

# Streaming de Video en Vivo por Internet

Streaming on Live for Internet

Esequiel Zavala Huavel<sup>1</sup>, Lita Soto Nieto<sup>2</sup>, Luis Martínez Ulloa<sup>3</sup>, Henry Vivanco Urco<sup>4</sup>

*Facultad de Ingeniería Electrónica y Eléctrica, Universidad Nacional Mayor de San Marcos, Lima Perú*

**Resumen**– El streaming se refiere a la entrega de flujos de datos, generalmente con contenidos multimedia. Ello trae ventajas que no son posibles utilizando procedimientos comunes de descarga de ficheros. Así como se realiza el streaming de contenidos estáticos, puede realizarse el Streaming de contenido en vivo..

**Abstract**– Streaming refers as delivery of data streams, generally multimedia content. This approach have advantages impossible by common procedures like file download. As making static content streaming, can be addressed live content streaming.

**Palabras clave**- Streaming en vivo, video, códec, contenedor, multimedia, FireWire, webcam, plataforma de enseñanza, videolan, VLC.

**Keywords**- Streaming on Live, video, codec, container, multimedia, FireWire, webcam, learning plataform, videolan, VLC.

## I. INTRODUCCIÓN

Se define el streaming como la distribución de audio o vídeo por Internet de forma continua (sin interrupción) y sin necesidad de ninguna descarga previa [1]. En el presente trabajo veremos en forma práctica como llevar a cabo este objetivo usando herramientas multiplataforma de software libre.

Para transmitir video a través de Internet puede lograrse de forma sencilla siguiendo este material que puede servir como una guía.

Las instituciones educativas hacen uso de medios diversos como el vídeo, por tanto el streaming puede ampliar y mejorar la prestación de conocimientos. Ya se viene usando el streaming en diversos eventos educativos, televisando en directo, lo cual permite seguirlos a distancia en el mismo momento en el que se

está realizando, con la ventaja que supone el poder interactuar con la teleaudiencia a través de algún medio de mensajería (como el servicio Twitter), realizando preguntas o comentando las distintas exposiciones.

La tecnología de streaming se utiliza para aligerar la descarga y ejecución de audio y vídeo por Internet, ya que permite escuchar y visualizar los archivos mientras se están descargando [2].

Si no utilizamos streaming, para mostrar un contenido multimedia en la red, tenemos que descargar primero el archivo entero en nuestro ordenador y más tarde ejecutarlo, para finalmente ver y oír lo que el archivo contenía. Sin embargo, la técnica del streaming permite que esta tarea se realice de una manera más rápida y que podamos ver y escuchar su contenido durante la descarga.

Existen otras ventajas que se pueden obtener de la difusión de medios mediante streaming, como:

- Saltar adelante o atrás sin haber descargado todo el stream gracias a la descarga progresiva en HTTP.
- Perfilado del ancho de banda de la descarga: Con esto logramos que la descarga consuma poco más de la tasa de bits del medio, para optimizar los recursos de ancho de banda.
- Adaptación en vivo de la tasa de bits, calidad y tamaño de los medios: en caso que el ancho de banda disponible lo permita, pudiendo conmutarse a un flujo del mismo medio original pero codificado a otro bitrate o en otra relación de aspecto para mejorar de la experiencia del usuario.

Para entregar video puede hacerse uso de diversas fuentes, como pueden ser archivos de video, medios (DVD, BlueRay), tarjetas de captura de video, equipos de video USB (receptores de televisión, webcams), equipos de video IEEE1394 (cámaras de video).

La conexión FireWire para Video Digital o DV, IEEE 1394, fue introducida por Apple y sirve para transmitir datos a alta velocidad entre equipos digitales. Tradicionalmente las cámaras y equipos como laptops incluyen ya estas conexiones FireWire, en el caso de las PC de escritorio pueden traer el hardware de FireWire integrado o bien como una

<sup>1</sup> Esequiel Zavala Huavel, E-mail: ezavalah@unmsm.edu.pe

<sup>2</sup> Lita Soto Nieto, E-mail: lsoton@unmsm.edu.pe

<sup>3</sup> Luis Martínez Ulloa, E-mail: lmartinez@ecologic-peru.com

<sup>4</sup> Henry Vivanco Urco, E-mail: hvivanco@gmail.com

Recibido: Abril 2011 / Aceptado: Junio 2011

tarjeta opcional con terminales o conectores tipo FireWire, tal como se muestra en la Fig.1.

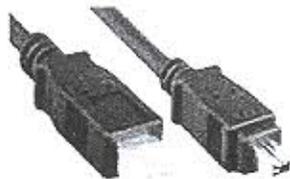


Fig.1. Conectores FireWire.

La obvia ventaja que tiene el FireWire sobre el video analógico es la pérdida nula de calidad entre la cámara y los procesos subsiguientes.

En este caso vamos a emitir contenido dentro de la red institucional, por lo que tenemos un ancho de banda suficiente para el streaming. Es importante entender que a mayor audiencia de nuestra transmisión, mayor deberá ser nuestra capacidad de transmisión. Si la transmisión será únicamente para la red local, no es necesario el IP público y bastará con el IP privado que tiene el equipo para conectarse a la red para que los demás usuarios puedan recibir la señal. Podemos tener ciertas limitaciones respecto a los equipos de comunicaciones y los filtros de internet.

En principio no es necesario contar con un servidor especial para colocar archivos de audio o vídeo con descarga streaming. Cualquier servidor normal puede mandar la información y es el cliente el que se encarga de procesarla para poder mostrarla a medida que va recibiendo, ver Fig.2.

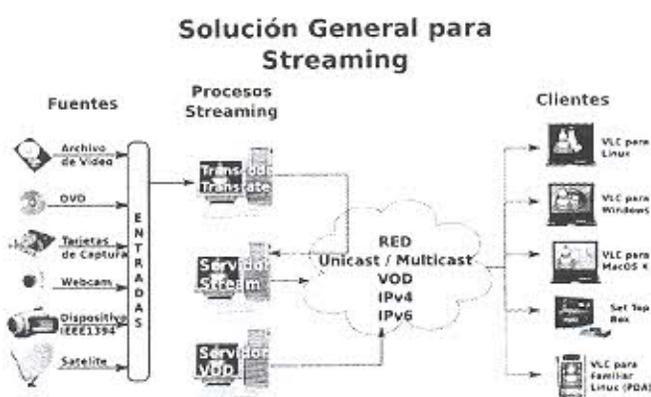


Fig.2. Flujo de datos en el Streaming

Sin embargo, existen servidores especiales preparados para transmitir streaming. Aunque en muchas ocasiones no es necesario utilizarlos nos pueden ofrecer importantes prestaciones como mandar el medio a mayor o menor calidad dependiendo de la velocidad de nuestra línea.

En determinados casos, como la puesta en marcha de una radio o la transmisión de un evento en vivo a través de la Internet, será imprescindible contar con un servidor de streaming con más ancho de banda, al que mandaremos la señal y este, enviará a todos los clientes.

## II. ANTECEDENTES

En el desarrollo de este trabajo se ha utilizado aplicativos y sistemas de Software Libre. Técnicamente se pudo haber aplicado alguna solución propietaria, pero reduce la cantidad de usuarios potenciales, por lo que la solución encontrada debía evitar este inconveniente. Otra ventaja es que podemos prescindir del monitor, ya que tanto la captura, la compresión como el streaming en si no implican necesariamente la visualización en el origen, en tanto que si lo soporta. Esto es, una solución flexible. Tan flexible que podemos activarlo usando la línea de comandos para automatizar el proceso.

Se inició con la transmisión de video en vivo usando una webcam USB, y enviando el flujo de audio y video sin codec, sin contenedor, usando el protocolo UDP, a todo el segmento de red local, usando la dirección de broadcast como origen para el servidor; los clientes se conectaron a la dirección de broadcast para recibir el flujo. Este enfoque consume demasiados recursos de la red, además que es filtrado rápidamente por los equipos de red; por lo tanto, no puede usarse en un entorno de producción.

Una vez desarrollado el primer paso, tenemos suficiente para abordar otro tipo de dispositivo de video, la videocámara con conexión Firewire o IEEE1394, éste con mayor resolución, ratio de fotogramas, calidad, ancho de banda y una conexión distinta. Aprovechamos algunos aplicativos que facilitaban la conexión entre los datos en bruto FireWire y los dispositivos compatible con la interfaz "Video For Linux" (V4L y V4L2). Específicamente, el programa Dvgrab. Para ello, debemos conocer la forma de activar VLC mediante la línea de comandos usando todos sus parámetros.

## III. MARCO TEÓRICO

En el streaming, la información es procesada en tiempo real, pasando por diferentes procesos sucesivos y en algunos casos simultáneos; este flujo de datos se ilustra en la Fig. 3.

- El flujo de audio en bruto es comprimido usando un codec de audio como MP3, Vorbis o AAC. Son formatos del codec

- El flujo de video en bruto es comprimido usando un codec de video como H.264 o WebM
- Los flujos de audio y video codificados son ensamblados en un flujo de bits contenedor como FLV, ASF o MOV(son formatos de contenedores)
- El flujo de bits es enviado desde un servidor de streaming a un cliente usando un protocolo de transporte, como HTTP, MMS o RTP
- El cliente de streaming puede interactuar con el servidor de streaming usando un protocolo de control, como MMS o RTSP



Fig. 3. Flujo de datos entre procesos del Streaming

El streaming involucra 3 conceptos básicos interrelacionados para su óptimo rendimiento en un entorno determinado.

#### A. Estructura del flujo de datos

La información dentro del flujo tiene una estructura compuesta por los objetos siguientes:

##### 1. Encapsulado o Contenedor

Un encapsulado o contenedor, es un formato que permite almacenar simultáneamente datos de distinto tipo en forma multiplexada. En el caso del video digital, un contenedor es un formato que incluye uno o más canales de video y audio (a veces subtítulos). Algunos contenedores son exclusivos de ciertos codecs. Los contenedores más conocidos son Mpeg, Ogg, MOV, AVI, MP4, Matroska, WebM.

##### 2. Codec

Son los algoritmos que permiten codificar y decodificar tanto el video como el audio que está dentro del contenedor, es decir, el formato en el que se van a guardar o guardado estos datos. Estos codecs comprimen la información para hacerla manejable dentro de los parámetros de ancho de banda, aún contemplando cierta pérdida de fidelidad. Los codecs de video más conocidos son DivX, XviD (dentro del estándar MPEG-4), MPEG-1 (VCD) y MPEG-2 (SVCD y DVD). Los codecs de mayor uso de audio son MP3, OGG y AC3.

##### 3. Frame

Los codecs separan el flujo en unidades llamadas frames o marcos. Cada frame puede tener características diferentes respecto a la tasa de bits o bitrate, según el códec busca optimizar el ancho de banda cuando la información que porta pueda comprimirse en mayor medida que el resto, por poseer más información redundante.

#### B. Los Protocolos de Transporte

Los protocolos de datagramas, como UDP, envían los flujos como series de paquetes pequeños. Esto es simple y eficiente; sin embargo, no hay un mecanismo dentro del protocolo para garantizar la entrega. Corresponde al aplicativo receptor detectar pérdidas o corrupción y recuperar la data usando técnicas de corrección de errores. Si la data se pierde, la recepción del stream puede sufrir una caída.

El Protocolo de Streaming en Tiempo Real (RTSP), el Protocolo de Transporte en Tiempo Real (RTP) y el Protocolo de Control de Transporte en Tiempo Real (RTCP) fueron específicamente diseñados para hacer streaming sobre redes. RTSP se ejecuta sobre una variedad de protocolos de transporte, mientras los dos últimos se constituyen sobre UDP.

Otro enfoque que parece incorporar tanto las ventajas de utilizar un protocolo de Internet estándar y la capacidad de ser utilizado para el streaming de contenido en vivo, es el streaming con bitrate adaptivo sobre HTTP. Está basado en la descarga progresiva sobre HTTP, pero contrariamente al enfoque previo, aquí los frames son muy pequeños, de tal modo que pueden ser comparados al streaming de paquetes, tanto como en el caso de usar RTSP y RTP [3].

Los Protocolos confiables, como el Protocolo de Control de Transmisión (TCP) garantiza la entrega correcta de cada bit del flujo del medio. Sin embargo, logran esto con un sistema de reintentos y tiempos de espera, que los hace más complejos de implementar. Los clientes pueden minimizar este efecto precargando data de despliegue [4].

Los protocolos Unicast mandan una copia separada del flujo de datos del servidor a cada destinatario. Unicast es la norma para muchas conexiones de Internet, pero no crece bien cuando muchos usuarios quieren ver el mismo programa concurrentemente.

Los protocolos Multicast fueron desarrollados para reducir la replicación de los datos. Dependiendo de la infraestructura de red y tipo, la transmisión multicast puede o no ser factible. Una desventaja potencial del multicasting es la pérdida de la funcionalidad de video en demanda. El flujo constante de material de radio o televisión usualmente se opone a la habilidad del destinatario de controlar la reproducción. Sin embargo, este problema puede ser mitigado por elementos como

servidores de caché, Decodificadores Set-top Box y Appliances y reproductores de medios con buffer.

Multicast IP provee formas de enviar un único flujo de medios a un grupo de destinatarios en una red de computadores. Un protocolo multicast, usualmente IGMP se usa para administrar la recepción de flujos Multicast a los grupos de destinatarios en una red local. Uno de los retos en la implementación de Multicast IP es que los ruteadores y firewalls entre redes locales deben permitir el paso de paquetes destinados a grupos Multicast. Si la organización que sirve el contenido tiene control sobre la red entre el servidor y destinatarios (por ejemplo intranets educativas, de gobierno y corporativas), los protocolos de ruteo como Protocol Independent Multicast, pueden ser usados para entregar flujos de contenido a múltiples segmentos de red.

Los protocolos P2P (de par a par o de iguales) sirven para flujos pregrabados a ser enviados entre computadoras. Esto evita que el servidor y sus conexiones de red se conviertan en cuellos de botella. Sin embargo, ello aumenta los problemas técnicos, de rendimiento y calidad, como ha quedado visto en la práctica.

C. Transcodificación

Usualmente se parte con el flujo de datos de audio y video en bruto, para codificarlo según un codec particular. Hay casos en los que se parte con un flujo de medios ya codificado con un codec origen y se debe lanzar el contenido usando otro codec destino o a un bitrate distinto, por lo general menor, ver Fig.4. Esta operación es necesaria cuando hay un servidor de streaming intermediario entre el flujo original y el destino, además de existir una dificultad respecto al ancho de banda del flujo original. De ese modo funcionan los servidores públicos de streaming que poseen un gran ancho de banda y una infraestructura informática que no podríamos abordar fácilmente. También es útil cuando nos dirigimos a un público especial (como los Smartphone) donde la recepción es más probable usando un codec en particular.



Fig. 4. Transcodificación de un flujo de datos.

En la práctica, el estándar adoptado por la industria es el uso de codecs como H.264 y MP3 en contenedor MP4 sobre HTTP, con apoyo de redes de despliegue de contenidos (CDN). Ha sido elegido fundamentalmente por su bajo ancho de banda y buena calidad de visualización. Se usa masivamente para la difusión de streaming por páginas como Youtube. Por tanto, lograr trabajar con estos formatos es el objetivo final del presente trabajo.

IV. METODOLOGÍA DE IMPLEMENTACIÓN

Hemos utilizado para el lado del Servidor, el programa VLC versión 1.1.4, funcionando sobre Ubuntu Linux Desktop Edition versión 10.10 Maverick, webcam USB y videocámara con Firewire como fuentes de video en vivo. Para los clientes, hemos usado el programa VLC 1.0.6 en Ubuntu Desktop 10.04 y 1.1.4 en Ubuntu Desktop 10.10 y en Windows XP Profesional SP2.

Al iniciar el programa elegimos la opción del menú "Medio->Emitir", Fig. 5.

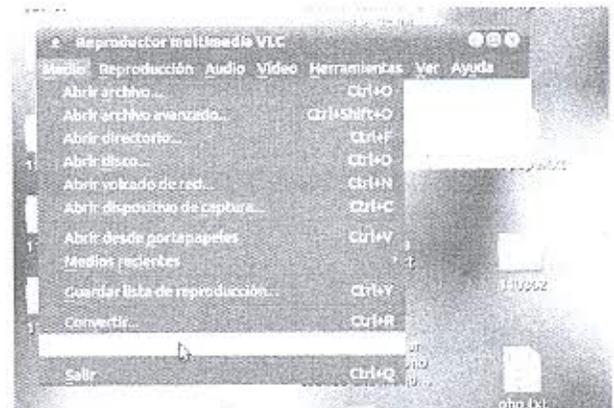


Fig. 5. Menú Medio.

Escribimos la ruta del archivo del dispositivo de video, en este caso, es "/dev/video0" y presionamos el botón "emitir", Fig. 6.



Fig. 6. Dialogo abrir medio.

A partir de aquí se nos presenta un asistente con los datos de configuración, Fig. 7.



Fig. 7. Asistente "salida de emisión"

En la pestaña destinos debemos elegir el protocolo HTTP. Si marcamos la opción "mostrar en local", el video se presentará en el servidor mientras se emite a la red, ver Fig. 8.



Fig. 8. Opción "destinos".

Añadimos HTTP como destino. Dejamos el número de puerto en 8080 y en la ruta ponemos la que deseamos por ejemplo "/video". Ver Fig. 9.

En las opciones de transcodificación, agregaremos un perfil al que llamaremos "MP4 H264 MP3", ver Fig. 10.

Elegimos el encapsulamiento MP4. En la pestaña video activamos la casilla correspondiente y elegimos el Codec H.264 como en la Fig. 11.



Fig. 9. Opciones de HTTP.

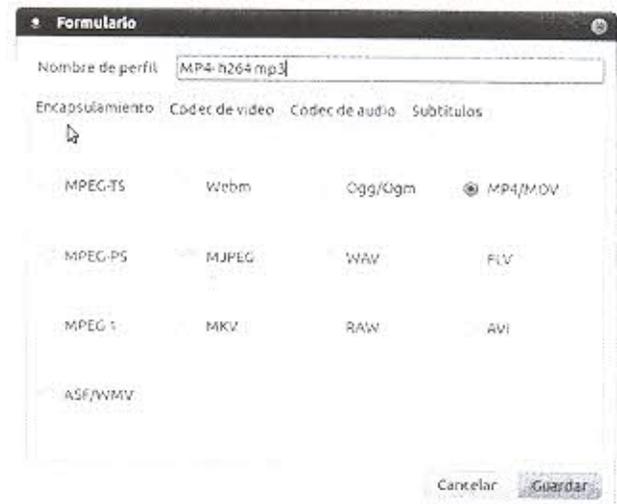


Fig. 10. Creación de nuevo perfil.

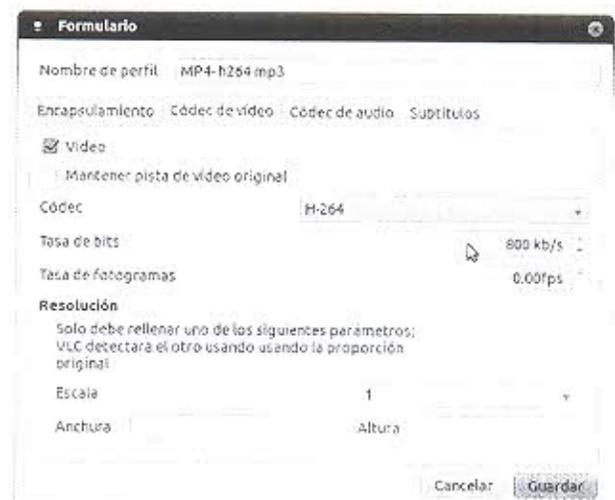


Fig. 11. Código de Video.

Luego en la pestaña de código de audio habilitamos esta característica y elegimos el Código MP3. En ese punto hacemos clic en "Guardar", Fig. 12.

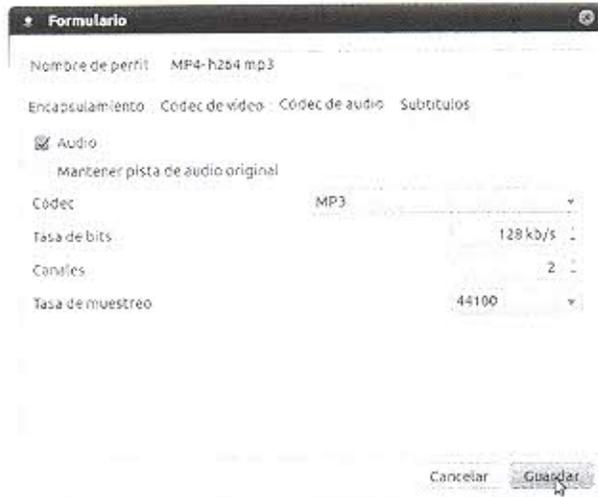


Fig. 12. Código de Audio.

Este perfil guardado es el que luego utilizamos para transferir el flujo, con los parámetros de transcodificación adecuados para el formato MP4. Dicho formato es un estándar de facto, lo cual nos asegura compatibilidad con otros reproductores y equipos. Aplicamos este perfil y hacemos clic en el botón emitir, resultando en la emisión del stream, como en la Fig. 13.



Fig. 13. Emitiendo el stream

Para recibir el flujo en otro equipo, abriremos una nueva instancia de VLC. Usamos la opción del menú “Medio->Abrir volcado de red”, como en la Fig. 14.

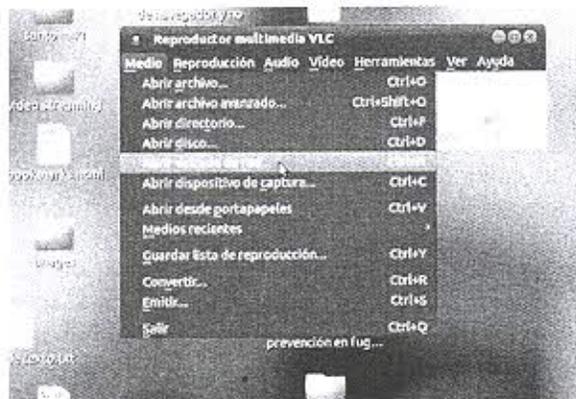


Fig. 14. Abrir volcado de red.

Ingresamos el URL de la transmisión, en este caso, http que es el protocolo que elegimos, 172.16.36.138

es el número IP del servidor, 8080 es el puerto de transmisión y "/video" es la ruta del medio transmitido. Ver Fig. 15.

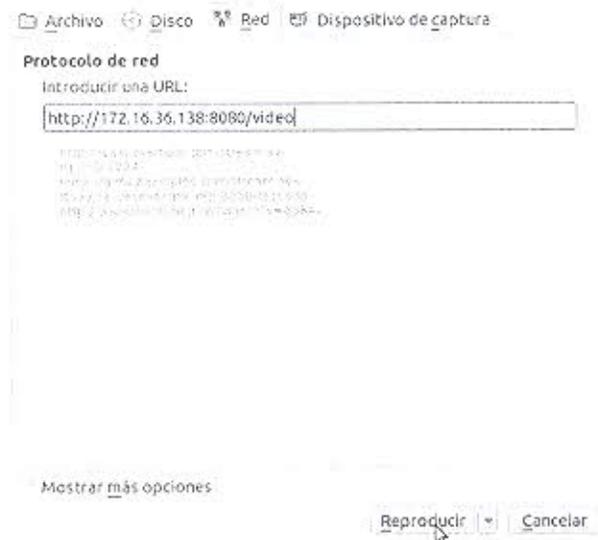


Fig. 15. Abrir volcado de red.

En la Fig. 16 se muestra la salida del stream desde otro equipo. Esto fue realizado usando la Interfaz Gráfica de Usuario [5]. Sobre esta base podemos abordar el streaming del dispositivo Firewire.



Fig. 16. Recepción del flujo de la webcam.

En el caso de la videocámara FireWire, nos conectamos al dispositivo “/dev/raw1394” usando el programa “dvgrab” y de la salida estándar de este redireccionamos a la entrada estándar VLC. Este se sirve de esta entrada para realizar la codificación y difundir el flujo.

El programa dvgrab toma el flujo de la entrada FireWire y genera un flujo de audio y video en formato DV [6], que puede almacenarse o bien redireccionarse como en este caso, a la entrada del VLC.

La línea de comando necesaria entonces implica dos cosas: redireccionar el flujo al programa y procesar la data. Esto se logra con lo siguiente:

```
dvgrab-|vlc---no-sub-autodetect-
file:demux=rawdv-sout
'#transcode{vcodec=mp4v,acodec=mpga,vb=2
000,ab=128}:standard{access=http,mux=ogg
,dst=172.16.36.138:8080}'
```

Se le está indicando al sistema que transfiera el flujo al programa que a continuación codificará y emitirá los datos. A continuación abrimos VLC en otro equipo, direccionando la escucha como en el caso anterior. Con esto se consiguió la emisión y recepción deseada, Ver Fig. 17.



Fig. 17. Recepción de flujo HTTP de la videocámara.

corriente porque se trata de un sistema que no requiere grandes conocimientos técnicos para su manipulación, ya que su manejo es simple, transparente y porque su costo empieza a ser asequible.

En el campo universitario, un sistema de streaming proporciona importantes beneficios como una mayor integración entre grupos de trabajo y cuerpos académicos de distintos campus beneficiando tanto a investigadores, funcionarios y usuarios externos.

El personal administrativo, los estudiantes, así como a la comunidad en su conjunto tendrán acceso a recursos pedagógicos de la propia institución o de cualquier parte del mundo.

La producción y realización de video streaming requiere de equipos multidisciplinarios con conocimientos básicos en informática, producción de audio y vídeo y aspectos de pedagogía, particularmente en educación a distancia. En la elaboración de los materiales gráficos, documentales o electrónicos de apoyo a las presentaciones es necesaria la planeación adecuada y con el tiempo suficiente que permita una realización de alta calidad.

No cabe duda que la transmisión de contenido multimedia a través de la web será cada vez más importante. La tecnología de streaming es un mercado con futuro y grandes compañías ya están luchando por el mercado. La velocidad de Internet aumentará con el tiempo y con ella aumentará la calidad de las transmisiones. Si bien la radio y la televisión por Internet ya existen y son viables, aún falta cerrar la brecha digital que permitirá que todas las personas puedan tener a su alcance el streaming.

## V. CONCLUSIONES

El fenómeno de la educación a distancia comienza a cobrar un desarrollo importante desde la utilización del video y muy recientemente la telemática como medios pedagógicos. Sin embargo sus orígenes se pueden encontrar en las necesidades individuales y profesionales y en las aspiraciones de crecimiento educativo y social.

El streaming ha pasado de ser una tecnología cara y exclusiva de grandes instalaciones a convertirse en una herramienta multimedia más.

Se espera que en los siguientes años, la utilización de esta tecnología, juntamente con otros sistemas multimedia, se introduzca en las universidades, en los centros de enseñanza a distancia, en las escuelas e institutos de formación profesional y jugará un papel importante en la formación continua en las empresas. Además, permitirá la enseñanza personalizada a domicilio. El uso de este sistema se hará cada vez más

## REFERENCIAS

- [1] Antonio Delgado. "Qué es el streaming". Fundación Eroski. 2009.  
<http://www.consumer.es/web/es/tecnologia/internet/2009/09/18/187866.php>
- [2] Juana Díaz. "Televisión y streaming educativos". Educ@ConTIC. 2010  
<http://www.educacontic.es/blog/television-y-streaming-educativos>
- [3] Ch. Z. Patrikakis, N. Papaoulakis, Ch. Stefanoudaki, M. S. Nunes, "Streaming content wars: Download and play strikes back". pp 218 – 226, Venice, Italy, 2009.
- [4] Krasic, C. and Li, K. and Walpole, J. "The case for streaming multimedia with TCP", pp 213--218, Springer, 2001.
- [5] Alexis de Lattre, Johan Bilien, Anil Daoud, Clement Stenac, Antoine Celleir, Jean-Paul Saman.

“Documentation: Streaming How To New”.  
VideoLan.2006.  
[http://wiki.videolan.org/Documentation:  
Streaming\\_HowTo\\_New](http://wiki.videolan.org/Documentation:Streaming_HowTo_New)

- [6] Daniel Kobras. “dvgrab”. Debian GNU/Linux system. Center for Computer Research in Music and Acoustics. Stanford University, 2006.  
<http://ccrma.stanford.edu/planetccrma/man/man1/dvgrab.1.html>