UNIVERSIDAD TECNOLOGICA DE BOLIVAR

(UTB)

El modelo DiffServ como mecanismo para garantizar QoS sobre MPLS

MARCEL PALOMINO FLOREZ

PROYECTO INTEGRADOR PRESENTADO COMO REQUISITO

PARCIAL PARA OPTAR POR EL TITULO DE ESPECIALISTA EN TELECOMUNICACIONES

Cartagena de Indias, Colombia

Abril del 2012

UNIVERSIDAD TECNOLOGICA DE BOLIVAR

(UTB)

Este Proyecto Integrador fue aprobado por la Universidad como

Requisito parcial para optar al título de Especialista en Telecomunicaciones

_____

GONZALO LOPEZ

Especialización en telecomunicaciones

_____

MARCEL PALOMINO FLOREZ

AGRADECIMIENTOS

Muy sinceramente expreso mi gratitud al coordinador del programa Gonzalo López por su comprensión y colaboración en el transcurso de la especialización y posteriormente.

A mi esposa, por su apoyo incondicional.

Mil y mil gracias.

INDICE DE CONTENIDOS

INDICE DE FIGURAS

INDICE DE ANEXOS

GLOSARIO

AF *Assured Forwarding*. Reenvío asegurado.

BBE *Better than Best Effort*. Servicio de flujo de datos IP.

BW *BandWidth*. Ancho de banda.

CBQ *Class Based Queing*, Es un término general que se refiere a cualquier mecanismo basado en clases

CEF *Cisco Express Forwarding*. Conjunto de funcionalidades de los routers Cisco para poder ejecutar MPLS

CPU *Central Processing Unit*. Unidad de proceso central.

CR-LDP *Constraint-based Routing* LDP. Encaminamiento basado en restricciones del protocolo de distribución de etiquetas.

CSC *Class Selector Codepoint*. Código selector de clase.

DE *Default Behaviour*. Comportamiento por defecto. Tipo de PHB.

DiffSer *Differentiated Services*. Servicios diferenciados.

DS *Byte Differentiated Services* de un paquete IPV4

DSCP *Differentiated Services Codepoint.* Código de servicios diferenciados, campo que forma parte del byte DS de un paquete IPV4.

EF *Expedited Forwarding*. Reenvió acelerado. Tipo de PHB.

ER-LSP *Explicit-Routed* LSP. Encaminamiento explícito LSP.

LDP *Label Distribution Protocol.* Protocolo de distribución de etiquetas.

LER *Label Edge Router.* Encaminador de etiquetas frontera.

LSP *Label Switched Path.* Camino de conmutación de etiquetas.

LSR *Label Switching Router.* Encaminador de conmutación de etiquetas.

MPLS *Multiprotocol Label Switching*. Multiprotocolo de conmutación de etiquetas.

OSPF *Open Shortest Path First*. Protocolo abierto del primer camino más corto.

PE Router de acceso de una red no MPLS a un dominio MPLS.

PHB *Per-Hop-Behaviour.* Comportamiento por salto.

PHP *Penultimate-Hop-Popping*. Proceso en el que el último router de un dominio MPLS retira la etiqueta y envía un paquete IP sin etiqueta.

QoS *Quality Of Service*. Calidad de servicio.

RFC *Request For Comments*. Documento de especificaciones del IETF

TCP *Transmision Control Protocol.* Protocolo de control de la transmisión.

TE *Traffic Engineering.* Ingeniería de Tráfico.

TOS *Type Of Service*. Tipo de servicio.

TTL *Time To Live.* Tiempo de vida.

UDP *User Datagram Protocol.* Protocolo de datagramas de usuario.

Video Streaming El streaming es una tecnología que le permite emitir audio y video por Internet tanto en directo como en diferido

INTRODUCCIÓN

Internet se ha transformado en los últimos años en una red de muy alta difusión en cuanto al número de usuarios conectados. Esto ha sido visto por parte de los operadores como una oportunidad de ofrecer nuevos servicios a dichos usuarios además del tradicional servicio de email, ftp y navegación Web. Algunos de estos servicios son por ejemplo servicios de telefonía, videoconferencia, televisión, radio, etc. Estos nuevos servicios presentan requerimientos diferentes en cuanto a volumen de tráfico, calidad de servicio y seguridad.

El paradigma en que se ha basado el envío de paquetes en una red IP (protocolo base de Internet) ha sido la denominada política 'best effort'. Best effort implica que el usuario envía paquetes y la red y esta hace su mejor esfuerzo para hacerlos llegar al destinatario. Con este principio no es posible ofrecer servicios con requerimientos fuertes de Calidad de Servicio (QoS) en cuanto a pérdidas retardos o jitter como exigen por ejemplo los servicios de voz o video interactivo. Protocolos superiores a IP (como TCP) han procurado solucionar el problema de la pérdida de paquetes básicamente reenviando paquetes si estos no llegan a destino. Esto resuelve los problemas de la transferencia tradicional de datos, pero este tipo de protocolos no puede ser usado para la transferencia de servicios interactivos en línea, en los que no es posible esperar por una retransmisión.

La comunidad de Internet ha realizado esfuerzos diversos en los últimos años para romper el paradigma actual y aproximarse a la calidad de servicio brindada por Red Pública Telefónica (PSTN). El problema que hoy se plantea es diseñar la nueva arquitectura, las políticas, las metodologías y las herramientas necesarias para desplegar una red multiservicio capaz de asegurar los requerimientos de QoS necesarios para cada uno de los servicios ofrecidos. Un problema adicional es que este cambio debe ser gradual porque de otro modo no sería aplicable, ya que los operadores deberían perder una inversión en tecnología aún no amortizada en muchos casos.

Muchos de los esfuerzos realizados para transformar IP en una red de servicios convergentes están aún en su fase experimental y no han logrado imponerse de forma masiva. En paralelo nuevas propuestas surgen frecuentemente, fruto de una fuerte investigación en esta área. Aspectos básicos sobre cómo asegurar calidad de servicio en Internet, cómo medirla o estimarla, qué protocolos o tecnologías usar para brindar estos servicios aún generan controversias. Eso abre las puertas a un campo donde hoy se encuentra un fuerte desarrollo académico y comercial.

1. DESCRIPCIÓN DEL PROBLEMA

A medida que avanza y se expande la red de Internet y las telecomunicaciones en general, las exigencias en cuanto a rendimiento, ancho de banda, procesamiento y demás recursos de red y equipos se hace cada vez mayor. La seguridad de los datos que son transportados a través de ésta compone de igual manera un hilo vital para las grandes compañías. Se ha venido trabajando ardua y progresivamente para garantizar velocidad y seguridad en nuestros datos, se han desarrollado innumerables investigaciones y mecanismos para garantizar que la información sea recibida de la manera correcta, sin vulneraciones, con el menor retardo posible y sin pérdida de la información. Sin embargo, al momento de enviar información por la red, ésta puede tomar una o múltiples rutas que, probablemente se encuentren congestionadas por una u otra razón. Cuando esto ocurre, si los datos no fueron tratados de manera adecuada por el emisor, tendrán simplemente que hacer "cola" obviando la importancia que tengan o el nivel de urgencia que se requiera aumentando de esta manera la exposición de nuestros datos a problemas de seguridad y lentitud en la entrega. Para superar esto, la información debe ser manejada con ciertos niveles y parámetros de Calidad de servicio (QoS) que garanticen rapidez, seguridad y tratamiento especial basado en las reglas asociadas o establecidas.

Es sabido que el protocolo IP no aporta mucho a la entrega correcta de los paquetes, por lo que se le asoció a protocolos de capas superiores en el modelo OSI la tarea de controlar si los paquetes son recibidos en el destino o no. Esto resuelve de algún modo, el servicio de transmisión de datos tradicional pero no las transmisiones en tiempo real o aquellas que no puedan esperar un retardo como el que supone el control de flujo TCP.

Se han creado varios mecanismos para brindar Calidad de Servicio en redes IP, uno de los cuales es el modelo IntServ (Integrated Services) o Servicios Integrados, en donde se procura asegurar recursos para cada flujo de datos a lo largo de la red, asegurando de ésta manera la calidad de servicio. Sin embargo, al presentar problemas de escalabilidad, este modelo se ha dejado atrás. En procura de mejorar dichos inconvenientes, surgió el modelo DiffServ (Differentiated Services) o Servicios Diferenciados. Este modelo, soluciona los problemas de escalabilidad agregando flujos por clase y dando a cada una la prioridad y el tratamiento correspondiente a la misma.

Con todo esto, protocolo TCP y calidad de servicio con DiffServ, se pensaría que se lograría un tratamiento adecuado de la información, pero lo verdadero es que a pesar de ello, la problemática continúa debido al tráfico excesivo que se crea o que se origina en algunos tramos de red debido a las "decisiones" de los protocolos de enrutamiento. Para descongestionar la red en dichos puntos, se utiliza Ingeniería de Tráfico, lo que equivale a distribuir el tráfico de la red de manera eficiente. En un principio se creyó que las redes IP sobre ATM eran la solución en este sentido pero pronto hubo muchos inconvenientes y problemas de escalabilidad. Surgió entonces MPLS.

La arquitectura MPLS garantiza la utilización proporcional de los recursos de la red a través de caminos virtuales y preestablecidos para cada flujo. Sin embargo, MPLS por sí mismo sigue sin poder diferenciar tráfico por lo que una integración con el modelo DiffServ aseguraría a esta arquitectura ciertos parámetros de calidad de servicio realizando una distinción y priorización del tráfico de datos. Dando como resultado una opción de arquitectura de red segura, eficiente en el manejo del tráfico y con altos niveles de calidad de servicio, lo que constituye una alternativa muy llamativa para su implementación en la infraestructura que lo requiera.

La integración de MPLS y DiffServ y la conveniencia de usar un modelo u otro es un tema aun en desarrollo.

Con el fin de observar el funcionamiento y la puesta en marcha de los mecanismos DiffServ para otorgar QoS, se elaborará un laboratorio en el cual se accede y ejecuta un archivo de video almacenado en un Host desde otro ubicado en otro extremo de una red MPLS. El archivo será ejecutado en diferentes entornos: el primero, consiste en ver el video sin calidad de servicio y sin tráfico en la red; el segundo, consiste en ejecutar el archivo de video sin métodos de calidad de servicio pero con congestión en la red; y, por último, se ejecuta el video bajo condiciones de red saturada o congestionada e implementando DiffServ para otorgarle QoS a la red.

El principal objetivo del laboratorio es observar cómo se deteriora la información al atravesar una red congestionada y sin mecanismos QoS y comparar esta situación al otorgarle parámetros de calidad de servicio a la red. Además se podrá ver de una manera más específica y didáctica el funcionamiento real de la arquitectura DiffServ proporcionando niveles de QoS a la red MPLS.

2. OBJETIVOS

2.1 GENERAL

Analizar y describir el funcionamiento de la Arquitectura MPLS integrada con el modelo de calidad de servicio DiffServ en redes IP

2.2 ESPECIFICOS

- Sintetizar el comportamiento de QoS proporcionado por la arquitectura MPLS
- Presentar las ventajas del modelo DiffServ
- Identificar los beneficios de una integración de MPLS con DiffServ
- Describir de manera clara el proceso de entrada-salida de los datos en una red MPLS con QoS bajo DiffServ
- Observar el funcionamiento del Modelo DiffServ para dar calidad de servicio sobre una red IP MPLS a través de la implementación de un laboratorio práctico en GNS3

3. MARCO TEORICO

3.1 MPLS

MPLS (siglas de Multiprotocol Label Switching) es un mecanismo de transporte de datos estándar creado por la IETF y definido en el RFC 3031. Opera entre la capa de enlace de datos y la capa de red del modelo OSI. Fue diseñado para unificar el servicio de transporte de datos para las redes basadas en circuitos y las basadas en paquetes. Puede ser utilizado para transportar diferentes tipos de tráfico, incluyendo tráfico de voz y de paquetes IP.

MPLS fue originalmente propuesto por un grupo de ingenieros de Cisco, con el nombre de "Tag switching", fue entregado a la IETF y luego del proceso de estandarización cambió de nombre.

Cómo funciona

Es de notar que la conmutación IP es realizada en la capa 3 y está basada en la dirección ip destino (en algunos casos también en la ip de origen); si miramos una tabla de enrutamiento sólo vemos la asociación "red destino" - "próximo salto".

El enrutamiento en sí, impone restricciones y ciertos cuidados en nuestras redes, como por ejemplo que en la asignación de direcciones ip no haya colisiones (dos segmentos de red no pueden tener las mismas direcciones).

Lo interesante de MPLS es que la conmutación de paquetes está basada en etiquetas y se realiza entre la capa 2 y la capa 3 (no depende del encabezado ip), estas etiquetas son agregadas antes del ingreso a la red MPLS y son removidas cuando los paquetes salen de ella. MPLS funciona adicionando a los paquetes un header MPLS, que contiene una o más etiquetas, esto es llamado "label stack".

Cada etiqueta contiene 4 campos:

* 20 bits - Valor de la etiqueta.
* 3 bits - Campo experimental reservado para usos futuros.
* 1 bit - Final de la pila. Si tiene el valor 1 entonces es la última etiqueta de la pila.
* 8 bits - Campo TTL (time to live)

Arquitectura de una red MPLS

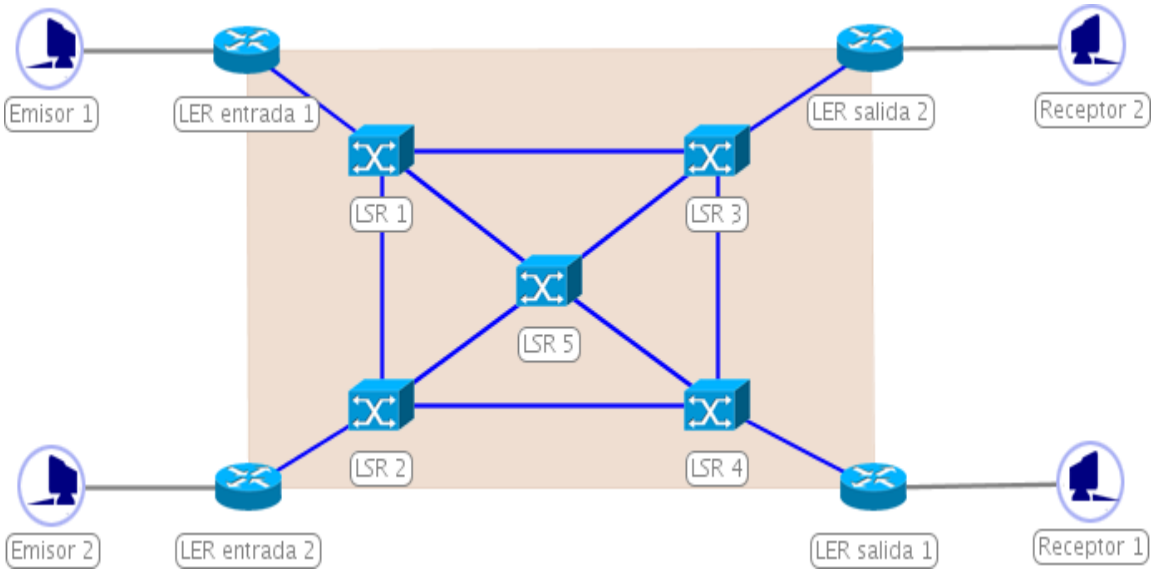

Figura 1. Arquitectura MPLS

- **LER** (Label Edge Router): elemento que inicia o termina el túnel (agrega y quita las etiquetas). Es el punto de entrada/salida a la red MPLS. Un router de entrada se conoce como Ingress Router y uno de salida como Egress Router. Ambos se suelen denominar Edge Label Switch Router, ya que se encuentran en los extremos de la red MPLS.

- **LSR** (Label Switching Router): elemento que conmuta etiquetas.

- **LSP** (Label Switched Path): nombre genérico de un camino MPLS (para cierto tráfico o FEC), es decir, del túnel MPLS establecido entre los extremos. Se debe tener en cuenta que un LSP es unidireccional.

- **LDP** (Label Distribution Protocol): un protocolo para la distribución de etiquetas MPLS.

- **FEC** (Forwarding Equivalence Class): nombre que se le da al tráfico que se encamina bajo una etiqueta. Subconjunto de paquetes tratados del mismo modo por el conmutador.

Posibles usos

Los sectores que más provecho pueden sacar de MPLS, son los proveedores de servicio (carriers), las grandes empresas e instituciones gubernamentales (o sea, las grandes redes). Algunas empresas medianas pueden contratar un servicio de VPNs, basado en MPLS de algún

proveedor de servicio, aunque la parte divertida la realiza el proveedor. Los usos más importantes son:

*** *MPLS-VPN***

Con MPLS pueden realizarse robustas VPNs, más escalables y menos costosas que otras alternativas como IPSec, ATM o frame relay; y además agrega QoS.

* ***Ingeniería de tráfico / QoS / Congestión***

El enrutamiento IP tradicional suele llevar a sobrecargar los caminos más cortos (a veces caminos más largos pueden tener menor congestión y menor delay).

Respecto a este problema MPLS puede ser utilizado para:

- Maximizar la utilización de los enlaces y los nodos
- Garantizar el nivel de delay (respetar los SLAs)
- Minimizar el impacto de las fallas.

Los principales protocolos para realizar ingeniería de tráfico con MPLS son CR-LDP y RSVP-TE.

* ***Integración de redes diversas: ATM, Frame relay, IP, Ethernet y ópticas***

Mantener una red, es más barato que mantener muchas. Con MPLS podemos armar una red de transporte universal.

3.2 La arquitectura de los Servicios Diferenciados (DiffServ)

En esta arquitectura, los paquetes son clasificados y marcados para recibir un trato particular en cuanto al envío en cada salto. Sofisticada clasificación, marcado, política y operaciones de acondicionamiento necesitan sólo ser implementadas en los bordes de la red o en los hosts.

Esta arquitectura logra escalabilidad al implementar un complejas funciones de clasificación y condicionamiento sólo en los nodos del borde de la red, y aplicando conductas por salto a los agregados del tráfico que han sido apropiadamente marcados usando el campo DS en las cabeceras de IPv4 o IPv6.

Es mantenida una distinción entre:

- el servicio provisto a un agregado de tráfico,

- las funciones de condicionamiento y los comportamientos por salto, usados para realizar los servicios,

- el valor del campo DS, usado para marcar paquetes para seleccionar el comportamiento en cada salto, y

- los mecanismos de implementación particulares del nodo que realizan un comportamiento por salto.

Esta arquitectura sólo provee servicio diferenciado en una dirección del flujo de tráfico y es por ende asimétrica. Antes de proseguir y entrar en detalle con el funcionamiento de DiffServ y el análisis de sus componentes, vamos a introducir una breve terminología para así se puede entender con más claridad lo expuesto más adelante.

3.2.1 Terminología

Behavior Aggregate (BA, también llamado a veces "agregado de tráfico", TA): es una colección de paquetes con el mismo DSCP (DiffServ Code Point) atravesando un enlace en una dirección.

BA classifier: es un clasificador que selecciona paquetes basado solo en el contenido del campo de DS.

Enlace de frontera: es un enlace que conecta los nodos de borde de dos dominios.

DS Behavior Agregate: una colección de paquetes con el mismo código DS, cruzando un enlace en una dirección particular.

Código DS: un valor específico de la porción DSCP del campo DS, usado para seleccionar un PHB.

DS-Compliant: capaz de soportar funciones y comportamientos de servicios diferenciados.

Dominio DS: un dominio capaz de tener DS; un conjunto contiguo de nodos que operan con un conjunto común de políticas de aprovisionamiento de servicios y definiciones PHB.

Nodo de Egreso DS: un nodo DS límite en su rol de manejar tráfico a medida que éste deja el dominio DS.

Nodo de Ingreso DS: un nodo DS límite es su rol de manejar tráfico a medida que éste entra al domino DS.

Nodo Interior DS: un nodo DS que no es un nodo DS límite.

Campo DS: es el octeto TOS de la cabecera de IPv4 o el octeto de la Clase de Tráfico de IPv6. Los bits del campo DSCP contienen el DS codepoint, mientras que los restantes bits no están en uso (se ampliará este tema más adelante).

Dropping: es el proceso de descartar paquetes basándose en reglas específicas; políticas.

Marking (marcado): es el proceso de seteo del DS codepoint en un paquete, basándose en reglas definidas; pre-marcado y re-marcado.

Metering (mediciones): es el proceso de medir las propiedades temporales de una corriente de tráfico seleccionada por un clasificador (classifier).

Microflow (microflujo): es un conjunto de datos, enviados unidireccionalmente entre dos aplicaciones, únicamente identificado por una quíntupla: protocolo de transporte, IP origen, IP destino, puerto origen y puerto destino.

Per-Domain-Behavior (PDB): se define como el trato esperado que un agregado de tráfico va a recibir de borde a borde de un dominio DiffServ.

Per-Hop-Behavior (PHB): define el tratamiento en cada nodo. Es una descripción del comportamiento de reenvío observado exteriormente; puede ser implementado por distintos mecanismos.

Policing: el proceso de descarte de paquetes dentro de un arroyo de tráfico en concordancia con el estado de un correspondiente medidor (meter) cumpliendo un determinado perfil.

Acuerdo del Nivel de Servicio (SLA): un contrato de servicio entre un cliente y un proveedor de servicio que especifica el servicio de envío que un cliente debe recibir.

Shaping (conformador): el proceso de retardar paquetes dentro de un flujo de tráfico, haciendo que conforme cierto perfil de tráfico ya definido.

Traffic Conditioner (acondicionador de tráfico): una entidad que realiza las funciones de condicionamiento del tráfico y que puede contener medidores, marcadores, droppers y conformadores. Están típicamente dispuestos en nodos de borde solamente.

Traffic Conditioning Agreement (TCA): un acuerdo especificando reglas de clasificación y perfiles de tráfico correspondientes, y mediciones, marcado, descarte y/o reglas de conformación que son aplicables a los arroyos de tráfico seleccionados por el clasificador.

3.2.2 DiffServ y paquetes MPLS

En la cabecera de los paquetes MPLS, tenemos el campo EXP para controlar el QoS. Como hemos podido observar, la cabecera IP tiene 6 bits destinados al DSCP para clasificar los distintos paquetes, pero la cabecera MPLS solo dispone de 3 bits de EXP. Por lo tanto se tendrán

que mapear las distintas 64 clases en las 8 que permite MPLS. Esto no es un gran problema, ya que 8 clases de servicio suelen ser más que suficiente.

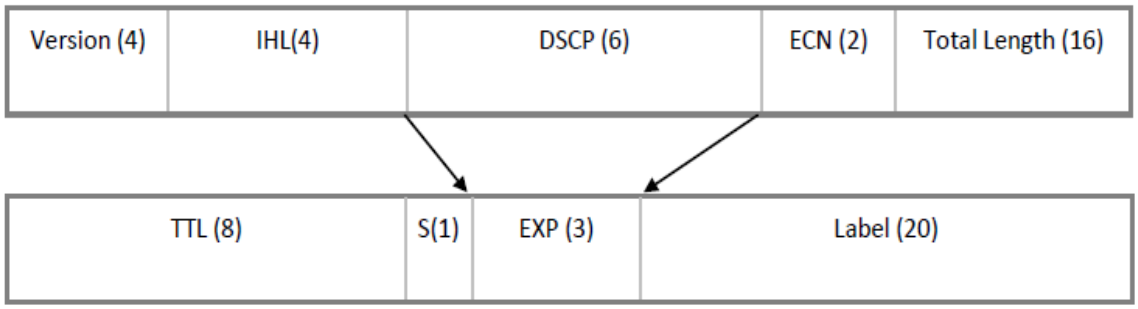

Figura 2. Campo DSCP vs EXP

Por defecto, cuando un paquete llega a la red, el router MPLS de ingreso encapsula el paquete IP con su etiqueta correspondiente y, el campo EXP con los 3 primeros bits del campo DSPC (los 3 bits más significativos). Luego, cuando el paquete MPLS viaja por la red, se va copiando el valor del campo EXP en la etiqueta más externa de la Pila de Etiquetas. Así pues, el mapeo que se realizará será el siguiente:

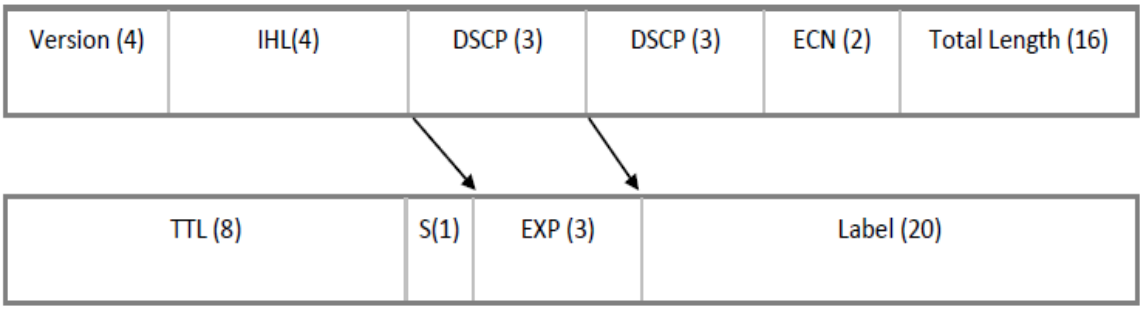

Figura 3. Campo DSCP vs EXP

Cabe destacar, que paquetes con distintos DSCP, pero con los 3 primeros bits de este iguales, tendrán el mismo valor de EXP, y por lo tanto serán tratados de igual forma por la red MPLS. Par que esto no ocurra, podemos definir un PHB para que modifique el valor del EXP en función de todo el valor del campo DSCP (6 bits). Entonces, cuando un paquete llegue a una red MPLS, el PHB asignará un valor preestablecido al campo EXP del nuevo paquete MPLS, y otro PHB podrá actuar para ese valor de EXP.

Es importante darse cuenta que los PHB definidos para los DSCP no tendrán efecto dentro de una red MPLS, ya que la ventaja de esta red es que no revisa los valores del paquete IP, y solo mira los valores del MPLS. Por lo tanto se tendrán que definir PHB para los valores de EXP.[1]

[1] DiffServ: Servicios Diferenciados. Monografía de Evaluación de Performance en Redes de Telecomunicaciones. Adrián Delfino, Sebastián Rivero.

3.3 IGMP

El protocolo de red **IGMP** se utiliza para intercambiar información acerca del estado de pertenencia entre enrutadores IP que admiten la multidifusión y miembros de grupos de multidifusión. Los hosts miembros individuales informan acerca de la pertenencia de hosts al grupo de multidifusión y los enrutadores de multidifusión sondean periódicamente el estado de la pertenencia.

La última versión disponible de este protocolo es la IGMPv3 descrita en el [RFC 3376]

Todos los mensajes IGMP se transmiten en datagramas IP y tienen el formato mostrado en la figura adjunta. Los campos son los siguientes:

Tipo	Máximo tiempo de respuesta			Checksum
Dirección De Grupo				
Resv (Reservado)	S	QRV	QQIC	Número De Fuentes (N)
Dirección de origen				

Figura 4. Mensaje IGMP

Una red diseñada para reenvíos IGMP debe verse de la siguiente manera:

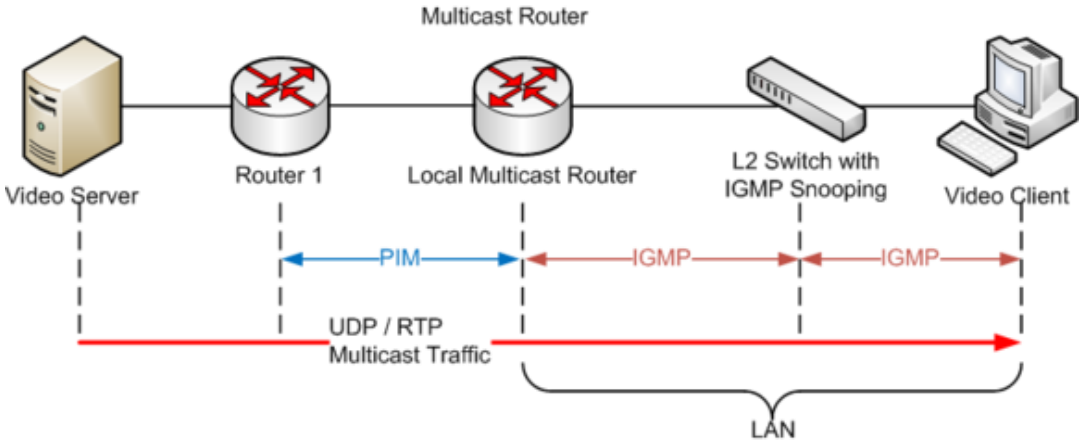

Figura 5. Ejemplo red IGMP

4. DESARROLLO Y RESULTADOS

El laboratorio y las pruebas que se llevarán a cabo a continuación corresponden a la configuración de una topología de red mostrada posteriormente cuya finalidad es estudiar a través de una transmisión de video Streaming el deterioro de la señal en una red con saturación y mostrar también la solución QoS para garantizar una buena calidad de experiencia en el usuario final. Se detallan apartes de la configuración MPLS, OSPF, IGMP y QoS DiffServ, como también puesta en marcha del software VLC en los extremos y la inyección de tráfico al canal a través de CAPSA. En los Anexos se presenta la configuración completa de cada uno de los enrutadores que participa en el proceso.

4.1 Herramientas (Software)

Para el desarrollo de este laboratorio y realizar las pruebas necesarias para tomar gráficas y mostrar los resultados inherentes a la calidad de servicio, se utilizaron herramientas de emulación de redes (GNS3), analizadores de tráfico (CAPSA COLASOFT), generadores de tráfico (CAPSA COLASOFT) y herramientas destinadas a la reproducción de video streaming (VLC) como se describen a continuación:

4.1.1 GNS3

GNS3 es una aplicación también realizada en Python que usa las librerías de Dynagen para crearle una interfaz gráfica (GUI). Sus principales funciones son editar el archivo de texto .net y realizar las operaciones del CLI hechas por Dynagen y Dynamips. Adicionalmente incorpora la capacidad de simular PCs.

La unión de Dynamips-Dynagen-GNS3 crea una plataforma que permite el fácil diseño de topologías de red complejas ya que se realizan tan sólo arrastrando los componentes y dibujando líneas entre routers de forma intuitiva. Por lo tanto, GNS3 es idóneo para el entrenamiento de estudiantes que desean familiarizarse con dispositivos de red.

Las capacidades más resaltantes que podemos obtener de GNS3 y que han servido como punto de partida para tomar la decisión de estudiar más a fondo este simulador son las siguientes:

- Se encuentra disponible de forma gratuita en la red.
- Es fácil de instalar ya que todos los programas que necesita para funcionar se encuentran en un solo paquete de instalación.
- Está en constante actualización y periódicamente se puede encontrar versiones de la aplicación más robustas y con nuevas funcionalidades.

- Permite la conexión Telnet a la consola de un router virtual, de forma fácil directamente desde la interfaz gráfica.
- Alternativamente también permite trabajar directamente desde consola de gestión de Dynagen.
- Permite la comunicación entre redes virtuales con redes del mundo real.
- Es apropiado para simular redes de grandes tamaños ya que permite que un cliente GNS3 pueda correr en una máquina diferente a la que contiene al emulador Dynamips, repartiendo el procesamiento entre diferentes PCs.
- Puede capturar los paquetes que pasan por enlaces virtuales y escribir los resultados de la captura en archivos que pueden ser interpretados por aplicaciones como Wireshark o tcpdumps.

GNS3 no es la única aplicación que brinda una GUI a Dynamips, existe otra con el nombre de *Dynagu*i que realiza la misma tarea pero que se encuentra actualmente en fase de desarrollo y que no llega a implementar todas las funcionalidades que posee GNS3.

4.1.2 COLASOFT CAPSA

Rastreador de paquetes experto diseñado para el análisis de protocolo y diagnóstico de redes. Analiza el tráfico de red de un equipo local o una red local. Con la habilidad de captura de paquetes en tiempo real y un análisis preciso de datos. Es capaz de mostrar estadísticas gráficas y todo tipo de información que facilite la administración.

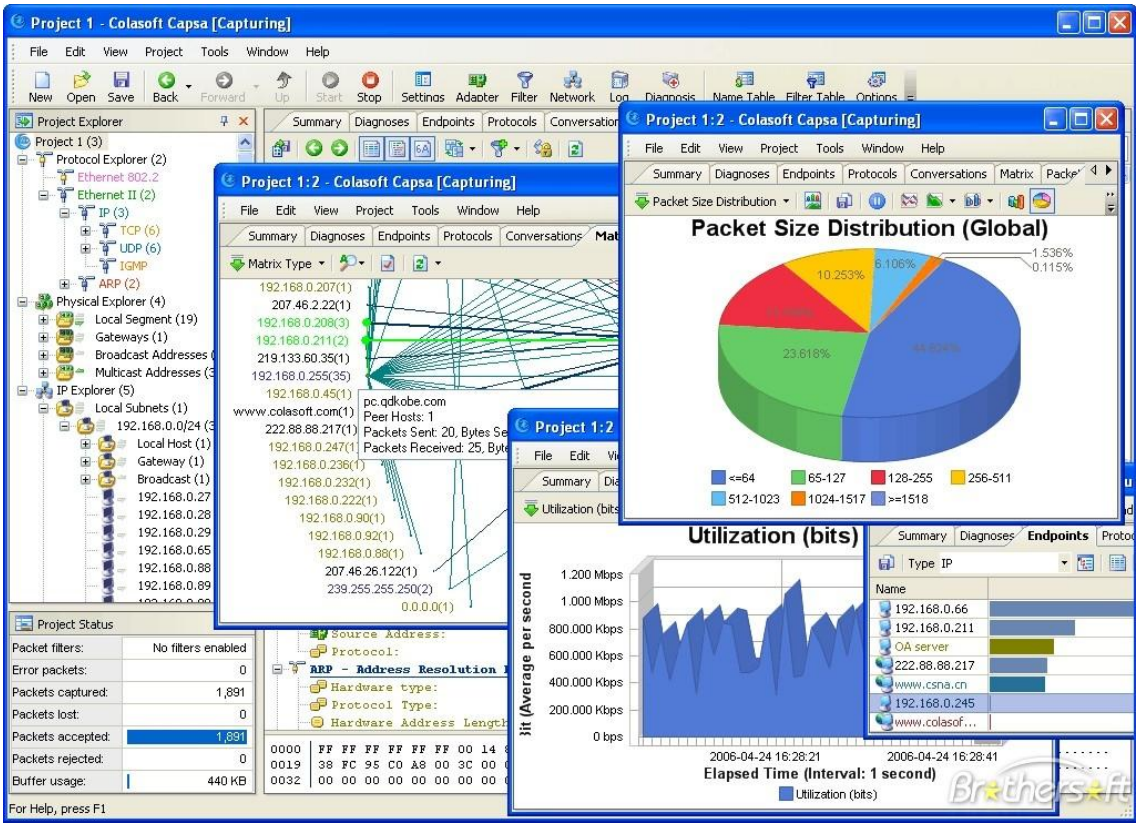

Figura 6. Entorno gráfico Colasoft Capsa

4.1.3 VLC Media Player

VLC media player es un reproductor multimedia y framework multimedia libre y de código abierto desarrollado por el proyecto VideoLAN. Es un programa multiplataforma con versiones disponibles para muchos sistemas operativos.

VLC es un reproductor de audio y video capaz de reproducir muchos códecs y formatos de audio y video, además de capacidad de streaming. Es software libre, distribuido bajo la licencia GPL.[2]

4.2 Topología de laboratorio GNS3

El laboratorio que se implementa para la elaboración de las pruebas finales de calidad de servicio sobre una red MPLS, consta de 6 routers de la serie 7200 interconectados entre sí y con interfaces capaces de soportar tráfico y paquetes MPLS. En los extremos de la red se ubican dos hosts remotos, los cuales, realizarán, el papel de Cliente y Servidor respectivamente. El servidor emitirá el video streaming a través del software VLC y, el Cliente reproducirá localmente ese video a través de la interfaz del mismo software.

En la transmisión del video, así como en la de cualquier dato en particular ocurren los mismos fenómenos inherentes a la comunicación de datos a través de una red de telecomunicaciones como son el retardo, el jitter, entre otros. Sin embargo, por las características mismas del contenido de esos paquetes, en este caso un video streaming, se hace más notable unos aspectos más que otros según sea el caso.

En la topología el objeto Cliente hace referencia a la maquina física ejecutando el Sistema Operativo Windows 7 Professional, 3.2 GHz de capacidad de procesamiento de 64 bits y 6 GB de memoria RAM.

El Servidor ha sido configurado en una máquina virtual a través del software VMWare con un sistema operativo Windows XP tradicional. En esta máquina alojamos nuestro video a emitir y la configuración del VLC actuando como Emisor.

[2] http://es.wikipedia.org/wiki/VLC_media_player

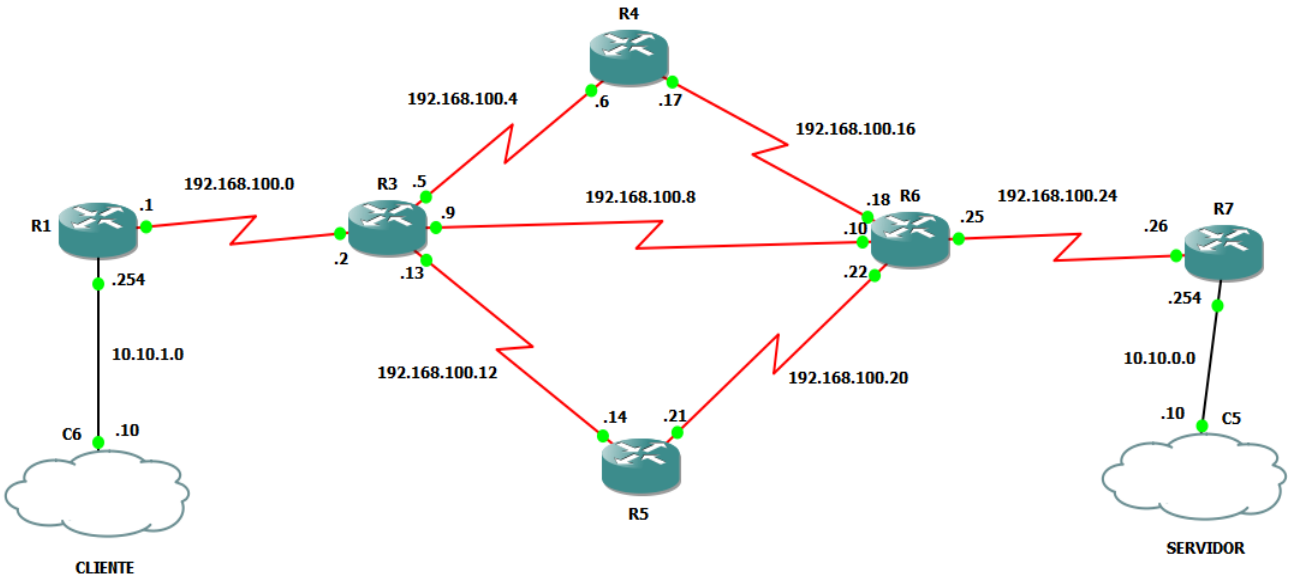

Figura 7. Topología red de laboratorio

4.3 Configuración OSPF

Se hace una configuración básica de enrutamiento a través del protocolo OSPF. No es necesario extenderse demasiado en este aspecto ya que solo se necesita que realice su labor de encaminamiento hacia el mejor destino según su algoritmo de una manera eficiente y sencilla. Con esta configuración las pruebas se pueden desarrollar con el mínimo de prestaciones que nos brinda este protocolo y sin agregar demasiada carga de información en la convergencia entre routers.

En todos los routers se crea una dirección de Loopback que actuará como ID del router para el proceso OSPF actual además de ser utilizado por MPLS para sus procesos de ingeniería de tráfico. Este paso es necesario y de suma importancia para la estabilidad y convergencia de nuestra red debido a que una interfaz virtual garantiza que los procesos OSPF se mantendrán arriba y en funcionamiento a diferencia de las interfaces físicas, las cuales, pueden caer súbitamente y los enrutadores deberán comenzar nuevamente un proceso de convergencia con el nuevo id.

Las redes conectadas directamente al enrutador deber estar asociadas al proceso OSPF a través del comando *network* para poder ser notificadas al resto de la red.

El comando no *auto-summary* nos sirve para visualizar todas las redes aprendidas y sin sumarización.

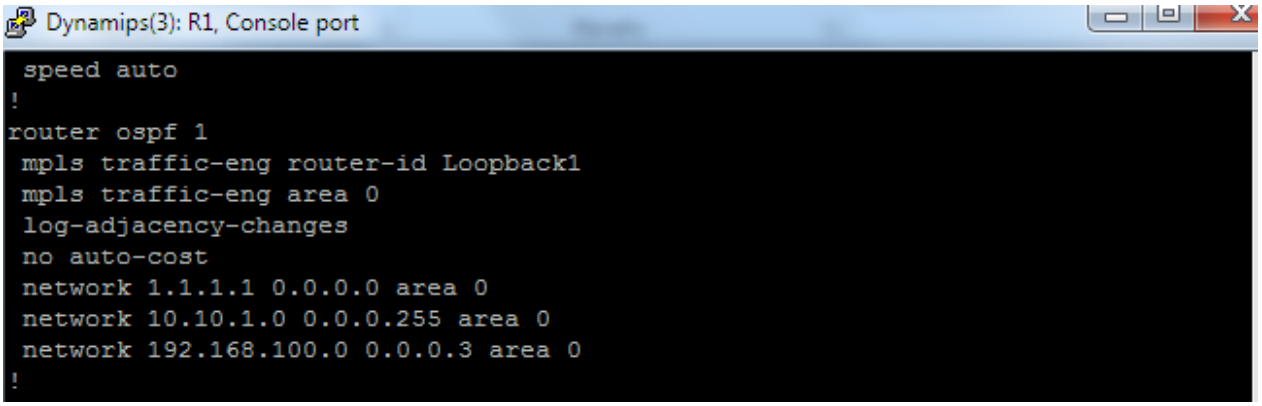

Figura 8. Muestra de configuración OSPF en R1

En la figura anterior se puede observar la configuración OSPF que se utilizó para R1. Las configuraciones del resto de routers se pueden observar en los Anexos del trabajo.

4.4 Configuración MPLS

Además de la configuración que MPLS hace dentro del proceso de enrutamiento, se debe primero habilitar cada una de las interfaces que participará en las rutas MPLS a través del comando *mpls ip* en el modo de configuración de dicha interfaz. Este comando hace que la interfaz participe en la comunicación MPLS y que trabaje con el encapsulamiento propio del protocolo, su tarea dependerá si es un router de frontera o no. Es sabido que los routers de frontera (LER) son los encargados de colocar y/o quitar el encapsulado MPLS, mientras que los routers internos (LSR) se basan en este encapsulado para llevar a cabo el encaminamiento respectivo:

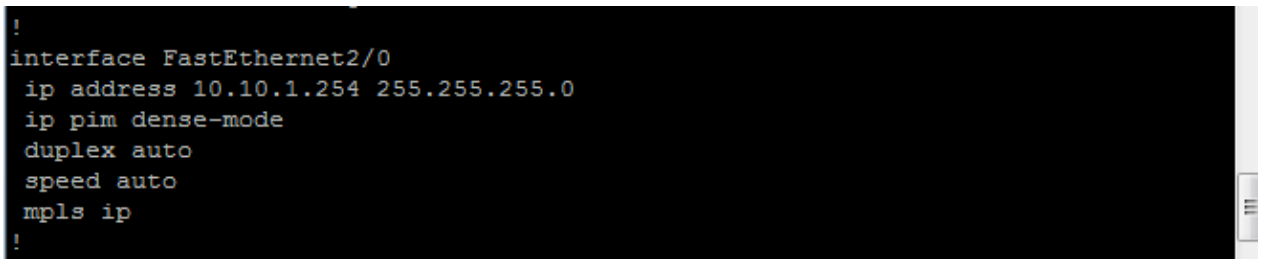

Figura 9. Muestra de configuración MPLS en una interfaz

4.5 Configuración IGMP

La mayoría de los routers no tiene habilitada su capacidad de realizar multicast a través del protocolo IGMP. En este caso y para este laboratorio es necesario configurarlo con el fin de poder desarrollar y llevar a cabo con éxito las pruebas con video streaming. Este tipo de transmisiones se hace en multicast y simplemente si no se tiene habilitado el protocolo en las interfaces y en los routers por donde se encaminará este tráfico, definitivamente no tendrá ningún éxito. No se podrá transmitir.

Por defecto un router cisco tiene IGMP deshabilitado en sus interfaces como se muestra:

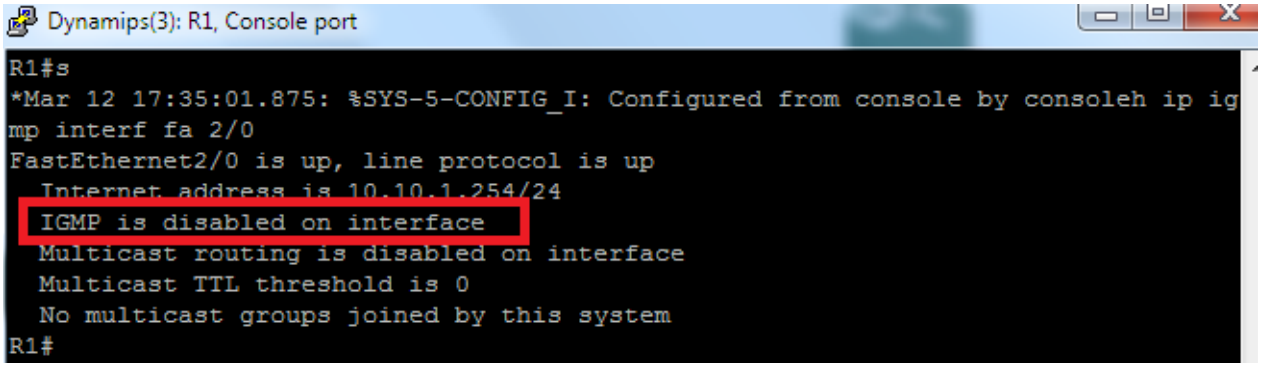

Figura 10. Verificación estado IGMP en router

Para habilitar el protocolo se procede de la siguiente manera. Primero se habilita en el modo de configuración global a través del comando *ip multicast-routing.* Y, posteriormente, se habilita en todas aquellas interfaces que participarán en la transmisión multicast con el comando *ip pim dense-mode:*

Figura 11. Configuración IGMP

Realizando la verificación se puede observar ahora que la interfaz tiene habilitado el protocolo IGMP y esta lista y esperando para transmitir también multicast:

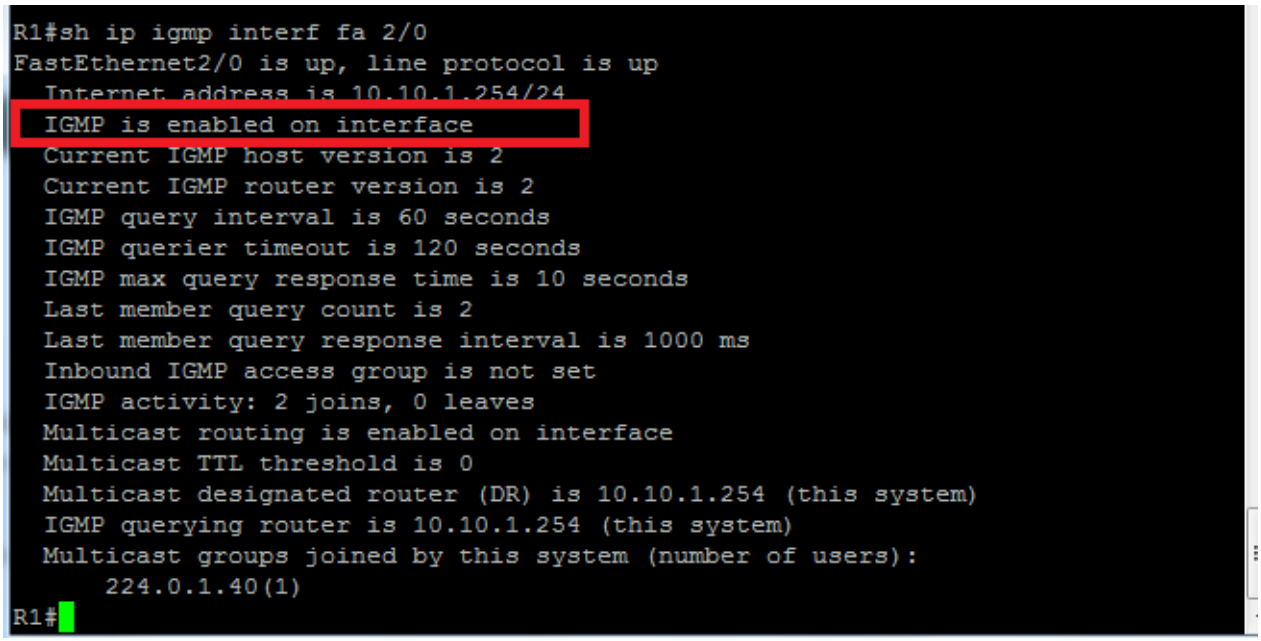

Figura 12. Verificación IGMP habilitado

Con esta configuración de IGMP sencilla se garantiza la retransmisión de los paquetes multicast que entren a las interfaces del router y la transmisión o emisión de nuestro video streaming seguirá su camino hacia el Cliente.

NOTA:

Luego que se tiene la topología lista y los routers funcionando correctamente, el protocolo OSPF distribuyendo todas las rutas y prueba de comunicación extremo a extremo, se procede a ejecutar el software VLC en los equipos PC de extremo, uno de los cuales actuará como emisor del video streaming y el otro como receptor. Primero el Servidor emitirá el video y luego en el Cliente se ejecutará VLC con el fin de poder capturar ese video transmitido. A continuación se elabora una guía paso a paso como es la ejecución de VLC en ambas maquinas (Servidor y Cliente):

4.6 Configuración VLC

Se instala el software VLC en la maquina física y en nuestro PC XP virtualizado. Y se procede a realizar los pasos que nos lleven a emitir correctamente un video y, por el otro lado, a poder recepcionarlo de manera adecuada.

4.6.1 En el Servidor:

Se ejecuta VLC y presionamos Medio/Emitir:

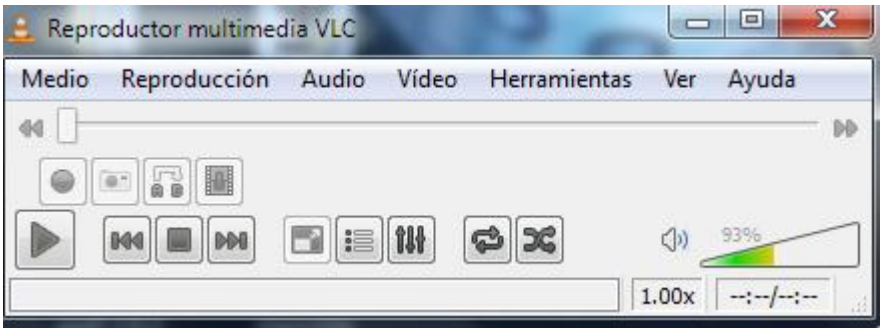

Figura 13. Interfaz inicial VLC

Se selecciona la opción Medio/Emitir para comenzar y seleccionar las opciones de la transmisión:

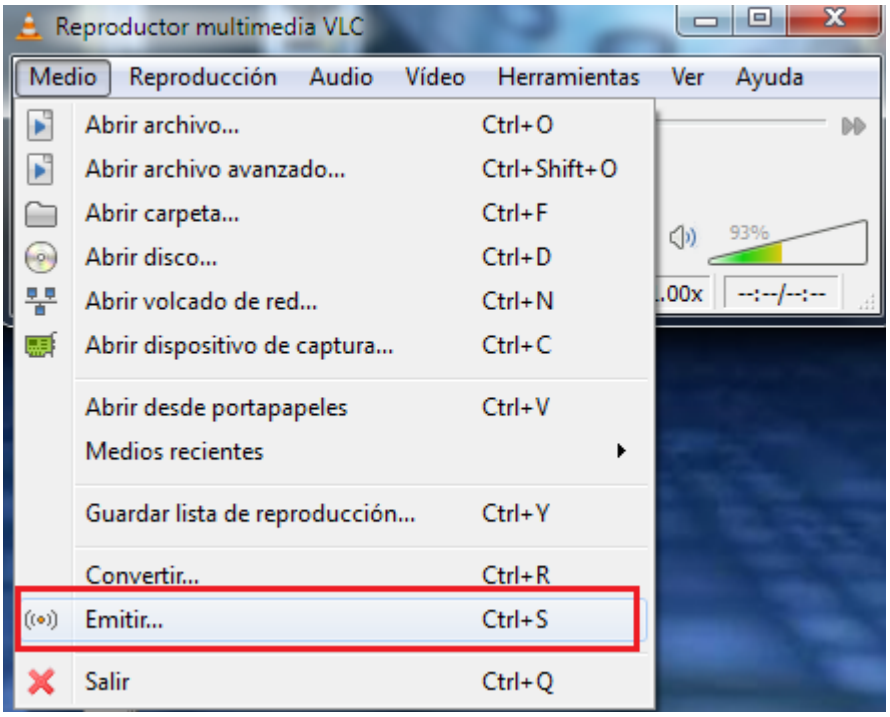

Figura 14. Opción Emitir video en VLC

En el siguiente cuadro añadimos el video o la lista de video que se desea emitir. En la pestaña Archivo se presiona el botón Añadir y se busca y selecciona el video que se desea transmitir:

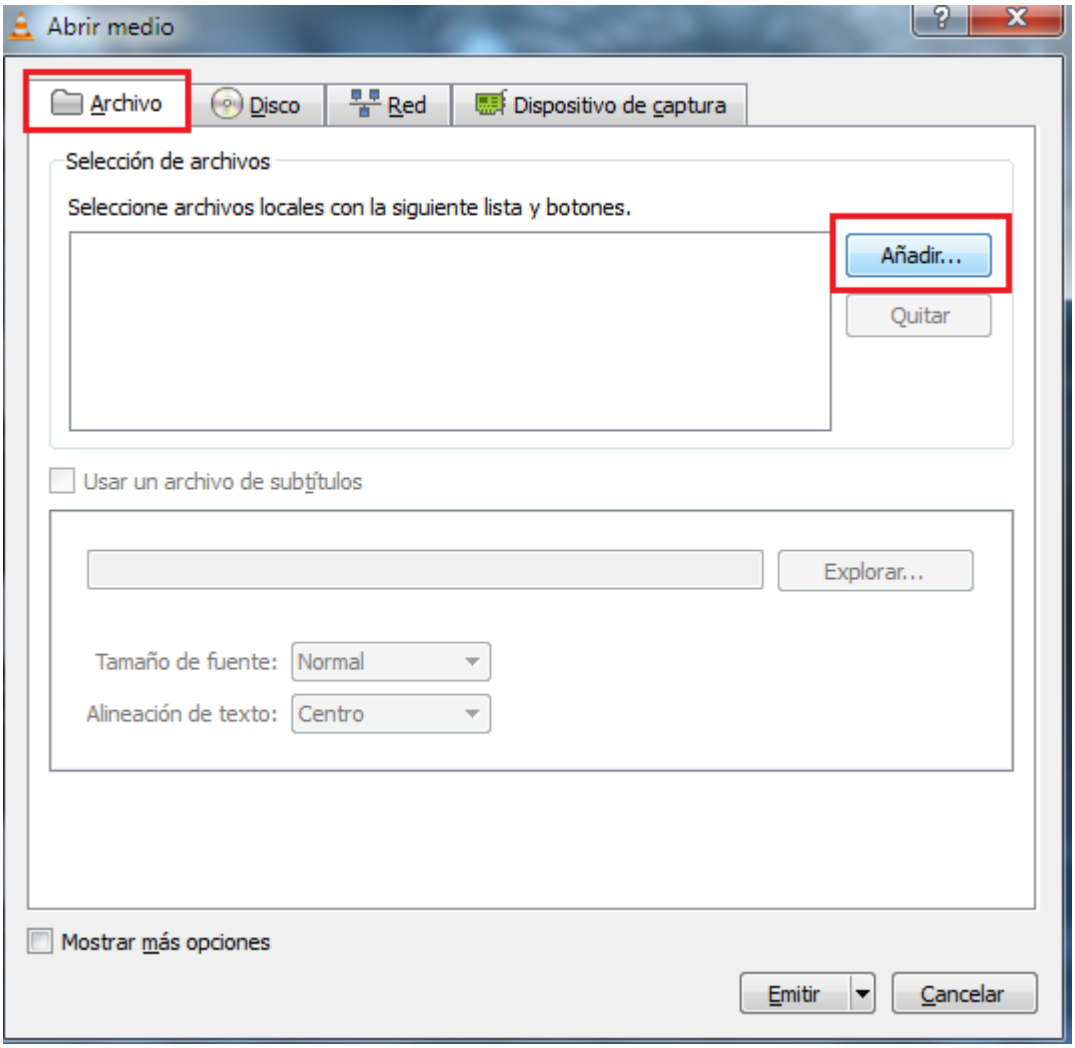

Figura 15. Seleccionar Archivo a transmitir

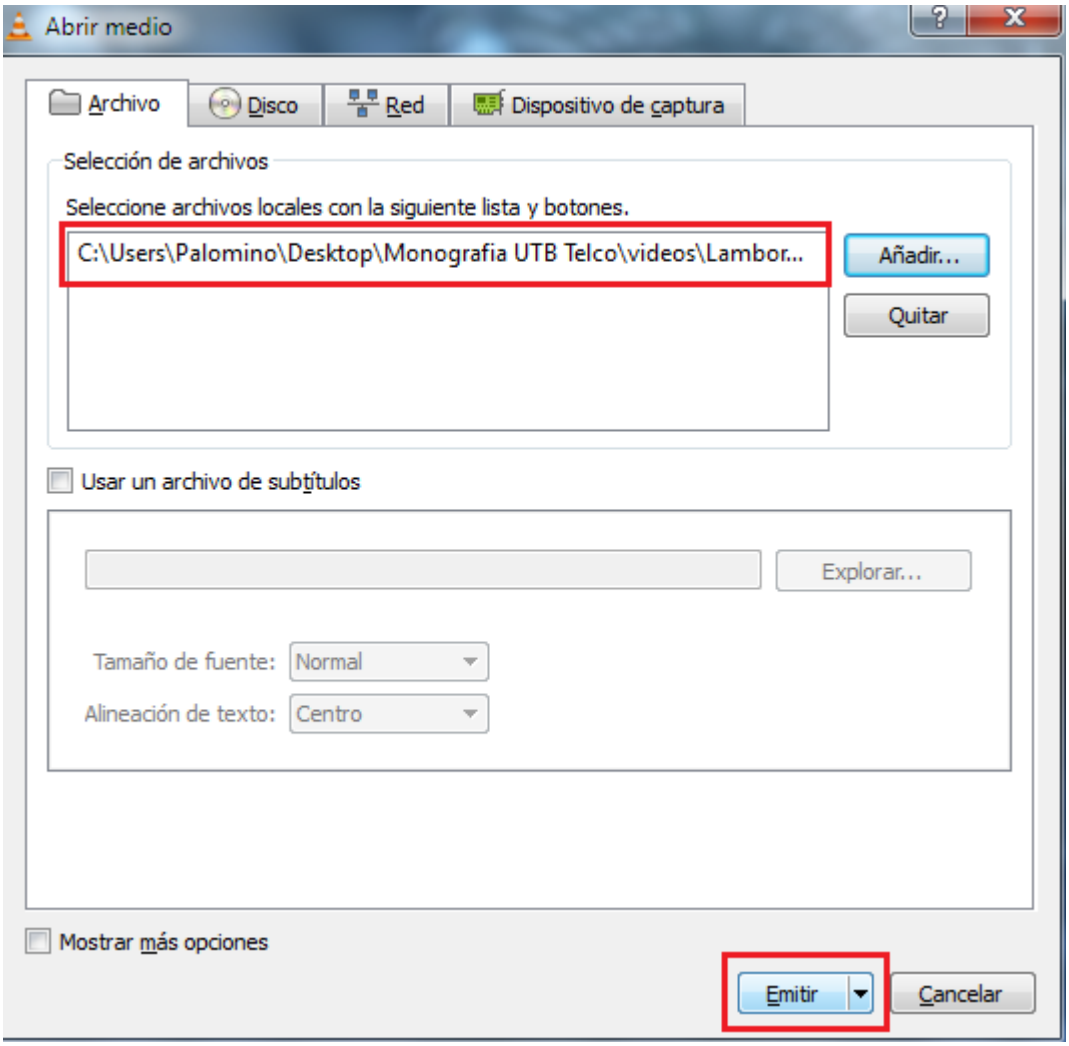

Figura 16. Emitir video seleccionado

Al seleccionar el video se presiona Emitir como se muestra en la figura anterior. En la ventana siguiente se presiona Siguiente para continuar con el proceso:

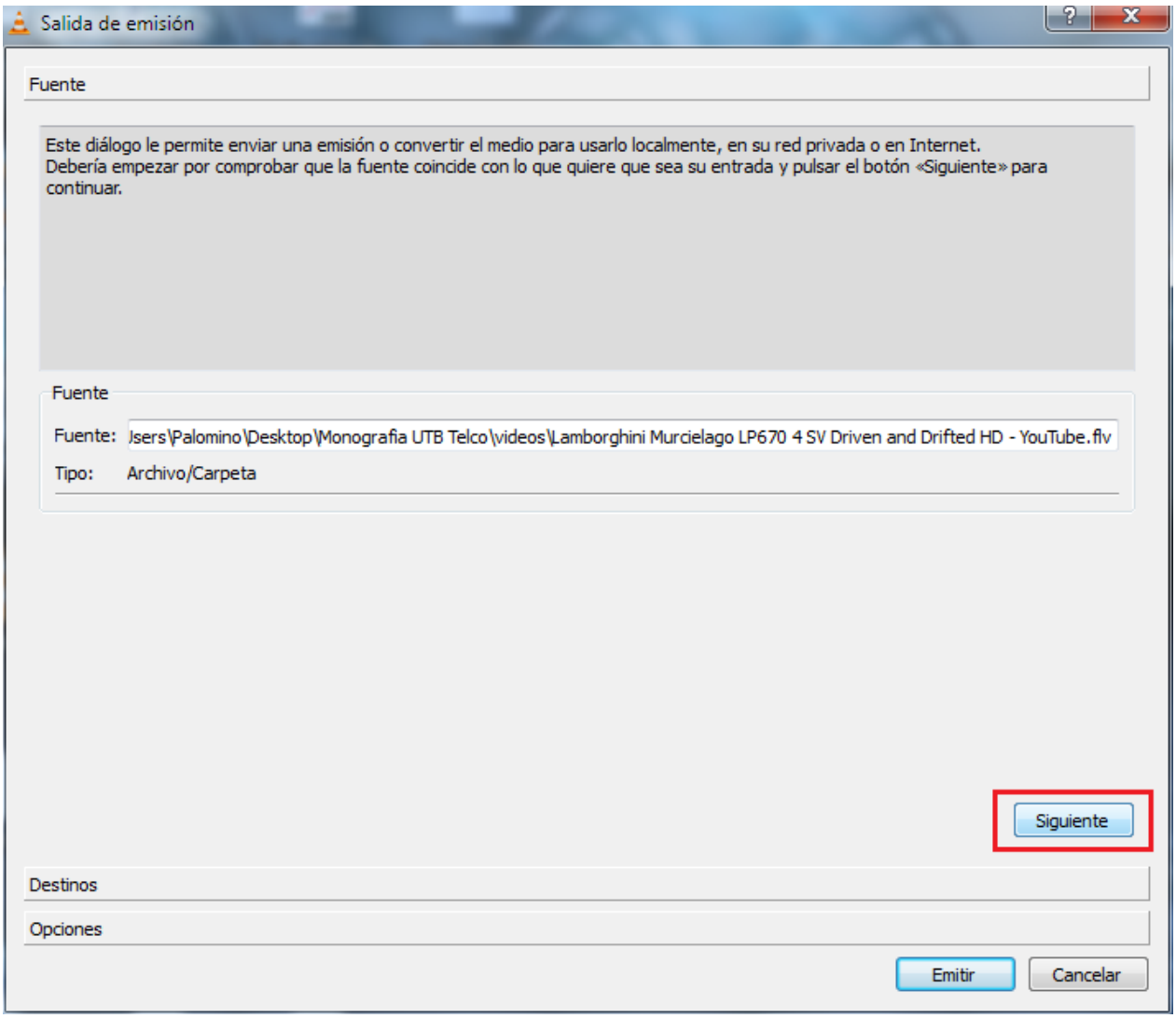

Figura 17. Ventana Fuente VLC

Posteriormente se selecciona el mecanismo de transmisión que se utilizará. En este caso se selecciona UDP. En este cuadro se puede seleccionar o activar la casilla “ver en el local”, con la cual se podrá observar localmente (en el Servidor) aquello que se está transmitiendo, es decir, si esta casilla está activa el VLC del Servidor también reproducirá el video que se está emitiendo.

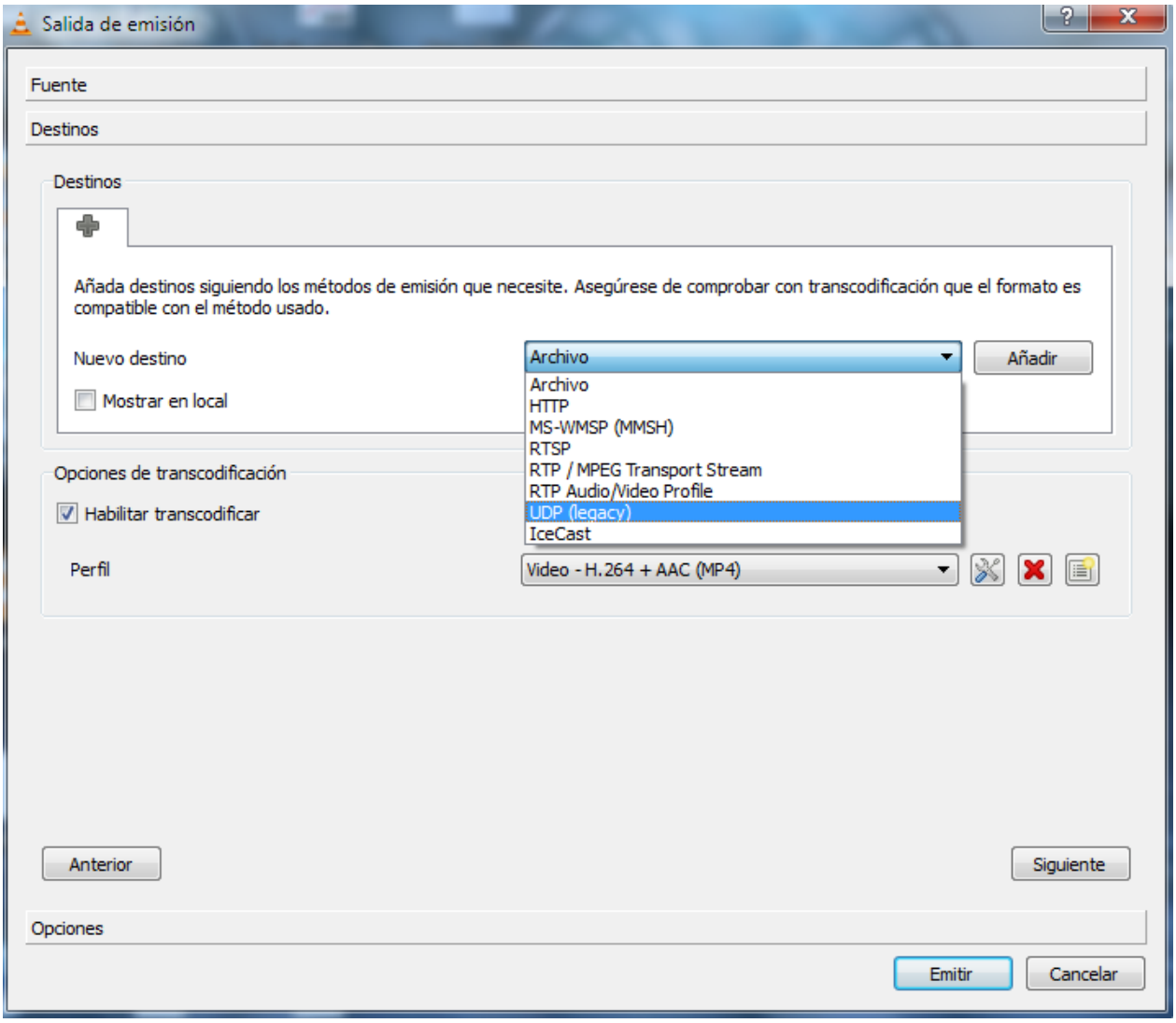

Figura 18. Selección del protocolo de transmisión

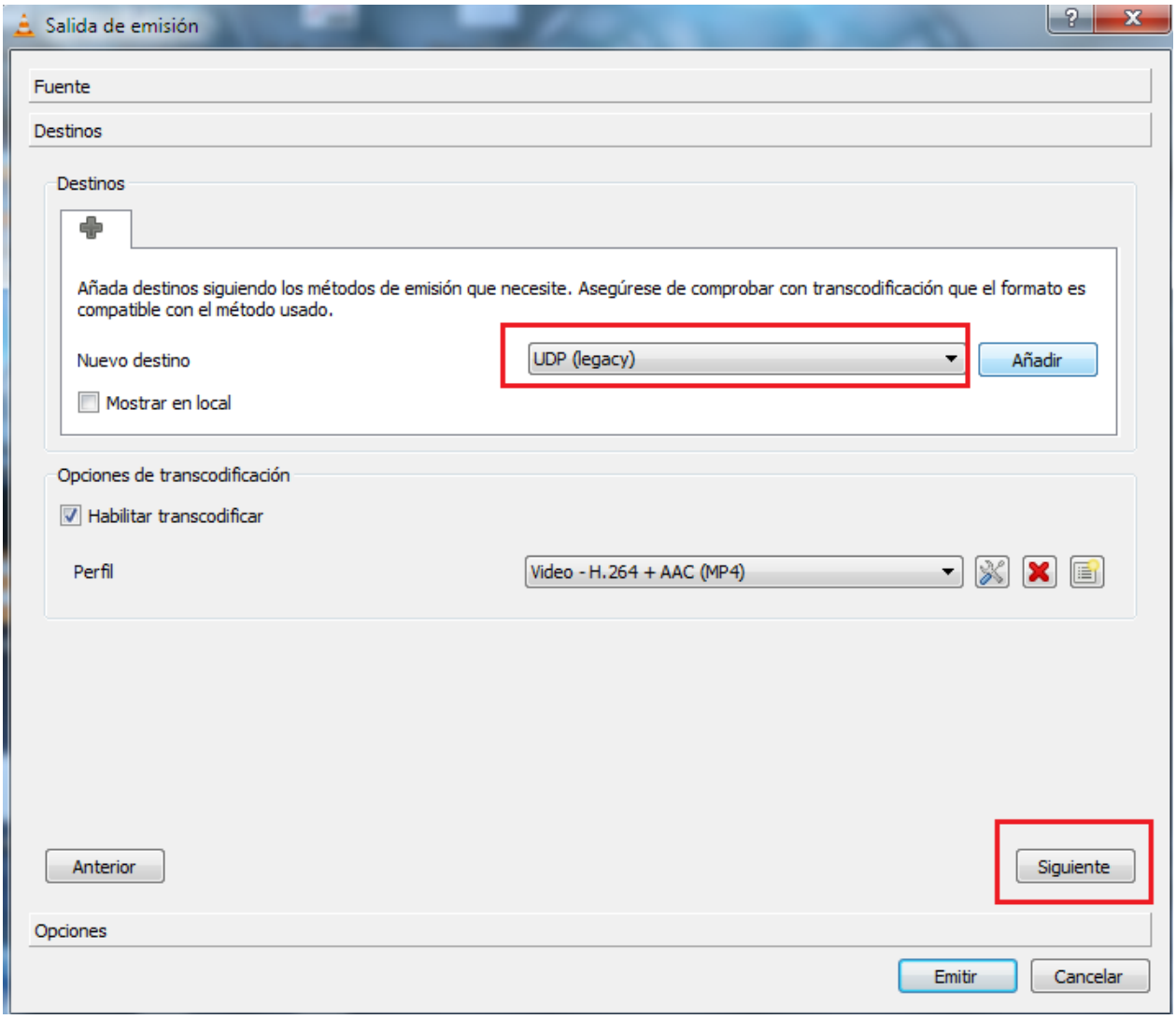

Figura 19. Selección del transcodificador

Luego se procede a indicar la dirección ip multicast que se utilizará para la transmisión y recepción de los datos y se continúa presionando Siguiente:

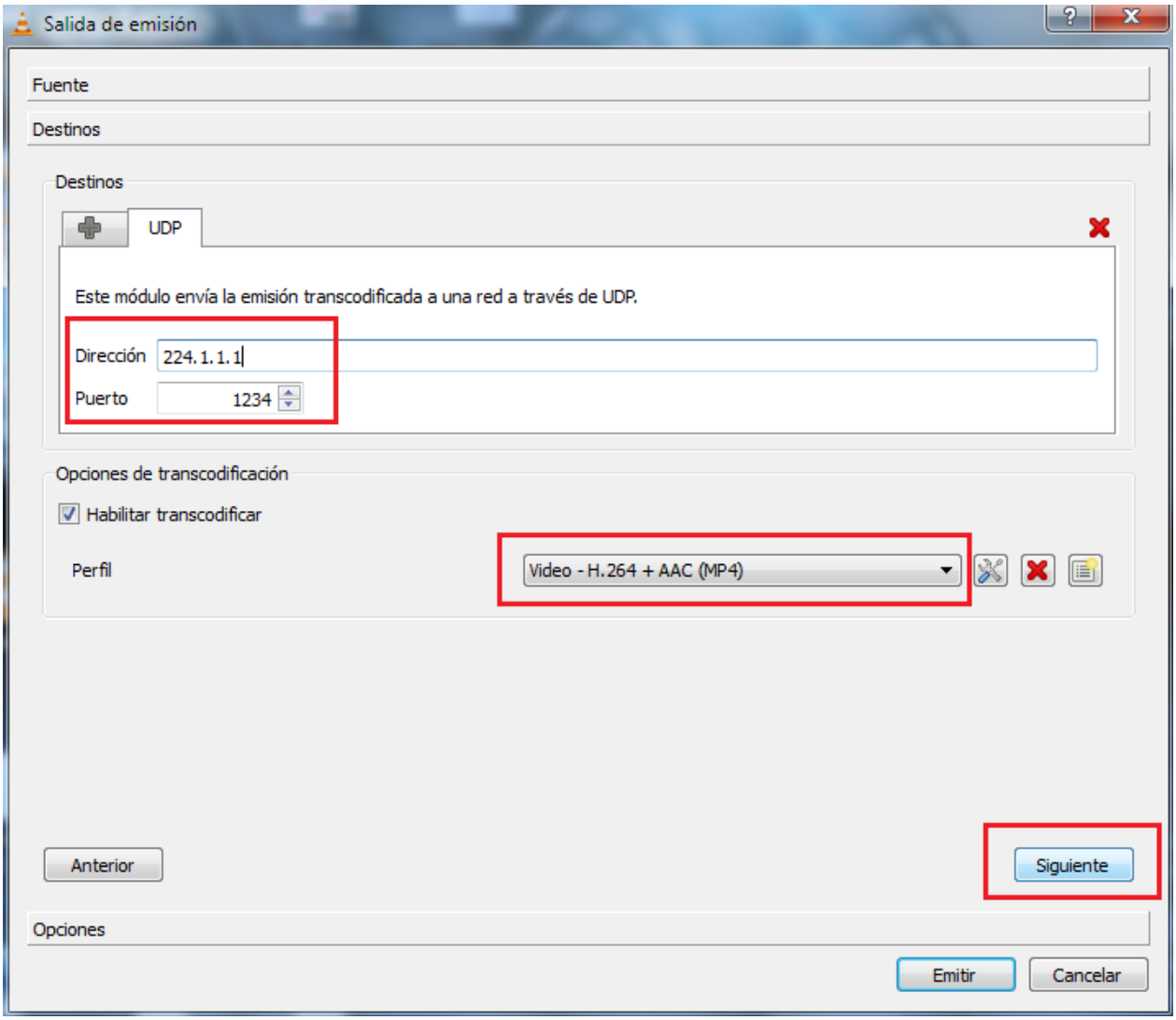

Figura 20. Ingreso dirección IP multicast

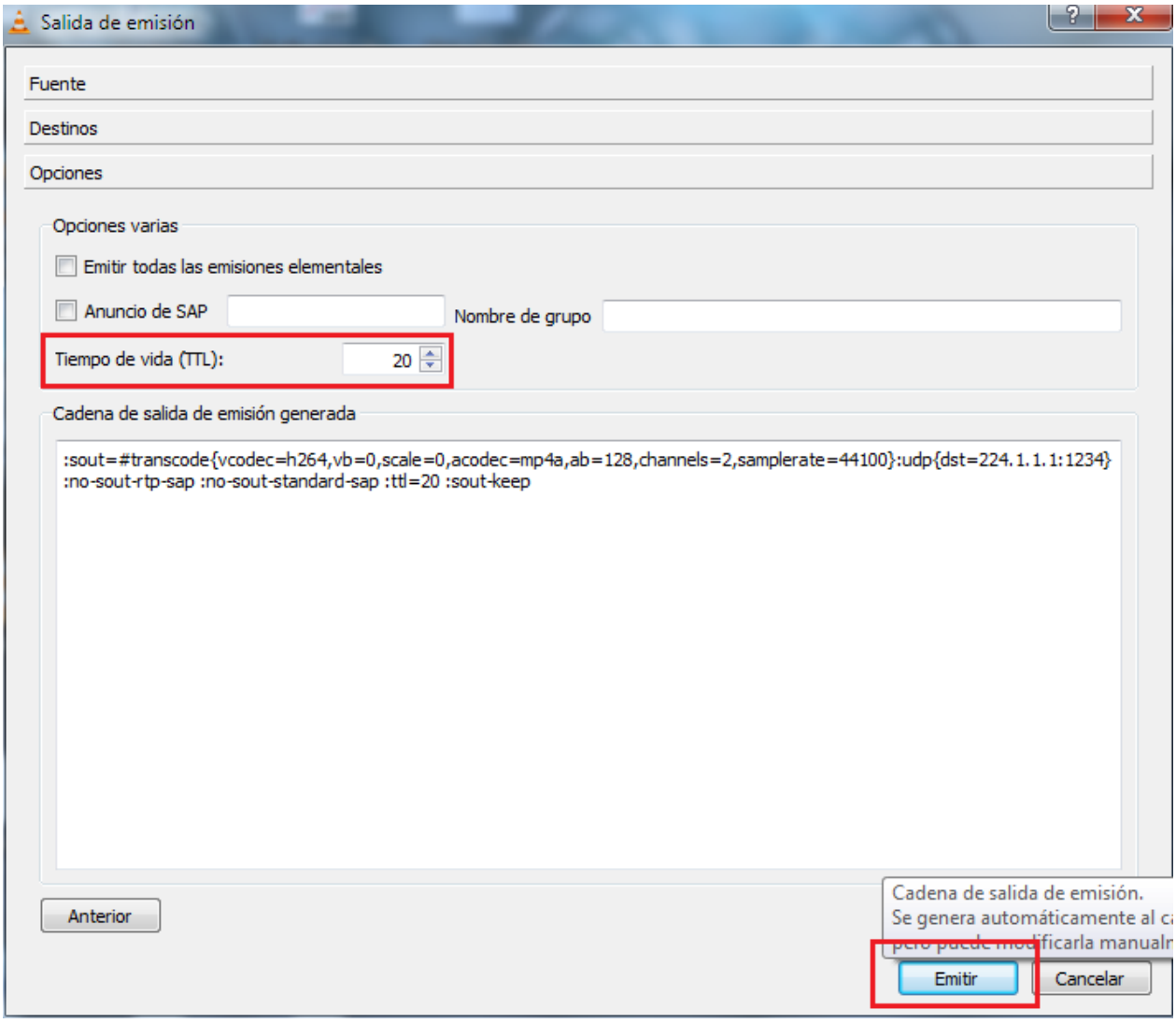

Figura 21. Ingreso de Time to Live (TTL)

En la imagen anterior es posible especificar en el campo TTL (Time to Live) la cantidad de saltos que puede llegar a tener la transmisión de los paquetes. Se recomienda tomar un poco más del valor real de saltos que tenemos en nuestra topología y evitar así posibles fallas o perdidas en la transmisión. Se presiona el botón Emitir posteriormente.

Y, finalmente, aparece nuevamente la ventana VLC pero esta vez emitiendo el video seleccionado:

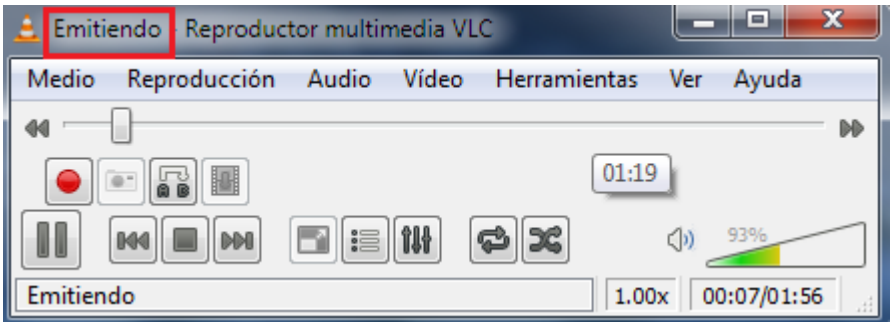

Figura 22. Interfaz VLV emitiendo

4.6.2 *En el Cliente:*

En el lado del Cliente se va a reproducir el video emitido por el servidor haciendo los siguientes pasos.

En la interfaz inicial de VLC se presiona Medio/Abrir volcado de red:

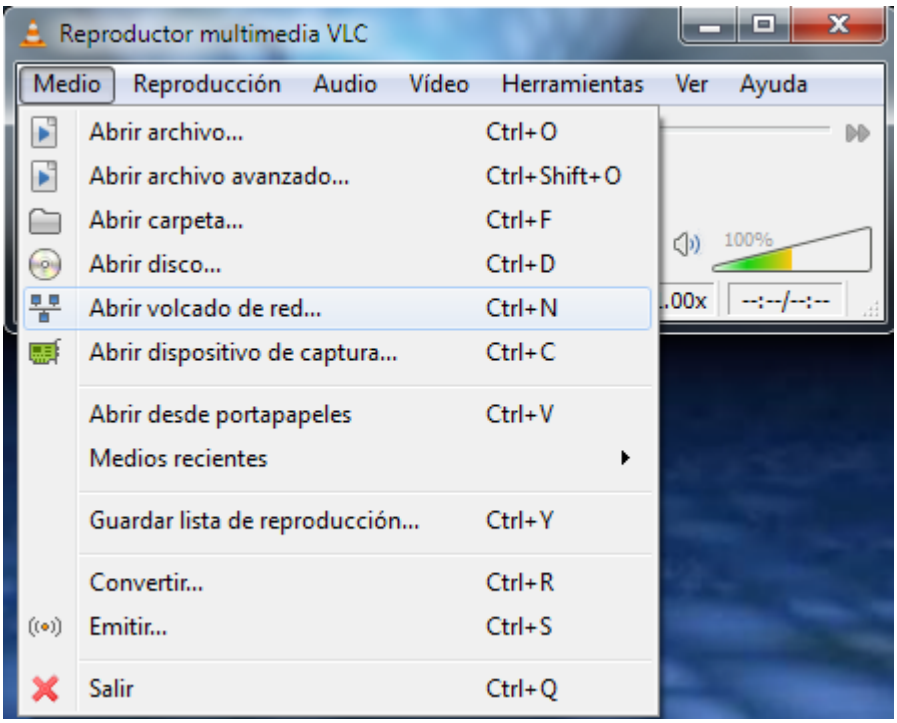

Figura 23. Abrir volcado de red en Cliente

En la pestaña Red se ingresa la dirección IP a través de la cual se está transmitiendo el video streaming, la misma que se digitó en el proceso Servidor y especificando el protocolo que se está utilizando para la transmisión, de la siguiente forma:

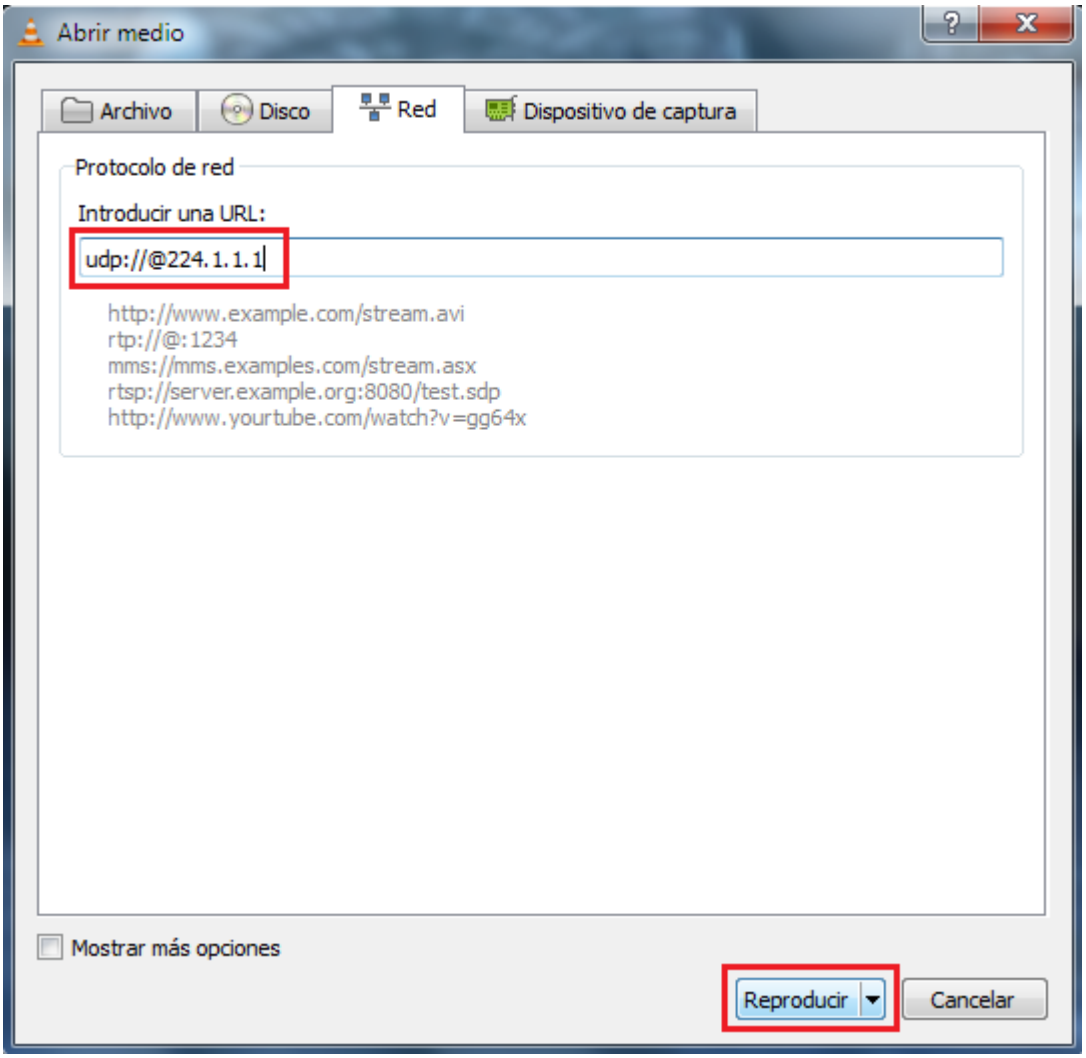

Figura 24. Ingreso dirección ip multicast para recepción de la transmisión

Se presiona el botón reproducir y en la imagen siguiente se visualiza como es la interfaz de video de VLC reproduciendo un video streaming a través del protocolo UDP:

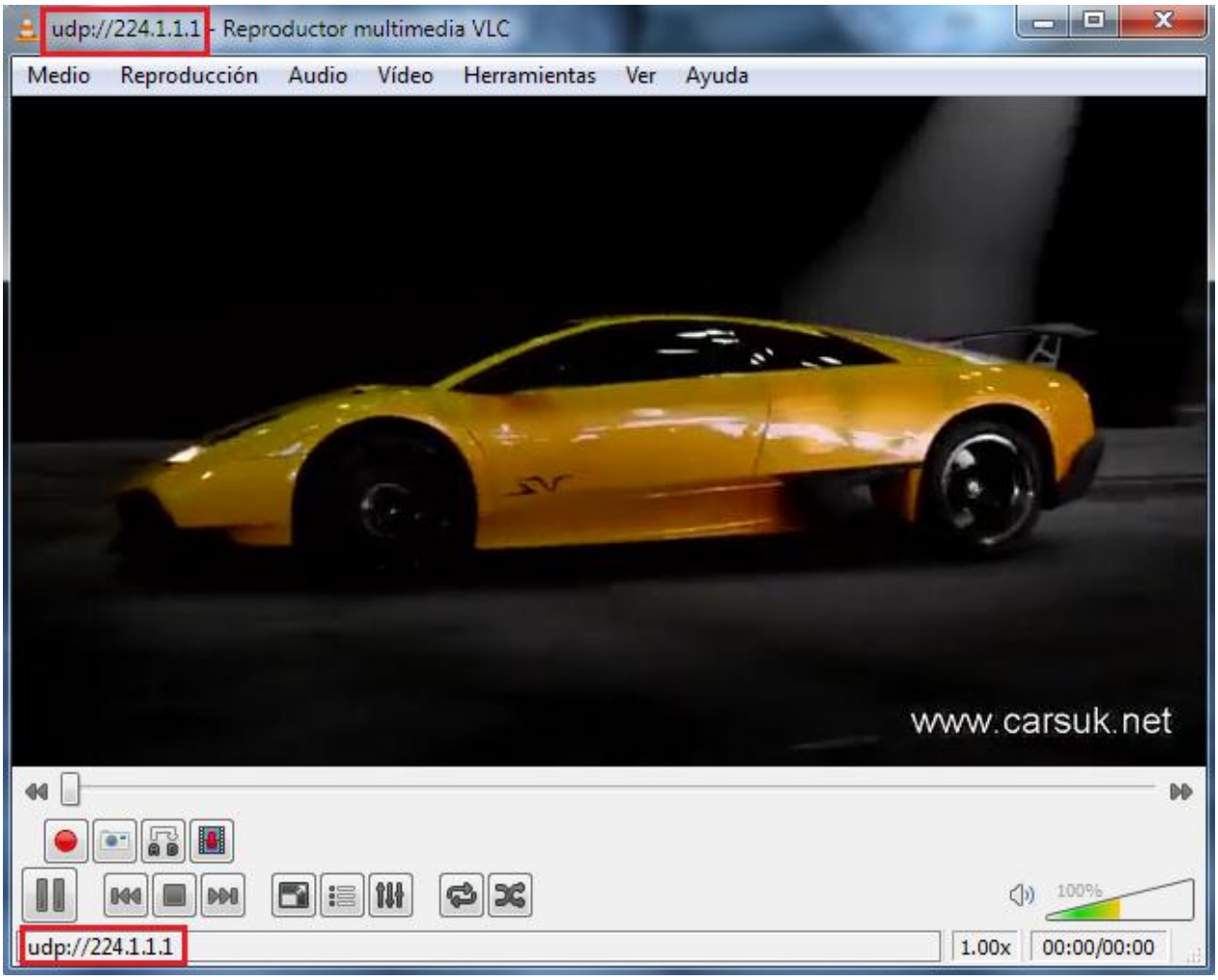

Figura 24. Interfaz VLC reproduciendo video Streaming

NOTA:

A continuación se realiza la primera prueba del Laboratorio que consiste en transmitir el video Streaming desde el Servidor y reproducirlos desde el Cliente con la configuración de red que se tiene hasta este punto, no hay saturación en el canal y no se posee aun mecanismos de calidad de servicio.

4.7 Prueba Reproducción del video streaming sin congestión y sin QoS

La primera prueba con la transmisión del video streaming, consiste en emitir el video sobre la topología GNS3 pura, sin tráfico pesado más allá de la comunicación de convergencia entre los

routers y sin calidad de servicio configurada manualmente. Como ya se detalló anteriormente, se emite el video desde la maquina XP y se recepciona en el equipo físico con Windows 7.

4.7.1 Resultado: Imágenes

El resultado de esta transmisión es un video casi sin pixelamiento. Se puede ver claramente todo el video y el audio del mismo acorde al tiempo de video.

Se toman imágenes de muestra para observar y dejar plasmado los resultados de la transmisión:

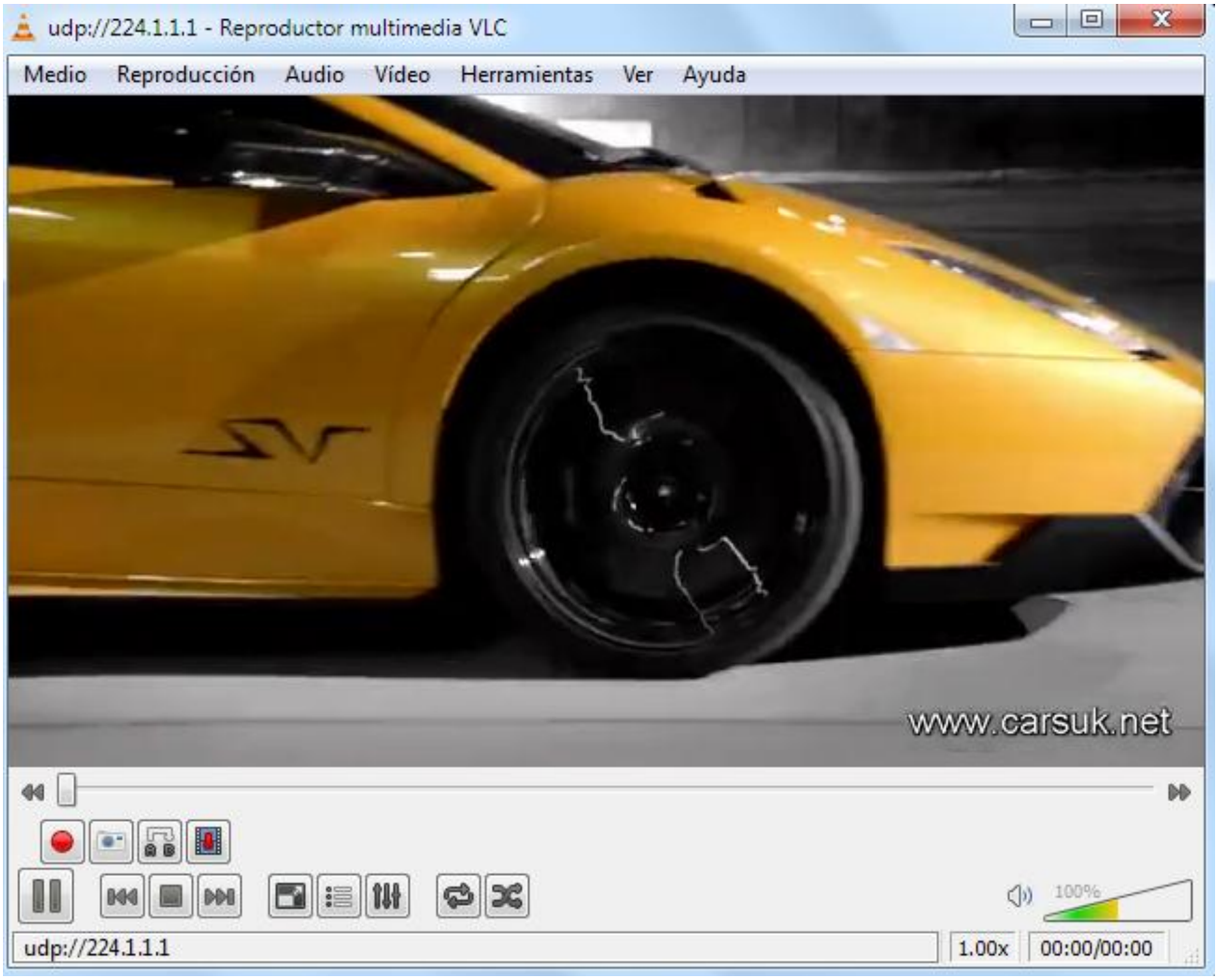

Figura 25. Imagen de muestra 1 de reproducción sin saturación

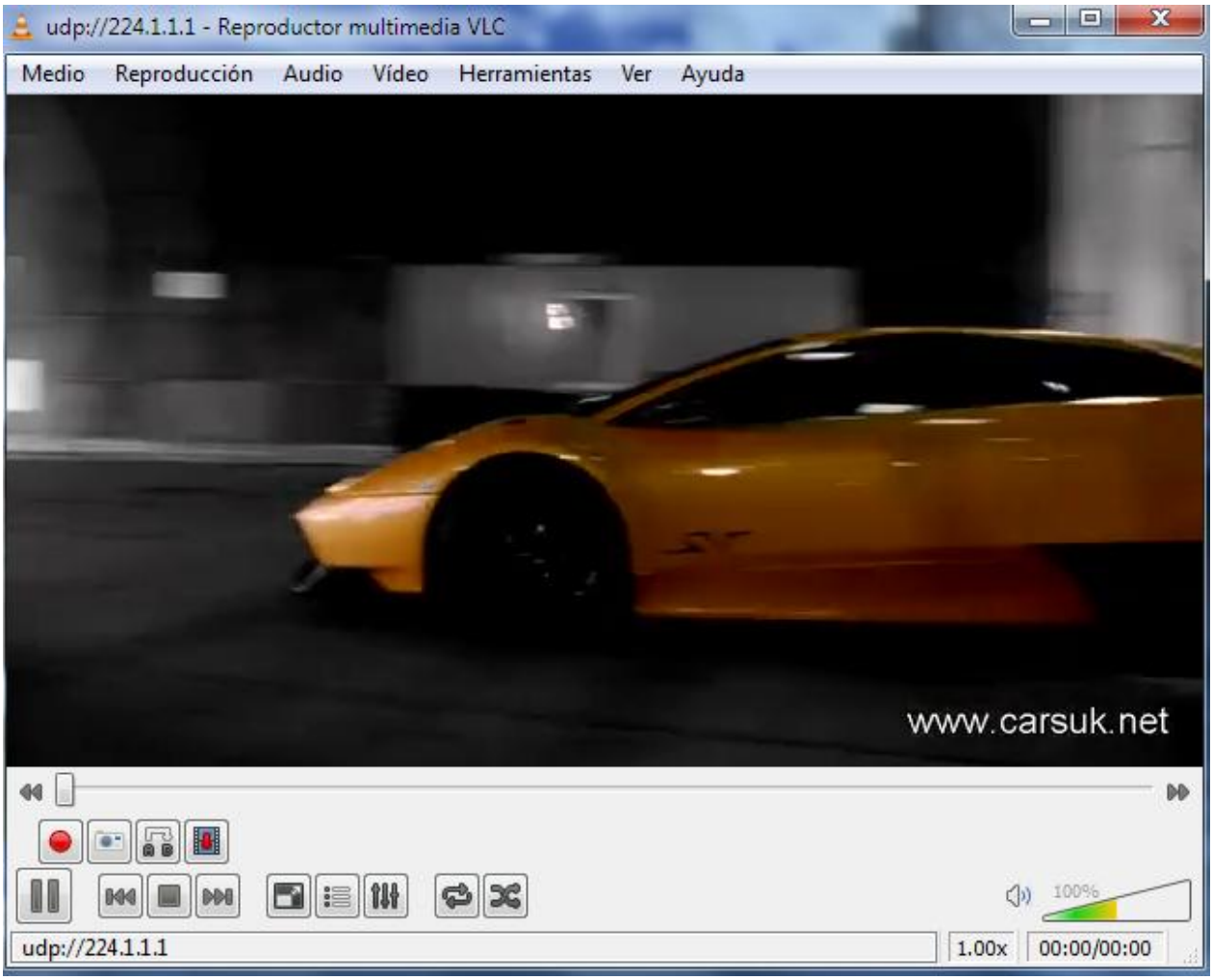

Figura 26. Imagen de muestra 2 de reproducción sin saturación

NOTA:

Con los pasos anteriores y la configuración de los routers que se tiene, los enlaces no tienen la cantidad de tráfico suficiente para deteriorar la imagen del video transmitido por el Servidor. Para poder lograr una pérdida de paquetes y un retraso entre ellos considerable, se hace necesario saturar la red utilizando, en este caso de laboratorio, un generador de tráfico el cual, insertará en el canal de comunicaciones una cantidad de tráfico a nuestra consideración al punto que permita percibir el deterioro de la señal.

4.8 Configuración Generador de Tráfico

Para generar tráfico y saturar de alguna manera la red de comunicaciones por la cual transitará los paquetes de video, se utiliza un módulo del software CAPSA de COLASOFT en donde se nos otorga la posibilidad de inyectar a nuestra interfaz de salida una gran cantidad de paquetes según nuestras especificaciones.

El siguiente es la interfaz inicial del generador de tráfico:

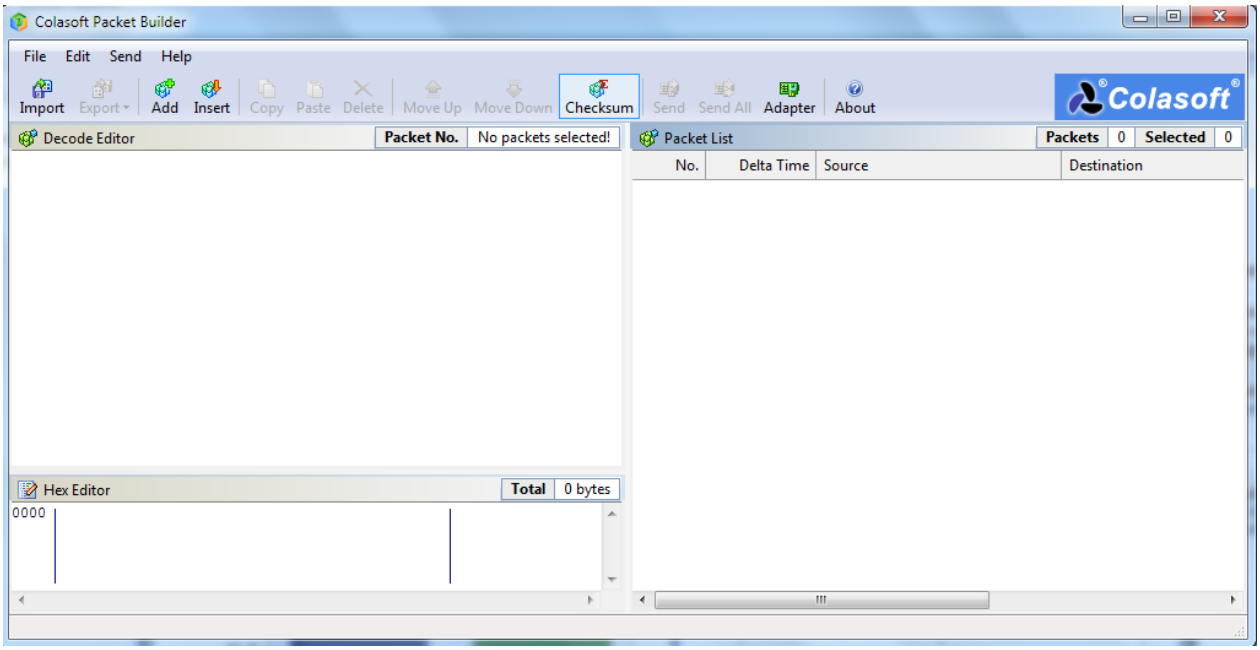

Figura 27. Interfaz inicial Generador de Trafico Capsa Colasoft

Se tiene la posibilidad de generar tráfico de paquetes TCP, UDP y para este caso utilizaremos el protocolo de resolución de direcciones ARP:

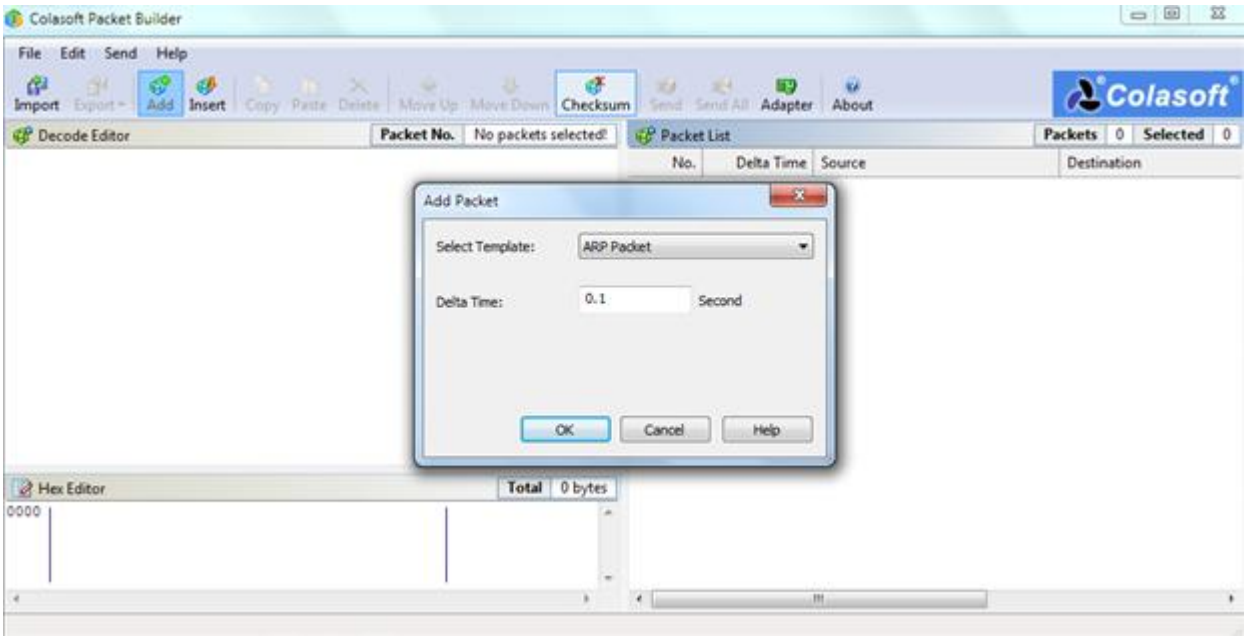

Figura 28. Tipo de tráfico a generar

Se debe escoger la interfaz de salida de este tipo de tráfico y para este caso se selecciona la interfaz virtual de bucle invertido utilizada para conectar una interfaz de router en GNS3 a nuestra interfaz LAN local en nuestro equipo físico:

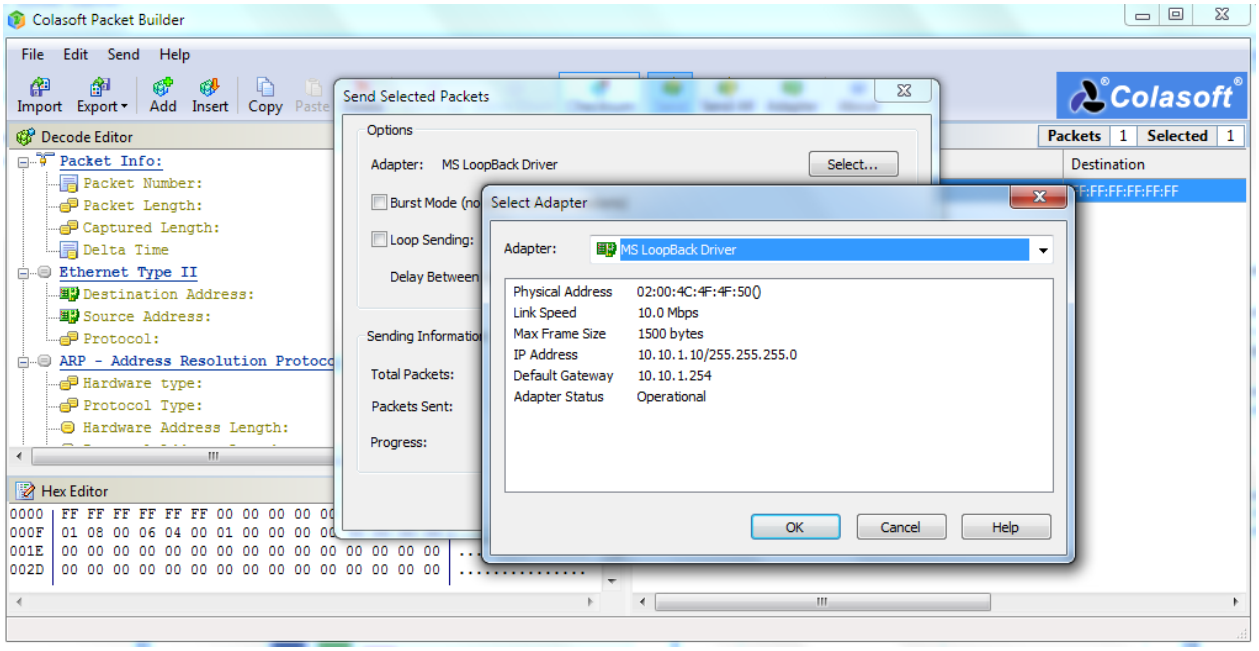

Figura 29. Selección de la interfaz de salida

Seleccionada la interfaz se configura la cantidad de paquetes que se va a inyectar y el tiempo entre cada una de estas transmisiones. Para este caso se utiliza un bucle infinito de generador de paquetes y un tiempo de intervalo de 20 milisegundos entre paquetes:

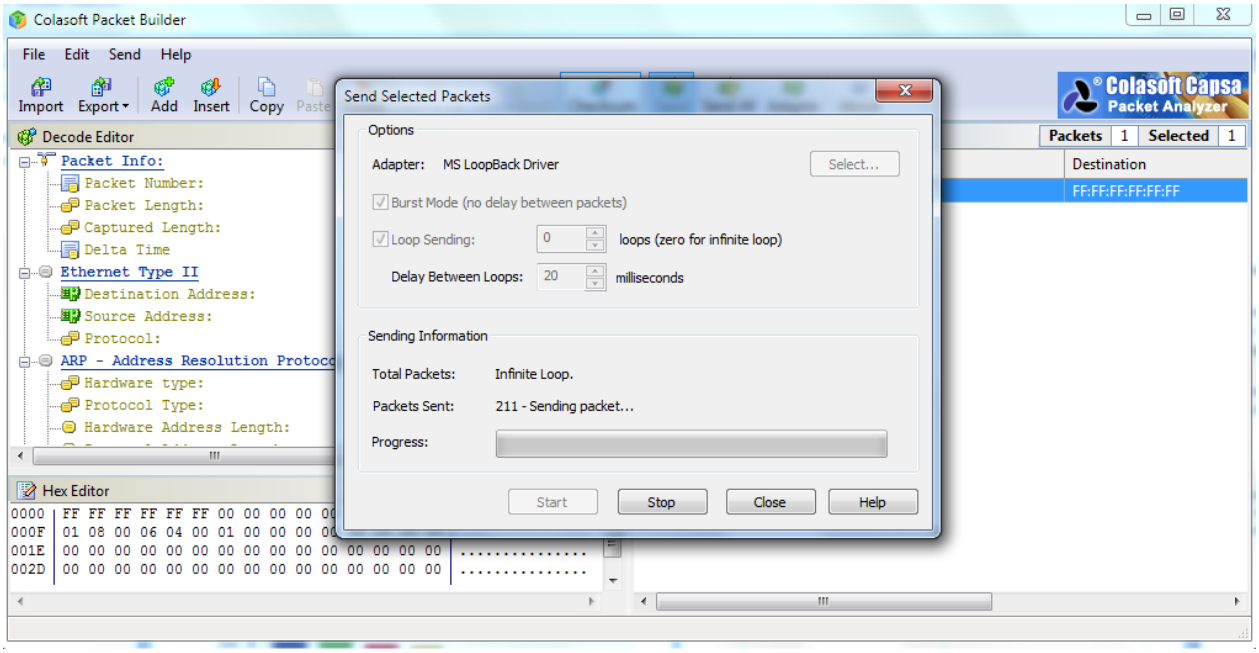

Figura 30. Generando tráfico

En las gráficas siguientes se puede ver la manera como es incrementado el uso del canal a medida que inyectamos tráfico pesado.

La prueba que se muestra es tomada de una tormenta ARP. A medida que se inyecta tráfico, la posibilidad de pérdida de paquete y de saturación en la red aumenta de manera progresiva:

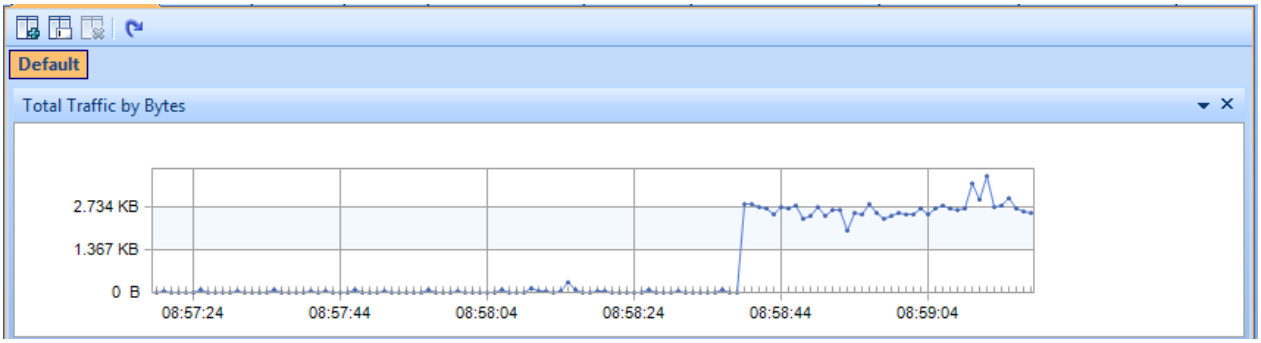

Figura 31. Grafica de saturación del canal

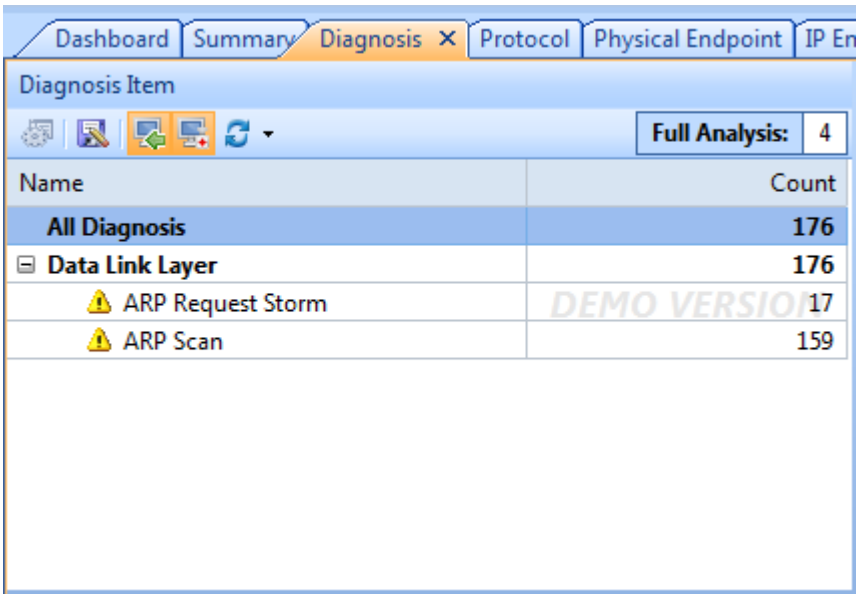

Figura 32. Detalle de los paquetes ARP inyectados al canal

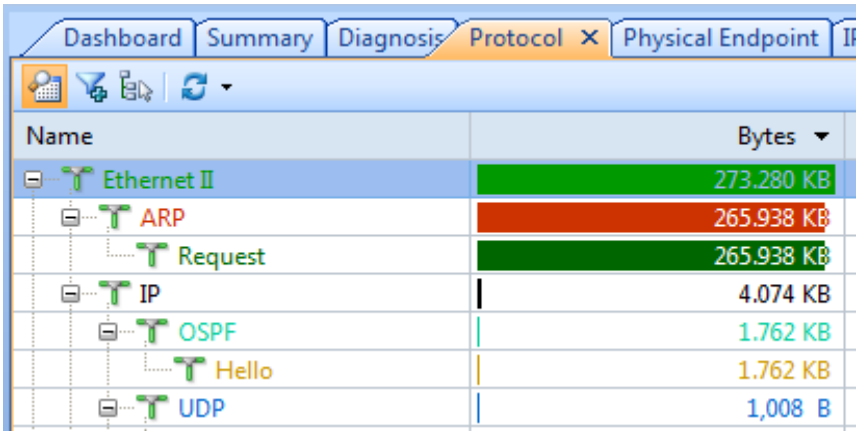

Figura 33. Tamaño del trafico ARP generado

En las imágenes anteriores se puede observar en el analizador de tráfico la cantidad de tráfico generado por una tormenta de broadcast.

NOTA:

Ahora que se sabe cómo inyectar tráfico a la red, se procede a transmitir nuevamente el video streaming esta vez con la red saturada parcialmente. Es decir, se satura la red y

luego se transmite el video desde el Servidor para ser reproducido posteriormente por el Cliente.

4.9 Prueba Generación de tráfico sin QoS y reproducción del video streaming

En la siguiente prueba se transmite el video streaming sobre una red congestionada y con tráfico abundante entre los routers, lo cual, deteriora notablemente la transmisión y la calidad de experiencia en el lado del Cliente.

Al haber una mayor cantidad de tráfico sobre el canal de datos, la posibilidad de pérdida de paquetes se incrementa notablemente así como también el retraso de los mismos.

Para poner a prueba nuestra red MPLS se inyecta una gran cantidad de trafico ARP y además, se lleva a cabo una prueba PING infinita con una variación en el tamaño del paquete por defecto para hacerlo más propenso a saturación y desgaste del ancho de banda de nuestras interfaces seriales.

4.9.1 Resultados: imágenes con red saturada sin QoS

En las imágenes que se muestran a continuación se puede ver con claridad el deterioro de la imagen de video transmitido en una red saturada y propensa a fallos y retrasos. La imagen se pixela notablemente y la perdida de paquetes hace que el video y el audio pierda momentos del mismo.

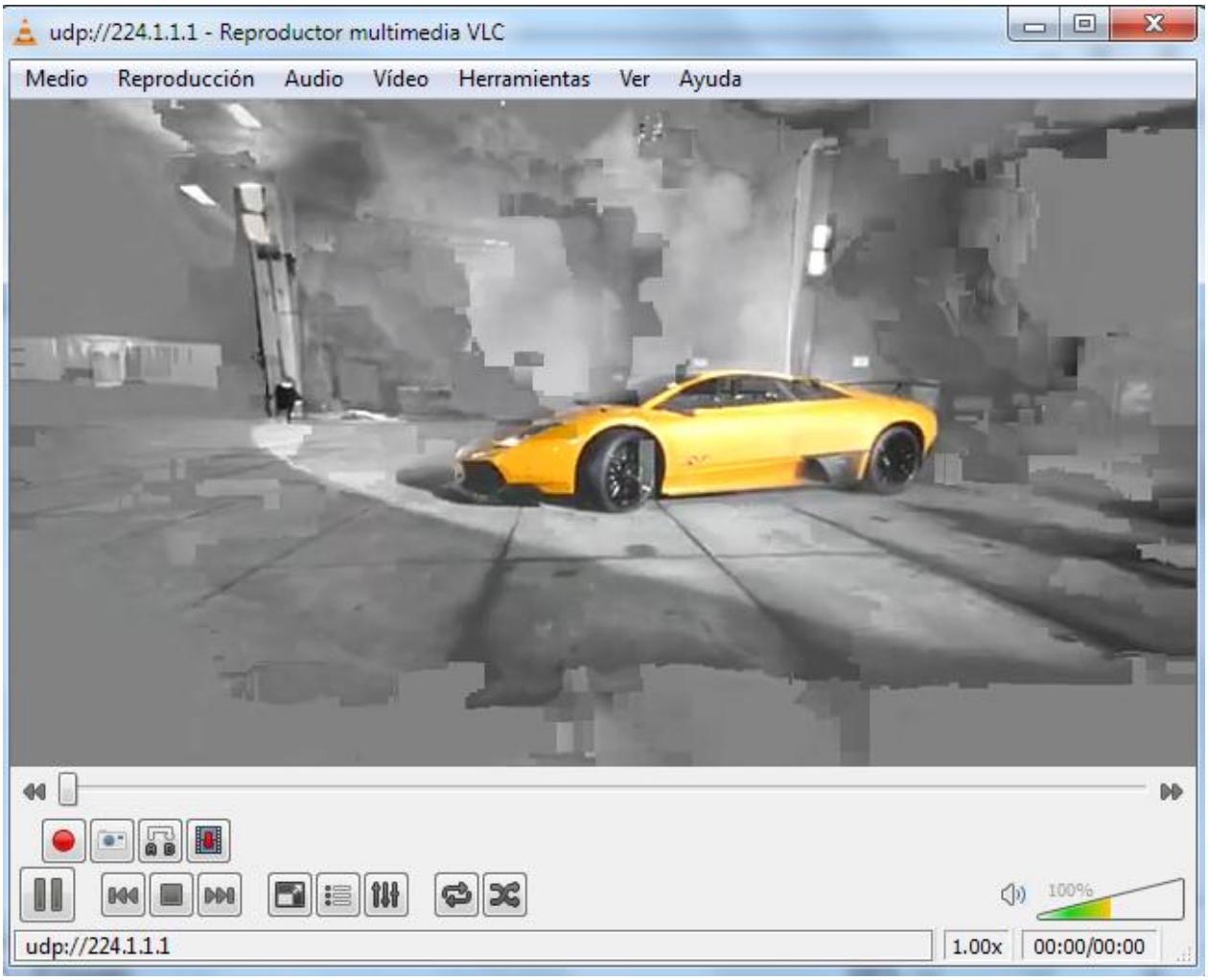

Figura 34. Imagen de muestra 1 de reproducción con saturación

Figura 35. Imagen de muestra 2 de reproducción con saturación

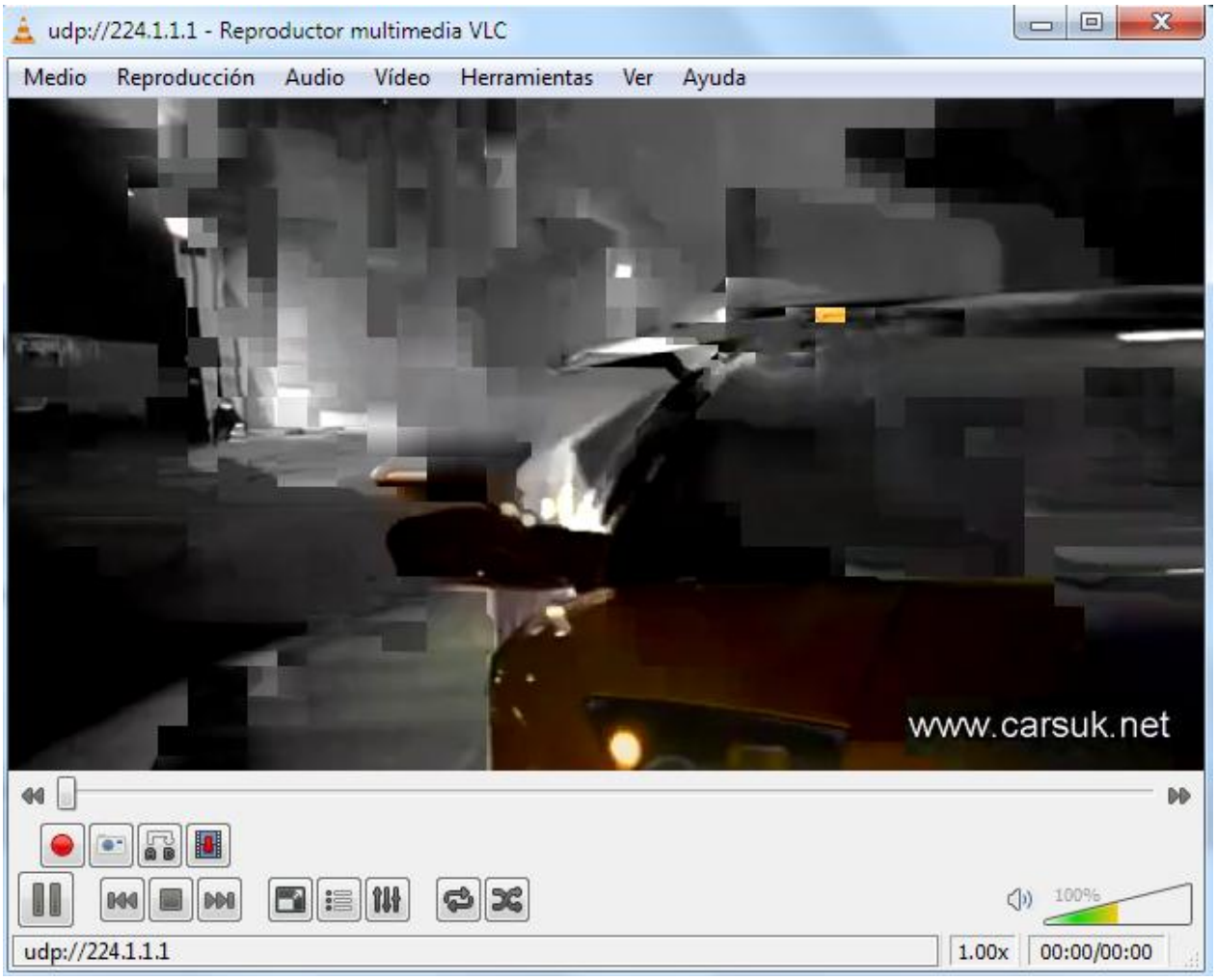

Figura 36. Imagen de muestra 3 de reproducción con saturación

NOTA:

Se pudo observar en los escenarios anteriores la calidad de la señal transmitida con y sin saturación en la red. A continuación se configurará los mecanismos de calidad de servicio en nuestros routers y se observa nuevamente los resultados.

4.10 Configuración QoS DiffServ

Con el fin de garantizar la calidad de experiencia en el Cliente, se configura algunos mecanismos de calidad de servicio sobre nuestra red MPLS.

Como es sabido, el paquete MPLS proporciona un campo Experimental (EXP) que ha sido estandarizado para ser utilizado en los mecanismos de calidad de servicio. Este campo es el que se debe modificar en cada paquete que sea necesario en nuestra política de QoS para ser revisado posteriormente en la interfaz de salida y otorgar a partir de este el mecanismo de Calidad de servicio determinado.

Se crea una clase llamada Stream la cual modificará el campo experimental del paquete MPLS basada en una lista de acceso que permite los paquetes UDP en cierto rango:

```
R7#sh class-map
 Class Map match-any class-default (id 0)
   Match any

 Class Map match-all Stream (id 1)
   Match access-group  101

 Class Map match-all Stream-EXP (id 2)
   Match mpls experimental topmost 1

R7#
```

Figura 37. Configuración de las clases

Posteriormente se crean las reglas dentro de unas *policy-map.* Para el caso se crearon dos: MARCACION, que está asociada a la clase Stream; y otra, POLITICA-QOS que asociamos a la clase Stream-EXP, la cual, otorga un porcentaje de ancho de banda para garantizar por allí la transmisión eficiente del video streaming:

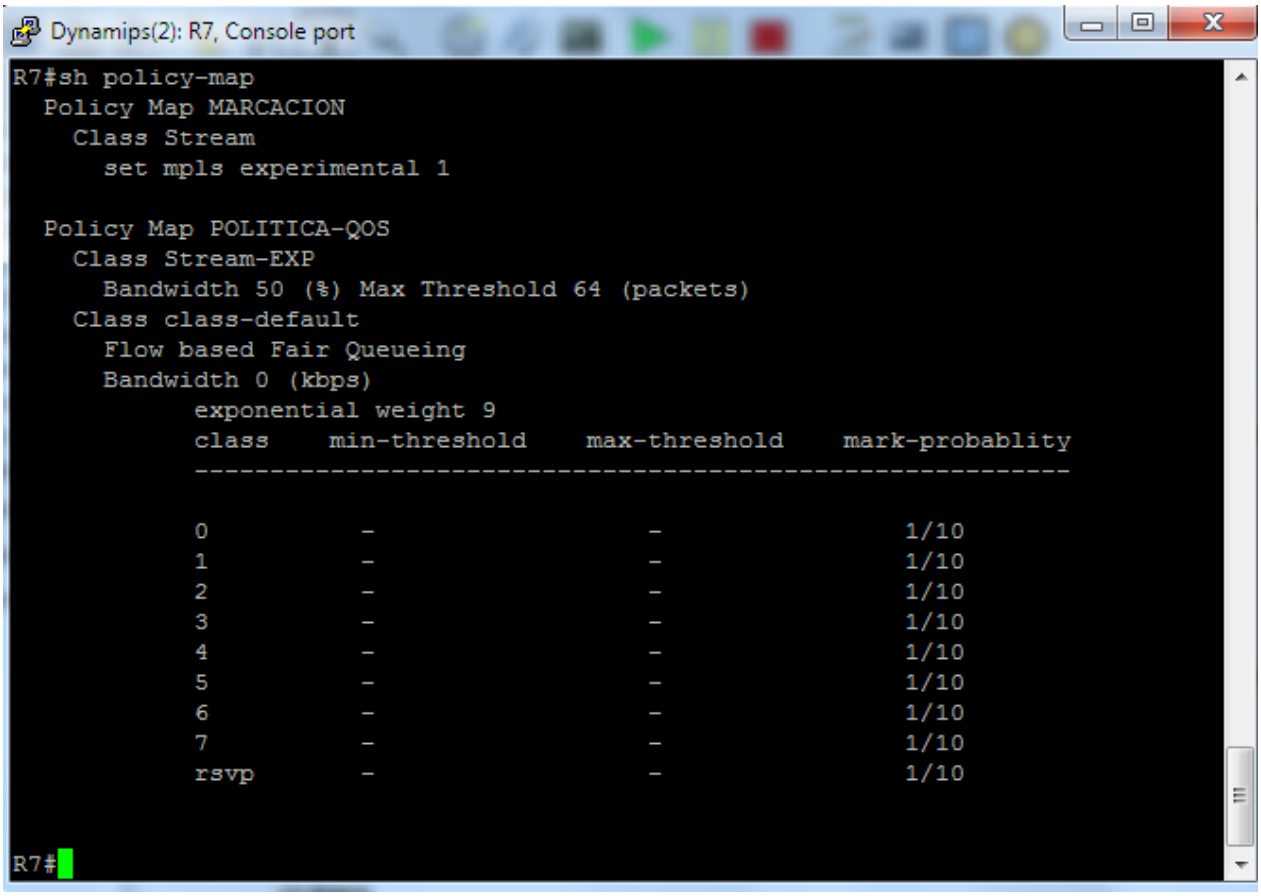

Figura 38. Configuración Policy-map

4.10.1 Funcionamiento Marcación en Policy-Map en interfaz de entrada

En este apartado se hace una verificacion del funcionamiento de las policy-map en la interfaz de entrada y constatando que este funcionando correctamente. A traves del proximo comando se observa la cantidad de paquetes que concuerdan con la politica de la clase y el tratamiento que se le dio.

Para esto se transmitió un video sobre los puertos UDP del rango de la lista de acceso con el fin de que, a traves del policy-map MARCACION se modificara su campo Experimental en 1 como se muestra:

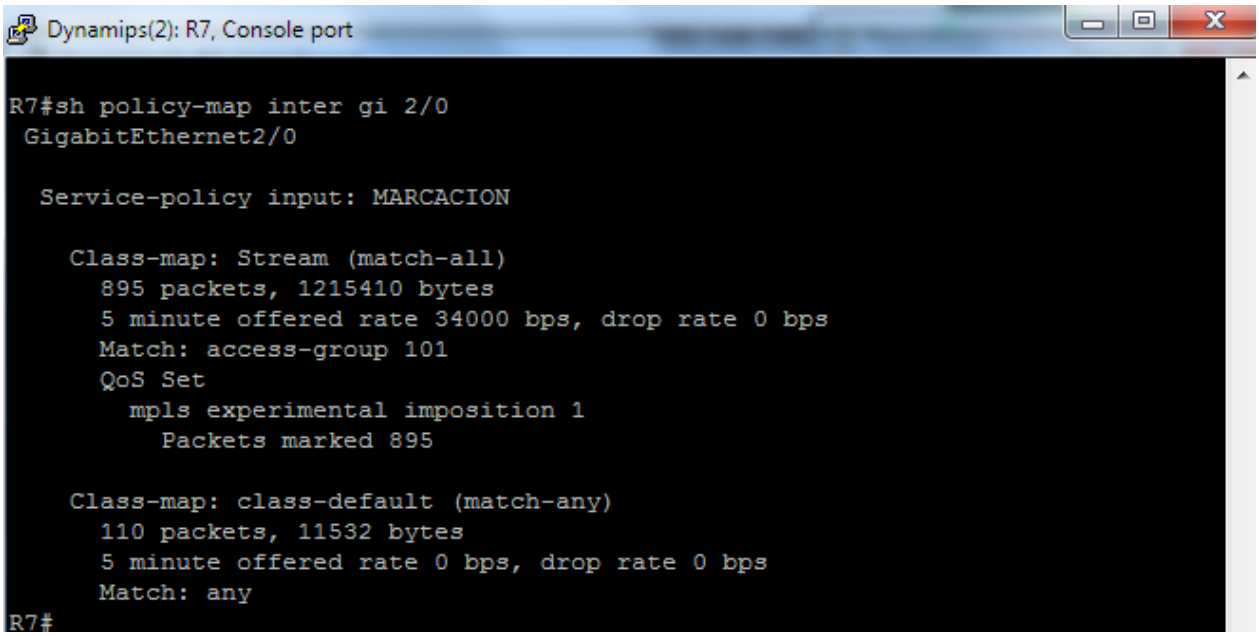

Figura 39. Muestra del funcionamiento de la política en la interfaz de entrada

4.10.2 Funcionamiento QoS en Policy-Map en interfaz de salida

En la imagen siguiente se muestra el correcto funcionamiento de la Policy-Map POLITICA-QOS asociada a la interfaz de salida de nuestro router de borde. Estas reglan actúan luego de haber realizado con éxito la marcación de los paquetes UDP con el campo Experimental en 1. Se verifican los paquetes que estén marcados de esa manera y asocian una política QoS preestablecida. Se observa con el siguiente comando la cantidad de paquetes que se le aplicaron las políticas:

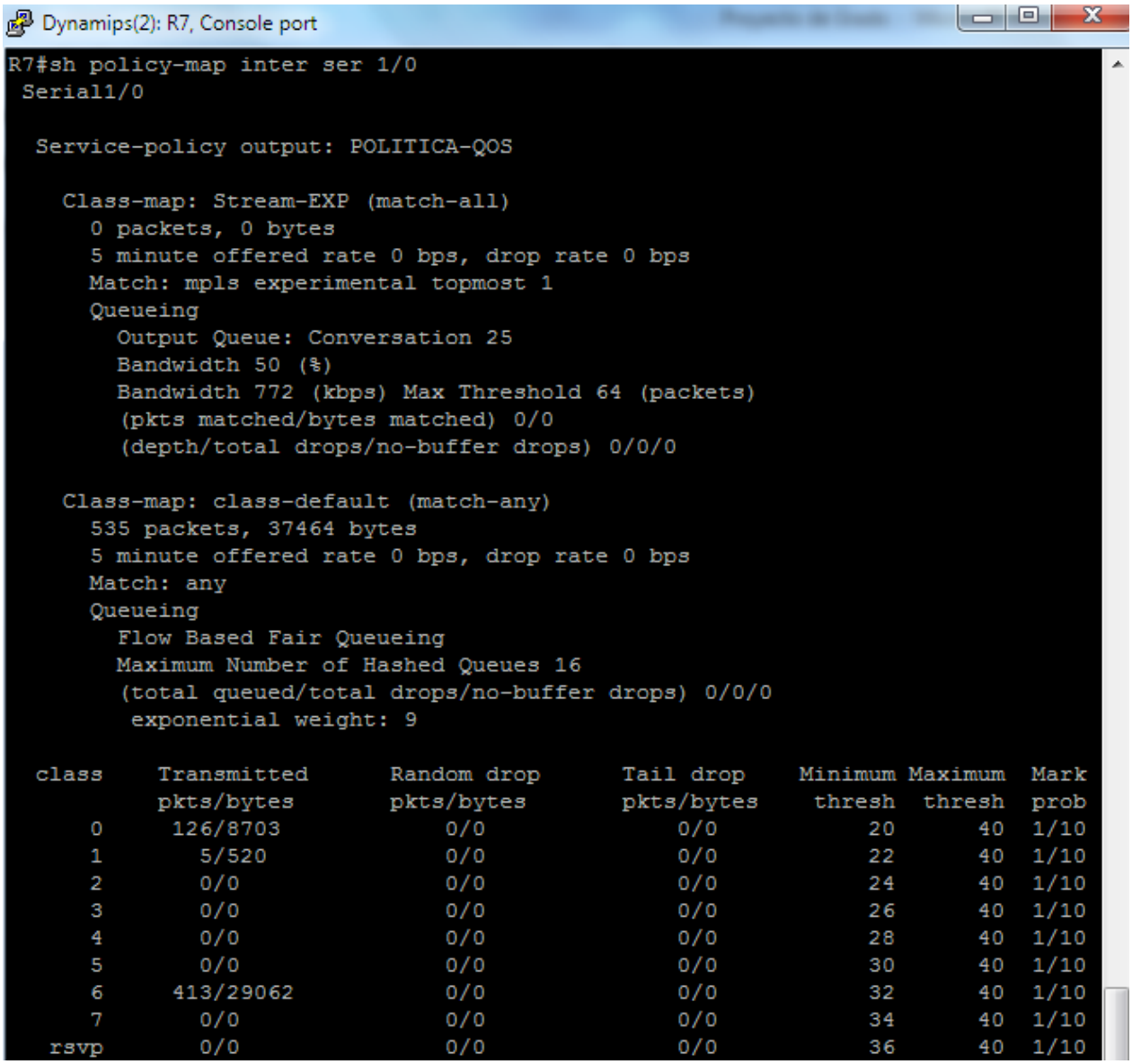

Figura 40. Muestra del funcionamiento de la política en la interfaz de salida

NOTA:

Al haber configurado los mecanismos de calidad de servicio adecuado para esta situación, nuevamente se inyecta y se satura el canal con gran cantidad de tráfico y se transmite nuevamente el video streaming. Se observan los siguientes resultados:

4.11 Prueba Reproducción de video streaming con red saturada y QoS

Con el tráfico ARP generado sobre nuestra canal de comunicaciones así como una prueba de ping de la muerte (se modifica el tamaño del paquete ICMP y se envía en un loop infinito), se transmite el video streaming observándose en la gráfica las ráfagas de la reproducción del video y de la voz:

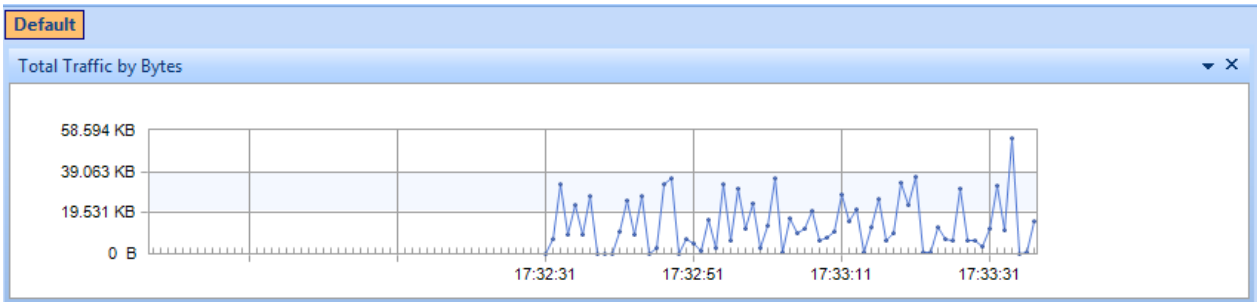

Figura 41. Reproducción con red saturada y QoS

Se hace una prueba PING y se prueba la congestión del canal de datos en este momento:

```
Haciendo ping a 10.10.0.10 con 32 bytes de datos:
Respuesta desde 10.10.0.10: bytes=32 tiempo=1550ms TTL=124
Respuesta desde 10.10.0.10: bytes=32 tiempo=2377ms TTL=124
Respuesta desde 10.10.0.10: bytes=32 tiempo=2498ms TTL=124
Respuesta desde 10.10.0.10: bytes=32 tiempo=1254ms TTL=124

Estadísticas de ping para 10.10.0.10:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 1254ms, Máximo = 2498ms, Media = 1919ms
```

Figura 42. Prueba PING con red saturada y QoS

Aunque esté ocurriendo esto en la red, el video alcanza al Cliente con los parámetros necesario para una reproducción optima y que no degrade la calidad de experiencia, como se muestra:

Figura 43. Imagen de muestra de reproducción con red saturada y QoS

Al finalizar la reproducción y el generador de tráfico se observa el despeje casi total de nuestro canal de datos:

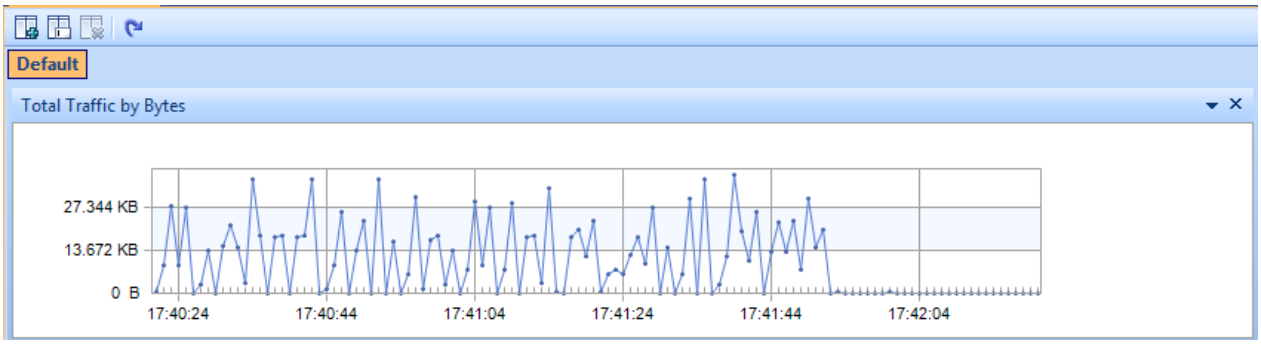

Figura 43. Análisis de tráfico al finalizar la reproducción

En una prueba PING desde el Cliente al Servidor los tiempos de respuesta se muestran, casi que inmediatamente, con un retardo mínimo para la red montada:

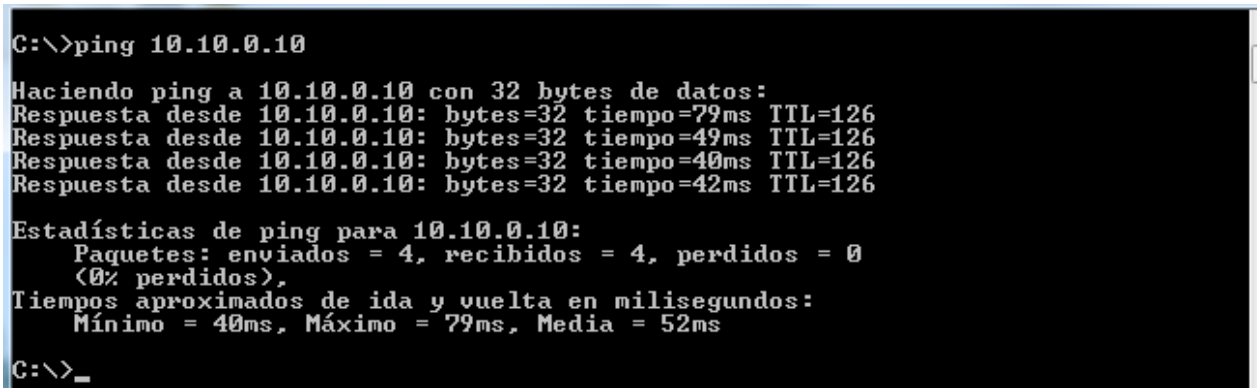

Figura 44. Prueba PING al finalizar la reproducción

5. CONCLUSIONES

La necesidad de ofrecer calidad de servicios en las nuevas aplicaciones y servicios hace necesario la adopción de modelos de QoS. Con la utilización de DiffServ se consigue una notable mejora en el rendimiento de las redes IP, pero todavía es mejorable, y quedan determinadas circunstancias, como la caída de enlaces, donde no se resuelve positivamente la situación. Existen otras soluciones o complementos como puede ser aprovechar las grandes posibilidades que MPLS ofrece en temas de calidad de servicio e ingeniería de tráfico. De esta forma, se consigue mejorar considerablemente el funcionamiento de las redes IP, como se ha podido observar en los resultados de las simulaciones efectuadas.

En el laboratorio prueba que se implementó para obtener resultados de la calidad de servicio en los enlaces MPLS se puede distinguir claramente la importancia de una buena implementación de estas políticas en nuestra de red comunicaciones y garantizar de esta manera un porcentaje de canal dedicado a nuestro tráfico más relevante.

En la experiencia obtenida con el desarrollo del laboratorio, se debe tener especial cuidado en la cantidad o porcentaje de ancho de banda que se desea otorgar a un tráfico en particular. Con mucho tráfico en el canal de comunicaciones, el ancho de banda garantizado al tráfico crítico constituye espacio por donde el resto de tráfico no podrá transitar. Esto deteriora el resto de comunicaciones por la pérdida de paquetes y descartes selectivos en los equipos enrutadores. De igual manera, si se reserva un ancho de banda menor al deseado o al peso del tráfico máximo que transita por el canal, la política de calidad de servicio no tendrá siempre éxito. Esto último se pudo recrear bajo la modalidad de prueba y error en el laboratorio. Se empezó a, progresivamente, aumentar el ancho de banda garantizado e ir observando el comportamiento de los datos (perdida de paquetes, retraso) a medida que se transmite el video streaming.

En una red de mayor escala y con gran cantidad de tráfico en volumen y en tipo, se pueden llevar a cabos los controles necesarios para hacer una correcta distinción de los mismos y controlar a través de la calidad de servicio el desempeño de los canales con relación al flujo del tráfico. Esto, sin lugar a dudas, constituye una herramienta clave para la escalabilidad de la red y el desempeño de la misma, manteniendo el tráfico pesado y menos importante con unas prioridades mínimas y el tráfico vital de la empresa con mayor grado de prioridad.

En general, se debió investigar profundamente ciertos aspectos referentes a la calidad de servicio sobre MPLS, multicast, emisión de video, entre otros temas para poder comprender la problemática planteada en el laboratorio y poder resolver paso a paso las diferentes circunstancias que se presentaban. Fueron pruebas muy didácticas que con los errores se hacían cada vez más comprensivas. Para lograr el resultado final y deseado se recrearon más de ocho ambientes de laboratorio, se hicieron pruebas con diferentes tipos de IOS de Cisco, diferentes tipos de enlaces de interconexión con el fin de poder observar de la mejor manera los resultados de QoS sobre un canal saturado.

Otro de los aspectos muy importantes y difíciles de dimensionar fue el funcionamiento del software GNS3. La gran cantidad de recursos de memoria y sobretodo de CPU consumida en la

puesta en marcha de los enrutadores hacen muy complicada la implementación de laboratorios a mayor escala. No solo necesitamos recursos disponibles para GNS3 sino que además estamos monitoreando el tráfico, generando reportes y gráficas, utilizamos software generador de tráfico, máquinas virtuales, software de transmisión de video en el equipo local y virtual, todo lo cual lleva nuestros equipos al límite de funcionamiento. Al principio, el laboratorio se implementó en un equipo con prestaciones promedio pero debido a los obstáculos descritos anteriormente y a la falta de disponibilidad de los equipos reales, se tuvo que realizar en una maquina tipo servidor de última generación con gran capacidad de procesamiento y memoria RAM para lograr capturar y analizar los resultados antes descritos en imágenes.

6. RECOMENDACIONES

Para el uso y más fácil observación de los resultados y del manejo de la encapsulación de los paquetes nivel a nivel de la capa de referencia OSI, se recomienda que el lector tenga conocimientos en Protocolos de enrutamiento, Enrutamiento IP y conceptos de MPLS.

Debido a la enorme cantidad de recursos de procesamiento y memoria que consume la puesta en marcha de un laboratorio con gran cantidad de routers en GNS3, se recomienda llevarlo a cabo en una maquina con las capacidades necesarias para tal fin: memoria RAM de 4 GB o superior y procesador de 2,8 GHz o superior. Todo esto debido a que con la ejecución de la topología el consumo de procesamiento se eleva al 100% muy fácilmente imposibilitando la correcta ejecución del laboratorio y la puesta en marcha de software de análisis necesarios para comprender lo que se está transmitiendo.

Si se tiene la posibilidad y la cantidad de routers con el IOS adecuado para ejecutar MPLS-DiffServ, se recomienda ejecutar estos laboratorios sobre los equipos reales.

A través del presente trabajo puede hacerse la recomendación de seguir adelante con el laboratorio incorporando más elementos o variando algunos otros y observar el comportamiento de la calidad de servicio en esos ambientes. Un ejemplo podría ser implementar la misma topología transmitiendo voz en vivo, tipo locución radial. También se podría inyectar diferente tipo de tráfico y aplicar calidad de servicio a más de un flujo. En caso de poseer conocimientos más sólidos en MPLS, el uso de la ingeniería de tráfico es un tema muy interesante para implementar.

BIBLIOGRAFIA

DiffServ: Servicios Diferenciados. Monografía de Evaluación de Performance en Redes de Telecomunicaciones. Adrián Delfino, Sebastián Rivero.

RFC 3031 Multiprotocol Label Switching Architecture

RFC 3443. Time to Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks.

RFC 3270 Multi-Protocol Label Switching (MPLS) Support of Differentiated Services

RFC 2475 An Architecture for Differentiated Services

RFC 2328, The Open Shortest Path First (OSPF) protocol.

Estados Unidos de América (USA). Cisco System Inc. Cisco IOS MPLS Configuration Guide. San José. 2008

Estados Unidos de America (USA). Cisco System Inc. Cisco IOS MPLS Quality Of Service White San José. 2001

Estados Unidos de América (USA). Cisco System Inc. Cisco IOS QoS Solutions Configuration Guide. San José. 2001

Estados Unidos de América (USA). Cisco System Inc. MPLS DiffServ Tunneling Modes. San José. 2008

Estados Unidos de América (USA). Cisco System Inc. Quality-of-Service, The Differentiated Services Model. San José. 2008

Estados Unidos de América (USA). Cisco System Inc. Multicast in a Campus Network: CGMP and IGMP Snooping. San José. 2008

ANEXO A

Running-config de Router 1

```
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R1
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
resource policy
!
ip subnet-zero
ip cef
!
!
!
!
ip multicast-routing
!
mpls label protocol ldp
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
interface Loopback1
 ip address 1.1.1.1 255.255.255.255
!
```

```
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 ip address 192.168.100.1 255.255.255.252
 ip pim dense-mode
 mpls ip
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/1
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/4
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/5
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/6
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/7
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
```

```
interface FastEthernet2/0
 ip address 10.10.1.254 255.255.255.0
 ip pim dense-mode
 duplex auto
 speed auto
 mpls ip
!
interface FastEthernet2/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
router ospf 1
 mpls traffic-eng router-id Loopback1
 mpls traffic-eng area 0
 log-adjacency-changes
 no auto-cost
 network 1.1.1.1 0.0.0.0 area 0
 network 10.10.1.0 0.0.0.255 area 0
 network 192.168.100.0 0.0.0.3 area 0
!
ip classless
no ip http server
no ip http secure-server
!
!
!
logging alarm informational
!
!
!
!
control-plane
!
!
!
!
!
!
gatekeeper
 shutdown
!
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
!
End
```

ANEXO B

Running-config R3

```
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R3
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
resource policy
!
ip subnet-zero
ip cef
!
!
!
!
ip multicast-routing
!
mpls label protocol ldp
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
interface Loopback3
 ip address 3.3.3.3 255.255.255.255
!
```

```
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 ip address 192.168.100.2 255.255.255.252
 ip pim dense-mode
 mpls ip
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/1
 ip address 192.168.100.5 255.255.255.252
 ip pim dense-mode
 mpls ip
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/2
 ip address 192.168.100.13 255.255.255.252
 ip pim dense-mode
 mpls ip
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/3
 ip address 192.168.100.9 255.255.255.252
 ip pim dense-mode
 mpls ip
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/4
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/5
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/6
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/7
 no ip address
 shutdown
```

```
 serial restart-delay 0
 no dce-terminal-timing-enable
!
router ospf 1
 mpls traffic-eng router-id Loopback3
 mpls traffic-eng area 0
 log-adjacency-changes
 no auto-cost
 network 3.3.3.3 0.0.0.0 area 0
 network 192.168.100.0 0.0.0.3 area 0
 network 192.168.100.4 0.0.0.3 area 0
 network 192.168.100.8 0.0.0.3 area 0
 network 192.168.100.12 0.0.0.3 area 0
!
ip classless
no ip http server
no ip http secure-server
!
!
!
logging alarm informational
!
!
!
!
control-plane
!
!
!
!
!
!
gatekeeper
 shutdown
!
!
line con 0
 exec-timeout 0 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
!
End
```

ANEXO C

Running-config R4

```
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R4
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
resource policy
!
ip subnet-zero
ip cef
!
!
!
!
ip multicast-routing
!
mpls label protocol ldp
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
interface Loopback4
```

```
 ip address 4.4.4.4 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 ip address 192.168.100.6 255.255.255.252
 ip pim dense-mode
 mpls ip
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/1
 ip address 192.168.100.17 255.255.255.252
 ip pim dense-mode
 mpls ip
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/4
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/5
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/6
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/7
 no ip address
 shutdown
```

```
 serial restart-delay 0
 no dce-terminal-timing-enable
!
router ospf 1
 mpls traffic-eng router-id Loopback4
 mpls traffic-eng area 0
 log-adjacency-changes
 no auto-cost
 network 4.4.4.4 0.0.0.0 area 0
 network 192.168.100.4 0.0.0.3 area 0
 network 192.168.100.16 0.0.0.3 area 0
!
ip classless
no ip http server
no ip http secure-server
!
!
!
logging alarm informational
!
!
!
!
control-plane
!
!
!
!
!
!
gatekeeper
 shutdown
!
!
line con 0
 exec-timeout 0 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
!
end
```

ANEXO D

Running-config R5

```
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R5
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
resource policy
!
ip subnet-zero
ip cef
!
!
!
!
!
mpls label protocol ldp
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
interface Loopback5
 ip address 5.5.5.5 255.255.255.255
```

```
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 ip address 192.168.100.14 255.255.255.252
 mpls ip
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/1
 ip address 192.168.100.21 255.255.255.252
 mpls ip
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/4
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/5
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/6
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/7
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
```

```
router ospf 1
 mpls traffic-eng router-id Loopback5
 mpls traffic-eng area 0
 log-adjacency-changes
 no auto-cost
 network 5.5.5.5 0.0.0.0 area 0
 network 192.168.100.12 0.0.0.3 area 0
 network 192.168.100.20 0.0.0.3 area 0
!
ip classless
no ip http server
no ip http secure-server
!
!
!
logging alarm informational
!
!
!
!
control-plane
!
!
!
!
!
!
gatekeeper
 shutdown
!
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
!
end
```

ANEXO E

Running-config R6

```
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R6
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
resource policy
!
ip subnet-zero
ip cef
!
!
!
!
ip multicast-routing
!
mpls label protocol ldp
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
interface Loopback6
 ip address 6.6.6.6 255.255.255.255
```

```
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 ip address 192.168.100.18 255.255.255.252
 ip pim dense-mode
 mpls ip
 no fair-queue
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/1
 ip address 192.168.100.22 255.255.255.252
 ip pim dense-mode
 mpls ip
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/2
 ip address 192.168.100.25 255.255.255.252
 ip pim dense-mode
 mpls ip
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/3
 ip address 192.168.100.10 255.255.255.252
 ip pim dense-mode
 mpls ip
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/4
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/5
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/6
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/7
```

```
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
router ospf 1
 mpls traffic-eng router-id Loopback6
 mpls traffic-eng area 0
 log-adjacency-changes
 no auto-cost
 network 6.6.6.6 0.0.0.0 area 0
 network 192.168.100.8 0.0.0.3 area 0
 network 192.168.100.16 0.0.0.3 area 0
 network 192.168.100.20 0.0.0.3 area 0
 network 192.168.100.24 0.0.0.3 area 0
!
ip classless
no ip http server
no ip http secure-server
!
!
!
logging alarm informational
!
!
!
!
control-plane
!
!
!
!
!
!
gatekeeper
 shutdown
!
!
line con 0
 exec-timeout 0 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
!
End
```

ANEXO F

Running-config R7

```
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R7
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
resource policy
!
ip subnet-zero
ip cef
!
!
!
!
ip multicast-routing
!
mpls label protocol ldp
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
interface Loopback7
 ip address 7.7.7.7 255.255.255.255
```

```
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 ip address 192.168.100.26 255.255.255.252
 ip pim dense-mode
 mpls ip
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/1
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/4
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/5
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/6
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
!
interface Serial1/7
 no ip address
 shutdown
 serial restart-delay 0
 no dce-terminal-timing-enable
```

```
!
interface GigabitEthernet2/0
 ip address 10.10.0.254 255.255.255.0
 ip pim dense-mode
 negotiation auto
 mpls ip
!
router ospf 1
 mpls traffic-eng router-id Loopback7
 mpls traffic-eng area 0
 log-adjacency-changes
 no auto-cost
 network 7.7.7.7 0.0.0.0 area 0
 network 10.10.0.0 0.0.0.255 area 0
 network 192.168.100.24 0.0.0.3 area 0
!
ip classless
no ip http server
no ip http secure-server
!
!
!
logging alarm informational
!
!
!
!
control-plane
!
!
!
!
!
!
gatekeeper
 shutdown
!
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
!
End
```