTSP y VLC.

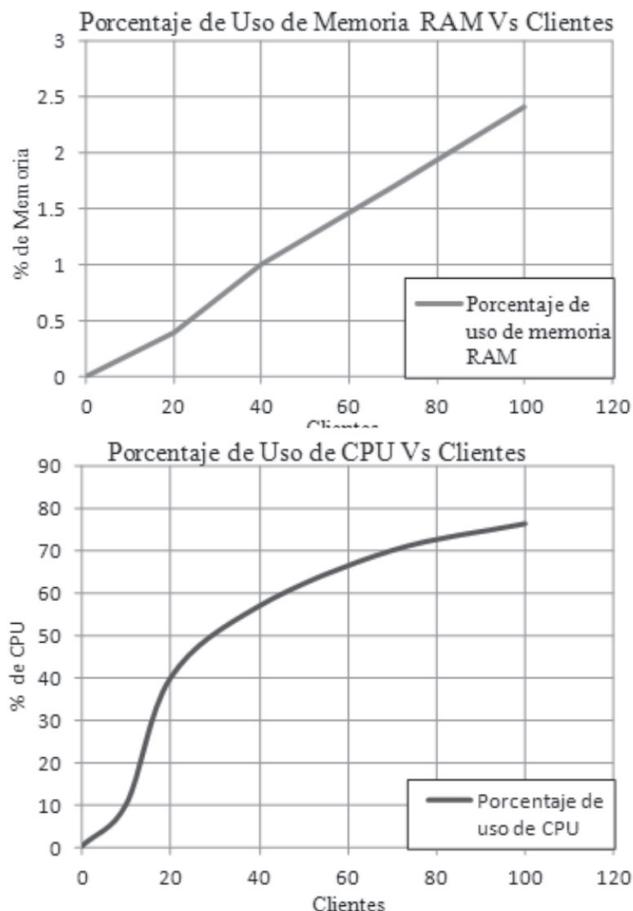


Figura 5. Pruebas de Uso de Memoria

Para lo anterior se desarrolló una herramienta en el lenguaje Python para ser ejecutada en el lado del cliente, cuyo propósito fue lanzar las conexiones de streaming de los clientes VLC y OpenRTSP, usando los conceptos de programación multihilo. A partir de las conexiones simultáneas de los clientes, la medición del consumo de memoria en el lado del servidor, se hizo mediante el comando "ps aux" de Linux, el cual brinda un reporte de la cantidad de memoria RAM y el porcentaje de CPU, que es usado por cada uno de los procesos activos del sistema operativo. Esta información fue filtrada mediante el lenguaje de programación awk (incluido en el sistema operativo Linux), que permite obtener los datos específicos de consumo del proceso live555.

De acuerdo a la Figura 5.a, la relación entre el número de clientes simultáneos y el porcentaje de memoria RAM usada es directamente proporcional, variando

aproximadamente en 0.5 % cada vez que se incrementa en 20 el número de clientes conectados al servidor de Streaming. Por su parte, en la Figura 5.b, se muestra la relación entre el número de clientes simultáneos y el porcentaje de uso de la CPU, tal como se muestra en esta figura la relación es directamente proporcional, teniendo una mayor variación de crecimiento hasta los 40 clientes (incremento en un 60%) y una variación más estable entre los 40 y los 100 clientes (incremento en un 20%).

A partir de las anteriores pruebas se puede concluir que según la tendencia marcada por las gráficas, el porcentaje de consumo de CPU provocado por peticiones superiores a 100 clientes sobre el servidor live555, puede generar problemas de procesamiento en la emisión de los contenidos, causando a su vez dificultades en la recepción adecuada de estos por parte de los clientes.

## Conclusiones y trabajos futuros

El escenario presentado en este trabajo recoge e integra las herramientas más adecuadas del mundo del software libre, para la implementación de servicios basados en videostreaming usando el protocolo RTSP.

El servidor de streaming de video LIVE555 mostró ser un software competitivo para la transmisión de información de video para diferentes dispositivos con distintos sistemas operativos. Su modo de operación está basado en el protocolo de transporte RTSP, por medio del cual vuelca el contenido del video en la red.

A través del software libre FFmpeg es posible editar un video para que este concuerde con las necesidades de transporte deseadas, como lo son formatos y codecs. Además, con FFmpeg se puede empaquetar los archivos de video en flujos de transporte, para que estos puedan ser reproducidos en diferentes dispositivos, como tablets, móviles o computadores.

El escenario de pruebas propuesto busca servir de guía para proyectos que requieran la implementación de servicios basados en videostreaming sobre diferentes entornos de aplicación, como son el sector salud y educación,

Se demostró el correcto funcionamiento del escenario de pruebas con diferentes sistemas operativos y diferentes dispositivos, mediante un único servidor de videostreaming, lo que permite su extensión a otros escenarios.

Como trabajo futuro se pretende considerar dentro del escenario de pruebas, módulos que permitan controlar la calidad de transmisión del contenido multimedia, de acuerdo a las características de la red.

Desarrollar estudios de tráfico que permitan dimensionar las redes, conocer las características telemáticas de los videos y los protocolos que los soportan.

Ampliar el escenario de pruebas para soportar otros códec, tal que permita desarrollar estudios de acuerdos de nivel de servicio, de calidad de servicio y calidad de la experiencia.

## Referencias

- [1] Rijo Sciara, Daniel. Fundamentos de Video Streaming. Facultad de Ingeniería Universidad de la Republica Montevideo – Uruguay. Monografía para el curso de postgrado: “Codificación de Imágenes y Video”. 2004
- [2] Soto, J y Belen, S. Las posibilidades didácticas y manejo de videostreaming en las clases de lengua y literatura. Facultad de Formación del Profesorado. Universidad de Extremadura. España. Revista Tejuelo, ISSN 1988-8430, N°. 4, pgs 84-101, 2008
- [3] Real Time Streaming Protocol Internet-Draft. Internet Engineering Task Force (IETF). Disponible en: <http://ietfreport.isoc.org/allids/draftietf-mmusic-rfc2326bis-03.pdf>
- [4] Mateos, D. y Reaño, S. Streaming de Audio/Video. Protocolo RTSP. Disponible en: [http://ibdigital.uib.es/greenstone/collect/enginy/import/Enginy\\_2007\\_08/Enginy\\_2007\\_08v01p015.pdf](http://ibdigital.uib.es/greenstone/collect/enginy/import/Enginy_2007_08/Enginy_2007_08v01p015.pdf)
- [5] Adobe. Real-Time Messaging Protocol (RTMP) specification, version 1.0. Portal Oficial de Adobe. Disponible en: <http://www.adobe.com/devnet/rtmp.html>
- [6] Campo, W., Arciniegas, J., Garcia, R. y Melendi, D. Analisis de trafico para un servicio de video bajo demanda sobre Redes HFC usando el protocolo RTMP. Universidad del Cauca y Universidad de Oviedo. Revista Informacion Tecnologica[online]. 2010, vol.21, n.6, pp. 37-48. ISSN 0718-0764. Disponible en: [http://www.scielo.cl/scielo.php?pid=S0718-07642010000600006&script=sci\\_arttext](http://www.scielo.cl/scielo.php?pid=S0718-07642010000600006&script=sci_arttext)
- [7] D. Torres, «Evolución y tendencia de la tecnología de streaming en Internet», 2009. [En línea]. Disponible en: <http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/estructura.pdf>. [Accedido: 17-sep-2013]
- [8] A. Acosta, M. S. Garcia-Vazquez, y J. Colores-Vargas, «MPEG-4 AVC/H.264 and VC-1 Codecs Comparison Used in IPTV Video Streaming Technology», en Electronics, Robotics and Automotive Mechanics Conference, 2008. CERMA '08, 2008, pp. 122-126.
- [9] S. Kim y Y. Yoon, «Video Customization System Using Mpeg Standards», en International Conference on Multimedia and Ubiquitous Engineering, 2008. MUE 2008, 2008, pp. 475-480.
- [10] T.-C. Lin, C.-W. Chen, C.-C. Lin, y T.-K. Truong, «Very Low Bit Rate Video Coding Using H.264 Codec and Cubic Spline Interpolation», en Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2007. IIHMSPP 2007, 2007, vol. 1, pp. 11-14.
- [11] M. A. R. Mustafa, M. M. Elahi, M. A. Hossain, y M. M. Islam, «Comparative performance analysis of MPEG4, FLV and 3GP multimedia file formats using wireless network parameters», en 2010 13th International Conference on Computer and Information Technology (ICCIT), 2010, pp. 306-311.
- [12] Z. Hao, C. Guang-Li, y C. Hua-Xiang, «Performance Improvement of DSS Based on High-Definition Video on Demand», en 2012 International Conference on Computer Science Service System (C3SS), 2012, pp. 761-764.
- [13] [En línea]. Disponible en: <http://www.catrasoftware.it/Streaming/CatraStreamingPlatform.htm>. [Accedido: 17-sep-2013].
- [14] L. Wang y C. Meinel, «Mining the Students' Learning Interest in Browsing Web-Streaming Lectures», en IEEE Symposium on Computational Intelligence and Data Mining, 2007. CIDM 2007, 2007, pp. 194-201.
- [15] F. Delgado, I. Quintana, J. Rufo, J. A. Rabadan, C. Quintana, y R. Perez-Jimenez, «Design and Implementation of an Ethernet-VLC Interface for Broadcast Transmissions», IEEE Commun. Lett., vol. 14, n.o 12, pp. 1089-1091, 2010.
- [16] N. Vun y M. Ansary, «Implementation of an embedded H.264 live video streaming system», en 2010 IEEE 14th International Symposium on Consumer Electronics (ISCE), 2010, pp. 1-4.
- [17] J. Bailey, «Live Video Streaming from Android-Enabled Devices to Web Browsers», University of South Florida, 2011.
- [18] Good Player For Android, <https://play.google.com/store/apps/details?id=com.hustmobile.goodplayer>

