ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

"ESTUDIO COMPARATIVO DE DOS SISTEMAS DE TRANSMISIÓN DE VIDEO STREAMING"

TESINA DE SEMINARIO

Previa a la obtención del Título de:

INGENIERO EN TELEMÁTICA

Presentada por

RICARDO ANDRÉS ROBALINO RONQUILLO

EDGAR ALBERTO JARA GÓMEZ

Guayaquil – Ecuador

2014

AGRADECIMIENTO

Al MSc. Marcos Millán, MSc. Patrica Chávez y MSc. Ignacio Marín por todos los conocimientos y consejos transmitidos. Al Sr. Vicente Ronquillo Quintero y Sra. Yolanda Figueroa Flores por todo el apoyo brindado.

DEDICATORIA

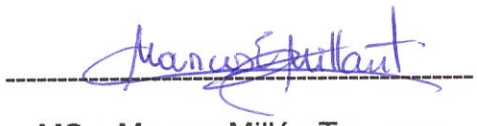
Le dedico este trabajo a Dios, a mi padre Ricardo Robalino Flores por ser un pilar fundamental en mi vida, por su esfuerzo y dedicación, a mi madre Silvia Ronquillo Quintero (+) que aunque no este fisicamente conmigo sé que siempre estará junto a mi en cada paso que dé, a mi hermana Michelle Robalino Ronquillo y mi enamorada Karen Mite Labre porque han estado siempre brindandome su apoyo. Gracias por sus consejos y por no dejarme decaer en el camino.

Ricardo Andrés Robalino Ronquillo

Le dedico este trabajo a Dios, a mi padre Edgar Jara Aguilar, mi madre Flor Gómez Flores, mi hermana Wendy Jara Gómez y a toda mi familia que siempre me apoyaron incondicionalmente y me ayudaron a encaminarme por el buen camino y asi lograr mis objetivos.

Edgar Alberto Jara Gómez

TRIBUNAL DE SUSTENTACIÓN



MSc. Marcos Millán Traverso

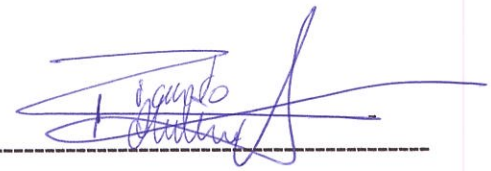
Profesor del Seminario de Graduación

MSc. Patricia Chávez Burbano

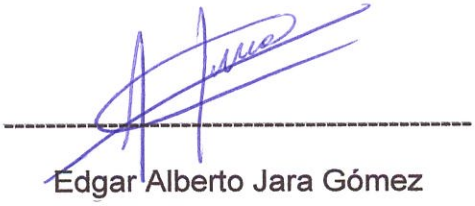
Profesora delegada del Decano

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta tesina me corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral"



Ricardo Andrés Robalino Ronquillo



Edgar Alberto Jara Gómez

RESUMEN

En esta tesina se presenta un estudio comparativo entre dos servidores de video transmisión en vivo (streaming), el cual esta dividido en 4 capítulos que comprenden el planteamiento del estudio comparativo, el análisis del problema, el marco teórico, implementación, análisis de los resultados, posterior a esto se revelan las conclusiones y recomendaciones.

Para esto se detallo los conocimientos básicos sobre el video digital tales como: Compresión, tipos de compresión, normas de compresión, códecs y formatos contenedores. Es necesario tener estos conocimientos debido a que estas son variables utlizadas para enviar el flujo de transmisión en vivo.

Se describio los conceptos básicos para realizar la transmisión en vivo y tipos de transmisión de video en vivo. En este trabajo se detallo las maneras en que se puede realizar la transmisión en vivo de video y sus diferentes protocolos.

Luego de tener todos los conocimientos necesarios, se estableció la metodología con la que se van a realizar las pruebas, presentación del escenario, software que se implementaron para poder recopilar los datos, asi como el análisis de los mismos con su debida interpretación, conclusiones y recomendaciones.

ÍNDICE GENERAL

ABREVIATURAS

ASF	Formato de Transmisión Avanzada
AVI	Intercalado de Audio y Video
CBR	Constante de tasa de bits
CD	Disco compacto
CPU	Unidad Central de Procesamiento
DMIF	Entrega de integración de marco de trabajo multimedia
DSM-CC	Comandos de los medios de almacenamiento digital y control
DVD	Disco de video digital
FourCC	Cuatro Caracteres de Código
GNU	Gnu No es Unix
HD	Alta Definición
HTTP	Protocolo de Transferencia de Hipertexto
IEC	Comisión Electrónica Internacional
IETF	Grupo de Trabajo de Ingeniería de Internet

IPMP	IP protocolo de medición
ISDN	Red digital de servicios integrados
ISO	Organización internacional de normalización
Kbps	Kilo bits por segundo
LCL	Contendor de grupaje
Mbps	Mega bits por segundo
Mbps	Mega bits por segundo
MOV	Formato de QuickTime Movie
MP4	Media Player 4
MPEG	Grupo de expertos en imágenes en movimiento
RFC	Solicitud por comentario
RGB	Rojo Verde Azul
RLE	Codificación de longitud de ejecución
RMVB	Tasa de bit variables de real media
RTCP	Protocolo de control de transporte en tiempo real
RTI	Red telefónica internacional

RTO	Objetivo de tiempo de recuperación
RTP	Protocolo de transporte en tiempo real
SAP	Sistemas, Aplicaciones, Productos en procesamiento de datos.
SDP	Protocolo de enchufe directo
SIF	Formato de entrada estándar
SMP	Memoria compartida paralela
SWF	Formato de web pequeño
TCP	Protocolo de transmisión de control
ITU	Union internacional de telecomunicaciones
UDP	Protocolo de datagrama de usuarios
VBR	Tasa variable de bits
VGA	Adaptador gráfico de video
VLC	Convertidor de Video LAN
VLS	Video Lan servidor
VOD	Video bajo demanda
WMV	Medio de video de Windows

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

En la actualidad la transmisión en vivo es una herramienta muy útil al momento de visualizar o escuchar contenidos como música, videos y documentos, pero existen gran cantidad de servidores de transmisión en vivo y en muchas ocasiones no tenemos el conocimiento adecuado para poder escoger el servidor que se ajuste a nuestras necesidades. En esta tesina nos enfocamos en la transmisión de video en vivo, que no es más que una transmisión multimedia que se realiza mediante una red y evita que el usuario descargue todo el contenido para luego poder observarlo.

El objetivo de esta tesina es comprender los conceptos básicos del video digital y de la transmisión de video en vivo. Después de esto se aplica estos conocimientos en la implementación de dos servidores con su respectiva función de transmisión en vivo y se establece un plan de pruebas que se aplica a cada servidor bajo el mismo escenario. Para realizar estas pruebas se arma una red local y se compara diferentes variables realizando el respectivo estudio comparativo. La captura de datos se realiza mediante el uso de herramientas libres como CACTI, ETTERCAP y WIRESHARK, se hizo uso de tres máquinas,

las cuales tuvieron los papeles de servidor, cliente y capturador de tramas de red. Cacti nos permitió monitorear el consumo de CPU y memoria en el servidor cada minuto, por otra parte ettercap y wireshark nos permitió capturar los paquetes transmitidos, poder calcular latencia y variación de retardo. Se utiizo Minitab como herramienta estadística para tratar la información y obtener datos concluyentes. Luego se realiza el respectivo análisis de los resultados con el cual se identifican fortalezas y debilidades de cada servidor. Con estos conocimientos ya se tiene la capacidad de decidir que servidor es el más eficiente en el escenario propuesto.

CAPÍTULO 1

PRESENTACIÓN DEL PROYECTO

En este capítulo se muestra el problema que se presenta al momento de elegir un servidor de transmisión de video en vivo, asi como la metodologia que se uso para realizar los experimentos y los objetivos generales-especificos que se plantearon en esta tesina.

1.1. **ANTECEDENTES**

Antes que la tecnología de transmisión en vivo apareciera en el año 1995 los usuarios tenían que descargar los archivos multimedia directo a su disco duro para poder visualizar o escuchar su contenido. Estos archivos, en especial los de videos, al momento de descargarlos generaban un tiempo de espera y espacio en el disco duro del usuario debido a que eran muy pesados. Para esto se creó la transmisión en vivo el cual nos permite visualizar el contenido sin la necesidad de descargar por completo el archivo haciendo más eficiente el uso de la red. Pero en la actualidad existe una gran variedad de servidores de transmisión de video en vivo para diferentes tipos de sistemas operativos, algunos son de pago y otros son gratuitos, pero todos tienen diferentes características al momento de transmitir y de administrar el servicio, esto hace complicado tomar la decisión de cual servidor sería el más óptimo para un determinado trabajo.

1.2. **DESCRIPCIÓN DEL PROYECTO**

Este proyecto define conceptos básicos sobre la transmisión de video en vivo, video digital y los servidores que se van a implementar, de esta

manera se estableció parámetros para poder realizar pruebas controladas en los servidores de transmisión de video en vivo . Se implementaron dos servidores de transmisión de video en vivo de código abierto los cuales trabajaron bajo el sistema operativo Linux y fueron ejecutados en la misma máquina, a cada servidor se les aplico las mismas pruebas controladas, luego de esto se procedio a la recopilación de los datos para poder analizarlos. De esta manera se obtuvo información para poder realizar el estudio comparativo entre los dos servidores.

1.3. **OBJETIVOS GENERALES**

Evaluar dos sistemas de transmisión de video en vivo de código abierto a través del protocolo HTTP.

1.4. **OBJETIVOS ESPECÍFICOS**

- Explicar definiciones básicas de la transmisión de video en vivo y video digital para poder comprender el uso del mismo.
- Establecer parámetros de configuración de las pruebas que se aplicarán en cada servidor de transmisión de video en vivo.

- Aplicar pruebas controladas a cada servidor de transmisión de video en vivo.
- Analizar los resultados de las pruebas realizadas a cada servidor de transmisión de video en vivo.
- Evaluar los estudios comparativos aplicados a cada servidor de video streaming.

1.5. **JUSTIFICACIÓN**

En la actualidad los usuarios no solo usan la red para comunicarse, también la usan para buscar otro tipo de contenido como: audio, video, documentos, fotos. 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. Y a través de un servidor de transmisión de video en vivo robusto, confiable y administrable se puede asegurar dar el mejor servicio posible a los clientes. En la actualidad existen muchos servidores de transmisión de video en vivo muy buenos que son gratuitos y de código abierto pero el problema es que las empresas que quieren tener este servicio no saben que características

debe tener este para trabajar de una manera eficiente. De aquí la importancia de tener los conocimientos para poder establecer que parámetros se tienen que tomar en cuenta en un servidor de transmisión de video en vivo para poder decidir cuál es el más eficiente en un escenario específico.

1.6. **METODOLOGÍA**

Para lograr comprender el tema se utilizado el método analítico, al descomponer el tema general en varias partes se logro estudiar más a fondo cada uno de sus elementos, en como se relacionan entre sí y con el todo. Luego se realizo experimentos de un hecho en particular, se observo y analizo su proceder. Para finalmente explicar y comprender su comportamiento en los experimentos y de esta manera se estableció el respectivo estudio comparativo. En la Figura 1.1 se muestra el diagrama de flujo donde se explica la metodología que se uso para realizar el estudio comparativo.

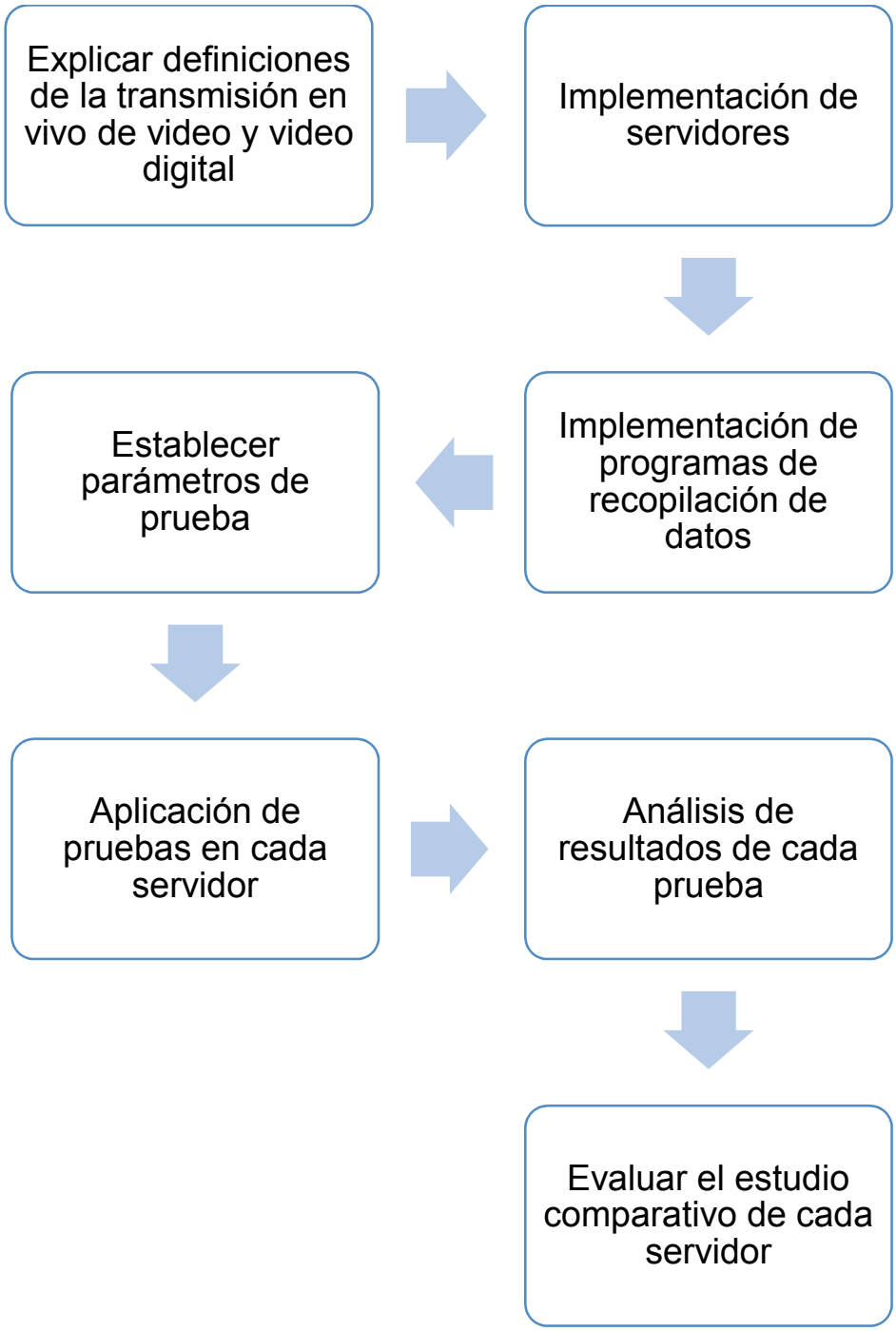


Figura 1.1 Metodologia usada para realizar el estudio comparativo entre dos servidores

CAPÍTULO 2

INTRODUCCIÓN AL VIDEO DIGITAL

En este capítulo se detalla los conocimientos básicos sobre el video digital tales como: compresión, tipos de compresión, normas de compresión, códecs y formatos contenedores. Es necesario tener estos conocimientos debido a que estas son variables utlizadas para enviar el flujo de transmisión en vivo.

2.1. **COMPRESIÓN O CODIFICACIÓN DE VIDEO**

Para poder realizar la compresión de video se realiza la manipulación de las semejanzas o redundancias que existen en una señal de video por medio de algoritmos de compresión que operan con la eliminación de redundancias en el temporal y / o espacial, por lo general existen redundancia temporal en los cuadros consecutivos de una secuencia de video, esto se debe a que generalmente contiene los mismos objetos con una pequeña diferencia de movimientos entre cuadros y la correlación entre las amplitudes de los pixeles de los cuadros cercanos se llama redundancia espacial, los componentes de color verde, rojo y azul que están en un cierto pixel a menudo también están correlacionados [2]. En la Figura 2.1 se muestra el proceso general que se realiza en la compresión de datos de un video.

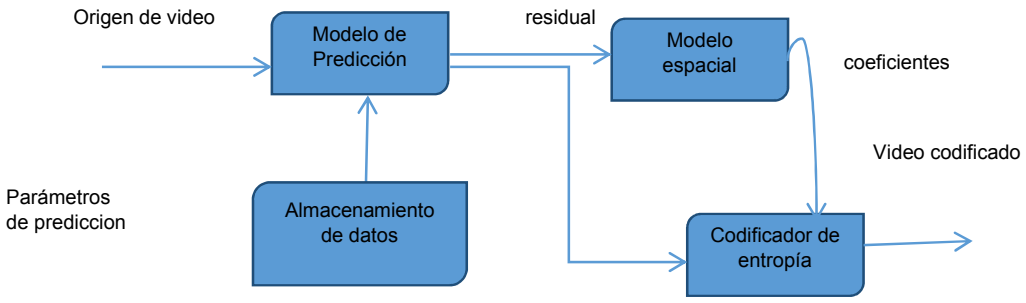


Figura 2.1 Proceso general de compresión de datos

Una de las principales metas de la compresión de video es reducir al mínimo la información irrelevante en una señal. Esto quiere decir que el algoritmo de compresión no gastará bits en información que sea irrelevante y sólo codificará características que tenga importancia perspectiva. La reducción de la redundancia en una señal de video es la parte fácil del trabajo, sin embargo el poder identificar la información que posea importancia perspectiva de la que no la tenga es la parte complicada, por lo tanto es algo difícil de explotar. Lamentablemente para lograr una mayor compresión se sacrifica información al decodificar la señal y esta no será idéntica a la original, pero hay otra codificacion en la cual no se pierde información aunque este no tiene tanto nivel de compresión que el anterior. Al existir en el mercado productos de diferentes fabricantes fue necesario defenir métodos y estándar de codificación en la compresión, de esta manera se fomenta la interoperabilidad y la competencia. Hay que recordar que los videos digitales tienen propiedades que influirán en la codificación del video, el almacenamiento en disco u otros dispositivos y en como se los transmite por la red. También afectará la decodificación del video por parte del programa que posee el cliente. [2]

- **Resolución:** Se refiere al número de pixeles del ancho y alto del video.

- **Número de bits por segundo:** Indica el número de bits por segundo que requiere el reproductor de video. Mientras mayor sea el número de bits, mayor será la calidad y viceversa. Existen videos que usan tasa constante CBR, o una tasa variable VBR.

- **Número de bits por pixel:** Existen diferentes tipos de proporciones de bits para cada color con el cual se codifica video. Una codificación 5:6:5 (rojo, verde, azul), el cual cada número representa el número de bits para cada color, este guarda el color en 16 bits. Al componente verde se le da un bit más porque el ojo humano es más sensible a este componente que a los demás.

2.2. **TIPOS DE COMPRESIÓN**

El principal propósito de la compresión es la reducción del tamaño de la información, procurando siempre tratar que esta reducción de tamaño no afecte al contenido original. Pero esta reducción puede afectar o no a la calidad de la información procesada, esisten dos tipos de compresión:

- **Compresión con perdida**: Al descomprimir datos estos no seran identicos al original pero siempre estaran a una distancia cercana del valor original, esto se aplica a datos los cuales no es de vital importancia recuperar de manera precisa la información, por ejemplo el color original puede tener una tonalidad de rojo y luego de descomprimir este color sera un rojo de otro tonalidad, pero cercano al original.

- **Compresión sin perdida:** Es un procedimiento en el cual el objetivo principal es comprimir cierta cantidad de información ocupando el menor espacio posible y al momento de descomprimir los datos, estos sean exactamente iguales a los datos antes de comprimidos, esto produce que el tiempo de compresión y descompresión sea mayor y se lo realiza con una tasa de bits variables, pero el nivel de compresión sera menor al nivel de compresión con perdidas, este tipo de compresión se lo utiliza especialmente en archivos ejecutables, imágenes y sonidos en los cuales no se permite perder datos por ejemplo cuando un medico observa un eco en tiempo real la descompresión se tiene que efectuar sin perdida ya que es indispensable que no se pierda ningun dato de la imagen.

2.3. **NORMAS Y ESTÁNDARES DE CODIFICACIÓN**

Hoy en día la necesidad de integrar el video a los servicios de telecomunicaciones, empresas, industria del entretenimiento y hogares ha causado que la tecnología del video digital sea algo necesario. Pero el problema de esto es que las imágenes y el video digital ocupan una gran cantidad de espacio de almacenamiento y una gran cantidad de ancho de banda, por este motivo se han desarrollado normas y estándares generales de compresión de video que interpretan el formato del archivo multimedia y hace posible su reproducción. Cada norma son documentos que especifican principalmente dos cosas, una representación codificada la cual se encarga de describir datos visuales en una forma comprimida y un método de decodificación de la sintaxis para lograr recuperar la información visual. Los estándares de compresión de video nos ayudan a garantizar la interoperabilidad o la comunicación entre los codificadores y decodificadores que son fabricados por diferentes empresas, logrando así que la gente confie con mayor rapidez en el producto y pueda tener un uso generalizado. [3]

2.4. **ESTÁNDAR DESARROLLADO POR LA ITU**

Bajo el auspicio de la ITU y la Organización Internacional de Normalización (ISO), fue publicado el H.261 primer estándar de compresión y descompresión de video en 1990, este fue desarrollado para videoconferencia en red digital de servicios integrados y fue diseñado para funcionar con tasas de bits en el rango de los 64 Kbit/s a los 1920 Kbit/s y posee entre 1 y 30 canales ISDN para la transmisión del video. Luego de esto vino el estándar H.263 que fue creado para mejorar la compresión alcanzada por su predecesor. Este estándar soporta nuevos métodos de codificación los cuales proporcionan mejor calidad de imagen usando bajas tasas de bits con un pequeño aumento en la complejidad. Posteriormente se implementaron mejoras que llevaron a una segunda versión del estándar llamada H.263+ y que proporciona 12 modos de operaciones negociables y caracterísiticas que mejoran la calidad de video. A la tercera versión de este estándar H.263++ se le agregaron más opciones para mejorar la calidad de video pero se convirtió en un estándar no muy manejable, debido al gran número de opciones y la necesidad de seguir soportando las funciones básicas del códec . Por último el estándar H.264 permite realizar videoconferencias con una calidad de video mejorada a la misma tasa de bits o la misma calidad

pero con una tasa aproximadamente la mitad de la tasa requerida anteriormente. [2]

2.5. **ESTÁNDAR DESARROLLADO POR LA ISO/IEC**

El MPEG es una serie de estandar los cuales se los describe mas a fondo en el ANEXO I, aquí nos enfocamos solo en el estandar MPEG-4. Este estandar se creo para atender los requerimientos de las aplicaciones multimedia y se adecuado para proporcionar una plataforma común a un amplio rango de aplicaciones multimedia tales como TV digital, juegos, transmisión de video en vivo, etc.

- **Manipulación basada en el contenido y edición de tren de bits:** Esta característica nos da la opción de editar objetos independientes de la escena, en otras palabras se puede realizar edición sin transcodificar la escena completa.
- **Herramientas de acceso a datos multimedia que se basan en el contenido:** Esta característica es muy útil al momento de indexar, buscar, cargar, etc. Con esto se puede dar al usuario una base para la interactividad.
- **Robustez en medios propensos a errores:** Sirve para implementar en medios donde la tasa de errores sea muy

elevada como las comunicaciones móviles, ya que el estándar implementa técnicas para reducir errores.

- **Codificación de trenes de datos concurrentes:** Es la habilidad de codificar eficientemente múltiples vistas de una escena, al mismo tiempo que se aprovecha de la redundancia, como por ejemplo las transmisiónes de escenas de video 3D.

- **Eficiencia mejorada de la codificación:** Ayuda a explotar las características de cada objeto multimedia y de la escena con la implementación de algoritmos para codificar y transmite con calidad en medios que poseen un ancho de banda muy pobre.

Como se puede observar en la tabla I MPEG-4 consiste de 19 partes.

Tabla I Partes de MPEG-4

Titulo	Número
Sistema	ISO/IEC 14496-1
Video	ISO/IEC 14496-2
Audio	ISO/IEC 14496-3
Pruebas de conformidad	ISO/IEC 14496-4
Programa de informe	ISO/IEC 14496-5
DMIF	ISO/IEC 14496-6
Programa de informe	ISO/IEC 14496-7
Transito por redes ip	ISO/IEC 14496-8
Hardware de informe	ISO/IEC 14496-9
Progreso de video	ISO/IEC 14496-10 (H.264)

Tabla I Partes de MPEG-4 (Continuación)

Titulo	Número
Descripción de escena	ISO/IEC 14496-11
Formato de archivo ISO	ISO/IEC 14496-12
Extensiones IPMP	ISO/IEC 14496-13
Formato de archivo MP4	ISO/IEC 14496-14
Formato de archivo H.264	ISO/IEC 14496-15
Extension de animación	ISO/IEC 14496-16
Formato de flujo de texto	ISO/IEC 14496-17
Fuente de compresión	ISO/IEC 14496-18
Sinterizador de secuencia de textura	ISO/IEC 14496-19
Representación de escena liejro	ISO/IEC 14496-20
Extenciones graficos MPEG-J	ISO/IEC 14496-21
Formato de fuente abierta	ISO/IEC 14496-22
Representación simbolica de musica	ISO/IEC 14496-23
Audio y sistema de interacción	ISO/IEC 14496-24
Modelo de compresión de grafico 3D	ISO/IEC 14496-25
Comformidad de audio	ISO/IEC 14496-26
Comformidad de graficos 3D	ISO/IEC 14496-27
Representación compuesta de fuente	ISO/IEC 14496-28

2.6. **THEORA**

Este es un códec de video libre el cual usa un método de compresión de videos con perdida, este esta basado en el códec de video VP3 que es producido por ON2 technologies y este dono el codigo fuente a la fundación Xhiph.Org el cual lo sigue desarrollando, lo más importante de este códec es el bajo consumo de CPU que requiere para poder transmitir. Theora es parte del proyecto ogg el cual busca

la integración de ficheros .ogg, donde este actúa como la capa de video. [4]

2.7. **FORMATOS CONTENEDORES DE VIDEO**

Los formatos contenedores de videos son tipos de formatos de archivos que almacenan información de audio, video y casi la mayoria metadatos, también hay que recordar que el formato contenedor no establece, ni la codificación, ni la compresión de los datos del archivo.

2.7.1. **AVI**

Este formato permite trabajar simultáneamente un flujo de datos de video y varios de audio para que cada fragmento de archivo tenga la información necesaria para poder reproducir algunos fotogramas junto con el sonido que le corresponde, es necesario que todos los flujos se reproduzcan simultáneamente para esto se requiere que se almacenen de manera entrelazada. Chunks son archivos AVI que se dividen en fragmentos bien diferenciados. Cada uno de estos chunk tiene un identificador que se denomina etiqueta FourCC. El que describe información respecto al archivo se denomina cabecera,

este se ubica en el primer fragmento. El segundo chunk es el que posee los flujos entrelazados de audio y video. Y por último puede existir un tercer chunk el cual actúa como índice para el resto de chunks. [5]

2.7.2. **OGG**

Es un formato contenedor que esta libre de patentes y es de codigo abierto, su desarrollador es la fundacion Xiph.Org y además este es el contenedor nativo para los códecs mutimedia que desarrola Xiph.Org. este códec puede contener audio, video y subtitulos en el mismo archivo, también permite reproducir este archivo, tanto en computadoras como en dispositivos moviles con la potencia de procesamiento necesaria. [4]

2.7.3. **MP4**

Se usa para el almacenamiento de los formatos audiovisuales determinados por ISO/IEC y el grupo de MPEG. MP4 es un estándar internacional MPEG-4 de ISO/IEC. Típicamente es utilizado para almacenar datos en archivos para ordenadores,

para la transmisión de flujos audiovisuales. Los códecs más recomendados para mp4 son MPEG-4, XviD, H264.

En la Tabla II se muestra una comparativa de los códecs más usados con la compatibilidad que poseen con los diferentes formatos contenedores.

Tabla II Compatibilidad de códecs con formatos contenedores

Formato Contenedor	**Códecs de videos**				
	THEORA	**MPEG-2**	**MPEG-4**	**H263**	**H264**
.mpg	Si	SI	SI	No	No
.ogg	Si	No	No	No	No
.wmv	Si	SI	SI	No	Si
.mp4	Si	Si	SI	No	Si
.mov	Si	Si	Si	Si	Si
.avi	SI	No	No	Si	Si
.flv	No	No	No	No	Si

CAPÍTULO 3

INTRODUCCIÓN A LA TRANSMISIÓN DE VIDEO EN VIVO

En este capítulo se describe los conceptos básicos para realizar transmisiones de video por medio de una red, además se detalla el proceso general que se tiene que realizar y los tipos de transmisión que se pueden realizar con esta tecnología.

3.1. **CONCEPTO DE LA TRANSMISIÓN DE VIDEO EN VIVO**

La transmisión de video en vivo es un conjunto de técnicas que permiten transmitir paquetes de datos multimedia en forma inmediata y continua esto permite al cliente poder visualizar el contenido sin la necesidad de esperar a que se descargue por completo el fichero. Esta tecnología esta orientada para el uso en internet, para esto se necesitan las herramientas adecuadas que en este caso será el plugin necesario en su navegador web o un software servidor-cliente para la transmisión de video en vivo. Pero hay que tomar en cuenta que el video digital es muy pesado, algo más de 200 MB por minuto en calidad de broadcast, lo cual es óptimo para transmitir por televisión, pero no es adecuado para su uso por internet, ya que ningún cliente va a querer descargar un video de apenas 30 minutos que supere 1GB con las velocidades comerciales actuales. Por esa razón se optó por reducir su tamaño y comprimirlos, por lo tanto, se implementó el uso de códecs y con esto se creó una solución que fue el video/audio en flujo (stream), a lo cual generalmente se lo llama transmisión de video en vivo. [3] En la Figura 3.1 se muestra el proceso general de la transmisión de video en vivo por medio de la red.

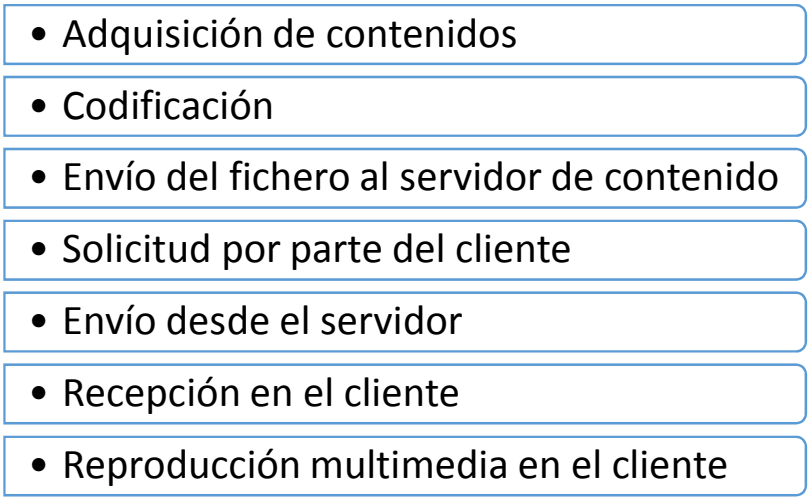


Figura 3.1 Pasos generales para transmitir un video por medio de la red

3.2. **TIPOS DE TRANSMISIÓN DE VIDEO**

Existen dos tipos de transmisiones de videos por medio de la red las cuales poseen diferentes características. En el ANEXO II se describe los diferentes esquemas de distribución que se puede realizar cuando se realiza estos tipos de transmisiones.

3.2.1. **TRANSMISIÓN DE VIDEO EN VIVO**

Este tipo de transmisión de video se refiere al flujo de contenido multimedia en tiempo real. En el que es necesario un software que nos permita editar, codificar y transmitir el contenido desde un servidor que enviará los paquetes a los clientes. Los

protocolos para transmitir en tiempo real por lo general están basados en UDP ya que TCP es un protocolo que permite una conexión y en caso de que ocurra algún error o se pierda algún dato en la transmisión, este se vuelva a retransmitir, lo cual causaría muchas molestias ya que puede ocasionar retardo que degradaria el flujo de trasmisión. La Figura 3.2 presenta el proceso general de una transmisión en vivo.

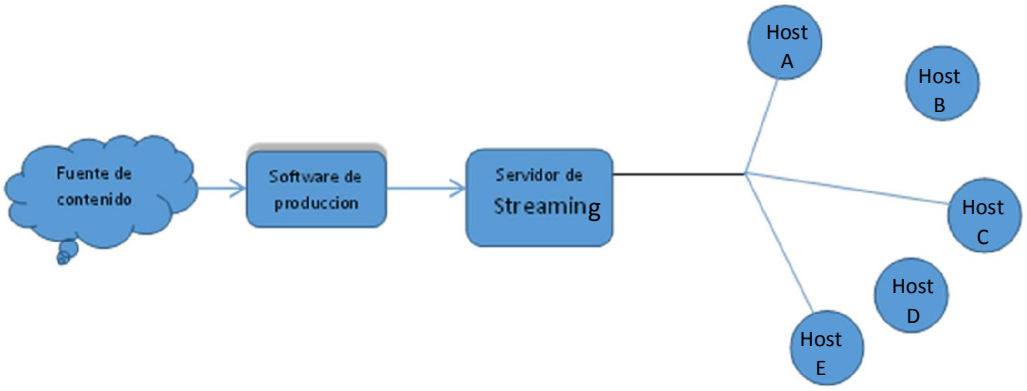


Figura 3.2 Transmisión de video en vivo

3.2.2. **TRANSMISIÓN DE VIDEO BAJO DEMANDA**

Este tipo de transmisión nos permite controlar la visualización del contenido en cualquier momento. Ya que el contenido esta almacenado en un servidor, este esta en capacidad de manejar conexiones individuales los cuales provienen de clientes que envían solicitudes de contenido y aquí es donde el servidor envía el flujo de contenido para que puede ser visualizado en la

máquina del cliente. Esto trae consecuencias ya que se manejan clientes diferentes y estos poseen su propio ancho de banda, por este motivo el número de clientes conectados al mismo tiempo depende directamente del ancho de banda disponible en la red. La Figura 3.3 presenta el proceso general de una transmisión bajo demanda.

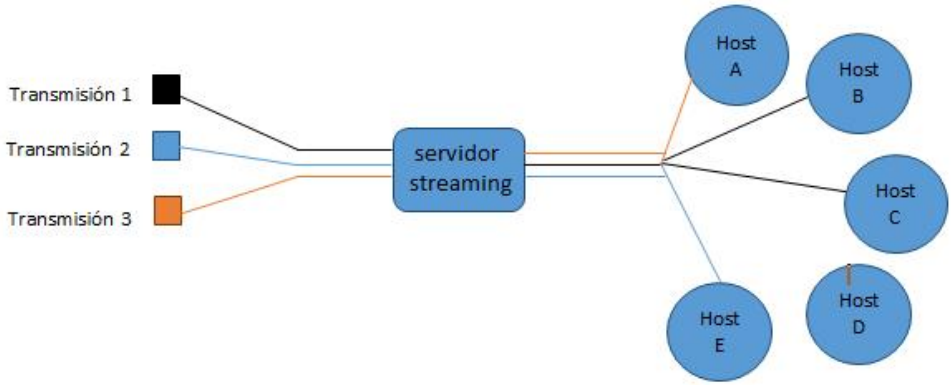


Figura 3.3 Transmisión de video bajo demanda

CAPÍTULO 4

PLAN DE PRUEBAS Y RESULTADOS

En este capítulo se establece la metodología con la que se va a realizar las pruebas, presentación del escenario, parametros que se escogieron para transmitir, software implementados para poder recopilar los datos, asi como el analísis de los mismos con su debida interpretación, conclusiones y recomendaciones.

4.1. **ESCENARIO**

Antes de realizar las pruebas se escogio dos servidores gratuitos los cuales se los describen en ANEXO IV y estos se implementaron en el servidor tal como se muestra en ANEXOS V, posterior a esto se utilizo una formula para determinar el número de repeticiones de las pruebas, la cual se especifica en ANEXOS VII, luego de esto se implemento una red LAN IPv4 formada por tres máquinas, una PC de escritorio que fue el servidor y dos laptops que actuaban como cliente y capturador de tramas de red respectivamente, las características de estas máquinas se describen en el ANEXO VI, se implemento la red LAN por medio inalambrico. En la Figura 4.1 podemos observar un ejemplo del escenario que se implemento para realizar las pruebas.

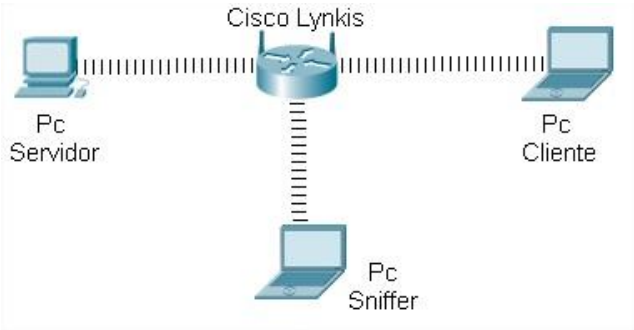


Figura 4.1 Escenario implementado para realizar las pruebas

4.1.1. **DESCRIPCIÓN Y JUSTIFICACIÓN DEL DESARROLLO DE LAS PRUEBAS**

En capítulos anteriores hablamos de como afecta los códec, formatos contenedores y protocolos al momento de transmitir el contenido multimedia, con base a esto se tomó decisiones para escoger las configuraciones de las pruebas. Para la realización de estas se utilizo los equipos mencionados anteriormente en este capítulo, se almaceno un video editado de tres minutos con formato .mp4, se realizo la prueba en cada servidor y para esto se hizo uso de un script que se encuentra adjunto en el ANEXO VIII, este nos permitio realizar las pruebas de forma automática sin intervencion de una persona, luego se usó Minitab cómo herramienta estadística para poder tener datos concluyentes. Se muestra la Tabla III en donde se puede observar la configuración que se uso para las pruebas:

Tabla III Parámetros usados para realizar las pruebas

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

Como se puede observar en la Tabla III se realizo las pruebas en los cuales las variables más importantes a considerar son la tasa de bits, la resolución, códec, formato contenedor, protocolo y fotograma por segundo. Se han realizado experimentos en el cual se demuestra que el ojo humano es capaz de ver imágenes fluidas después de los 24 fotogramas por segundo, por esa razón se decidió utilizar un valor constante de 25 fps. Un fotograma tiene un número de pixeles que tiene altura y anchura, cuanto más resolución tenga, más número de pixeles vamos a obtener, en este caso usamos una resolución de 640x480 que corresponde a las dimensiones de VGA.

La encapsulación del video fue .ogg ya que es un formato que esta libre de patentes y es de codigo abierto, fue diseñado para un alto grado de eficiencia en la transmisión de video en vivo y la compresión de archivos. El códec que se uso es Theora, ya que es un códec de video libre y posee un bajo consumo de CPU, otra de las razones para elegir Theora es que el servidor Icecast solo puede utilizar este códec para transmitir video. La tasa de bits que se uso para codificar los videos es de 800 kbps debido que es una calidad estandar y la resolución que se uso es de calidad VGA con lo cual el video se ve fluido.

Con respecto al protocolo que se uso fue HTTP ya que es un protocolo confiable que está basado en el protocolo de control TCP asegurándose que los paquetes lleguen a su destino y en secuencia, esto quiere decir que si en el camino se llega a perder algún paquete el protocolo lo va a retransmitir.

4.2. **RESULTADOS DE PRUEBAS REALIZADAS CON VLC**

Luego de realizar la n cantiidades de pruebas por medio de un script que se encuentra ubicado en ANEXOS VIII, se ha recopilado los siguientes datos con el software que anteriormente se ha mencionado. En la Figura 4.2 se puede observar la recopilación de datos del consumo porcentual del CPU en el servidor al momento de realizar la transmisión usando Icecast, en el cual podemos ver que en cosumo del CPU es muy elevado y al momento de dejar de transmitir este desciende drasticamente a niveles muy bajos.

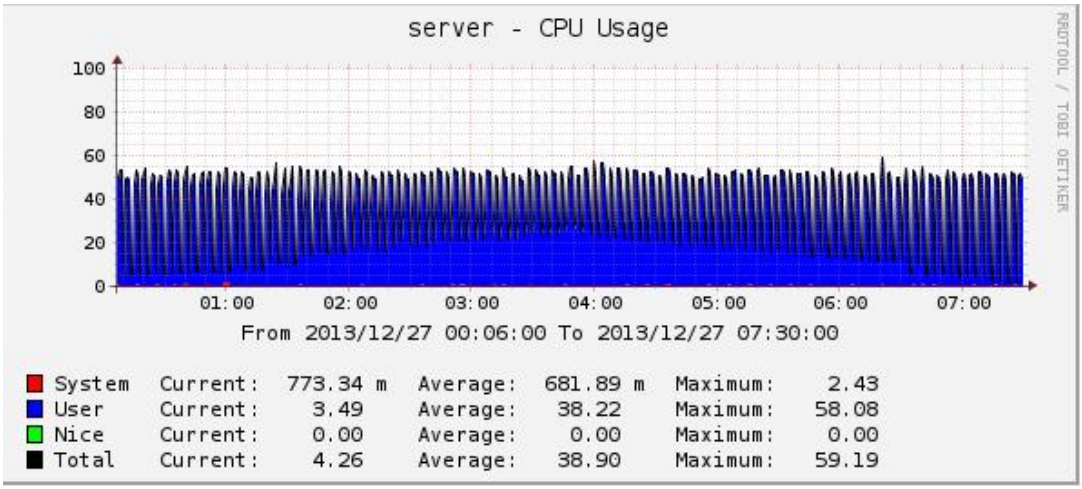


Figura 4.2 Consumo de CPU de VLC

En la Figura 4.3 tenemos el cosumo de memoria que presenta el servidor al momento de hacer la transmisión del video, como se puede observar la variación del consumo de memoria fisica es muy baja en compración con el consumo de CPU del servidor, al analizar estas gráficas podemos concluir que en la transmisión el que tiene que trabajar más es el CPU del servidor. En la Figura 4.4 se encuentra graficado la transmisión de datos en bits vs tiempo en el cual se puede observar que la transmisión fue de tres minutos y un minuto no se transmitio datos, aquí solo se muestra una pequeña parte de la prueba que se realizo.

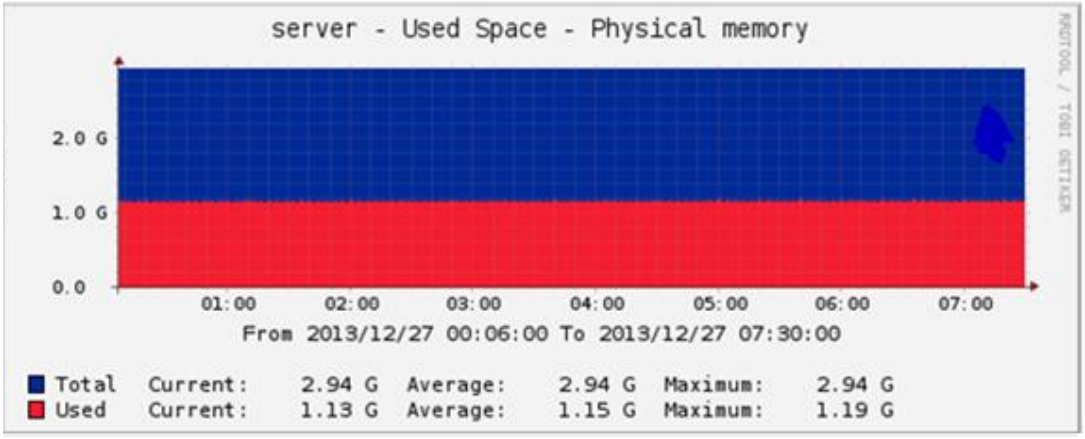


Figura 4.3 Consumo de memoria de VLC

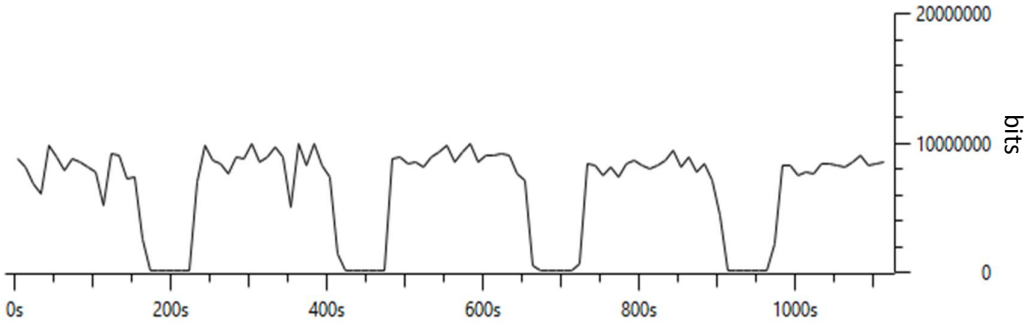


Figura 4.4 Tiempo vs bits de la transmisión en VLC

Luego de tratar los datos recopilados se uso Minitab como herramienta estadística para ayudarnos a tener datos concluyentes. En la Tabla IV podemos ver los datos estadísticos porcentuales de las pruebas de consumo de memoria cuando transmitimos con VLC y en la Figura 4.5 se puede observar que existen pocos intervalos, esto quiere decir que hay poca variación en los datos que se recopilo y también se puede observar al intervalo entre 39.25 a 39.50 el cual posee una frecuencia

más elevada por lo tanto es el dato que más se repitio en la 116 transmisiones del video.

Tabla IV Datos de consumo pocentual de memoria de VLC

Número de datos	Media	Desviación estandar	Varianza	Mínimo	Mediana	Máximo
113	39.374%	0.151%	0.0227%	38.545%	39.390%	39.718%

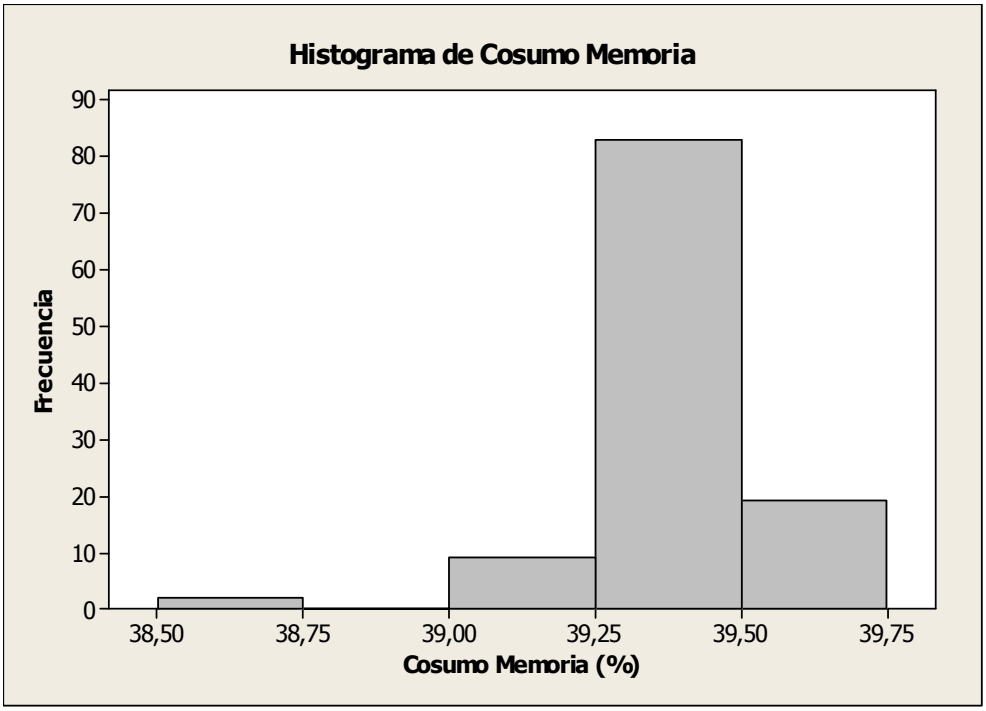


Figura 4.5 Histograma de consumo de memoria de VLC

A continuación se muestra la Tabla V donde están los datos estadísticos de las pruebas porcentuales de comsuno de CPU al momento de transmitir por medio de VLC y se puede observar que

tiene una desviación estandar no muy distante de la media. En la Figura 4.6 se puede ver que el intervalo con mayor frecuencia fue el de 45.12 a 46.16 esto quiere decir que fue el dato que más se repitio a lo largo de la realización de las pruebas.

Tabla V Datos de consumo pocentual de CPU de VLC

Número de datos	Media	Desviación estandar	Varianza	Mínimo	Mediana	Máximo
111	47.169%	2.453%	6.016%	42.51%	46.997%	52.13%

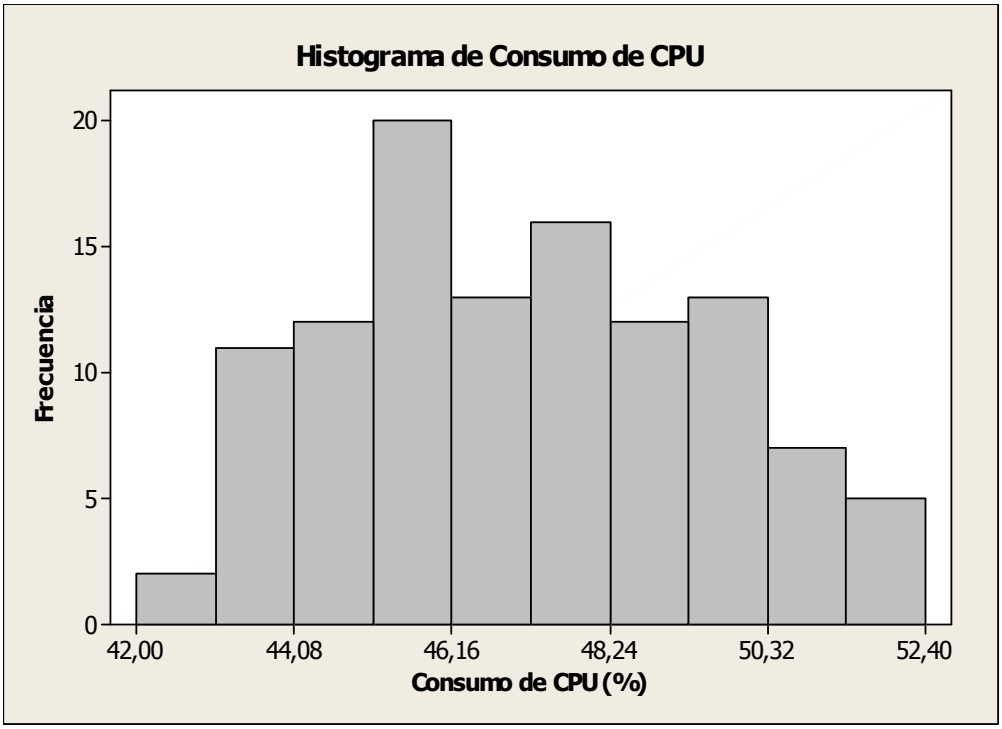


Figura 4.6 Histograma de consumo de CPU de VLC

En la Tabla VI se muestra los datos estadísticos de la latencia que existio en la transmisión de los paquetes usando protocolo HTTP yel

servidor VLC. En la Figura 4.7 se muestra el histograma de los datos recopilados donde se observa facilmente que hay un intervalo en el cual la frecuencia es mucho mayor que los demás y es un valor muy cercano a cero.

En la Figura 4.8 se muestra la gráfica de la latencia vs el tiempo donde la latencia esta dada en segundos, se escogio un tramo de tres minutos de la transmisión de paquetes que fue recolectada por medio del programa Wireshark, esto equivale a los tres minutos de transmisión que es lo que dura el video y de los tres minutos se grafico un punto cada diez segundos mostrando asi el resultado que se observa en la gráfica.

Tabla VI Datos estadísticos de la latencia de paquetes de VLC

Número de datos	Media	Desviación estandar	Varianza	Mínimo	Mediana	Máximo
5648	0.010950 s	0.000117 s	0.0000 s	0.010613 s	0.010962 s	0.016490 s

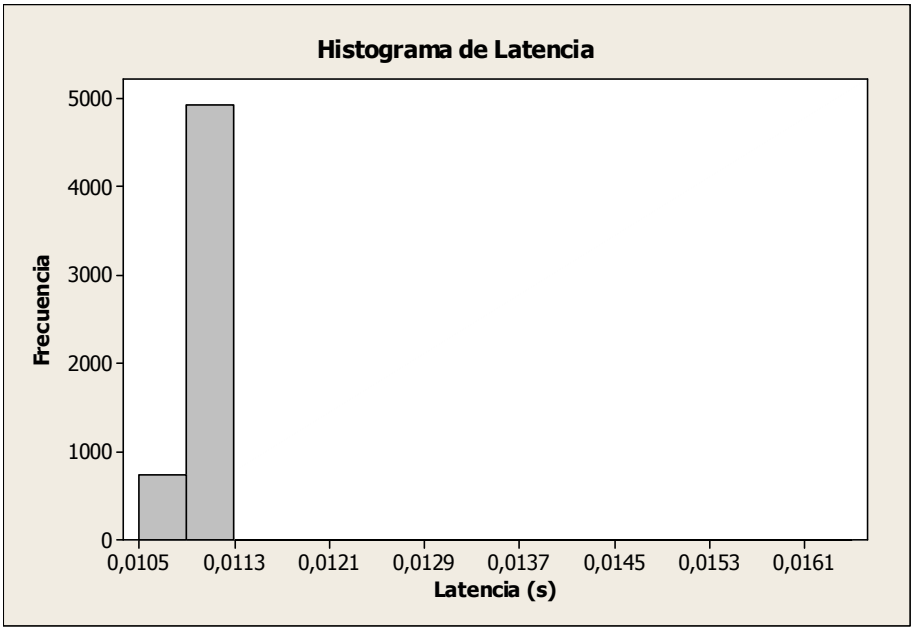


Figura 4.7 Histograma de latencia de pruebas realizadas con VLC

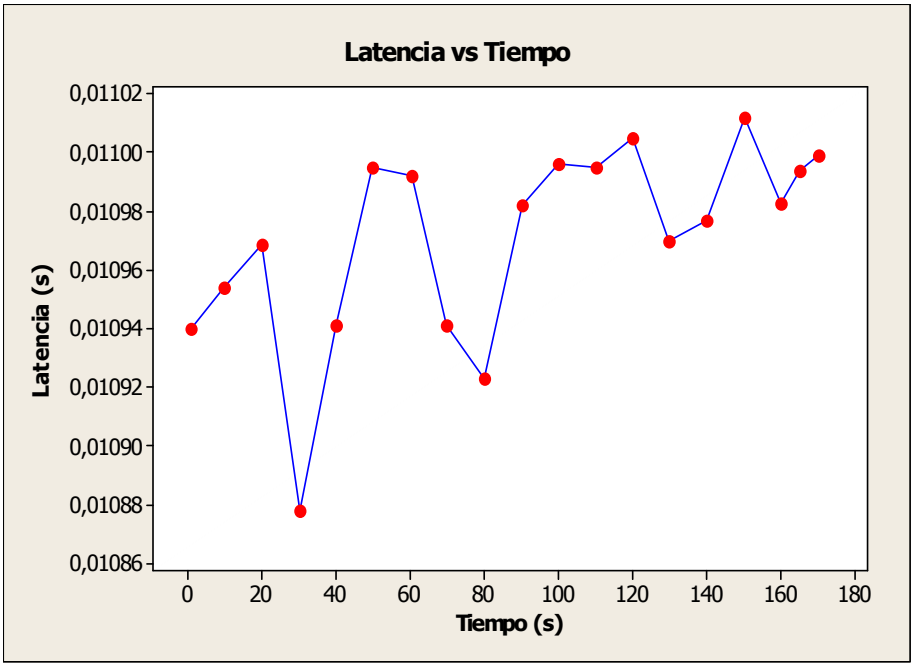


Figura 4.8 Latencia vs tiempo de la transmisión de VLC

En la Tabla VII se puede observar los datos estadísticos de la variación del retardo que existe cuando se transmite mediante el

protocolo HTTP usando como servidor VLC y en la Figura 4.9 se muestra su respectivo histograma donde se puede ver que posee un bajo retardo ya que la cantidad con mayor frecuencia es muy cercana a cero.

Tabla VII Datos estadísticos de variación de retardo de paquetes enviados desde VLC

Número de datos	Media	Desviación estandar	Varianza	Mínimo	Mediana	Máximo
5648	0.000015 s	0.000105 s	0.000000 s	0.00000 s	0.00005 s	0.005475 s

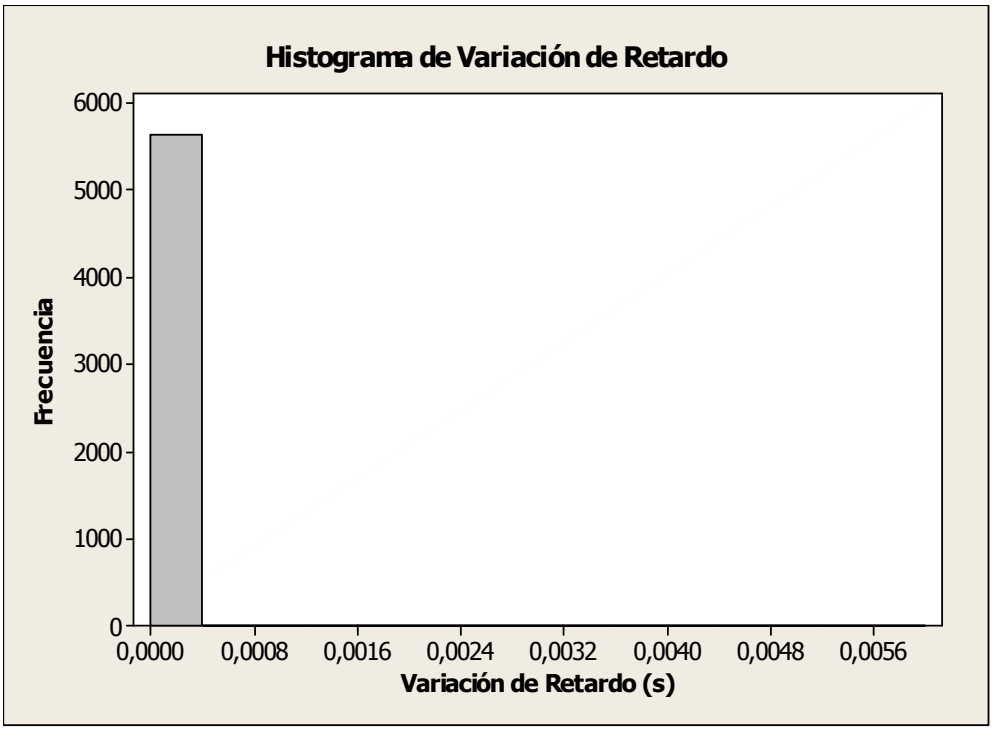


Figura 4.9 Histograma de variación de retardo en VLC

4.3. **RESULTADO DE PRUEBAS REALIZADAS CON ICECAST**

En la Figura 4.10 se muestra la captura de los datos usando Cacti en la prueba de consumo porcentual de CPU transmitiendo mediante Icecast y como se observa el cosumo de CPU en el servidor es muy elevado utilizando los mismos Parámetros para transmitir y el mismo equipo usado en las pruebas anteriores.

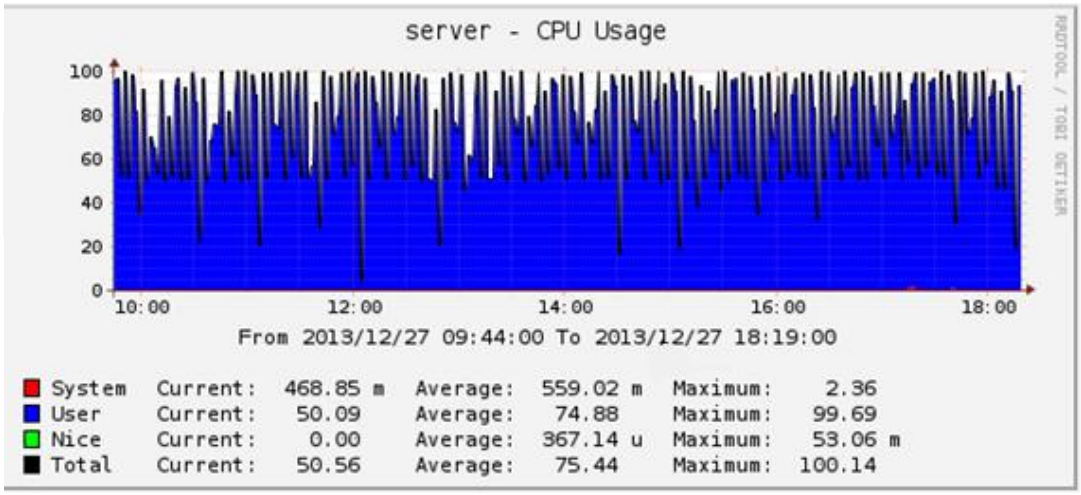


Figura 4.10 Consumo de CPU de Icecast

En la Figura 4.11 se puede observar el cosumo de memoria transmitiendo mediante Icecast, se puede ver que el cosumo de memoria es Mínimo y hay muy poca variaciones de consumo a lo largo de la duración de la prueba. En la Figura 4.12 se puede ver la gráfica

de bits vs tiempo de una parte de la prueba, donde se observa que hay una trasmisión de tres minutos, seguido de otra de un minuto donde no se transmite nada, también se observa que en los tres minutos que dura cada transmisión existe variación de bits con respecto al tiempo.

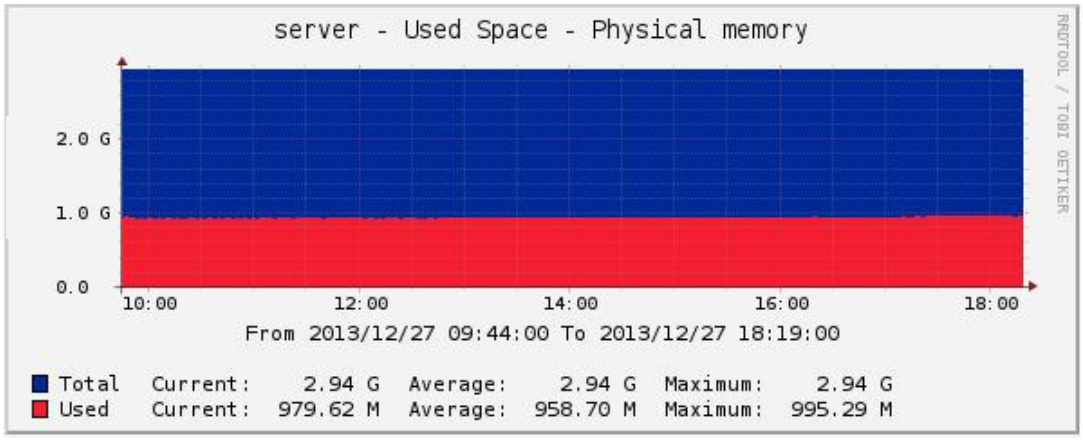


Figura 4.11 Consumo de memoria de Icecast

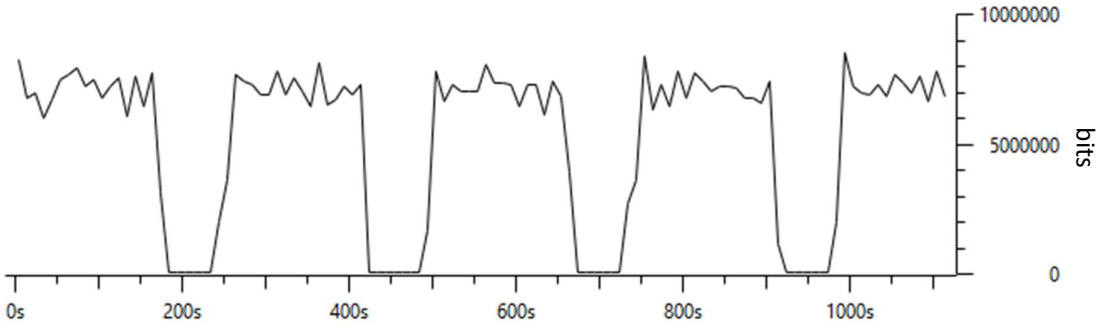


Figura 4.12 Tiempo vs bits en la transmisión de Icecast

Se obtuvo como resultado los datos mostrados en la Tabla VIII, se observa existe una desviación estandar un poco alejada de la media, esto se refleja en la Figura 4.10 donde se ve un cosumo de CPU irregular. En la Figura 4.13 se muestra el histograma de consumo de CPU el cual refleja que el intervalo entre 73.9 a 78.48 es el que tiene más frecuencia, esto quiere decir que fue el dato que más se repitio a lo largo de la prueba.

Tabla VIII Datos de consumo porcentual de CPU de Icecast

Número de datos	Media	Desviación estandar	Varianza	Mínimo	Mediana	Máximo
101	75.315%	9.270%	85.928%	51.858%	75.146%	96.222%

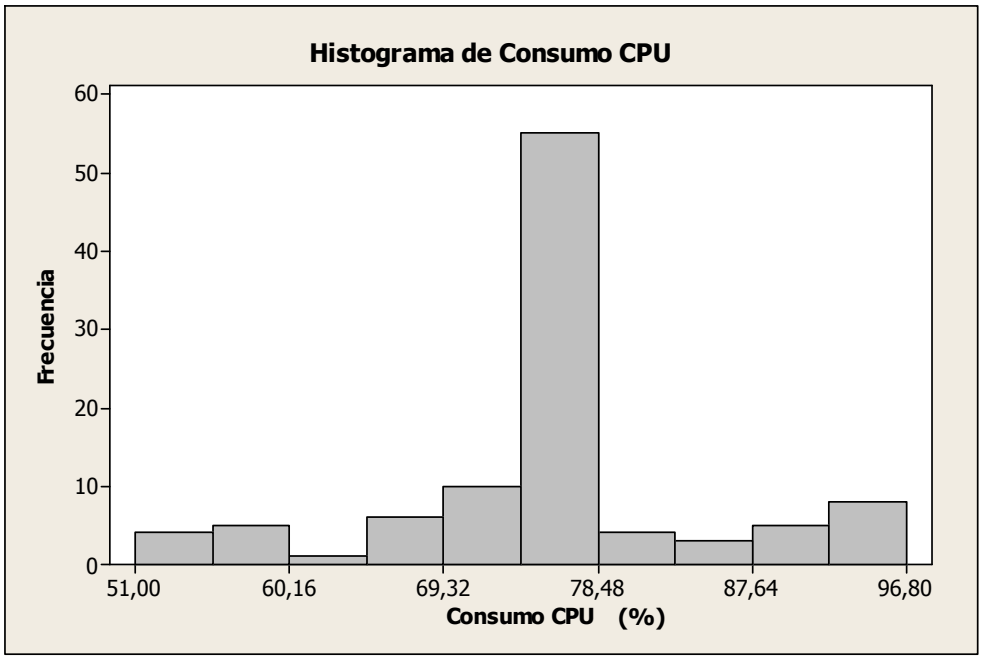


Figura 4.13 Histograma de consumo de CPU de Icecast

En la Tabla IX se muestra los resultados obtenidos del cosumo porcentual de memoria utilizando el servidor Icecast, como se puede observar se tiene una desviación estandar muy baja con respecto a la media que se calculo con Minitab, esto también se lo pudo observar en las pruebas realizadas con VLC y se reflejó en la Figura 4.11 donde se puede observar una variación minima en el consumo de memoria a lo largo de la duración de las pruebas. En la Figura 4.14 podemos observar el histograma de consumo de memoria el cual posee varios intervalos y se puede ver que varios intervalos poseen una frecuencia parecida entre ellos, esto quiere decir que existio un consumo constante de memoria.

Tabla IX Datos de consumo pocentual de memoria de Icecast

Número de datos	Media	Desviación estandar	Varianza	Mínimo	Mediana	Máximo
103	31.876%	0.297%	0.0882%	31.407%	31.810%	32.704%

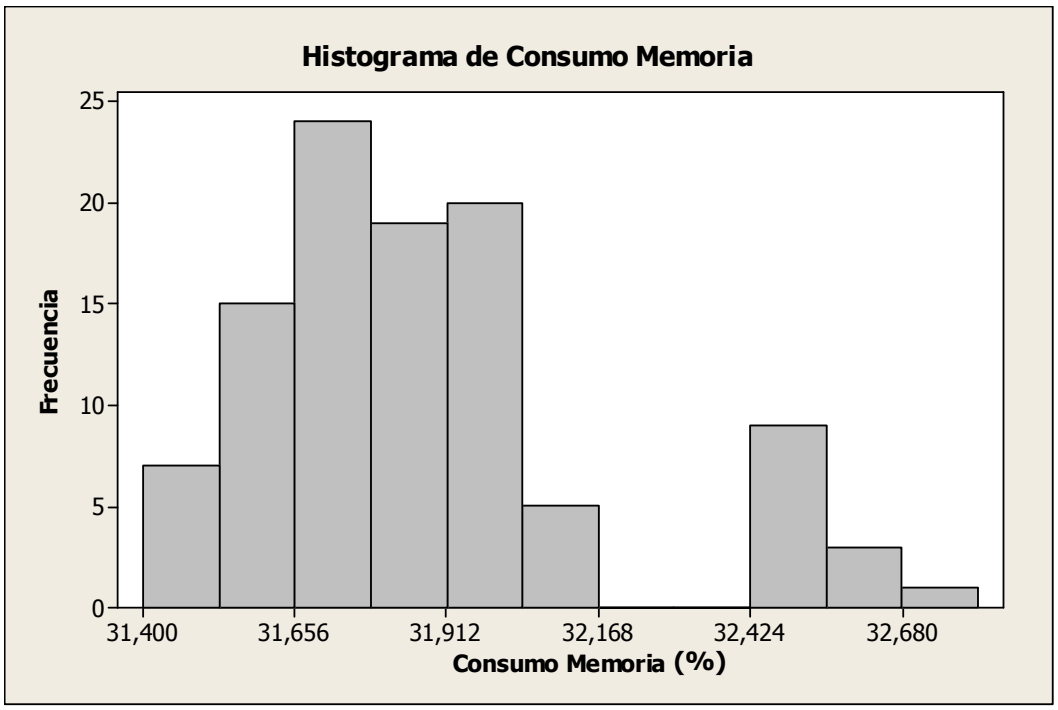


Figura 4.14 Histograma de consumo de memoria de Icecast

En la Tabla X se muestra los datos estadísticos sobre la latencia que existio en la transmisión de los paquetes usando protocolo HTTP con el servidor Icecast y en la Figura 4.15 se muestra el histograma de los datos recopilados donde se puede observar que un intervalo es el que posee mayor frecuencia comparado con los demás.

En la Figura 4.16 se muestra la gráfica de la latencia vs el tiempo donde la latencia esta dada en segundos y asi mismo como se realizo en la Figura 4.8 aquí también se escogio un tramo de tres minutos de

la transmisión de paquetes que fue recolectada por medio del programa Wireshark, esto es equivalente a los tres minutos de transmisión que dura el video que se uso para las pruebas y de los tres minutos se grafico un punto cada diez segundos mostrando el resultado que se observa en la gráfica .

Tabla X Datos estadísticos de la latencia de paquetes de Icecast

Número de datos	Media	Desviación estandar	Varianza	Mínimo	Mediana	Máximo
4768	0.026635 s	0.000038 s	0.00000 s	0.026437 s	0.026620 s	0.026759 s

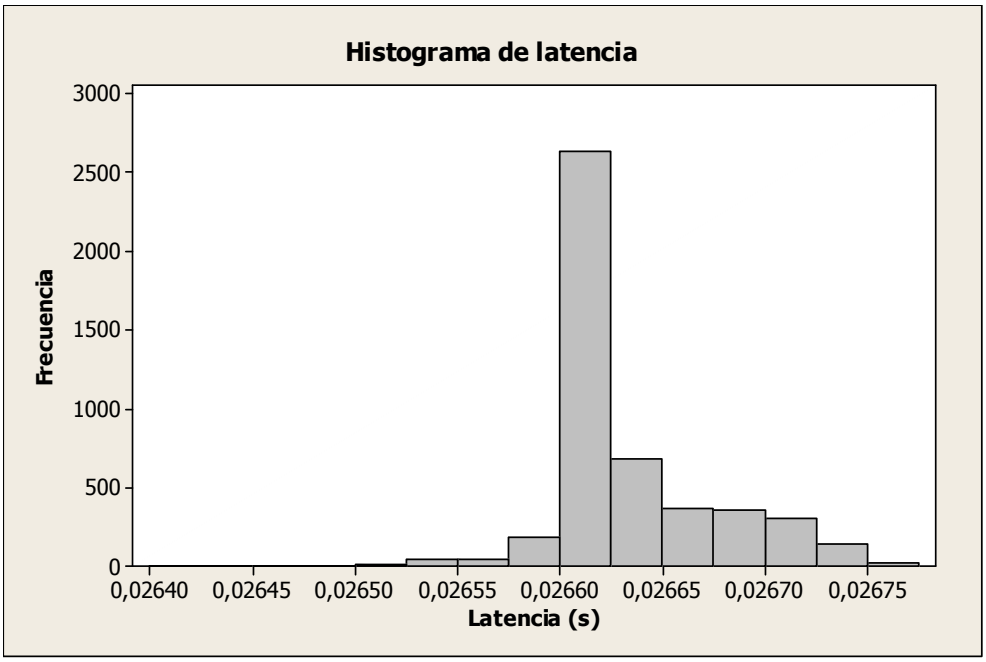


Figura 4.15 Histograma de latencia de pruebas realizadas con Icecast

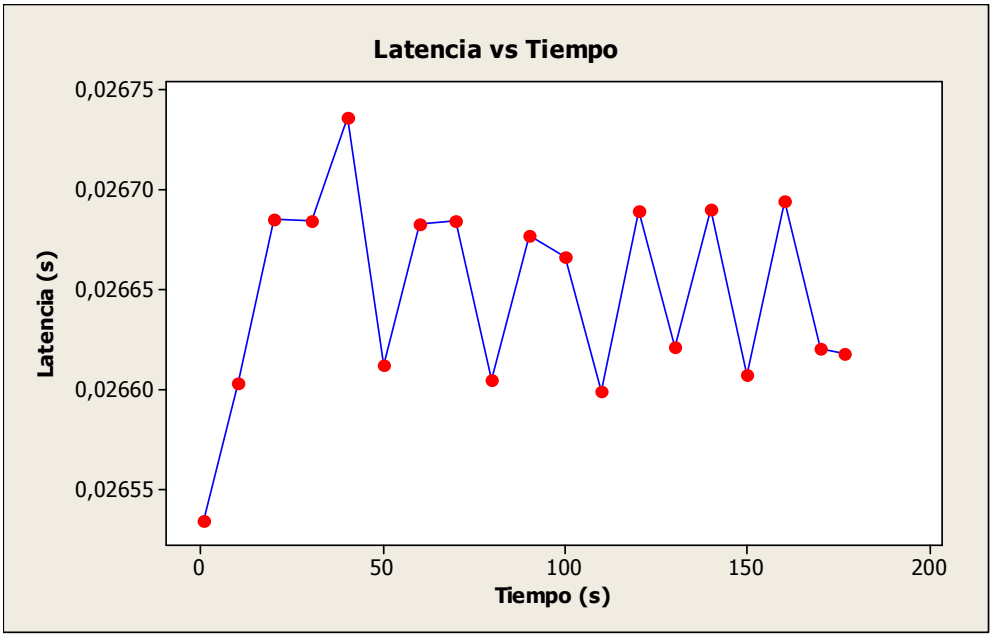


Figura 4.16 Latencia vs tiempo en transmisión usando Icecast

En la Tabla XI se puede observar los datos estadísticos del retardo que existe cuando se transmite por medio del protocolo HTTP usando el servidor Icecast y en la Figura 4.17 se muestra su respectivo histograma en el cual se puede concluir que existio poco retardo en la transmisión ya que el intervalo con mayor frecuencia es el más cercano a cero.

Tabla XI Datos estadísticos de variación de retardo de paquetes enviados desde Icecast

Número de datos	Media	Desviación estandar	Varianza	Mínimo	Mediana	Máximo
4768	0.000010 s	0.000026 s	0.00000 s	0.00000 s	0.00000 s	0.000182 s

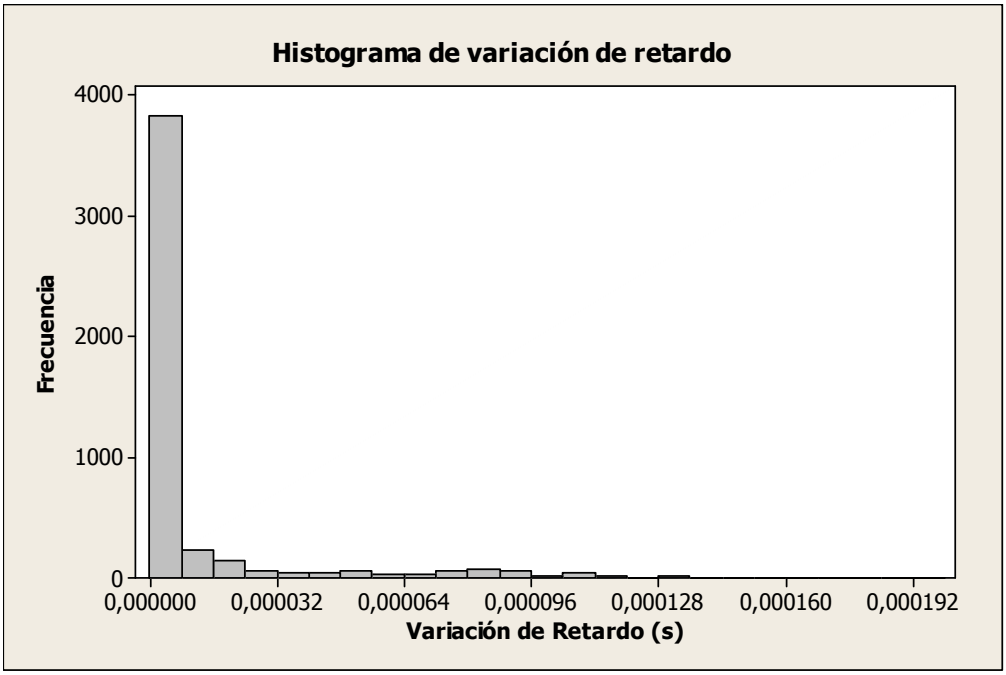


Figura 4.17 Histograma de variación de retardo de paquetes en Icecast

CONCLUSIONES

Con el escenario propuesto y las características que se implementaron al realizar las pruebas se puede establecer las siguientes conclusiones con respecto a la eficiencia de cada servidor implementado en el momento de transmitir un flujo de datos hacia un cliente.

1. Posterior al análisis de las Tablas IV, V y con ayuda de las Figura 4.2, 4.3 se observo que el uso de procesador es 7.79% más elevado comparado con el uso de memoria en VLC y al ver las Figura 4.10, 4.11 y al analizar las Tablas VIII, IX se noto que el comsumo de procesador en Icecast es 43.43% mayor al uso de memoria al momento de transmitir un video en vivo. Se concluyó que el recurso que más se consume al transmitir es el del procesador. VLC cosume 28.14% menos recursos de procesador comparado con Icecast al momento de transmitir, esto se verificó analizando las Tabla V y la Tabla XIII.

2. Con la ayuda de las Figura 4.8 y 4.16 se comparo las gráficas de latencia vs tiempo y con los datos observados en las Tablas X y VI se concluyó que Icecast tiene mayor latencia con una diferencia de 0.015685 segundos respecto a la latencia presente en VLC, pero VLC posee una desviacion estandar 0.000079 segundos mayor a la que posee Icecast esto quiere decir que su latencia varia mas con respecto a su media.

3. 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.

4. Al comparar la Tabla XI y la Tabla VII se observo que en las dos transmisiones existio poca variación en el retardo pero sin embargo en la transmisión realizada con VLC se observo que tenia 0.000005 segundos más de retardo que la que fue realizada con Icecast y este tiene 0.000079 segundos más de desviación estandar 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 obsevar la variación de retardo de los paquetes.

5. Después de analizar todas las tablas y datos que se recopilaron 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 transmisón asignados, pero VLC es más eficiente al momento de transmitir en cuanto a cosnumo 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.

RECOMENDACIONES

Posterior al proceso del estudio comparativo realizado entre los dos servidores se tiene que tener en cuanta algunas recomendaciones basadas en la experiencia que se tuvo al realizar este trabajo.

1. Se recomienda realizar pruebas de calidad de recepción, asi como también pruebas de cantidad de paquetes perdidos para poder determinar con seguridad que transmisión fue mas eficiente al momento de realizar la transmisión de video en vivo.

2. 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 transcodificar sea más facil de realizar y en un menor tiempo, asi el video es menos vulnerable a perder calidad.

3. 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 y si lo que se busca es una transmisión de calidad se aconseja el uso del códec H264 el cual da una muy buena calidad con un consumo, no tan bajo comparado con el de Theora, pero de menor consumo a diferencia de otros códec que entregan la misma o menor calidad.

ANEXO I

ESTÁNDAR DESARROLLADO POR LA ISO/IEC

Los estándares MPEG son independientes de la red específica, lo que nos brinda un punto de interoperabilidad en entornos de red heterogéneos. MPEG trabaja por fases, las fases se identifican con números, estas fases no describen diversas versiones de una única norma, al contrario son normas completamente distintas que se encargan de aspectos diferentes de la comunicación multimedia.

MPEG-1

EL primer estándar que el MPEG incluyó fue MPEG-1, este fue desarrollado para el almacenamiento y distribución de audio y video digital. Este es la base para los primeros videos CD además de ser un estándar popular para videos en internet que son transmitidos como archivos de extensión mpg.

La compresión de MPEG-1 usa técnicas de predicción con compresión de movimiento el cual consiste en reducir con un mínimo de información adicional, esto permite una reducción considerable en la cantidad de información necesaria para transmitir imágenes sucesivas. MPEG-1 no es un video entrelazado, es progresivo y logra alcanzar un bitrate de 1.5 Mbps. Para un efectivo uso la entrada de video primero se convierte al formato estándar de entrada MPEG SIF. MPEG-1 consiste de 6 capas como se muestra en la Tabla I. La capa III de MPEG-1-audio es el estándar más popular para la compresión de audio y es popularmente conocido como MP3 [2]

Tabla XII Partes de MPEG-1

Sistema	ISO/IEC 11172-1
Video	ISO/IEC 11172-2
Audio	ISO/IEC 11172-3
Tasa de bit bajos de audio	ISO/IEC 13818-3
Pruebas de conformidad	ISO/IEC 11172-4
Programas de simuacion	ISO/IEC 11172-5

MPEG-2

El estándar MPEG-2 es usada para muchas aplicaciones HDTV y almacenamiento en DVD entre otras y ahora soporta tasas de bits que van desde los 1.5 Mbps hasta los 60 Mbps. MPEG-2 amplía las técnicas de compresión de MPEG-1, para cubrir imágenes más grandes y de mayor calidad,

a expensas de una tasa de compresión más baja y por ende hace uso de ancho de banda más alto. MPEG-2 posee varias mejoras comparadas con su antecesor, entre estas tenemos: [2]

- Nuevos modos de predicción de campos y tramas para scanning entrelazado.
- Cuantización mejorada.
- Incrementos soportados por accesos aleatorios.
- Soporte resistente para incremento de errores.
- Múltiples programas con un multiplexor (MPEG-1 no puede hacer esto, y esto fue un driver principal para el MPEG-2).

Actualmente MPEG-2 consiste de 11 partes como se muestra en la Tabla II:

Tabla XIII Partes de MPEG-2

Sistema	ISO/IEC 13818-1
Video	ISO/IEC 13818-2
Audio	ISO/IEC 13818-3
Tasa de bit bajos de audio	ISO/IEC 13818-4
Pruebas de conformidad	ISO/IEC 13818-5
Programas de simulación	ISO/IEC 13818-6
Extensiones DSM-CC	ISO/IEC 13818-7
Codigo de audio avanzado	ISO/IEC 13818-8
Extensiones RTI	ISO/IEC 13818-9
DSM-CC de conformidad	ISO/IEC 13818-10
IPMP	ISO/IEC 13818-11

Anexo II

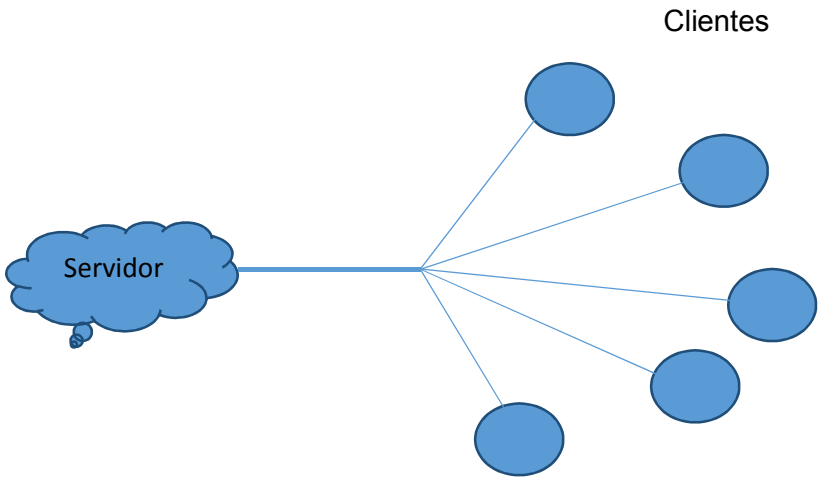
ESQUEMAS DE DISTRIBUCIÓN

Son las diferentes formas en la que se puede transmitir los paquetes de datos a uno a varios clientes al mismo tiempo, aquí se explica cada una de estos esquemas y se los detalla con un ejemplo.

BROADCAST

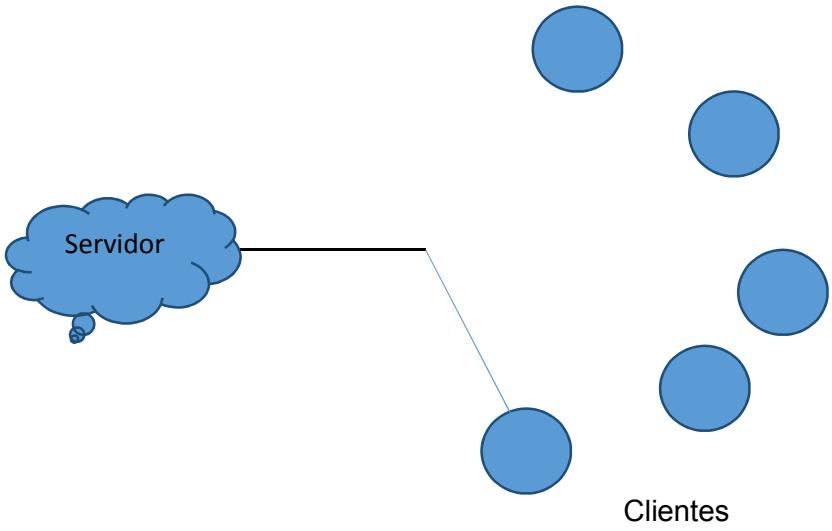
Este término se refiere a la transmisión de los paquetes de datos que van a ser recibidos por clientes que tengan dispositivos que pertenezcan a la red donde se emitió, este es dado para los medios de comunicación en masa. Generalmente ese tipo de distribución se ve limitado por los enrutadores de la red los cuales no soportan tramas broadcast. La Figura muestra un ejemplo de distribución broadcast.

`



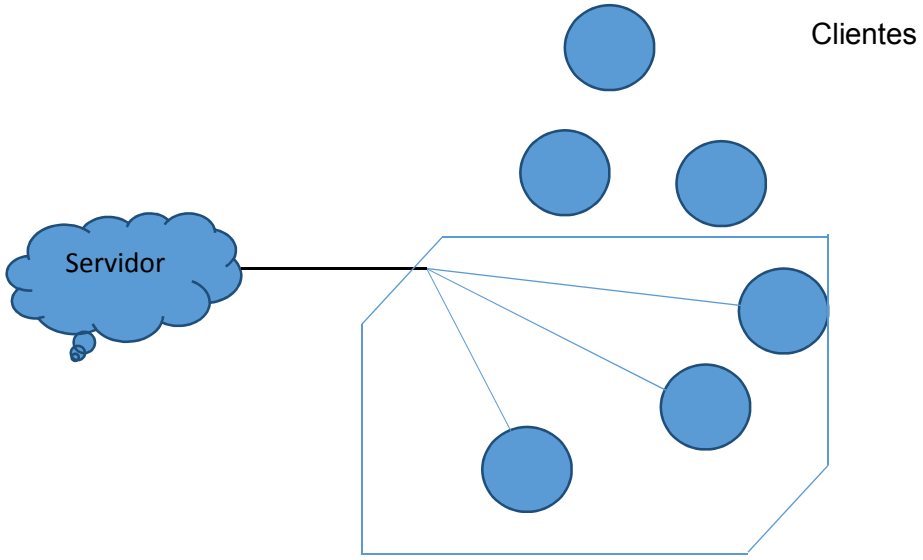
UNICAST

Al contrario que el broadcast el unicast es un término usado para cuando la transmisión de paquetes de datos es hacia un solo destino. Por lo que esto implica que se creará una conexión punto a punto con nuestros clientes y si tenemos n clientes, habrá n conexiones independientes enviando duplicados de la misma información. Cabe destacar que esto mejora la calidad de la transmisión ya que se usa una conexión independiente. La Figura muestra un ejemplo de distribución unicast.



MULTICAST

Esta distribución esta basada en generar grupos de clientes, el servidor puede enviar la misma información a múltiples destinos que este mismo escoje, pero usa el mismo ancho de banda para distribuir los paquetes de datos a sus clientes. Para esto las máquinas de los clientes deben estar configuradas para que puedan escuchar la información que se esta enviando al grupo, también cabe recalcar que todos los routers que se encuentran en el trayecto del servidor-cliente deben estar habilitados para la distribución multicast. La Figura muestra un ejemplo de distribución multiCast.


Clientes
Servidor
Grupo Multicast

Anexo III

PROTOCOLOS PARA LA TRANSMISIÓN DE VIDEO EN VIVO

Existen varios protocolos para realizar las transmisiones en vivo y para realizar una buena transmisión es necesario saber cuales son estos protocolos, aquí se detalla algunos de los mas importantes y basicos.

PROTOCOLO RTP

RTP es un estándar creado por la IETF para la transmisión confiable de voz y video a través de Internet. El protocolo UDP se usa para transportar el tráfico que es generado por RTP ya que este tiene menos pérdidas de paquetes pero es sensible a los retardos. Este protocolo nos permite tener un medio uniforme de transmisión sobre IP de datos que tienen limitaciones de tiempo real. Implementa los números de secuencia de paquetes IP para la reconstrucción de la información de voz o de video, incluso cuando la red subyacente cambie el orden de los paquetes . [6]

De manera más general, RTP permite:

- Identificar el tipo de información transmitida;
- Agregarle marcadores temporales y números de secuencia a la información transmitida.
- Controlar la llegada de los paquetes a destino.

PROTOCOLO RTCP

Es un protocolo adicional que permite el envío de datos de control y datos de mediciones que se realizan en la transmisión. RTCP se lo conoce como RTO. Básicamente la función de este protocolo es transmitir periódicamente los paquetes de control que realizan todos los participantes de la sesión, estos paquetes son enviados aproximadamente cada cinco segundos y poseen datos que nos permiten la verificación de las condiciones de trasmisión en el cliente. [7]

PROTOCOLO TCP

Es un protocolo de conexión segura ubicado en la capa intermedia entre la de internet y la capa de aplicación. TCP retransmite un paquete si detecta que este ha llegado con errores o se ha adelantado a otros paquetes. Para esto hace uso del número de secuencia para ordenar los segmentos TCP recibidos y detectar paquetes duplicados. Para poder realizar la comunicación las dos máquinas deben establecer una conexión, la máquina que esta solicitando la conexión la llamaremos cliente y la otra máquina que la receptará se llamará servidor, por eso se dice que es un ambiente cliente-servidor y esta comunicación se realiza de ambos lados. Para poder garantizar la conexión el protocolo agrupa al agregarle un encabezado a los paquetes de datos que permiten sincronizar las transmisiones. [8]

PROTOCOLO UDP

Este es un protocolo que se basa en el intercambio de datagramas y no necesita de una conexión previa ya que en el datagrama tiene la suficiente información de cabecera para poder llegar a su destino, uno de los aspectos mas importantes de este protocolo es que no tiene comfirmación, ni control de flujo,

esto quiere decir que los paquetes se pueden adelantar uno de otros y que no es seguro que el paquete llegue a su destino correctamente [9].

Anexo IV

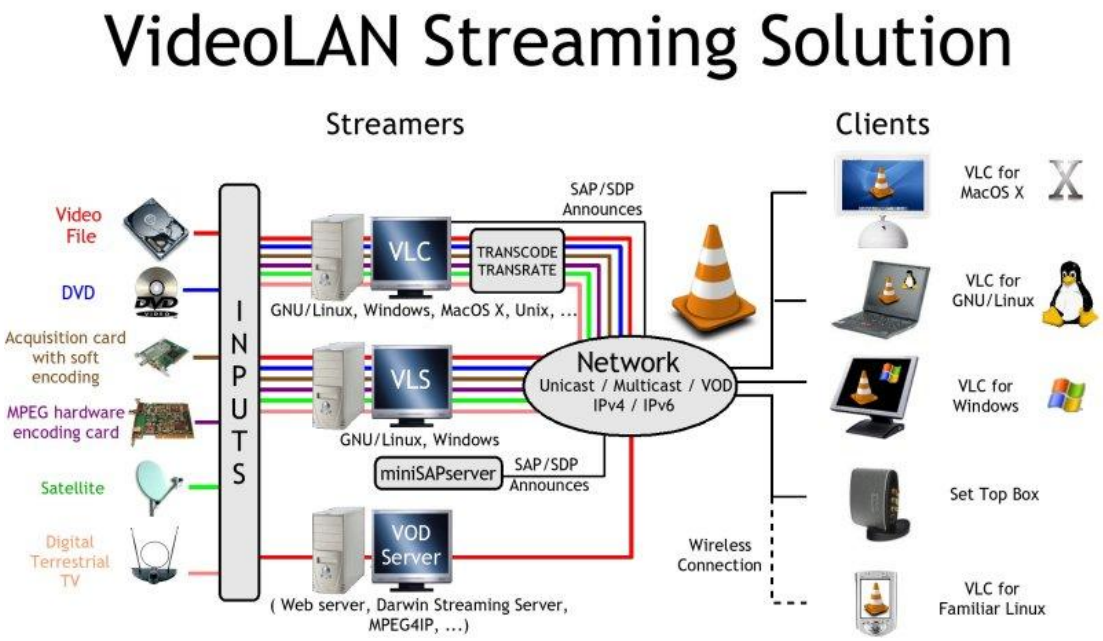
SERVIDORES DE TRANSMISIÓN DE VIDEO EN VIVO

Aquí se se detalla la arquitectura basica y formatos compatibles que poseen los servidores gratuitos que se han escogido para realizar el estudio comparativo, antes de realizar la implementación se da a conocer un poc mas sobre estos servidores.

VIDEO LAN

VLC es un reproductor de audio y video de código abierto que fue creado por el proyecto de video LAN, Organización sin fines de lucro compuesta por desarrolladores voluntarios, bajo el licenciamiento GNU General Public. Básicamente es un paquete de software que esta encargado de los medios de comunicación en un ordenador y a través de internet. Contiene una interfaz intuitiva y posee una arquitectura modular con la cual se puede incorporar fácilmente nuevos códecs, formatos de contenedor y protocolos de transmisión. Este programa es multiplataforma, sus desarrolladores han creado múltiples versiones para diferentes sistemas operativos, es capaz de reproducir muchos códecs al igual que muchos formatos de audio y

video. Posee la capacidad de transmisión de video en vivo y bajo demanda. [10] La siguiente Figura muestra la compatibilidad con los sistemas operativos y opciones de fuente de entrada de datos.

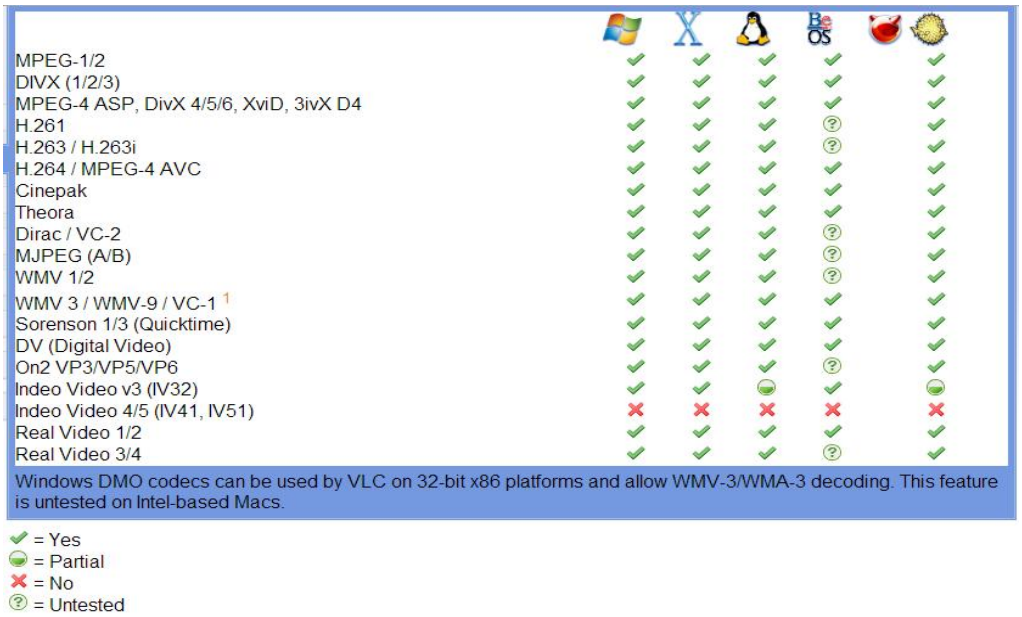


Arquitectura de VLC [1]

FORMATOS DE VIDEOS COMPATIBLES

El código esta escrito en C y es muy difícil de comprenderlo en su totalidad. Una de los componentes más importantes en VLC son los módulos, posee gran cantidad de ellos entre 200 a 400 dependiendo de la construcción, estos son los que dan a VLC la mayoría de las funciones que uno espera. Esta imagen muestra los formatos compatibles en VLC para

diferentes sistemas operativos. En la Figura se muestra la compatibilidad de códec que posee VLC en diferentes sistemas operativos.



MPEG-1/2	✔	✔	✔	✔		✔
DIVX (1/2/3)	✔	✔	✔	✔		✔
MPEG-4 ASP, DivX 4/5/6, XviD, 3ivX D4	✔	✔	✔	✔		✔
H.261	✔	✔	✔	?		✔
H.263 / H.263i	✔	✔	✔	?		✔
H.264 / MPEG-4 AVC	✔	✔	✔	✔		✔
Cinepak	✔	✔	✔	✔		✔
Theora	✔	✔	✔	✔		✔
Dirac / VC-2	✔	✔	✔	?		✔
MJPEG (A/B)	✔	✔	✔	?		✔
WMV 1/2	✔	✔	✔	?		✔
WMV 3 / WMV-9 / VC-1 [1]	✔	✔	✔	✔		✔
Sorenson 1/3 (Quicktime)	✔	✔	✔	✔		✔
DV (Digital Video)	✔	✔	✔	✔		✔
On2 VP3/VP5/VP6	✔	✔	✔	?		✔
Indeo Video v3 (IV32)	✔	✔	◒	✔		◒
Indeo Video 4/5 (IV41, IV51)	✘	✘	✘	✘		✘
Real Video 1/2	✔	✔	✔	✔		✔
Real Video 3/4	✔	✔	✔	?		✔

Windows DMO codecs can be used by VLC on 32-bit x86 platforms and allow WMV-3/WMA-3 decoding. This feature is untested on Intel-based Macs.

✔ = Yes
◒ = Partial
✘ = No
? = Untested

Compatibilidad de codecs de VLC con los sistemas operativos [1]

Existen los siguientes módulos entre otros:

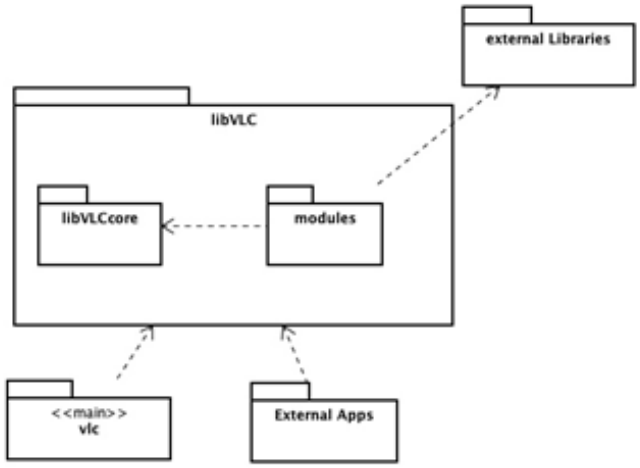
- libVLCcore: Es el núcleo central de la estructura. Se distribuye en la capa orientada a objetos para C, módulo de carga/descarga y un conjunto de funcionalidades relativas multimedia: entrada, multiplexación, demultiplexación, salida de audio, salida de vídeo.

- Módulos: Proporcionan funciones concretas del marco. Los módulos se clasifican de acuerdo a sus capacidades. Hay módulos para la gestión de los insumos (archivos, red, cd), módulos de códecs (mp3, divx,...), módulos de interfaz gráfica de usuario (texto, web, telnet, macosx nativo).

- Bibliotecas externas: Como hay muchos módulos, hay gran cantidad de dependencias externas. Hay una página de desarrolladores de VLC en Wiki que intenta mantener estas dependencias.

- VLC (principal) - Es el principal del reproductor. Se inicializa la interfaz de usuario libvlc y el lanzamiento.

  En la Figura se muestra la arquitectuta de alto nivel de VLC.



ICECAST

Icecast es un servidor de transmisión de video en vivo que es distribuido bajo la licencia GPL de GNU y este es mantenido por la Fundacion Xiph.org sin fines de lucro, el cual desarrolla principalmente formatos de la familia Ogg, principalmente estos son softwares libres que están diseñado para competir con otro formatos como MPEG-4, Windows Media Video. Al implementar este servidor podemos transmitir audio y con ayuda de complementos se puede implemetar una radio en internet, en nuestro caso Icecast lo usamos como servidor de video para esto se necesito instalar un complemento llamado ffmpeg2theora, el cual es una aplicación muy potente que permite usar todo tipo de formato de video de entrada y este lo transforma al formato .Ogv el cual es compatible con Icecast. [11]

PLATAFORMAS COMPATIBLES

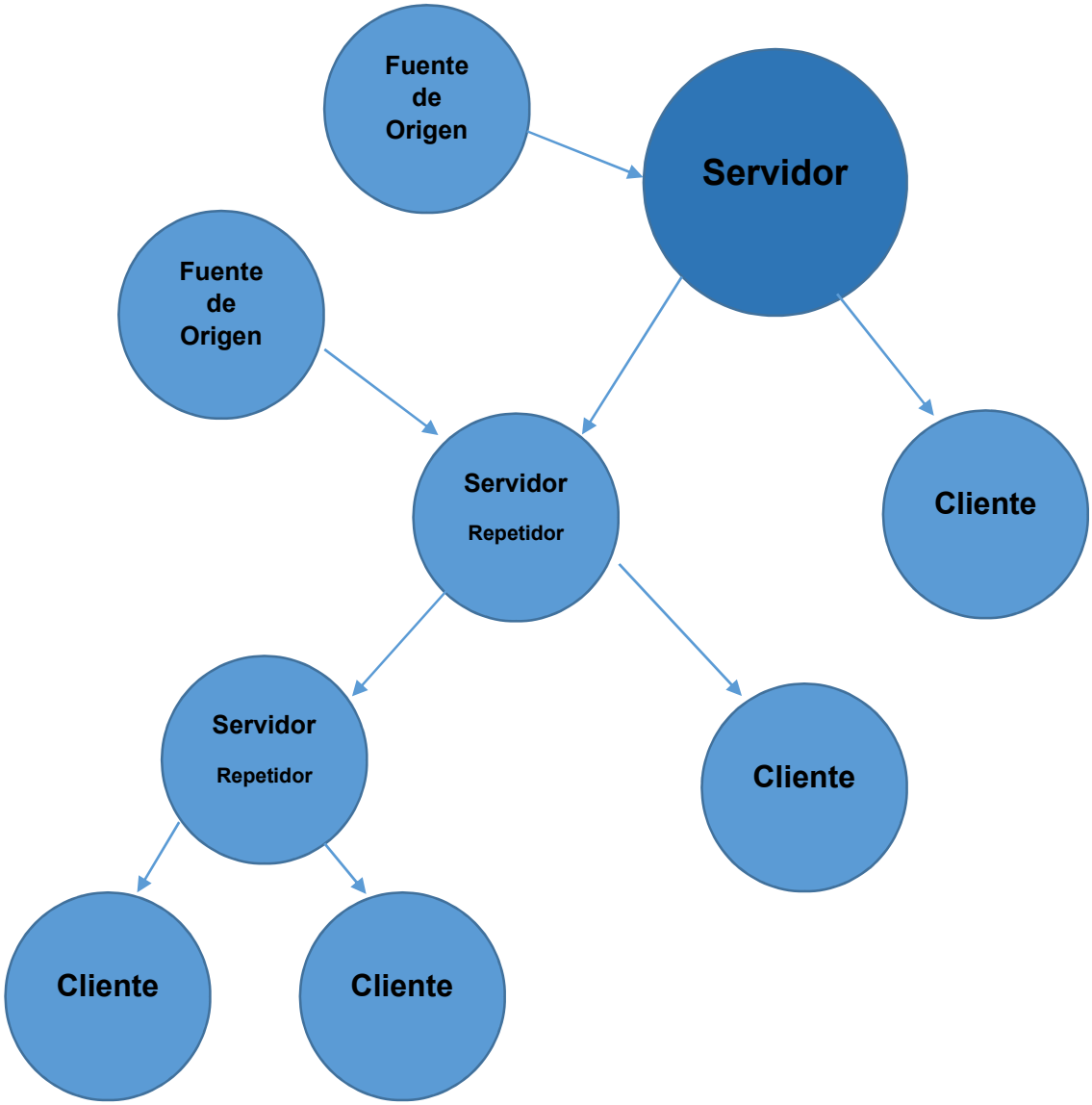
En la actualidad Icecast es compatible con las plataformas Unix y Windows . Respecto a lo que plataformas de Windows se refiere es compatible desde Windows 2000 en adelante y dentro de lo que es Unix tenemos:

- Linux (todas las distribuciones)
- FreeBSD
- OpenBSD
- Solaris

ARQUITECTURA BASICA

La arquitectura basica de Icecast es muy simple y obedece a lo que es Servidor, Cliente y Servidor Repetidor. La mayor parte de trafico va a traves del protocolo HTTP, estos flujos de datos solicitan peticiones mediante Get HTTP este servidor solo abre un puerto el cual es utilizado para escuchar todas las transmisiones. Para entender un poco más esta arquitectura podemos observar la Figura en la cual se aprecia el cliente, fuente de origen que tiene como función contener los archivos multimedia,

que luego seran enviado cuando el servidor los requiera. Por medio de una linea de comando en la cual la especifica la ip, puerto y contraseña de Icecast que fue colocada al momento de la instalacion del programa podemos transmitir de forma remota estos archivos por medio del servidor. Otra parte de la arquitectura es el servidor de repeticion el cual actua como un cliente más, pero este a la vez actua de servidor repitiendo la transmisión a diferentes clientes. [11]


Fuente de Origen
Servidor
Fuente de Origen
Servidor Repetidor
Cliente
Servidor Repetidor
Cliente
Cliente
Cliente

ANEXO V

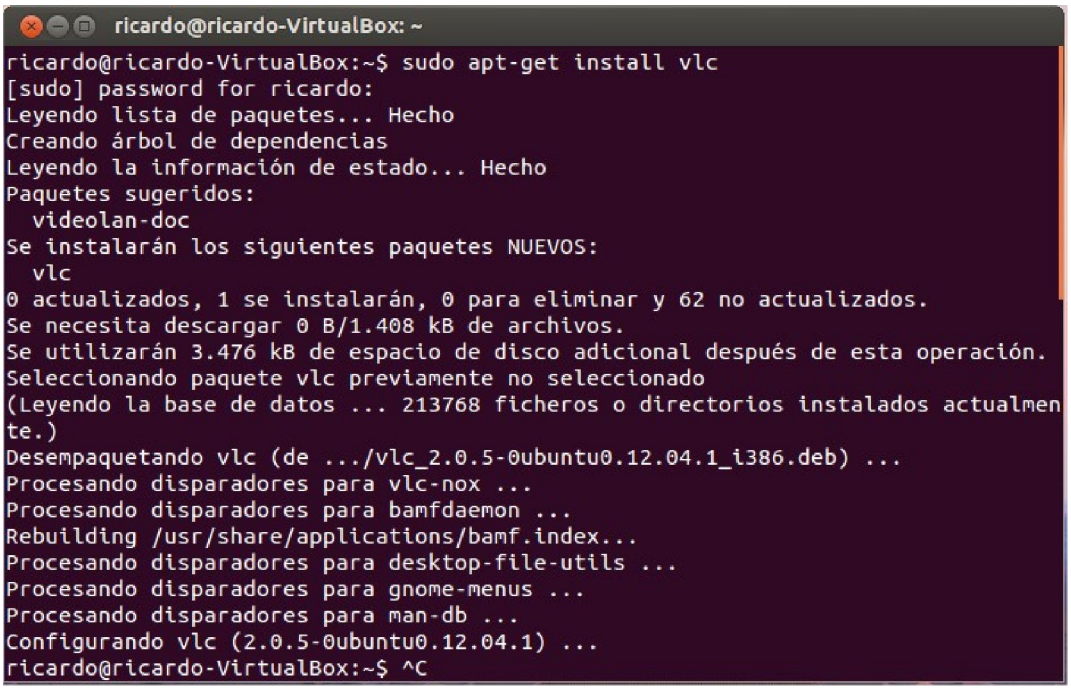
IMPLEMENTACIÓN DE SERVIDORES DE TRANSMISIÓN DE VIDEO EN VIVO

Uno de los objetivos de esta tesina es realizar el estudio comparativo de servidores de transmisión de video en vivo, para estos vamos a instalar dos tipo de servidores, VLC y Icecast.

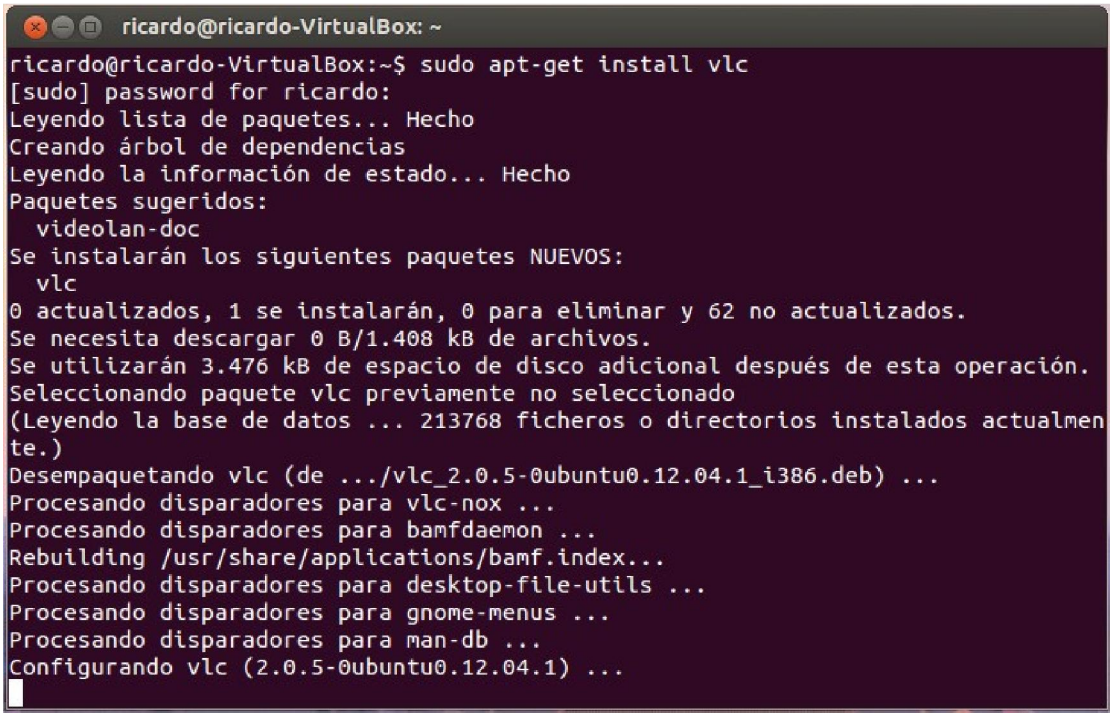
IMPLEMENTACIÓN DEL SERVIDOR VLC

La implementación del servidor VLC será el sistema operativo linux (Ubunto 12.10) y el cliente será VLC que es compatible con Windows o Linux. Este es uno de los software más completos, sencillos y uno de las más usados ya que tiene la versatilidad de poder usarse en modo servidor y modo cliente.

1- Empezaremos la instalación en ubuntu por línea de comando como se puede observar con la línea de comando sudo apt-get install vlc se instalara todos los paquetes de vlc



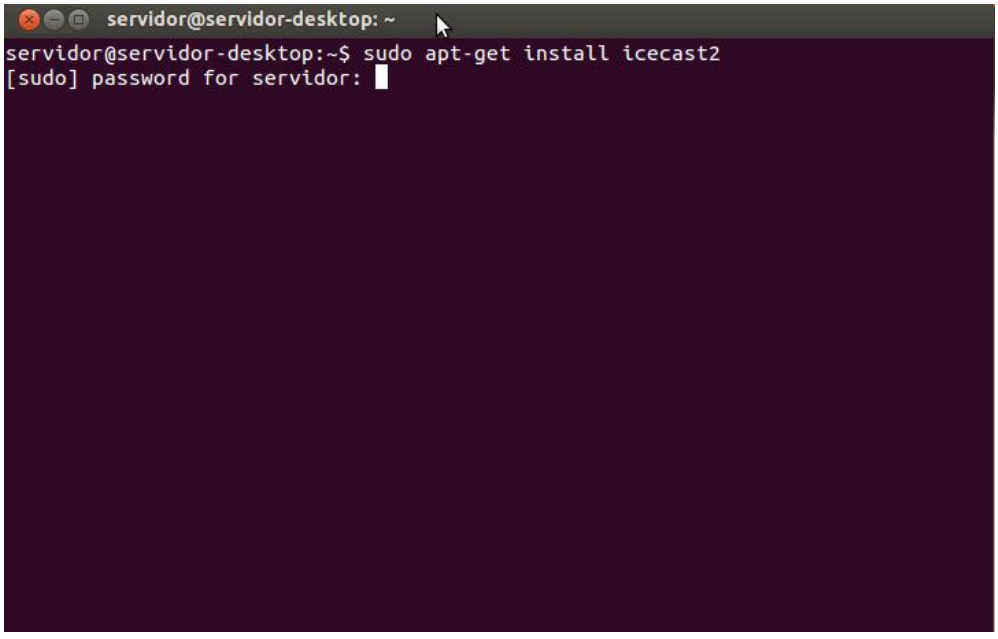
2- Adicional a esto instalaremos un plugin de vlc para Mozilla



IMPLEMENTACIÓN DEL SERVIDOR ICECAST

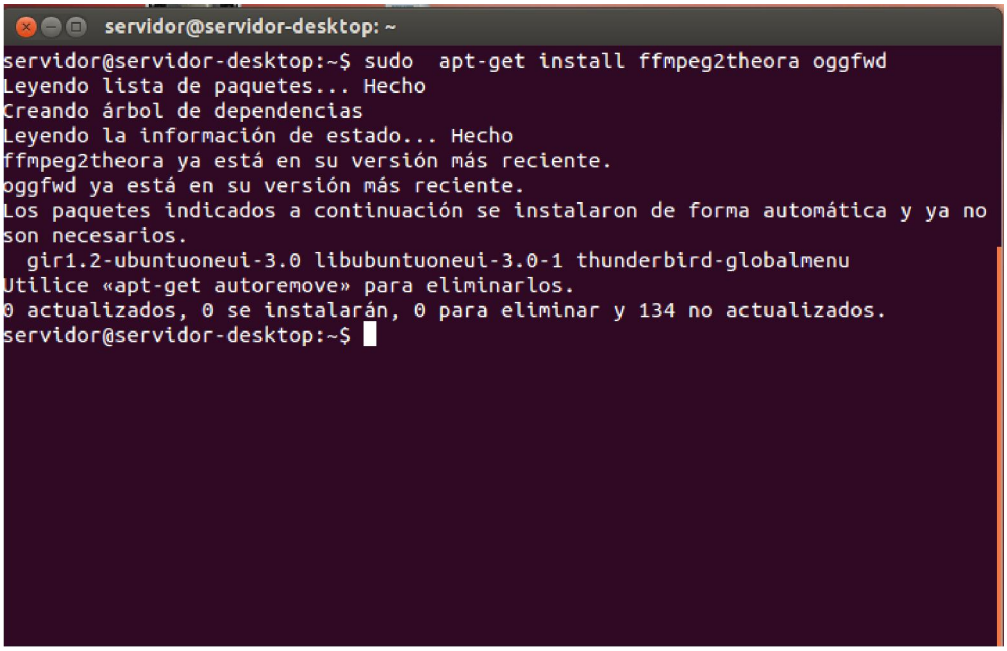
El servidor icecast es originalmente usado para hacer transmisiones de datos de audio, pero con ayuda de un plugin icecast es capaz de transmitir audio y video, acontinuación se va a mostrar la instalacion en linux mediante linea de comando.

1- El primer paso es instalar el servidor como tal:



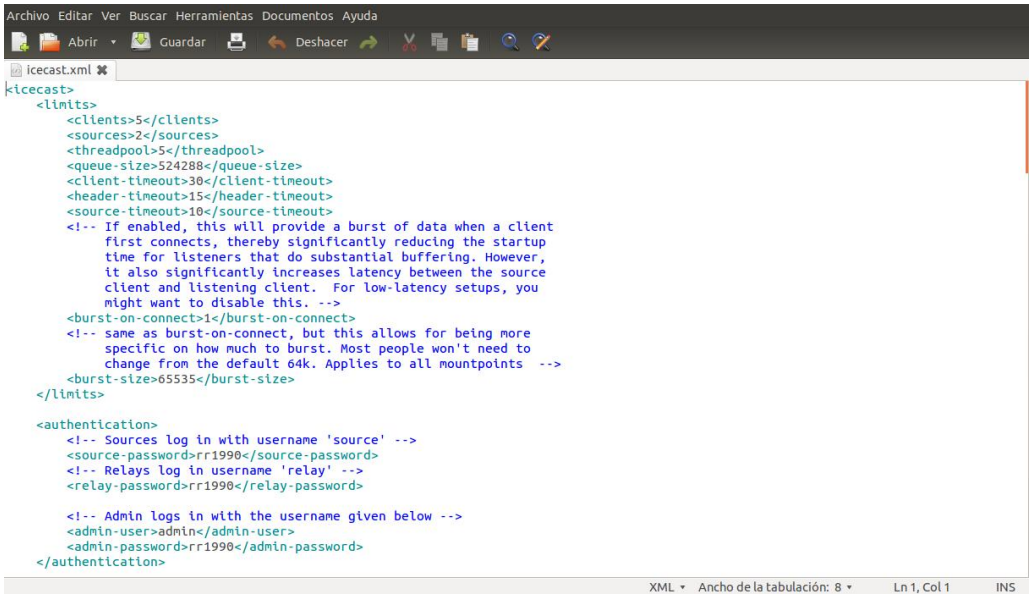
2- Instalamos el servicio de conversion de video theora y un plugin que nos permite transformar otros formatos al formato ogv para poder ser

transmitido.

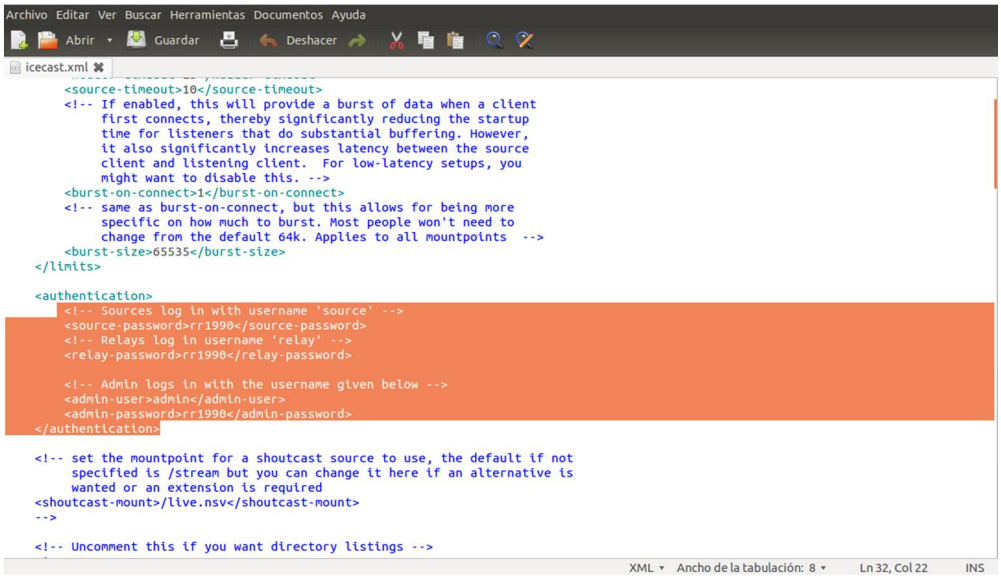


3- Configurar icecast para esto vamos al archivo de configuración .

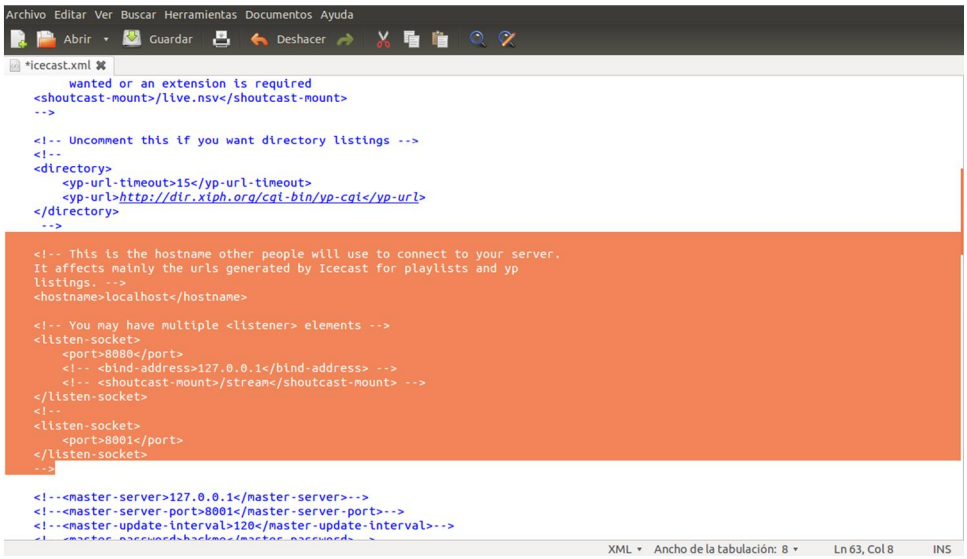
4- Aquí se configura tiempo de espera, número de clientes, número de transmisiones, entre otras cosas.



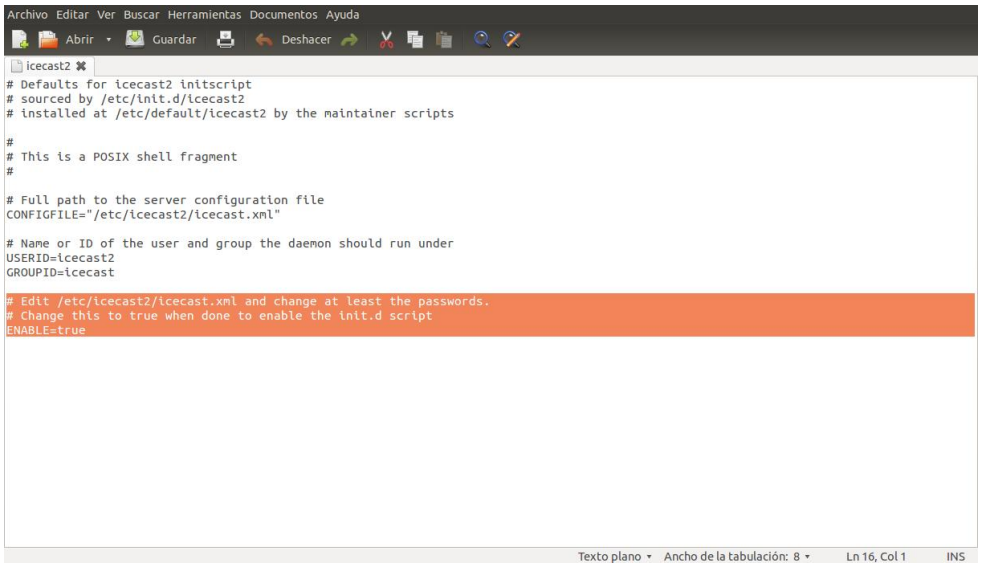
5- En el mismo archivo pero más abajo se encuentra las lineas donde se configura la clave y el admin para la cuenta y para la transmisión.



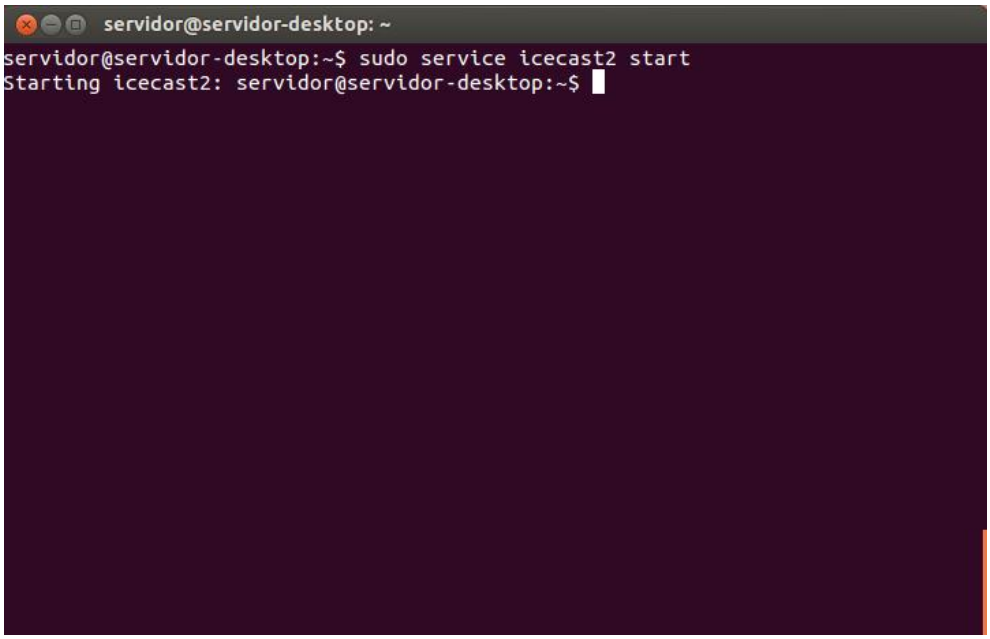
6- En el mismo archivo configuramos el ip del servidor y el puerto por el cual se va a transmitir.



7- Luego de esto vamos a la ruta /etc/default/icecast2 y configuramos este archivo que por default esta false, lo modificamos y ponemos true



8- Una vez realizada todos estos pasos iniciamos el servicio y esta todo listo para usar icecast


servidor@servidor-desktop: ~
servidor@servidor-desktop:~$ sudo service icecast2 start
Starting icecast2: servidor@servidor-desktop:~$

ANEXO VI

EQUIPOS PARA IMPLEMENTAR ESCENARIO

Es importante dellatar las características de cada hardware usado al realizar las pruebas ya que esto puede influenciar en los resultados de los datos que se van a obtener, hay que resaltar que todas las pruebas fueron realizadas con el mismo hardware.

SERVIDOR

El servidor fue una computadora de escritorio en la cual fueron instalados IceCast y VLC. Se almaceno el contenido multimedia para así poder transmitirlo al cliente, dicha computadora tiene un sistema operativo Linux Ubuntu 12.04 y posee las siguientes características en hardware: CPU Intel Pentium Dual Core (R) 3.40 GHz, memoria RAM de 3 GB y un disco duro de 100 Gb.

CLIENTE

Para el cliente se uso una laptop Toshiba Portege R835-p94 con Sistema operativo Linux Ubuntu 12.04 con Procesador Intel core I5 2,5GHz, 8 Gb de RA y 640 Gb disco duro, aquí se ejecuto los reproductor multimedia para poder visualizar el contenido enviado por el servidor.

CAPTURADOR DE TRAMAS DE RED

Se utilizo una laptop Toshiba Satélite S845 con sistema operativo Linux Ubuntu 12.04 con procesador intelCore I5 2.5Gb, 8Gb de RAM y esta se uso como capturador de tramas de red, el motivo de usar otra máquina solo para funcionar como capturador de tramas de red es para no alterar los datos de comsumo de recursos que queremos obtener entre el servidor y el cliente.

PUNTO DE ACCESO

Para crear la red LAN utilizaremos un Linksys WRT54G2 de cisco con una velocidad de transferencia de datos de 54 Mbits/s por medio inalámbrico. Para realizar las pruebas vamos a usar este medio con la finalidad de tener la facilidad de que el capturador de tramas de red pueda escuchar la conversacion entre el servidor y el cliente para asi poder obtener la información requerida.

ANEXO VII

CALCULO DE CANTIDAD DE

REPETICIONES EN LAS PRUEBAS

Se determino un número n = 116 pruebas con un nivel de confianza del 95%, un poder estadistico del 88% y una diferencia minina del 10%, usando la formula: [12]

$$n = (W - W^2 * Z_\beta + 1{,}4 * Z_\alpha)/W^2$$ [12]

Donde,

n = Número mínimo de muestras, observaciones o réplicas que deben efectuarse en el estudio.

Zα = Valor correspondiente al nivel de confianza asignado (Riesgo de cometer un error tipo I).

Zβ = Valor correspondiente al poder estadístico o potencia asignada a la prueba (Riesgo de cometer un error tipo II).

W = Rendimiento mínimo esperado, eficiencia mínima esperada o diferencia mínima observable.

El procesador posee varios campos de consumo en los cuales se basan en tres Parámetros principales que son:

%user : Porcentaje de uso en el nivel de usuario

%system: Porcentaje de uso en el nivel del sistema (kernel)

%nice: Porcentaje del tiempo que un proceso de usuario se ejecuta con la prioridad variada con nice

ANEXO VIII

SCRIPT UTILIZADO PARA TRANSMISIONES

EN LAS PRUEBAS

```
while [ $Num -le 110 ]; do
     echo "\$Num: $Num"
     cvlc
     file:///home/servidor/Escritorio/WillIAm_Britney_Spears_Scream_S
     hout.mp4 --start-time 111 --sout
     '#transcode{vcodec=theo,scale=1,width=640,height=480,acodec=mpga
     ,vb=800,ab=128}:standard{access=http,mux=ogg,url=192.168.1.108:8
     080}'    vlc://quit

    sleep 60
    let Num=$Num+1
done

sleep 900

NumICE=0

while [ $NumICE -le 110 ]; do
     echo "\$NumICE: $NumICE"
     ffmpeg2theora -V 800 -s 111 -x 640 -y 480 -A 128 /home/servidor
     /Escritorio/WillIAm_Britney_Spears_Scream_Shout.mp4 -o /dev/
     stdout | oggfwd 192.168.1.104 8000 rr1990 /out.ogv

    sleep 60
    let NumICE=$NumICE+1
done
```

BIBLIOGRAFÍA

[1] «videoLan Organization,» [En línea]. Disponible: http://www.videolan.org/vlc/. [Último acceso: 26 10 2013].

[2] I. E. G. Richardson, H.264 and Mpeg-4 Video Compression, Aberdeen: John Wiley & Sons Ltd, 2003.

[3] W.-t. T. S. J. W. John G. Apostolopoulos, «Video Streaming: Concepts, Algorithms, and Systems,» Hewlett-Packard Laboratories, Palo Alto, 2002.

[4] Wikipedia, «Wikipedia,» [En línea]. Disponible: www.wikipedia.com. [Último acceso: 09 2013].

[5] J. Giménez, «Social Net Blog,» 19 IV 2012. [En línea]. Disponible: http://www.socialnetblog.com/conoce-los-diferentes-formatos-de-video/. [Último acceso: 09 VI 2013].

[6] IETF, «RFC2326 Real Time Streaming Protocol,» Abril 1988. [En línea]. Disponible: http://www,ietf.org/rfc/rfc2326.txt. [Último acceso: octubre 2013].

[7] IETF, «RFC 2960 Stream Control Transsmision Protocol,» octubre 2000. [En línea]. Disponible: http://www.ietf.org/rfc/rfc2960.txt. [Último acceso: Octubre 2013].

[8] IETF, «RFC 793 TRANSMISSION CONTROL PROTOCOL,» Septiembre 1981. [En línea]. Disponible: www.ietf.org/rfc/rfc793.txt. [Último acceso: octubre 2013].

[9] IETF, «RFC 768 User Datagram Protocol,» 28 agosto 1980. [En línea]. Disponible: http://www.ietf.org/rfc/rfc768.txt. [Último acceso: Octubre 2013].

[10] P. CAMPANELLI, «enjoyTheArchictecture,» 2011. [En línea]. Disponible: http://www.enjoythearchitecture.com/vlc-architecture. [Último acceso: 16 VII 2013].

[11] Xiph.org, «icecast,» 6 2 2013. [En línea]. Disponible: www.icecast.org/docs/. [Último acceso: 22 12 2013].

[12] W. A. Lozano-Rivas., «DETERMINACIÓN DEL NÚMERO MÍNIMO DE OBSERVACIONES EN INVESTIGACIÓN,» Investea, Bogota, 2011.

[13] P. M. L. P. A. M. R. Almudena Diaz, «Estudio practico del rendimiento del servicio de Streaming de Video sobre redes moviles GPRS/UMTS,» Complejo Tecnologico, Campus de la computacion, Malaga, España.