

Estudio Comparativo entre Dos Sistemas de Transmisión de Video Streaming

Ricardo Andrés Robalino Ronquillo ⁽¹⁾ Edgar Alberto Jara Gómez ⁽²⁾ Marcos Efraín Millán Traverso ⁽³⁾
Facultad de Ingeniería en Electricidad y Computación
Escuela Superior Politécnica del Litoral (ESPOL)
Campus Gustavo Galindo, Km 30.5 vía Perimetral
Apartado 09-01-5863. Guayaquil-Ecuador
ricanrob@espol.edu.ec ⁽¹⁾ ajara@espol.edu.ec ⁽²⁾ mmillan@espol.edu.ec ⁽³⁾

Resumen

Este documento presenta el estudio comparativo de dos plataformas de transmisión de video en vivo, en una distribución del sistema operativo Linux, implementado mediante líneas de comando. El estudio se realizó debido a la gran cantidad de sistemas que se encuentran en el mercado, tanto gratuitos como de pago, esto dificulta al usuario al momento de decidirse por un sistema de transmisión de video en vivo que se ajuste a sus necesidades. Por este motivo se implementó un plan de pruebas para poder llevar a cabo el estudio comparativo entre Icecast y VLC los cuales son servidores de transmisión de video en vivo gratuitos, estas transmisiones se realizaron mediante el protocolo HTTP(TCP) con un códec libre, llamado THEORA. Se analizaron diferentes variables como son: consumo de recursos del Servidor (Procesador, Memoria) y en la transmisión se analizó la Latencia y Variación de retardo.

Palabras Claves: Linux, Theora, VLC, Icecast, Retardo, Variación de Retardo

Abstract

This document presents a comparative study between two live video streaming platforms in Linux Operative System distribution using the command line interface, the research was done due to large amount of systems, both paid and free that you can be found in the market. This the most difficult for the user when is time to choose one. In this research we have implemented the respective test plan in order to do the comparative study between both of them, the live video streaming were made by a HTTP (TCP) protocol with a free codec, called THEORA, and several variables were taken in consideration such as: Server Resources (CPU, RAM memory), Delay and Jitter.

Keywords: Linux, Theora, VLC, Icecast, Delay, Jitter.

1. Introducción

A mediados de los años noventa, el usuario que deseaba observar contenido mediante red tenía que descargar completamente los datos de un archivo, para luego poder observarlo. La descarga de estos archivos en especial los de video generaba un tiempo de espera y ocupaba espacio en el disco duro del usuario, para evitar esto se creó una nueva tecnología llamada streaming que quiere decir transmisión, esto permitió al usuario observar el contenido del archivo mientras se lo está descargando evitando así las molestias que se presentaban anteriormente y haciendo más eficiente el consumo de la red. Esta tecnología se ha vuelto indispensable en la red, por medio de la transmisión en vivo se ha abierto un nuevo mercado de servicios de tráfico de datos y distribución de contenidos que están destinadas a dar soluciones sencillas y eficientes

para la distribución global de contenido de audio y video.

Estos Sistemas de transmisión se implementaron en la misma máquina y se les aplicó a cada servidor el mismo plan de pruebas, de esta forma obtuvimos datos importantes para poder realizar la comparación de ciertas variables como, consumo de recursos de procesador y memoria en el servidor, de esta manera se pudo determinar que servidor es más eficiente y eficaz al momento de transmitir un video bajo el escenario y parámetros que se implementaron.

2. Metodología

Para lograr comprender este tema se utilizó el método analítico, que no es más que descomponer el tema general en varias partes, con esto se logró

estudiar más a fondo cada uno de sus elementos. Luego de esto se realizó experimentos de un hecho en particular, se observó y analizó su funcionamiento, para finalmente explicar y comprender su comportamiento en los experimentos y de esta manera se estableció el respectivo estudio comparativo.[1]



Figura 1. Metodología usada para realizar el estudio comparativo entre dos servidores

En la Figura 1 se muestra el proceso que se usó para realizar la comparativa, se explica los conceptos básicos para realizar una transmisión de video y los servidores que se utilizaron, posterior a esto se realizó la instalación de estos servicios mediante línea de comando en el sistema operativo Linux, también se instaló una serie de programas en la computadora que actuaba de husmeador, estos fueron utilizados para recopilar los datos requeridos, luego se estableció los parámetros de las pruebas que se aplicaron a los servicios, una vez establecido los parámetros de la transmisión se procedió a realizar las pruebas mediante un script para poder automatizar el proceso, una vez obtenido los datos estos fueron ordenados y clasificados con ayuda de *Excel* para utilizar *Minitab* como herramienta estadística y poder obtener datos y graficas de las variables observadas, los cuales se analizó y se pudo realizar el respectivo estudio comparativo.

3. Plan de pruebas

Se implementaron dos servicios de transmisión de video en vivo gratuitos *Icecast* y *VLC*, se realizó la transmisión del mismo video y se le aplicó los mismos parámetros a la transmisión, se lo realizó de esta manera para poder comparar variables como, consumo de recurso de procesador, memoria y comparar la latencia y variación de retardo que existió en la transmisión. Para poder realizar estas pruebas se implementó un escenario como se muestra en la Figura 2, este estaba conformado de tres máquinas que actuaban de servidor, cliente y husmeador. Se conectaban a la misma red por medio inalámbrico a través de un enrutador.

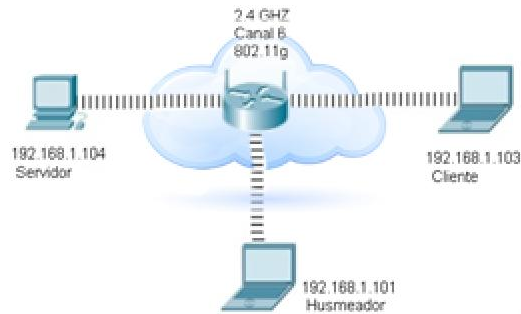


Figura 2. Escenario implementado para realizar pruebas

Posterior a esto se seleccionó un video formato mp4 en calidad HD y se determinó el número de veces que se tenía que repetir la transmisión para poder observar cualquier diferencia entre cada una de estas. Se hizo uso de *Cacti* el cual es un programa para monitoreo de dispositivos, este recopila datos cada minuto, por lo tanto la prueba consistió en realizar la transmisión de video en vivo por tres minutos, luego dejar pasar un minuto sin realizar la transmisión y volver a repetir el proceso, esto se repitió más de 100 veces, por este motivo se implementó un script el cual nos permitió automatizar estas pruebas. También hizo uso de *Wireshark* que nos ayudó a recopilar los datos que se enviaban al momento de realizar la transmisión entre servidor y cliente, con esta información se pudo obtener la latencia de la transmisión y la variación del retardo. [2]

Tabla 1. Parámetros asignados a las transmisiones

Parámetros	Características
Tasa de Bits	800kbps
Fotograma por segundo	25fps
Resolución	640x480
Protocolo	HTTP
Códec	Theora
Encapsulamiento	.ogg

Se les asignaron algunos parámetros importantes a la transmisión como se puede ver en la tabla 1, la tasa de bits fue de 800kbps debido a que la resolución aplicada fue de 640 x 480 pixeles la calidad del video era aceptable y si aumentáramos la tasa de bits íbamos a necesitar más ancho de banda y esto provocaría un mayor consumo de procesador debido a la transcodificación, el video original tenía ya establecido 25 fps y si aumentáramos o disminuíamos este valor íbamos afectar al momento de que el cliente vea el video final.[3] El protocolo elegido fue http ya que es un protocolo confiable, seguro y en algunas empresas bloquean el tráfico UDP por motivos de seguridad o políticas internas.[4] El códec *Theora* fue diseñado para la eficiencia al consumir procesador en el momento de la transmisión, es libre de patentes y de código abierto, al igual que el formato contenedor *.ogg* que fue diseñado para un alto grado de eficiencia

en la transmisión de video en vivo y la compresión de archivos.[5]

4. Resultados

Luego de realizar la n cantidades de pruebas por medio de un script, se recopiló los siguientes datos con los programas que anteriormente se ha mencionado. En la Figura 3 y 4 se observa los histogramas de consumo porcentual de procesador que existió al momento de realizar las pruebas con ambos servidores, *VLC* consume menos procesador en comparación a *Icecast* y es más estable en su consumo ya que *VLC* obtuvo una variación menor a la que obtuvo *Icecast*

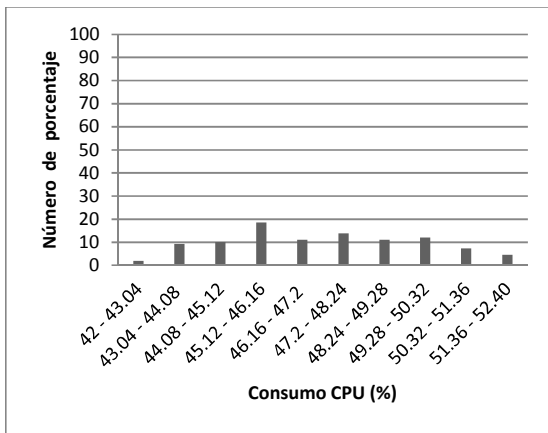


Figura 3. Histograma de consumo de CPU con *VLC*

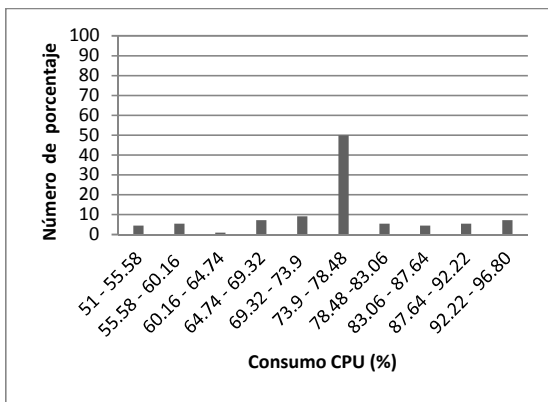


Figura 4 . Histograma de consumo de CPU con *Icecast*

En la Figura 5 y 6 se puede ver los histogramas de consumo de memoria que existió en ambos servidores al momento de realizar las transmisiones, *VLC* consume mayor recursos de memoria que lo que consume *Icecast* y su desviación estándar es mayor a la de *Icecast* con una diferencia mínima, esto quiere decir que *Icecast* fue más estable al momento de consumir memoria.

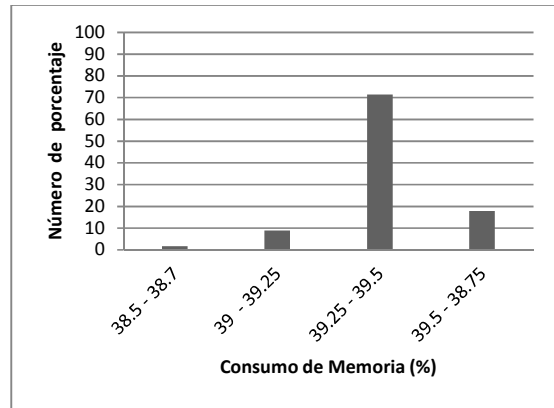


Figura 5. Histograma de consumo de memoria con *VLC*

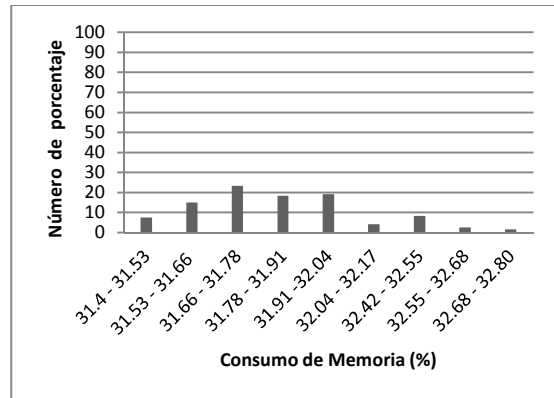


Figura 6 . Histograma de consumo de memoria con *Icecast*

Con ayuda de *Wireshark* recopilamos datos de la transmisión, en la Figura 7 y 8 tenemos la comparación de los histogramas de latencia que existió en ambos servicios al momento de realizar las transmisiones, *VLC* obtuvo menos latencia en comparación del otro servicio, pero la desviación estándar de la latencia en *VLC* fue mayor a la de *Icecast* con una diferencia mínima.

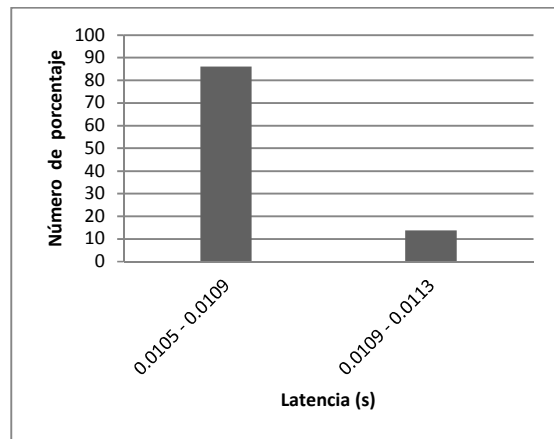


Figura 7 Histograma de consumo de latencia con *VLC*

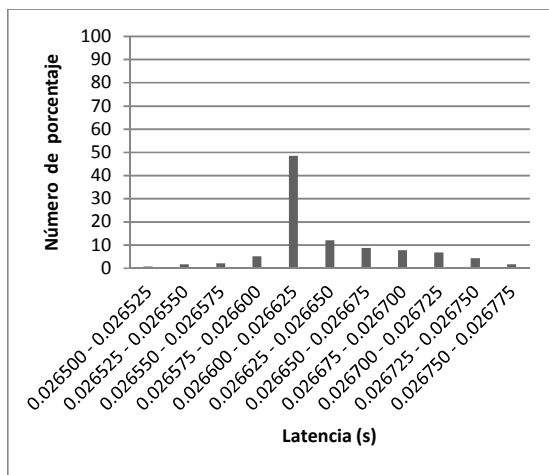


Figura 8 Histograma de consumo de latencia con Icecast

5. Conclusiones

Se determinó que el uso de memoria en Icecast es 7.79% más elevado comparado con el uso de memoria en VLC, se notó que el consumo de procesador en Icecast es 43.43% mayor al de VLC al momento de transmitir un video en vivo.

Se concluyó que Icecast tiene mayor latencia con una diferencia de 15.69ms respecto a la latencia presente en VLC, pero VLC posee una desviación estándar 79 μ mayor a la que posee Icecast esto quiere decir que su latencia varía más con respecto a su media.

Luego de examinar los diferentes tipos de códecs y formatos contenedores, se concluyó que el consumo de recursos, especialmente de procesador es proporcional al uso del códec que se escoja y este no afecta de mayor manera al consumo de memoria al momento de realizar una transmisión de video en vivo.

Se observó que en las dos transmisiones existió poca variación en el retardo pero sin embargo en la transmisión realizada con VLC se observó que tenía 5 μ s más de retardo que la que fue realizada con Icecast y este tiene 79 μ más de desviación estándar con respecto a la media que la que posee Icecast, pero la diferencia entre los tiempos de Icecast y VLC son muy pequeños, por lo cual no se puede decir que uno es mejor que el otro con solo observar la variación de retardo de los paquetes.

Después de analizar todos los datos que se recopilaban a lo largo de las pruebas se concluyó que tanto el servidor VLC como Icecast poseen una transmisión de calidad tomando en consideración los parámetros de transmisión asignados, pero VLC es más eficiente al momento de transmitir en cuanto a consumo de recursos se refiere, ya que este consume

28.14% menos de recursos de procesamiento que los que consume Icecast y el consumo de memoria de los dos servidores son muy bajos. Por este motivo se concluye que bajo el escenario de trabajo y parámetros asignados a las pruebas, VLC es más eficiente como servidor de transmisión de video en vivo.

6. Recomendaciones

Se recomienda realizar pruebas de calidad de recepción, también pruebas de cantidad de paquetes perdidos para poder determinar con seguridad que transmisión fue más eficiente al momento de realizar las transmisiones de video en vivo.

Se aconseja usar como servidor de transmisión de video en vivo una máquina que posea un procesador Core 2 Duo en adelante para que la función de transcódec sea más fácil de realizar y en un menor tiempo, así el video es menos vulnerable a perder calidad.

Al momento de escoger un códec para una transmisión en la cual lo primordial sea el bajo consumo de recursos del servidor y de la red, se recomienda el uso de Theora.

7. Agradecimientos

Agradecemos infinitamente por el apoyo y el tiempo prestado para la realización de este artículo a la MSc. Patricia Chávez y al MSc. Ignacio Marín

8. Referencias

- [1] W.-t. T. S. J. W. John G. Apostolopoulos, «Video Streaming: Concepts, Algorithms, and Systems, » Hewlett-Packard Laboratories, Palo Alto, 2002.
- [2] W. A. Lozano-Rivas., «Determinación del Número Mínimo de Observaciones en Investigación,» Investea, Bogota, 2011.
- [3] I. E. G. Richardson, H.264 and Mpeg-4 Video Compression, Aberdeen: John Wiley & Sons Ltd, 2003.
- [4] ETF, «RFC 793 TRANSMISSION CONTROL PROTOCOL,» Septiembre 1981. [En línea]. Disponible: www.ietf.org/rfc/rfc793.txt. [Último acceso: octubre 2013].
- [5] Xiph.org Foundation, «Xiph.org» 6 2 2013. [En línea]. Disponible: <https://www.xiph.org/doc/> [Último acceso: 22 12 2013].