

**La realidad actual del streaming de video. El streaming tradicional vs
alternativas actuales**

Autor: Mariano García Clavería
Director del Trabajo de Grado: Jorge Oribe
Categoría: Trabajos finales
Universidad Austral

La realidad actual del streaming de video. El streaming tradicional vs alternativas actuales

Abstract. En los últimos tiempos se ha evidenciado una presencia masiva de sitios web que proporcionan video, incurriendo a diferentes técnicas que suelen categorizarse dentro de lo que sería streaming de video. Sin embargo, no siempre, dichas prácticas son la aplicación de streaming de video en su forma tradicional, al contrario, son alternativas que basándose en los conceptos tradicionales de streaming pretenden proporcionar un servicio similar condicionado por los costos, tecnología y realidad actual, que cada sitio eventualmente sufre. Por consiguiente en el presente estudio se pretendió resumir las diferentes alternativas que existen hoy a la ahora de proveer un video, evaluar sus ventajas y desventajas y estudiar qué sucedía en sitios como Netflix y Youtube.

1. Introducción

Cuando uno habla de Streaming de video, lo más probable que uno posea una definición vaga de lo que realmente estos términos quieren decir. En un primer momento, podríamos asociarlo con lo que sería la transmisión de video, cosa que no es incorrecto, pero estaríamos dando una definición incompleta de estos conceptos. Podríamos agregar también, que si hablamos de video, transmitir video, eso incluiría la transmisión de audio y entonces diríamos que Streaming de video consiste en la distribución de audio o video. Pero nuevamente, nos quedaríamos con una definición incompleta. Es por eso, que es en el concepto de Streaming donde debemos centrar nuestra atención. ¿Qué significa la palabra Streaming? 'Stream' significa 'chorro' o 'flujo' y alude a la descarga de un fichero que no tiene principio ni final: el caso más claro es una transmisión en directo. Una emisión en diferido no deja de ser un fichero muy grande y que se puede transmitir por HTTP, FTP o cualquier otro medio habitual de descarga. La cuestión es que un stream debe ser transmitido de modo que cualquiera pueda conectar con él en cualquier momento, y no sólo al principio de la transmisión. De tal forma, que la visualización de los contenidos en el cliente se da sin la necesidad de esperar la descarga completa de un fichero.

Con la recepción de una pequeña parte el cliente es capaz de entregar su contenido al usuario, mientras continúa recibiendo la corriente de datos (streaming) que irá mostrando posteriormente. La parte almacenada actúa como "colchón" entre el ancho de banda irregular que caracteriza a las redes TCP/IP y el débito continuo que requieren las transmisiones de audio y video [1].

Por lo que ahora podríamos decir, que Streaming es una tecnología de transmisión a través de la red, en la que no existe descarga de la información en un disco local y que por ende, la información que se envía a través de la red al cliente se reproduce en tiempo real al recibirla. Tal es así, que esta tecnología de transmisión también puede ser usada para otros fines. Por ejemplo, se hace uso de streaming en aplicaciones de control en donde sensores van enviando datos que deben reproducirse en tiempo real. Luego, en la medida que la información que sea transmita sean

contenidos multimedia, tales como audio o video, hablaremos de Streaming de Video.

De esta forma, a continuación se describirá lo que es el streaming tradicional, luego diferentes alternativas de streaming. Una vez comprendido lo anterior, se seguirá con un análisis de las ventajas y desventajas de cada uno de los streamings con una tabla descriptiva que complementara dicho análisis. Por último, se evaluará el comportamiento del streaming proporcionado por Netflix y Youtube para saber que está sucediendo realmente cuando vemos un video.

2.1 Streaming Tradicional

El proceso de streaming consiste en la entrega de uno o varios medios multiplexados hacia un cliente en tiempo real, usando una red con un determinado ancho de banda (que no tiene porqué ser necesariamente grande). En el proceso de streaming NO hay ningún fichero que se descarga al ordenador del cliente, sino que el medio se reproduce conforme se está recibiendo, y a su vez el medio se recibe a la velocidad adecuada para su reproducción. Esto contrasta con las descargas progresivas, en las que el fichero sí queda descargado en disco y además se recibe a la mayor velocidad posible, con el fin de terminar el proceso de descarga lo antes posible. En un proceso de streaming estándar de audio y vídeo sincronizado, las peticiones de servicio por parte de los clientes se pueden manejar utilizando el protocolo RTSP (RealTime Streaming Protocol). Este protocolo se encarga de controlar el stream de contenido multimedia en dos direcciones, de forma que los clientes pueden pedir al servidor hacer cosas como rebobinar la película, saltar al siguiente capítulo, etc. Esto se puede conseguir con streaming ya que el medio no se descarga linealmente sino que se reproduce conforme se obtiene, y se permiten saltos en la reproducción, consiguiendo un acceso aleatorio al medio, incluso en saltos hacia delante. Por otra parte, los datos del medio (el stream que contiene típicamente audio y vídeo sincronizados) se pueden transportar usando el protocolo estándar RTP (Real-Time Transport Protocol), que es un protocolo de transporte que permite la transmisión de información multimedia en tiempo real sobre cualquier tipo de red (aunque su uso más habitual es sobre redes IP usando el protocolo UDP) [2].

De esta forma quedan definidos dos canales de comunicación entre los clientes y el servidor de streaming:

- Un canal para el control de sesión(RTSP)
- Un canal para la transmisión de la información.(RTP/UDP/TCP)

Adicionalmente, también encontramos el protocolo SDP(Protocolo de Descripción de Sesiones), que describe los parámetros de inicialización de los flujos multimedia y en aquellas aplicaciones que se basan en la transmisión simultánea de flujos de datos de audio y video, se utiliza el protocolo de control de transporte en tiempo real (RTCP, real- time transport control protocol) para sincronizar los dos flujos de datos multimedia antes de realizar la operación de decodificación, además de proporcionar información sobre la calidad del servicio. Diremos luego, que hablaremos de Streaming Tradicional en la medida que se utilicen los protocolos antes descriptos [3] [4] [5].

2.2 Un poco de Historia

Si bien la incorporación de contenido multimedia siempre fue una idea que estuvo presente, en los primeros tiempos los altos costos sumado a las limitaciones técnicas constituían una gran dificultad que atentaba con su progreso. Las principales cuestiones técnicas en lo que respecta a Streaming consistían en:

- Tener una capacidad de procesamiento suficiente así como también un ancho de banda que soporte la requerida tasa de datos.
- La creación de paths de interrupción de baja latencia en los sistemas operativos con el fin de prevenir un subdesbordamiento de bufer.

Ya en los principios de los 90, con la apertura del campo multimedia por Apple en 1991 y con un usuario de internet beneficiado por:

- Mejoras del ancho de banda, particularmente a finales de la década de los noventa.
- Aumentos en los accesos a las redes, especialmente Internet.
- Uso de protocolos y formatos estándares como TCP/IP, HTTP.
- La comercialización de Internet

El progreso comenzó a ser notorio. Sin embargo, fue recién con la aparición de RealNetworks en 1995 donde comenzó a existir una solución disponible para el gran público de transmisión multimedia sobre Internet. En 1995 RealNetworks logró transmitir por primera vez el audio de un evento, un partido de baseball entre los Yankees y los Seattle Mariners. Tal es así, que el streaming, como tecnología, surge en 1995, para la transmisión de audio, y poco más tarde, en la transmisión de vídeo, ampliando las posibilidades de Internet, al permitir acceder a contenidos audiovisuales sin necesidad de descarga previa. De tal forma que, con posteriores avances en lo que respecta a redes de transmisión de datos y computadoras hogareñas más poderosas con sistemas operativos más modernos, se hizo más práctico y accesible para usuarios comunes. Se podría decir también que el uso del streaming tuvo un crecimiento exponencial en los últimos años tanto por su uso para el entretenimiento, como también por considerarse un vehículo utilizado por las organizaciones para promocionar, vender sus productos y servicios, así como también por su utilización en las comunicaciones.

2.3 Streaming Alternativo

Este tipo de distinción es la que utilizaremos para distinguir el steaming tradicional de aquel que es proporcionado por un servidor web. Es decir, es el caso en el que el servidor de streaming no se encuentra separado de servidor web. En este tipo de streaming todos los pedidos se gestión por pedidos HTTP, no encontraremos en este caso que se está utilizando un terminado protocolo para controlar el streaming y otro protocolo para el envío de datos, como sucede en el streaming tradicional. Dentro del streaming alternativo podemos encontrar distintas opciones:

- **Descarga trivial:** Cuando un usuario hace click sobre un hipervínculo de una página que apunta a un archivo de audio o video, el browser sigue el mismo proceso que para un archivo de texto o imagen. Por lo tanto, el HTTP del browser establece primero una conexión TCP con el http del servidor que aparece en el link. A continuación, envía una solicitud del contenido del archivo especificado en el hipervínculo mediante un mensaje de petición GET. El servidor responde enviando el contenido del archivo en un mensaje de respuesta GET. Al recibirlo, el browser decide a partir del campo Content-Type de la cabecera del mensaje invocar el reproductor de video y, al mismo tiempo, le pasa el contenido del archivo comprimido. El reproductor multimedia procede a descomprimirlo y envía el flujo de bytes resultante a la tarjeta de sonido. Como puede evidenciarse, la desventaja de este método es que el browser debe recibir el contenido del archivo completo y esto puede producir un retardo inaceptable si el tamaño del archivo es considerablemente grande [6].
- **Descarga Progresiva:** Se da cuando el contenido del archivo es enviado directamente al reproductor en lugar de hacerlo a través del navegador. De esta forma el reproductor procede a obtener su contenido de manera normal mediante HTTP/TCP. Al recibir el contenido del fichero, el reproductor simplemente dirige el flujo comprimido hacia la memoria de reproducción. Tras un retardo predefinido para permitir que la memoria se llene parcialmente –diez segundos en el caso del audio- comienza a leer el flujo desde la memoria y, tras descomprimirlo, envía el flujo resultante hacia la tarjeta de audio o de video. Se realiza una descarga progresiva de la información, de manera que cuando se empieza a disponer de información, se pueda empezar a reproducir. Se descarga usando el máximo ancho de banda que disponen cliente y servidor, y no hay ningún control para evitar cortes en la reproducción: el medio se va almacenando en disco conforme se descarga, pero si el ancho de banda es más reducido que el necesario para la reproducción, la información se reproduce “a saltos”, ya que se va reproduciendo conforme llega [7].
 - **Ventajas:**
 - Fácil de configurar.
 - **Desventajas:**
 - El video entero es descargado a menos que el usuario cierre el navegador.
 - El video se descarga completamente por lo que el contenido no está protegido.
 - El usuario no puede reproducir el video desde cualquier momento. Solo podrá ver aquella parte que se encuentre descargada.
- **HTTP Pseudo-streaming:** Basado en el anterior, lo que pretende es simular lo que sería el streaming bajo demanda, agregando la posibilidad de adelantar o retroceder la reproducción. Es decir que las partes que se saltean

no se descargan y permite reducir el ancho de banda que en determinadas situaciones se pierde. Pseudo-streaming requiere adaptaciones tanto para el lado del cliente como para el lado del servidor. Para el lado del servidor, existen plugins disponibles para Apache, lighttpd. Mientras que del lado del cliente, es necesarios reproductores customizados que permiten resincronizar el video, leer metadata, etc. Si se diera el caso de que estamos viendo un video que podemos adelantar y que luego lo podemos encontrar en la cache del browser, luego, el servidor está haciendo uso de HTTP Pseudo-streaming

- **Ventajas:**
 - Posibilidad de interactuar con el stream
 - Mejor utilización del ancho de banda
- **Desventajas:**
 - Necesita implementaciones tanto del cliente como del servidor
 - El video no está protegido, es posible encontrarlo en al cache del browser.
- **Dynamic Adaptive Streaming over HTTP (DASH):** La idea central es dividir el video en pequeñas partes y proveerlo por HTTP. Luego esas partes son combinadas en el lado del cliente y reproducidas. Este método soporta video bajo demanda y live streaming permitiendo entregar el stream correcto teniendo en cuan saturada este la red. Existen numerosas implementaciones de este método con diferentes nombres, todas basadas en la idea antes mencionada: HTTP Live Streaming (Apple), Smooth Streaming (Microsoft) HTTP Dynamic Streaming (Adobe), Adaptive Bitrate (Octoshape).
 - **Ventajas:**
 - Sobre HTTP ofrece una solución muy robusta.
 - Mejoras en cuanto a la protección del contenido dado que el video es dividido en pequeñas partes.
 - **Desventajas:**
 - Las diferentes implementaciones no son del todo compatible.
 - Requiere un desarrollo adicional del lado del cliente.

2.4 Streaming Tradicional vs. Streaming Alternativo

La primera distinción que hay que llevar a cabo a la hora de realizar la comparación es el protocolo de transporte sobre los cuales se dan cada una de las alternativas, el cual impacta notablemente en servicio de streaming que se está proporcionando.

El streaming tradicional tiene la particularidad de poder utilizarse sobre UDP mientras que el streaming alternativo dado que es sobre HTTP, en todas sus versiones, trabaja sobre TCP.

Como sabemos TCP/IP es usado como la capa de transporte en Internet. Los archivos se bajan a la cache del browser tan rápido como el sistema lo permita. TCP incorpora control de flujo para manejar la tasa con la que baja información. No existe

una tasa predeterminada para el envío de los datos. TCP incrementará dicha tasa hasta que una pérdida significativa de los paquetes que se han enviado indique que la red se encuentra congestionada. Y en dicho punto, como consecuencia, la tasa de envío disminuirá. Otra desventaja que presenta TCP para su uso en streaming de video, es que TCP hace uso del mecanismo de ventana deslizante para el control del flujo de datos. Esto da como resultado que los paquetes se procesen ni bien llegan, por lo que si la información llega muy rápido puede darse un buffer overflow. En tal caso, el cliente notificará al servidor que disminuya la tasa de envío, de forma de evitar dicho inconveniente o la necesidad de memorias de reproducción grandes para ocultar el efecto de la pérdida de un segmento al usuario.

Supongamos que uno quiere transmitir un stream codificado a 40 kbit/s. La transmisión TCP podría empezar a 10 kbit/s. Luego podría llegar a 100 kbit/s, tasa sobre la cual se congestionaría la red y por ende, quedaría definida la tasa máxima. Supongamos ahora, que un nuevo usuario se conecta a la red y la transmisión se acelera nuevamente a 30 kbit/s. En ningún momento la tasa de transferencia se asemeja a la tasa de codificado del stream que se quiere transmitir. Lo que da lugar a que el usuario note que el video se ha parado [8].

En el caso de que se de una recepción incorrecta, si se utilizara UDP, la mejor opción en la interpolación de los dato o algún otro tipo de compensación, pero en este caso al estar implementado sobre TCP, se reenvía el paquete en cuestión.

A su vez, en steaming Tradicional se gana velocidad, dado que este posee un menor retardo que TCP a costa de sacrificar la confiabilidad que TCP ofrece, pero que es solucionada utilizando el protocolo RTCP y RTP.

Hay que destacar también, la posibilidad del streaming tradicional de proveer el contenido por medio de técnicas multicast, ideal para la difusión de medios en vivo. Por otro lado, con el uso de RTCP, se podrá sincronizar flujos de datos de audio y video antes de realizar la operación de decodificación. Incluso brindar la posibilidad multiservidor y la capacidad de agregar un nuevo stream en una presentación en vivo. Se podrá informar al emisor los paquetes que se pierden, se podrá cambiar el tipo de codificación, datos el tamaño del buffer e incluso negociar el método de transporte más adecuado antes de comenzar la transferencia del flujo datos [9] [10].

Mientras que si nos centramos en el streaming alternativo podemos afirmar que es una alternativa más fácil de configurar, permite dado que esta implementada sobre HTTP, llegar a una mayor audiencia, y por sobre todas las cosas ofrece un servicio que combinado con un servidor web permite dar una solución acorde sin la necesidad de grandes inversiones. Obviamente dispuesto a perder determinados beneficios y la escalabilidad que el streaming tradicional ofrece.

2.5 Resumen Comparativo

A continuación, en la ilustración 1 se presenta un cuadro comparativo que permite resumir las diferencias existentes a la hora de utilizar diferentes tecnologías de streaming.

| Streaming | | | | |
|--|--|-------------------------|--|---------------------------------------|
| | Streaming Tradicional | | Streaming Alternativo | |
| | | | Progressive Download | HTTP Pseudostreaming |
| Protocolo de Aplicación | RTP-RTSP-RTCP | | HTTP | |
| Protocolos de Transporte | UDP | TCP | TCP | |
| Protocolo de Red | IP | | | |
| Soporte Multicast | SI | No | | |
| Soporte Unicast | Si | | | |
| Ante una perdida de un paquete | El nivel de aplicación determina que hara con el paquete perdido | Retransmisión del mismo | | |
| Capacidad de Multiplexación de varios Stream | SI | | NO | |
| Problemas de NATEO y FIREWALL | Si | NO | NO | |
| Quality Service | SI | | NO | |
| Permancia en cache | No | Si | SI | Solo en fragmentos de tamaño reducido |
| Soportado por Browsers | No | | SI | |
| | Reproductor | | Necesario para la Reproducción Browser+Plugin / Reproductor | |
| Control sobre la transmisión | Si | No | SI | SI |
| Necesidad de Servidores | Servidores de Streaming | | Servidores Web | |
| Inmediates | Se reproduce conforme llega | | | |
| | Solo usa el ancho de banda que necesita | | Utilizan máximo ancho de banda para descarga | |
| Capacidad de Adaptación al ancho de banda | Si | NO | No | SI |
| Opción de modificar algoritmo de compresión en curso | Si | NO | | |

Ilustración 1: Tabla Resumen que compara las diferentes tecnologías de streaming

4. Casos de Estudio

Se describen luego, los resultados obtenidos luego de haber analizado el funcionamiento del streaming proporcionado por Youtube y Netflix.

4.1 Youtube

Cuando analizamos como funciona realmente el streaming propuesto por Youtube ya a primera vista es posible evidenciar que hace uso de la tecnología de Adobe. El video llega incrustado como parte de un archivo cuya extensión es .swf . Entonces cuando queremos ver un video en youtube, lo que está sucediendo es que este nos devuelve un .swf que se descarga a la cache del browser y luego es ejecutado por Adobe Flash Player. De esta forma, este .swf se encarga de embeber el video en el browser, customizar como se ve el reproductor y controlar el comportamiento del video entre otras cosas. Siendo el parámetro video_id, el que indica el id del video a reproducir. Se pudo ver también, que a la hora de reproducir un video no se hace uso del protocolo RTSP. Todas las peticiones que se realizaron para interactuar con el video, particularmente el reproducir y el adelantar se llevan a cabo mediante http

request.

Observamos que, al querer adelantar el video, lo que sucede es que se cancela la descarga de la parte del video que actualmente se está reproduciendo y comienza a descargarse y reproducirse la nueva parte del video desde la posición seleccionada por el usuario. En dicho request, es el atributo range, el que nos determina la posición de comienzo del video. Más precisamente el range se refiere al código de tiempo de cada frame (fotograma o cuadro, es una imagen particular dentro de una sucesión de imágenes que componen una animación). A su vez, si dicha url la decodificamos, la pegamos en la barra de dirección del navegador es posible descargar el video desde esa posición.

Se descarta también la posibilidad de encontrar paquetes RTP dado que sobre TCP no tendrían sentido. Y esto se debe a que RTP es una capa diseñada para actuar sobre protocolos no confiables proveyendo determinados beneficios como los que proveen los protocolos confiables (garantiza orden, timestamp) sin las desventajas que trae TCP, por lo que aplicar este tipo de combinación sería totalmente desacertado. Incluso utilizando Wireshark, no fue posible encontrar evidencia de paquetes con el protocolo RTP.

La sospecha de que se estuviera haciendo DASH queda totalmente descartada dado que no se encontraron archivos de extensión .f4v ni .F4M. El primero serían los fragmentos que se divide el video que se esta streameando, en cuyo contenido se encuentra el video y el segundo describe el medio en cuestión, pudiendo de esta forma ir determinando como debería ir variando el bitRate. Tal como establece la especificación de Adobe.

De todos modos, se pudo comprobar que independientemente de la resolución de video, en todos los casos siempre se experimento una descarga a las mayores velocidades posibles y nunca cambiando la calidad de video durante la reproducción. La calidad de video en youtube es fija, si bien se puede cambiar voluntariamente a lo largo del video, esta se mantendrá constante y será el ancho de banda del cliente quien determine como se verá un video. A mayor ancho de banda, mayor será el bitrate que el cliente podrá streamear. Y si se diera un cambio en la calidad del video, producto de que youtube.com está haciendo DASH, el contenido total del video descargado utilizando un ancho de banda y otro ancho de banda diferente debería ser distinto. Sin embargo, utilizando 1Mb o 3Mb siempre, en cada una de las resoluciones, el tamaño del archivo final es aproximadamente el mismo.

Por lo que podemos concluir que youtube.com no muestra sus videos mediante streaming tradicional ni tampoco HTTP Dynamic Streaming. Youtube.com lleva a cabo pseudo-streaming para la entrega y reproducción de sus videos. No tiene en cuenta el ancho de banda del cliente y tampoco envía los videos al bit rate que son codificados, usa en todos los casos la máxima velocidad de descarga. Los videos quedan alojados en la cache del browser y todo su mecanismo de streaming se pasa en HTTP, incluso las acciones de play y adelantar. En todos los casos hace uso de TCP.

4.2 Netflix

Analizar como NetFlix provee su contenido multimedia, por lo menos en Windows, nos lleva a estudiar la tecnología de streaming denominada Silverlight. Silverlight, es un producto desarrollado por Microsoft, que permite crear aplicaciones

web, particularmente RIA (Rich Internet Application). A pesar de que Silverlight tiene infinidad de utilidades nos concentraremos en la tecnología utilizada para llevar a cabo el streaming que es Smooth Streaming. Esta tecnología no es otra cosa que DASH, y es lo que realmente utiliza Netflix a la hora de proveernos un video. De esta forma, Netflix posee sus películas codificados en varias calidades, y detectando nuestro ancho de banda, descargará aquel que se adecue a nuestra velocidad de bajada. Esto implica que ver un video con Smooth Streaming sea mucho más suave y rápido que con otros métodos de streaming tradicionales.

Se evidencia que al comenzar a ver una película se descarga un archivo .xap, que al igual que sucedía con youtube que era un .swf, este determinará el funcionamiento del reproductor. Si inspeccionamos ese .xap, no es otra cosa que un archivo comprimido que posee todo lo necesario para poder correr una aplicación con el plugin Silverlight. Si lo descomprimos encontraremos siempre como mínimo dos archivos, un manifest y el archivo propio que contiene el código necesario para la ejecución del reproductor (.dll). Sin embargo, lo que realmente sucede con Netflix es que el .xap suele guardarse en la cache del browser, tiene solo un archivo un .dll y luego, se baja el manifest por separado [11].

Este manifest, describe los diferentes encoding que posee la película tanto para el audio como para el video, identificándolos con un Id y la url del host donde se encuentran alojados. En el caso estudiado se detectaron 6 video que van desde los 0,235 Mbps a 1,750 Mbps y tres tipos de audio (español, inglés y portugués) con sus respectivos bitRates.

A continuación, comenzarán las diferentes solicitudes de los fragmentos de video de acuerdo al ancho de banda disponible. Tal es así que se irán generando request de este tipo:

<http://959.ar.akam.nflxvideo.net/la50/156/1768216156.ismv/range/2216419-2448828?token=133...>

En los cuales, según el ancho de banda disponible será el fragmento que se va a recibir, conectándose al servidor correspondiente de acuerdo a lo descrito en el manifest. Y será el parámetro range el que indique el offset que se necesitará para poder reproducir el video en el momento correcto. Por lo tanto, en el request anterior se está accediendo al host: 959.ar.akam.nflxvideo.net y luego /la50/156/1768216156.ismv/ indica el tipo de codificación que se está solicitando. En este caso, según lo descrito en el manifest se estaría bajando un fragmento de un encoding de 0,375 Mbps con una resolución de 384x288p, en formato .ismv.

Este formato no es otra cosa que un fragmento de un video en formato MPEG-4. Este fragmento es definido por la tecnología Smooth como un MPG-4 Movie Fragment y es almacenado con otro archivo que permite su acceso de forma rápida y aleatoria. De esta forma, cuando un cliente solicita un fragmento en determinada ubicación, el servidor encuentra de forma dinámica el Movie Fragment box con el MPG-4 contiguo y solo envía este a través de la red [12]. En otras palabras, el servidor no tiene una película dividida en millones de chunks, sino que estos son creados dinámicamente por cada solicitud. Lo que da lugar ahorros en lo que respecta a manejo de archivos. Todo esto da lugar a que el video se vaya descargando en pequeñas partes y no en una sola unidad como en la descarga progresiva, que es el caso de Youtube.

Hay que destacar el hecho, de que antes de comenzar a reproducir la película,

se van descargando fragmentos de diferente bitRate en diferentes conexiones TCP, luego de un tiempo, solo quedarán dos conexiones TCP, una encargada de obtener el audio y otra el video. Como se notará, siempre que sea posible se pretenderá descargar aquellos fragmentos de mayor resolución. A su vez, si bien antes dijimos que luego de un tiempo solo perduraban dos conexiones, una para audio y otra para video, se puede ver que en determinados momentos ambas conexiones descargan video. Y esto se debe seguramente, para poder estar transmitiendo siempre contenido en ambas conexiones. Si la conexión de audio no está siendo utilizada para la transmisión de audio entonces se aprovecha y se envía contenido de video.

Hay que aclarar que Netflix no nos permite definir un tipo de resolución de bajada, en todos los casos transmitirá aquel fragmento de mayor resolución de acuerdo al ancho de banda disponible. Sin embargo, si se reduce la ventana del reproductor a tal punto de no poder ver el video, la resolución de los fragmentos será al mínimo BitRate. Por otro lado, si lo que pretendemos es adelantar la película, esta operación también se realiza mediante una http request, el cual una vez solicitado, puede demorar un tiempo la reproducción dado que antes, el reproductor pretenderá cargar determinado contenido en el buffer. Cada vez que adelantamos, se produce una nueva conexión TCP.

Por lo que entonces, podemos concluir que Netflix, hace uso de una típica tecnología de streaming adaptativo, el video/audio es cortado en segmentos pequeños, por lo general denominados chunks. Cada chunk son por lo general de 2 a 4 segundos de duración, los cuales, por un lado se los puede ver como un grupo de imágenes y por el otro, hay que afirmar, que cada chunk no tiene dependencia con el pasado o futuro del resto de los otros chunks. Por ende cada uno puede ser, decodificado independientemente de cualquier otro chunk. De esta forma, cada chunk es descargado utilizando HTTP progressive download.

Ahora, la parte que da lugar a que podamos afirmar que Netflix, utiliza adaptative streamings es debido a que el browser va solicitando chunks de diferente bitrate. De esta forma, como los servidores web tienden a enviar la información a la mayor velocidad posible, el cliente puede estimar el ancho de banda y decidir descargar chunks más grandes o más chicos, dado que el video se encuentra codificado para distintas definiciones, por lo que cada envío se encuentra condicionado al ancho de banda del usuario. Toda esta información, es posible obtenerla desde el manifest, antes explicado y un correcto funcionamiento del reproductor Silverlight [13].

5. Conclusión

A lo largo de este trabajo, se pudo darle un marco de referencia a lo que se entiende cuando hablamos de streaming. Evidenciando un streaming tradicional, basado en protocolos tradicionales diseñados para la transferencia de datos sujetos a limitaciones de tiempo real y un streaming alternativo que sobre el protocolo HTTP pretende dar una solución similar.

De esta forma el streaming tradicional se caracterizó en poder enviar video y que este llegue al usuario sin pausas, que este pudiera controlar su recepción así como también, poder recibir video en tiempo real con la mejor calidad de acuerdo a su ancho de banda y poder ir notificando la calidad del servicio que estaba recibiendo.

Mientras que el streaming alternativo, se presenta como una solución capaz de simular el mismo comportamiento que el streaming tradicional, pero de una forma más sencilla aunque con determinadas limitaciones.

Se destaca particularmente la opción de streaming adaptativo como la opción que hoy en día, puede que no esté implementada en todos los sitios que ofrecen videos, pero es la tecnología que basándose en los protocolos tradicionales supo adaptarlos y aplicarlos al contexto actual. Permitió desligarse de la necesidad de requerir un servidor de streaming dedicado, y centrar el tráfico entre cliente y servidor a través de HTTP. Lo que dio lugar a tener un mayor alcance (mayor audiencia), dado que el tráfico HTTP puede alcanzar la mayoría de los dispositivos e incluso a diferencia de otros protocolos, los firewalls no los bloquean y, por sobre todas las cosas, sentarse sobre la base de una tecnología más escalable.

Un ejemplo del uso de esta tecnología fue Netflix, que haciendo uso de Smooth Streaming hacia llegar su contenido a sus usuarios. Sin embargo, Smooth Streaming era solo una de las tantas formas para ofrecer streaming adaptativo. Hay otras alternativas, Smooth Streaming era una de ellas, pero también estaban las opciones propuestas por Adobe, Octashape o Apple.

De todos modos, si bien hoy en día, existen actores fundamentales y cada uno ofrece su tecnología similar por separado, no existe una estandarización definida en lo que respecta a Streaming adaptativo, pero se está en camino a eso. Puede también que en un futuro cercano, sean los mismos browsers los que incorporen de manera nativa las funcionalidades de streaming.

Tal es así, que la importancia del streaming de video seguirá creciendo, y se deberá seguir investigando, controlando y evaluando las tendencias del mercado de tal forma de poder elegir la tecnología correcta y que proporcione un stream de la mejor calidad según sean las condiciones técnicas de enlace que tenga el grupo de usuarios. La tecnología será fundamental, determinará aspectos claves en los costos y en la complejidad del sitio que la implemente. Pero tampoco habrá de olvidarse, que el stream a ofrecer deberá estar orientado a la audiencia que lo consuma y desde donde lo consuma. Por lo que determinar que una tecnología es mejor que otra o utilizar este protocolo o aquel dependerá de la audiencia a la que se quiera llegar, la complejidad que se esté dispuesto asumir, los costos relacionados con dicha complejidad, la calidad del servicio que se quiera proporcionar y por sobre todas las cosas el tipo de aplicación que se esté desarrollando. De todos modos, en cualquier variante que se lleve a cabo el streaming, se podrá notar de fondo que dicha implementación perseguirá siempre los mismos conceptos sobre los cuales los protocolos tradicionales se establecieron.

Luego se puede afirmar que el mundo del streaming no se limitaba solo a los protocolos tradicionales, y tampoco sus alternativas acotaban la cuestión. Existían y existen numerosos aspectos que no fueron considerados en este trabajo pero que también juegan y condicionan el modo de obtener un correcto desempeño a la hora de llevar a cabo streaming de video. Con solo mencionar cuestiones como buffering, tipo de codecs, formatos y demás dan noción de lo complejo que se convirtió el mundo del streaming.

6. Bibliografía

- [1] "Utilización de video Streaming." J. Aramberri y J. Lasa. N.p., n.d. Web. 07 Aug. 2012. <<http://www.rediris.es/difusion/publicaciones/boletin/58-59/ponencia10.html> />. Sitio
- [2] "RTSP: FAQ." Computer Science - Columbia University. N.p., n.d. Web. 07 Aug. 2012. <<http://www.cs.columbia.edu/~hgs/rtsp/faq.html> >. Sitio
- [3] "Streaming de Audio y Vídeo" Grupo de Redes de Computadoras – Universidad Pontificia de Valencia. N.p., n.d. Web. 07 Aug. 2012. <<http://www.grc.upv.es/docencia/tdm/practicas/P3.pdf> >. Práctica
- [4] "RTSP" Computer Science - Columbia University. H. Schulzrinne, A. Rao, R. Lanphier. N.p., 2 Feb. 1998. Web. 07 Aug. 2012. <<http://www.cs.columbia.edu/~hgs/rtsp/draft/draft-ietf-mmusic-rtsp-09.pdf> >. Paper
- [5] " RTP: A Transport Protocol for Real-Time Applications" The Internet Engineering Task Force (IETF). H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, Jul 2003. Web. 07 Aug. 2012. <<http://www.ietf.org/rfc/rfc3550.txt> >. Paper
- [6] "Video Delivery in HTTP." Roman10. N.p., n.d. Web. 07 Aug. 2012. <<http://www.roman10.net/video-delivery-in-http/> />. Sitio
- [7] "Redes de Computadoras e Internet". Fred Halsall, Editorial Pearson
- [8] "Streaming Media Bible". Steve Mack, Editorial Hungry Minds.
- [9] "RTCP, RTP Control Protocol." Network Sorcery, Inc. N.p., n.d. Web. 07 Aug. 2012. <<http://www.networksorcery.com/enp/protocol/rtcp.htm> >. Sitio
- [10] "6.1 RTCP Packet Format." Freesoft.org. N.p., n.d. Web. 10 Aug. 2012. <<http://freesoft.org/CIE/RFC/1889/14.htm> >. Sitio
- [11] " An Experimental Evaluation of Rate-Adaptation Algorithms in Adaptive Streaming over HTTP". Saamer Akhshabi, Ali C. Begen, Constantine Dovrolis, Feb 2011. Web. 07 Aug. 2012. <<http://www.cc.gatech.edu/~dovrolis/Papers/final-saamer-mmsys11.pdf>>. Paper
- [12] "Live and On Demand Video with Silverlight and IIS Smooth Streaming_FINAL" Microsoft N.p., n.d. Web. 10 Aug. 2012. <http://download.microsoft.com/download/3/A/4/3A4A066C-6543-4BC1-A8B..._and_On_Demand_Video_with_Silverlight_and%20IIS_Smooth_Streaming_FINAL.pdf />. Sitio
- [13] "Analysis of Netflix's security framework for 'Watch Instantly' service " Pomelo, LLC TechMemo. N.p., Mar 2009. Web. 07 Aug. 2012. <<http://pomelolc.files.wordpress.com/2009/04/pomelo-tech-report-netflix.pdf>>. Paper.