

# "Transmitiendo video sobre Internet usando software libre"

Lic. Javier Díaz, [javierd@info.unlp.edu.ar](mailto:javierd@info.unlp.edu.ar)

Lic. Claudia Banchoff Tzancoff, [cbanchof@info.unlp.edu.ar](mailto:cbanchof@info.unlp.edu.ar)

LINTI, Facultad de Informática, Universidad Nacional de La Plata

## Introducción

Internet se ha convertido en un nuevo medio de comunicación. El correo electrónico, las salas de chat y la mensajería instantánea, prácticamente, en algunos entornos, han reemplazado a la comunicación telefónica y, mucho más aún, al correo postal incluyendo han posibilitado una nueva modalidad de trabajo a distancia usando videoconferencias y herramientas colaborativas..

A través de Internet las personas no sólo intercambian información textual e imágenes, sino que, en la actualidad es común escuchar y ver conferencias, que radios y programas de TV emitan su programación por la web e inclusive, en las salas de chat las personas pueden transferir su voz o su imagen a través de las webcam o control remoto de aplicaciones robotizadas, etc.

Si bien es común incluir en sitios o aplicaciones web información multimedial, especialmente audio y video, su uso requiere un conocimiento más profundo sobre algunos aspectos que tiene que ver con la transferencia sobre redes de archivos que, por lo general, suelen ocupar gigabytes y sobre las distintas alternativas de aplicaciones existentes.

Una de las bases sobre la cual se sostiene Internet es la utilización de estándares abiertos. Así es posible consultar los conocidos RFC[1] donde se encuentran las especificaciones de los protocolos que permiten el intercambio de información de casi todos los datos que recorren la red.

Aunque existen muchas formas de almacenar archivos de video en un disco y transferirlos por una red, muchos utilizan formatos propietarios que requieren de una aplicación especial (y propietaria) para visualizarlos y transmitirlos.

La intención de este trabajo es plantear una alternativa basada en código abierto[2] que utiliza formatos estándares para manejar video sobre Internet. Se analizan algunos productos existentes que permiten tanto la reproducción de video como el envío y almacenamiento de los mismos.

## Transmisión de video

Como un video se refiere a visualización de imágenes en movimiento, existen diferentes formas de considerar la fuente de información de video:

- Fuentes en vivo o en tiempo real, donde los datos son capturados, transportados y visualizados mientras el evento está ocurriendo;
- Fuentes pregrabadas y almacenadas en el servidor, el cliente o en ambos

Una arquitectura para un servidor de video debería permitir varias formas de creación de los contenidos y varias formas de distribución, ya sea en tiempo real o en un modo "store and forward".

También es necesario contemplar las técnicas de concentración de datos, tales como compresión, que, como mencionará más adelante, permiten optimizar el transporte cuando hay enlaces lentos, obtener mayor calidad o acomodarse a restricciones de almacenamiento en distintos lugares.

Aún hoy, con los grandes avances realizados (tanto en software como en hardware) existen dos grandes obstáculos para el desarrollo de estas nuevas tecnologías:

La computación distribuida: Las redes, hoy en día son cada vez más veloces y los procesadores que las integran tienen alto poder de cómputo. Especialmente con el video, la cantidad de datos que se deben mover a través de la red es muy grande. Las WAN no son siempre confiables y el ancho de

banda disponible para usuarios individuales muchas veces es insuficiente para la mayoría de las aplicaciones de video. Además hay que tener en cuenta los algoritmos de ruteo, los cuellos de botella inesperados y la posibilidad de que los streams de datos de video digitalizado sean demorados y lleguen fuera de secuencia.

Manejo de contenido: Un problema particular con el video es la naturaleza de los archivos de video mismos. Los archivos de video por definición, son muy grandes. El problema es manejar estas cantidades de información. Por ejemplo, la mayoría de las bases de datos existentes basadas en el modelo relacional no están diseñadas para almacenar registros del orden de gigabytes. Por el lado del cliente, hay muy pocas PCs que tengan gigabytes de espacio libre en disco para "bajar" y almacenar un archivo de video. Al igual si tenemos en cuenta mover esta gran cantidad de información a través de la red.

Cuando se va a incluir este tipo de medio en una aplicación (sea un sitio web o una aplicación web), los puntos anteriormente mencionados deberán tenerse en cuenta de modo tal de plantear una arquitectura para un servidor de video adecuada.

La base de cualquier arquitectura para un servidor de video es un sistema de computación abierto y distribuido. Un servidor de video debe ofrecer alta confiabilidad, disponibilidad y escalabilidad. Debe soportar distintos tipos de redes, incluyendo Internet, y debe manejar y controlar grandes volúmenes de información compleja.

La arquitectura de Internet y sus protocolos asociados han removido una barrera creando una gran infraestructura de transporte disponible y muy bien publicitada. Los protocolos de red sofisticados proveen herramientas que permiten integrar información de video en documentos, mensajes y transacciones interactivas.

Las capacidades de los sistemas de red actuales necesitan que los clientes (PCs, por ejemplo) tengan cierto grado de procesamiento de gráficos que las hace un equipo más costoso.

Un servidor de video es un sistema mutimedial distribuido especializado cuyo propósito es la colección, almacenamiento, distribución y presentación de cuadros en movimiento. Los dos esquemas básicos para un servidor de video son:

"Download and play": En este esquema el archivo de datos completo debe ser transferido al destino antes que la información sea visualizada. El retraso en la transferencia de un archivo tan grande (el cual puede ser muy grande en líneas dial-up) plantea una situación a tener en cuenta si el usuario está esperando para comenzar a ver el video. En este caso, el cliente también requiere una gran cantidad de almacenamiento local para guardar estos archivos. "Download and play" es una técnica que requiere que videos previamente archivados sean bajados en el cliente antes que puedan ser vistos. Esto es justamente una desventaja ya que no existe la posibilidad de una pre visualización antes de la transferencia completa. Una ventaja es que el video transferido y almacenado puede ser visto las veces que sean necesarias sin volver a realizar la transferencia.

"Streaming": En esta técnica la señal de video es transmitida al dispositivo cliente y la presentación del video comienza después de un retraso momentáneo para almacenar los datos en un buffer. Este esquema reduce el tiempo para comenzar a ver el video y permite una presentación en tiempo real. También elimina la necesidad de gran capacidad de almacenamiento en el cliente. La transmisión efectiva de estas señales de video sobre una tasa de ancho de banda baja requiere una alta tasa de compresión para compensar la calidad de la presentación. El manejo y almacenamiento de los datos desde su origen al destino es minimizado. La técnica de streaming primero fue utilizada en aplicaciones de audio sobre Internet y fueron bastante aceptables aún en redes de baja velocidad. Como una ventaja importante y destacable se encuentra la reducción de los requerimientos de almacenamiento pero el más importante beneficio, para el usuario, es la posibilidad de visualizar el video en tiempo real, evitando largas esperas.

Streaming es en este momento la técnica más popular y aceptada para distribuir señales de video sobre redes de computadoras. Las aplicaciones basadas en video son aquellas que involucran la distribución de la señal de un servidor de medios a uno o más destinos (uno a uno o uno a muchos). Cualquier arquitectura de video streaming debe incluir creación o adquisición de la imagen, manejo y formateo (incluyendo compresión) del contenido, distribución y presentación (visualización) y control del visualizador.

Sin importar el mecanismo empleado para transmitir la información, el intercambio de datos continuos entre distintas computadoras utilizando redes de datos requiere servicios y protocolos apropiados para una transmisión exitosa. Existen varios requerimientos que deben exigirse tanto sobre los servicios como sobre los protocolos de red que son independientes del nivel en la arquitectura de la red. En general, se debe tener en cuenta que:

- el procesamiento de audio y/o video requiere de limitaciones importantes de tiempo. Es crucial que no exista demasiado retraso.
- el "jitter end-to-end" debe estar limitado. Esto es especialmente importante en aplicaciones interactivas (por ejemplo, la telefonía) donde si los valores del jitter son muy grandes se deberán emplear buffer muy grandes y retardos muy altos (delay).
- para la transferencia de los datos se debe pedir una buena performance de procesamiento y transferencia sobre el bus, y almacenamiento disponible para procesar el protocolo.
- si la aplicación es una aplicación colaborativa donde se transfieren datos multimediales, el sistema debería soportar conexiones multicast para ahorrar recursos
- los servicios deberían proveer mecanismos de sincronización
- los protocolos usados para la transmisión deberían coexistir con los protocolos de red usados actualmente para la transmisión de cualquier tipo de información.

Para lograr esto han surgido una serie de protocolos que ayudan al esquema tradicional en las cuestiones donde los protocolos básicos son inadecuados. Entre estos protocolos se puede mencionar: Resource Reservation Protocol (RSVP), Real Time Streaming Protocol (RTSP - RFC 2326), Real Time Protocol (RTP -RFC 1889), Session Description protocol (SDP -RFC 2327)

### **Estándares de compresión**

El video digital es utilizado por muchos tipos de aplicaciones distintas que deben ser capaces tanto de transmitirlo como de reproducirlo. Es indispensable que esta información se comprima tanto para ahorrar espacio de almacenamiento como ancho de banda en la transmisión. Hoy en día es común que distintas aplicaciones intercambien información multimedial entre ellas. Si cada aplicación define su propio formato de video particular esta tarea puede ser bastante complicada o imposible según el caso. Por tal motivo es indispensable que se cuente con formatos de datos estándares que faciliten el intercambio de datos.

Existen muchos formatos estándares para la codificación de video digital. Entre los más utilizados se encuentran la familia de MPEG<sup>1</sup> [3] (MPEG1, MPEG2 y MPEG4) , MJPEG<sup>2</sup> y los protocolos H.261 y H.263 (utilizados principalmente para aplicaciones de video conferencias).

En la siguiente tabla se muestran un resumen de las características de cada uno de ellos. Básicamente, cada uno orientado a un tipo de aplicación.

---

<sup>1</sup> La especificación y descripción completa de todos ellos se puede ubicar en <http://www.mpeg.org> en donde también se describen los estándares MPEG7 y MPEG21 aún no tan popularizados.

<sup>2</sup> Una generalización del formato de compresión de imágenes estáticas JPEG.

Formato original	Codificación	Bitrate	Almacenamiento
QCIF	MPEG-4 Simple	50-300kbps	400-2400KB/min
CIF	MPEG-4 Simple	500-1000 kbps	4-8MB/min
NTSC	MPEG-2 Main	2-6 Mbps	15-45 MB/min
HDTV	MPEG-2 High	15-30 Mbps	113-225 MB/min

### Reproductores y servidores de video con software libre

Entre las aplicaciones que se han analizado se encuentra MPEG4IP[4]: Un proyecto de código abierto basado en estándares que permite intercambiar formato mpeg4 sobre redes IP. Incluye herramientas que permiten la codificación, transmisión y reproducción de archivos off-line, on-line o permite realizar streaming en tiempo real. Se encuentra disponible en casi todas las plataformas. Utiliza Darwin Streaming Server para realizar streaming bajo demanda y se basa en los protocolos RTP/UDP, RTP/RTSP/TCP. Además de reproducir formato mpeg4, puede reproducir otros formatos estándares y propietarios.

La herramienta descrita anteriormente no es una aplicación web. Es una aplicación que utiliza Internet como medio de transmisión de datos pero no es posible ejecutarla desde un navegador web.

Como la intención primaria es encontrar una solución para adaptarla a un entorno de educación a distancia basado en web se ha analizado otra aplicación que se ejecuta sobre un framework escrito en Python denominado Zope[5]. Zope permite construir aplicaciones web brindando funcionalidades de manejo de autenticación de usuarios, sesiones, seguridad y acceso a bases de datos. Este framework cuenta con un número considerable de productos "importables" al framework dentro de los cuales se encuentra el MyMediaManager[6]. este producto permite generar presentaciones multimediales usando Zope. Este producto es más limitado que el anterior pero tiene la ventaja de poder ser ejecutado sobre navegador.

### Conclusión

La intención de este trabajo no es construir un servidor ni un reproductor de video sino analizar la posibilidad de integrar productos existentes en un único ambiente comprando su performance en distintos entornos.

Las pruebas se están realizando sobre una plataforma Linux Red Hat 7.3 y se prevé realizar una análisis comparativo de los resultados en un futuro inmediato.

### Referencias

- [1] <http://www.faqs.org>
- [2] <http://www.opensource.org>
- [3] <http://www.mpeg.org>
- [4] <http://mpeg4ip.sourceforge.net>
- [5] <http://www.zope.org>
- [6] <http://www.zope.org/Members/gittew/MyMediaManager>