
# ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

# ESCUELA DE POSTGRADO Y EDUCACIÓN CONTINUA

**Maestría en Interconectividad de Redes**

**"Aplicación de la Arquitectura DIFFSERV sobre redes MPLS para la provisión de QoS punto a punto en la transmisión de tráfico en tiempo real"**

**Presentado por:** Pamela Alexandra Buñay Guisñan

**RIOBAMBA – ECUADOR**

**2013**

# DERECHOS DE AUTORÍA

Yo, Pamela Alexandra Buñay Guisñan, soy responsable de las ideas, doctrinas y resultados expuestos en esta Tesis; y el patrimonio intelectual de la Tesis de Grado pertenece a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO.

_____

Ing. Pamela Alexandra Buñay Guisñan

# CERTIFICACIÓN

El Tribunal de Tesis certifica que:

El trabajo de investigación titulado**: "Aplicación de la Arquitectura DIFFSERV sobre redes MPLS para la provisión de QoS punto a punto en la transmisión de tráfico en tiempo real",** de responsabilidad de la señorita Pamela Alexandra Buñay Guisñan ha sido prolijamente revisado y se autoriza su presentación.

Ing. Ms.C. Danilo Pastor R. _____

**DIRECTOR**

Ing. Ms.C. Gloria Arcos M. _____

**MIEMBRO**

Dra. Narcisa Salazar. _____

**MIEMBRO**

**Escuela Superior Politécnica de Chimborazo**

**Riobamba, Marzo 2013**

# ÍNDICE GENERAL

## INDICE DE TABLAS

# INDICE DE FIGURAS

# DEDICATORIA

Dedico este trabajo de tesis a Dios, pues sin él nada podemos hacer, gracias por otorgarme la sabiduría y salud para lograrlo. A mi familia, profesores y amigos quienes siempre me apoyaron incondicionalmente a seguir adelante en mi preparación profesional, haciéndome ver mis errores para mejor como persona y lograr así mis metas y sueños.

Dios les bendiga.

Pamela A. Buñay G.

# AGRADECIMIENTO

Agradezco a Dios, a la Virgen, a mi familia quienes siempre me han apoyado para seguir adelante, sin decaer ante los fracasos y adversidades que se presentan; siempre animándome a conseguir mis sueños, metas, objetivos con esfuerzo, trabajo y responsabilidad. Agradezco a mi director de tesis y miembros de tribunal que me guiaron para culminar esta importante investigación. Además agradezco a mis amigos quienes siempre me ayudaron cuando más lo necesitaba, brindándome su amistad, confianza para salir adelante.

Pamela A. Buñay G.

# RESUMEN

La presente investigación tiene como objetivo aplicar la Arquitectura DIFFSERV sobre redes MPLS para proveer una transmisión de tráfico punto a punto en los laboratorios de la Academia Local de Redes Cisco de Escuela Superior Politécnica de Chimborazo.

Los métodos Científico y Deductivo fueron empleados para obtener los aspectos más importantes de la aplicación de la Arquitectura DiffServ sobre redes MPLS. Los tipos de investigación Experimental y Descriptiva fueron utilizados para la manipulación de la variable independiente y analizar las consecuencias de esta sobre las variables dependientes. Se analizó las características del objeto de estudio. Las técnicas observación, recopilación de información y pruebas se aplicaron a un ambiente de prueba implementado en los laboratorios de la Academia Cisco. En cada escenario se evaluaron los indicadores de calidad de servicio: retardo, jitter y pérdida de paquetes.

Para el análisis se utilizó la herramienta D-IGT 2.61 GUI 0.92 (Distributed Internet Traffic Generator) para tráfico IPV4, IPV6. Esto generó tráfico a nivel de red, transporte y aplicación. Luego de la recolección y tabulación de datos se pudo comprobar que la aplicación de DiffServ sobre las redes MPLS, redujo el jitter un 19.29% para VoIP y 73.63% para datos, perdida de paquetes un 85.34% VoIP y 72.34% datos, retardo un 80.19% VoIP y 73.39% en datos, con relación a las redes IP.

Se concluye que la aplicación de la arquitectura ¨ Servicios Diferenciados ¨ sobre redes MPLS fue adecuada porque las mediciones realizadas en los parámetros jitter, pérdida de paquetes y retardo estuvieron dentro de los umbrales adecuados. Se creó una guía para la implementación de la arquitectura DiffServ sobre redes MPLS en la transmisión de tráfico en tiempo real. Se recomienda que para asegurar Calidad de Servicio punto a punto, implementar MPLS como backbone con un modelo de Servicios Diferenciados (DiffServ) por las garantías de calidad de servicio.

# ABSTRACT

This research aims to implement Architecture DIFFSERV on MPLS networks to provide traffic transmission point to point in the laboratories of the Cisco Networking Local Academy at the Polytechnic University of Chimborazo.

Scientific and deductive methods were employed to obtain the most important aspects of the implementation of the architecture DiffServ on MPLS networks. Experimental research and Descriptive were used for manipulation of the independent variable. The characteristic of the objet study also were analyzed. Observation techniques, evidence gathering and test environment on the laboratories of the Cisco Academy were applied. In each scenario the service quality indicators: delay, jitter and packet loss were evaluated.

For the analysis, the D-IGT tool GUI 2.61 0.92 (Distributed Internet Traffic Generator) for IPT4, IPV6 traffic was used. This generated traffic at the network, transport and application level. After collection and tabulation of data it was found that the implementation of DiffServ over MPLS networks, reduced jitter for VoIP one 19.29% and 73.63% for data, one packet loss 85.34% 72.34% VoIP and data delay one 80.19 % and 73.39% for VoIP data in relation to IP networks.

As conclusion, the architecture “Differentiated Services” on MPLS networks was appropriate because the measurements made in the parameters jitter, packet loss and delay thresholds were within the appropriate boundaries. A guide for implementing the DiffServ architecture for MPLS networks in the transmission of real-time traffic also was created.

It is recommended that to ensure quality of service point to point, the MPLS backbone implementation with a model of Differentiated Services (DiffServ) must be used to ensure the quality of service.

# CAPÍTULO I
# INTRODUCCIÓN

## 1.1 PLANTEAMIENTO DEL PROBLEMA

Las tendencias de hoy en día son caracterizadas por un fuerte crecimiento de diversos servicios y por buscar simplificar la operación y gestión de una red.

Actualmente el desarrollo de estas redes se ha enfocado hacia la de Calidad de Servicio (QoS). El objetivo es evitar que la congestión de determinados nodos de la red afecte a algunas aplicaciones que requieran un especial caudal o retardo, como son las aplicaciones en tiempo real de videoconferencia y voz sobre ip.

Las redes IP reparten paquetes según el servicio "best effort" (BE) "lo más posible, lo antes posible". En los paquetes con este tipo de servicio los routers del núcleo solo miran la cabecera y buscan en la tabla de ruteo para definir el siguiente salto. En este tipo de redes si llega a ocurrir congestión, los routers retardan o desechan los paquetes, siendo escalable para datos; pero no para otro tipo de aplicaciones como son las de tiempo real (telefonía, radio, televisión, video conferencia, etc.) que tienen otras exigencias por ejemplo no debe existir retardos variables o pérdidas de datos.

Los problemas muchas veces tienen que ver con la utilización de la red como por ejemplo velocidad, ancho de banda. Los factores que afectan a la calidad del servicio (QoS) son: la latencia, jitter, pérdida de paquetes. Muchas veces producidas por la información que no viaja siempre por el mismo camino y otras porque las comunicaciones son en tiempo real.

La presente investigación trata de solucionar el problema que existe en las redes donde la QoS debe manejar tráfico haciendo que el ancho de banda soporte los requerimientos de aplicaciones.

Una forma de mejorar sería dotar de más ancho de banda a la red, pero no es suficiente, porque que el tráfico es en ráfagas, produciendo congestiones temporales, retardos y pérdidas.

La solución para dotar a las redes IP de QoS sería la aplicación de la arquitectura de DiffServ (Servicios Diferenciados) para proporcionar QoS y la tecnología MPLS (Conmutación de etiquetas multiprotocolo) que sería útil para realizar Ingeniería de Tráfico.

La Arquitectura de Servicios Diferenciados proporciona QoS en redes IP de forma sencilla y escalable, la red clasifica el tráfico en distintas clases y les aplica una disciplina de servicio diferenciada con el objetivo de proporcionar distintos niveles de calidad de servicio.

La conmutación de etiquetas multiprotocolo proporciona las características de las redes orientadas a conexión a las redes no orientadas a conexión. Asigna a las tramas que circulan por la red una identificación que le indica a los routers la ruta que deben seguir los datos. Por lo tanto, MPLS sirve para la administración de la calidad de servicio.

## 1.2 JUSTIFICACIÓN

Con la llegada de aplicaciones como son: telefonía, radio, televisión, video conferencia, etc., ha motivado para distinguir los distintos niveles de servicio al tráfico de paquetes. Hoy en día la clave está en dotar a la red de una mayor inteligencia para obtener QoS diferenciada, dando a ciertos paquetes un mejor trato.

Se pretende integrar la tecnología MPLS y la arquitectura DiffServ, MPLS actuará en el nivel de enlace de red proporcionando un método de envío rápido por conmutación de etiquetas mientras que DiffServ permitirá diferenciar tráfico en el tratamiento del envío.

Mediante MPLS los datagramas de cada flujo tendrán una etiqueta única que permitirá una conmutación rápida en los routers intermedios (solo se mirará la etiqueta, no la dirección de destino). Está técnica brindará beneficios a las redes basadas en IP como: VPN, ingeniería de tráfico y calidad de servicio.

DiffServ clasifica los paquetes en categorías, se proporcionará mecanismos de calidad de servicio para reducir la carga en dispositivos de la red a través de un mapeo entre flujos de tráfico y niveles de servicio. La diferenciación de servicios se logrará mediante la definición de comportamientos específicos para cada clase de tráfico entre dispositivos de interconexión. De esta manera a través de DiffServ plantearemos asignar prioridades a los diferentes paquetes que son enviados a la red.

Se realizará los escenarios de prueba en los laboratorios de la Academia Local Cisco ESPOCH con equipos Cisco que cuenta la misma. Los escenarios serán los siguientes:

- Escenario con redes IP
- Escenario con redes MPLS
- Escenario con redes MPLS aplicando la arquitectura DiffServ.

Se medirá cada uno de los parámetros que determinan la calidad de Servicio (QoS) y se realizará las comparaciones con cada escenario.

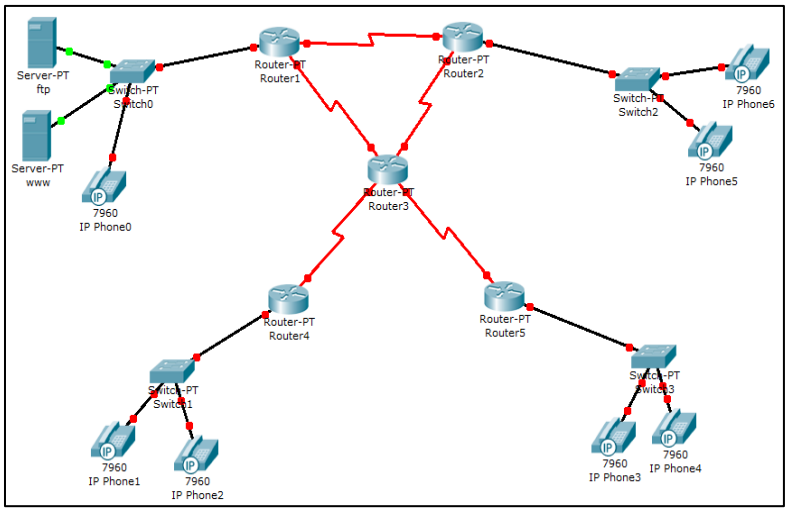



**Figura 1.1: Escenario de Prueba**

## 1.3 OBJETIVOS

### 1.3.1 General

Aplicar la Arquitectura DIFFSERV sobre redes MPLS para proveer de QoS punto a punto en la transmisión de tráfico en tiempo real.

### 1.3.2 Específicos

- Estudiar la arquitectura DiffServ sobre redes MPLS para entender su funcionamiento.
- Determinar los parámetros adecuados que influyen en la calidad de Servicio de tráfico en tiempo real.
- Establecer un ambiente de simulación para la provisión de la calidad de Servicio en redes MPLS en la transmisión de tráfico en tiempo real aplicando la arquitectura DiffServ.
- Definir una guía para la implementación de la arquitectura DiffServ sobre redes MPLS en la transmisión de tráfico en tiempo real.

### HIPÓTESIS

La aplicación de la arquitectura ¨ Servicios Diferenciados ¨ sobre redes MPLS proveerá una adecuada Calidad de Servicio punto a punto en la transmisión de tráfico en tiempo real.

# CAPÍTULO II
# REVISIÓN DE LITERATURA

## 2.1 Conceptos Básicos de Calidad de Servicio

### 2.1.1 Calidad de Servicio

La provisión de calidad de servicio (QoS) garantizada por parte de las redes de comunicación en un ámbito global es actualmente uno de los campos de investigación en activo, principalmente debido a la creciente importancia que cobra un conjunto de aplicaciones telemáticas, destacando entre ellas las dedicadas a la tele-educación, que precisan de esa garantía para su correcto funcionamiento. Una de las principales propuestas que se contemplan a la hora de definir marcos integrados de provisión de QoS propone la utilización de IP sobre ATM, de manera que se aprovechen tanto el control sobre los parámetros de QoS que proporciona ATM, como la gran expansión y conectividad de que goza IP.

### 2.1.2Conceptos de Calidad de Servicio

ITU E.800: “Efecto global de las prestaciones de un servicio que determinan el grado de satisfacción de un usuario al utilizar dicho servicio”

IETF RFC 2386: “Conjunto de requisitos servicio que la red en del debe cumplir el transporte de un flujo.”

### 2.1.3 ¿Qué es Calidad de Servicio?

La QoS puede ser definida como un conjunto de tecnologías que permiten a los administradores de red manejar los efectos de la congestión del tráfico usando óptimamente los diferentes recursos de la red, en lugar de ir aumentando continuamente capacidad. En este punto es necesario prestar una atención especial al hecho de que la QoS no es aumentar ancho de banda sino distribuirlo de acuerdo a las necesidades de la empresa [1].

QoS se define como un proceso de entrega fiable de datos que incluye importantes aspectos de red, tales como: pérdidas de datos, retardo mínimo o latencia, variación (jitter), distancia mínima entre dos puntos terminales y/o máxima eficiencia en el uso del ancho de banda [2].

La calidad de servicio (QoS) es un conjunto de estándares y mecanismos que aseguran la calidad de funcionamiento de un servicio que determina el grado de satisfacción de un usuario de un servicio especialmente para las aplicaciones críticas como voz y video en tiempo real, telefonía IP, videoconferencia, audio y video en demanda, ya que representan grandes ahorros al no tener que usar la red de telefonía pública

### 2.1.4 Parámetros de Calidad de Servicio

Los parámetros de QoS en red de transporte, tal y como se están estandarizando en IETF, se definen sobre un flujo IP:

| Parámetro | Unidades | Significado | Ejemplo |
|---|---|---|---|
| Ancho de Banda (bandwidth) | Kb/s | Indica el caudal máximo que se puede transmitir | 2 Mbps |
| Retardo (delay) o latencia (latency) | ms | El tiempo medio que tardan en llegar los paquetes | 80 mseg |
| Jitter | ms | La fluctuación que se puede producir en el Retardo | ± 20 mseg |
| Tasa de pérdidas (loss rate) | % | Proporción de paquetes perdidos respecto de los enviados | 0,1% |

**Figura 2.1: Parámetros de Calidad de Servicio**

### 2.1.5 Requerimientos de Calidad de Servicio de las aplicaciones

Cada aplicación precisa de unos parámetros de servicio particulares para que sea posible que se establezca con las garantías suficientes


**Figura 2.2:[3] Requerimientos de las aplicaciones**

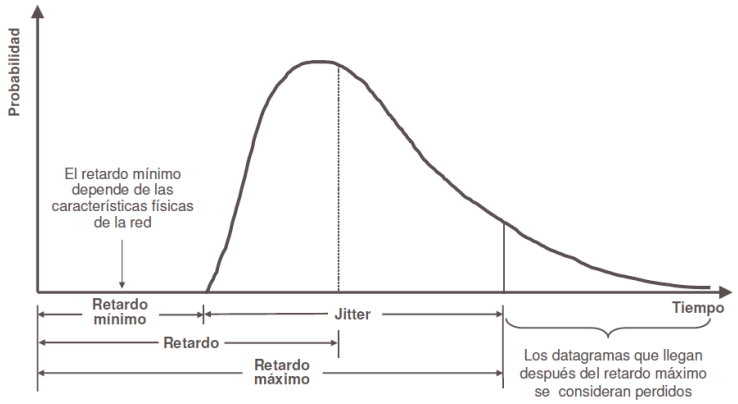



**Figura 2.3:[3] Relación entre la probabilidad de llegada de los datagramas y los parámetros de QoS**

| Aplicación | Fiabilidad | Retardo | Jitter | Ancho de Banda |
|---|---|---|---|---|
| Correo electrónico | Alta (*) | Alto | Alto | Bajo |
| Transferencia de ficheros | Alta (*) | Alto | Alto | Medio |
| Acceso Web | Alta (*) | Medio | Alto | Medio |
| Login remoto | Alta (*) | Medio | Medio | Bajo |
| Audio bajo demanda | Media | Alto | Medio | Medio |
| Vídeo bajo demanda | Media | Alto | Medio | Alto |
| Telefonía | Media | Bajo | Bajo | Bajo |
| Vídeoconferencia | Media | Bajo | Bajo | Alto |

**Figura 2.4: [3] Requerimientos de Calidad de Servicio de las aplicaciones**

**(*) La fiabilidad alta en estas aplicaciones se consigue automáticamente al utilizar el protocolo de transporte TCP**

### 2.1.6 Requisitos para una comunicación en tiempo real

- Bajo jitter
- Baja latencia
- Capacidad de adaptación dinámica a condiciones de tráfico y red cambiantes
- Buen rendimiento para grandes redes y gran cantidad de conexiones
- Requisitos modestos para los buffers dentro de una red
- Utilización de la capacidad de manera altamente efectiva  bits de cabecera paquete
- Baja redundancia de procesamiento por paquete dentro de la red y en el sistema final.

### 2.1.7 QoS en los routers

- Disciplina de colas para dar preferencia a los según la QoS establecida
- Selección de ruta según las características de QoS de cada posible ruta.

- Invocar tratamiento QoS en la subred del siguiente salto.

### 2.1.8 Modelos de Servicio

Los modelos de servicio describen la interfaz entre la red y sus usuarios en la arquitectura de asignación de recursos.

#### 2.1.8.1 Best effort(No QoS)

Best effort(No QoS)Se define como “el mejor esfuerzo no existe una garantía de calidad de servicio (QoS). No existe una preasignación de recursos, ni plazos conocidos, ni garantía de recepción correcta de la información.

**Limitaciones**

- Retardo extremo a extremo: Especialmente crítico en aplicaciones audio como por ejemplo videoconferencia, telefonía.
- Variación de retardo (jitter): Multimedia es generada a tasa constante y deben reproducirse de igual forma.
- Pérdidas: El tráfico multimedia es tolerante a pérdidas (tasas < 2% suelen pasar inadvertidas).

#### 2.1.8.2 Integrated services (Hard QoS):

Es una arquitectura propuesta para Internet, con el objetivo de dar QoS a flujos basados en: reserva de recursos y control de admisión.

Un flujo declara un patrón de tráfico (TSpec) y QoS deseada, las mismas que son enviadas a cada router mediante el protocolo RSVP. Los router verifica si disponen de recursos, reservándolos en caso afirmativo y rechazando la sesión en caso negativo

#### 2.1.8.3 Differentiated services (Soft QoS)

Esta arquitectura se basa en dividir el tráfico en clases, controlar la cantidad de tráfico que cada cliente envía a la red de cada clase de tráfico y asegurar requerimientos de QoS utilizando en cada enlace políticas de scheduling y dropping. En este modelo se establecen acuerdos con el cliente SLA (Service Level Agreements), en el cual entre otras cosas se le garantizan para ciertas clases de tráfico ciertas garantías de QoS siempre que el cliente envié el tráfico dentro de un cierto perfil (normalmente detenido por valores de media, pico y tamaño máximo de burst).

## 2.2 MPLS

### 2.2.1 Introducción

El crecimiento imparable de la Internet, así como la demanda sostenida de nuevos y más sofisticados servicios, supone cambios tecnológicos fundamentales respecto a las prácticas habituales desarrolladas a mitad de los años 90. Esto impulsó en un inicio a los proveedores de servicios de Telecomunicaciones a desplegar en sus infraestructuras una combinación de enrutadores IP con conmutadores ATM/Frame Relay, una vez consolidada la tecnología TCP/IP, esta combinación propiciaba un equilibrio frente a las necesidades de crecimiento de la época. Este modelo de red adoptado presentó limitaciones de interoperabilidad con otras redes, dificultad de gestionar estas conexiones y un alto crecimiento en equipamiento.

La Arquitectura MPLS (MultiProtocol Label Swiching)[4] es una nueva arquitectura que habilita a realizar Ingeniería de Tráfico en redes IP [5]. La tecnología MPLS (Multiprotocol Label Switching) también es conocida como tecnología de la capa 2.5, porque realiza un encapsulado intermedio entre la capa de enlace (capa 2) y la capa de red (capa 3). En este encapsulado se introduce una etiqueta de 4 bytes, que permite a los routers utilizar técnicas de conmutación. El utilizar el etiquetado por debajo de capa 3, permite que MPLS pueda funcionar independientemente del protocolo de capa 3 utilizado, de ahí lo de “multiprotocol”. Esta arquitectura de etiquetado es flexible y permite anidar etiquetas, es decir, introducir una trama MPLS dentro de otra.

El objetivo de MPLS es separar la parte de encaminamiento de la parte de conmutación en el reenvío de los paquetes, de forma que mientras la parte de encaminamiento es compleja y lenta (tiempos de convergencia, cálculo de rutas), se realiza independientemente de la parte de conmutación, que es rápida y simple.

La característica principal de MPLS que habilita a realizar ingeniería de tráfico es la de ruteo explícito. El ruteo explícito permite establecer caminos (Label Switchwd Path, LSP) predefinidos para los paquetes. Esto se realiza desde los routers de la frontera de la red. MPLS retoma en este sentido las bases sobre las que se diseñó ATM, al establecer “caminos virtuales” para los flujos agregados.

#### 2.2.2 Definición

MPLS es un estándar emergente del IETF [6] que surgió para consensuar diferentes soluciones de conmutación multinivel, propuestas por distintos fabricantes a mitad de los 90.

Definida en el RFC 3031[7], es una tecnología orientada a paquetes muy flexible. Si la situásemos en el modelo ISO/OSI (International Standard Organization / Open System Interconnection) se encontraría en la capa 2.5, entre la capa de enlace y de red, o sea, entre la capa 2 y 3. El hecho de

que se encuentre entre dos capas, le proporciona el nombre de “Multi Protocol”. Este hecho le da la ventaja de poder usar las característicasde los protocolos de las capas adyacentes sin ninguna restricción. Además de esto, MPLS ofrece adaptación total a IP. Esto es de gran importancia porque actualmente el mundo se mueve con este protocolo.

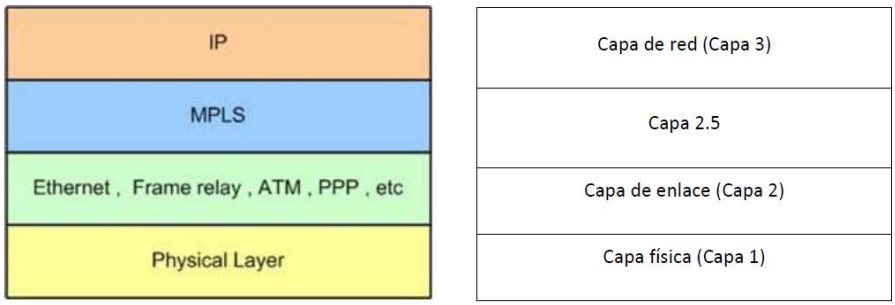



**Figura 2.5:Posición de MPLS en el modelo OSI**

### 2.2.3 Características

MPLS permite aplicar técnicas de Ingeniería de Tráfico para encontrar la mejor ruta no necesariamente la más corta en algunos casos, pero que garantiza la llegada de los flujos de tráfico evitando cuellos de botella y caída de los enlaces.

Las principales aplicaciones de MPLS son funciones de ingeniería de tráfico (a los flujos de cada usuario se les asocia una etiqueta diferente), routing basados en políticas (Policy Routing), servicios de VPN, servicios que requieren QoS , etc y según la aplicación, se necesitarán más o menos etiquetas, desde 1 hasta 3 o 4.

### 2.2.4 Importancia de MPLS

Multiprotocol Label Switching es una tecnología en telecomunicaciones que supone la mecanización de las redes de alto rendimiento utilizados en las telecomunicaciones a los datos directos dentro de los nodos de la red mediante el uso de los caminos más cortos en lugar de direcciones largas. Esto hace que sea posible para evadir sistemas complejos y las búsquedas que participan en la tabla de rutas.

Soporta cualquier tecnología a nivel físico y enlace sin necesidad de adoptar por completo una nueva infraestructura de red para cada servicio lo que representa para los operadores abaratamiento en costos de implementación y mantenimiento. Todo ello de manera más flexible y con menores costes de planificación y gestión para el administrador, y con mayor calidad de servicio para los clientes.

### 2.2.5 Componentes

Los elementos básicos son:

LER, Label Edge Router (Ruteador Etiquetador de Borde)
LSR, Label Switching Router (Ruteador de Conmutación de Etiquetas)
LSP, Label Switched Path (Ruta Conmutada de Etiquetas)
FEC, Forward Equivalence Class (Clase Equivalente de Envío)
LIB, Label Information Base (Base de Información de Etiquetas)
LDP, Label Distribution Protocol (Protocolo de Distribución de Etiquetas)

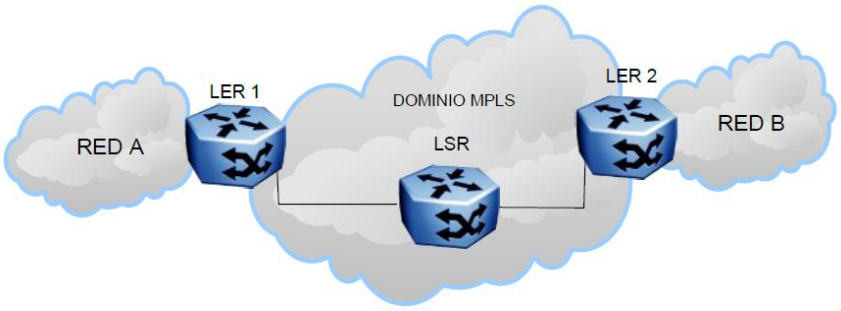



**Figura2.6:Red básica con MPLS**

Una red MPLS está constituida por dispositivos capa 3 que soportan MPLS y son los LER y los LSR, básicamente con las mismas características físicas, la diferencia radica en el modo de trabajo configurado por el administrador. Además al igual que los routers IP convencionales intercambian información sobre la topología de la red mediante los protocolos de enrutamiento como: OSPF, BGP, IS-IS entre otros y son capaces de manejar tablas de envío, estas últimas para la conmutación local de las etiquetas en el dominio MPLS.

**Label Edge Router (LER)**

Los LER son los routers frontera que operan en los bordes de una red MPLS. Desempeñan las funciones de encaminamiento tanto para un dominio MPLS como para un dominio no MPLS (otras redes). El LER analiza y clasifica el paquete IP entrante considerando hasta el nivel 3, es decir, considerando la dirección IP de destino y la QoS demandada, a esta clasificación por conjuntos de paquetes se le denomina FEC. Una vez analizado el paquete IP se añade una cabecera MPLS y en uno de sus campos denominado Etiqueta se le asigna un valor de acuerdo a su clasificación FEC.Es decir, el LER en vez de decidir el siguiente salto, como haría un router IP normal, decide el camino entero a lo largo de la red que el paquete debe seguir.

**Label switching router (LSR)**

Es un enrutador de alta velocidad especializado en el envío de paquetes etiquetados por MPLS.Los LSR están ubicados en el núcleo de la red MPLS para efectuar encaminamiento de alto

rendimiento basado en la conmutación por etiqueta, considerando únicamente hasta el nivel 2. Una vez que le llega un paquete a una de sus interfaces del LSR, éste lee la etiqueta de entrada en la cabecera MPLS y busca en la tabla de conmutación la etiqueta e interfaz de salida y reenvía el paquete por el camino predefinido escribiendo la nueva cabecera MPLS. Si un LSR detecta que debe enviar un paquete a un LER (según el FEC), extrae la cabecera MPLS; como el último LER no conmuta el paquete, se reducen así cabeceras innecesarias.

**FEC (Forwarding Equivalence Class)**

El FEC es la agrupación de etiquetas que permite la asociación de un conjunto de paquetes sobre el mismo camino y con un destino comúntravés de la red (LSP). Se determina una vez a la entrada a la red MPLS en un router LER.

Todos los paquetes de un mismo FEC se tratan de la misma forma hacia su destino, y cuantos más FECs tengamos, mayor granularidad para diferenciar entre distintos tipos de flujos. Aunque el hecho de tener más FECs nos afecta en la escalabilidad de la red, y por lo tanto, tendremos que llegar a un compromiso entre el número de FECs y la eficiencia de la red.

Para clasificar a los paquetes dentro de un mismo FEC se lo hace en base a criterios como:

- Dirección IP de origen, destino o direcciones IP de la red.
- Número de puerto de origen o destino
- Campo protocolo de IP (TCP, UDP, ICMP60, etc.)
- Valor del campo DSCP de DiffServ
- Etiqueta de flujo en IPv6

Cada FEC tiene un camino específico a seguir a través de la red MPLS y es independiente en cada router. Puede darse el caso que para una misma dirección IP haya más de un FEC a través del mismo LSP (Label Switched Path), lo que significa que paquetes con un mismo destino pueden pertenecer a FECs distintos si se tienen que tratar de forma distinta.

**Label Distribution Protocol (LDP)**

LDP es el protocolo más extendido para la distribución de etiquetas y comunicación de ellas a los LSRs. Está definido en el RFC 3036, funciona sobre TCP y usa las tablas de enrutamiento IP existentes creadas por el protocolo de enrutamiento, como OSPF, para propagarse. Define los mecanismos para la distribución de etiquetas, permite a los LSR descubrirse e intercambiar información sobre las asociaciones FEC/Etiqueta que se han realizado y sobre todo para mantener la coherencia de las etiquetas utilizadas para los distintos tipos de tráfico que conmutan. Evita que a un LSR le llegue tráfico con una etiqueta que no se encuentra en su tabla, asegurando la rapidez en la conmutación de los LSR.

La distribución de las etiquetas usa uno de los dos siguientes métodos:

**Unsolicited Downstream:** En este método, el LSR distribuye su información sobre las etiquetas cuando las tiene disponibles, aunque no se la hayan solicitado.

**Downstream on Demand:** Solo se envía información sobre las etiquetas cuando el vecino LSR pide información sobre ella.

**Label Switched Path (LSP)**

Es la ruta sobre una red MPLSsigue un grupo de paquetes que pertenecen al mismo FEC. La ruta se crea concatenando los saltos que dan los paquetes para el intercambio de etiquetas en los LSR utilizando mensajes LDP como son:

Descubrimiento: mediante mensajes “hello” de un LSR a otro LSR.

**Sesión:** dos LSR establecen y mantienen la comunicación.

**Anuncio:** para dar a conocer a otro LSR de las asociaciones FEC/Etiqueta.

**Notificación:** información de eventos y errores.

Existen dos métodos para el establecimiento de los LSPs:

**Ruta explícita:** A partir del primer LSR de salto se construye una lista de saltos específica utilizando los protocolos de señalización o de distribución de etiquetas

**Salto a Salto:** Cada LSR selecciona el próximo salto según el FEC que esté disponible.

**Label Information Base (LIB)**

Es la tabla de etiquetas que manejan los LSR. Relaciona la pareja (interfaz de entrada - etiqueta de entrada) con (interfaz de salida - etiqueta de salida).

La construcción de estas tablas se basa en las operaciones que realizan las etiquetas y son las siguientes:

**PUSH:** imposición de las etiquetas en un ruteador de ingreso LER.

**SWAP:** la etiqueta es cambiada por otra dentro del mismo rango que identifica un FEC en los LSRs.

**POP:** operación en la que se elimina la etiqueta en un LER al salir de la red MPLS.

**Forwarding Information Base (FIB)**

Es la tabla de rutas del router, pero con soporte hardware, basado en CEF. Esta tabla se actualiza automáticamente a petición de los protocolos de routing.

**Label Forwarging Information Base (LFIB)**

Es la tabla que asocia las etiquetas con los destinos o rutas de capa 3 y la interfaz de salida en el router, indicándole al router lo que tiene que hacer: poner o quitar etiqueta.

**PHP (Penultimate Hop Popping)**

Describe un método por el cual es posible retirar la etiqueta MPLS, es una alternativa de entrega de trama MPLS al final del circuito virtual. Consiste en quitar la etiqueta MPLS cuando se sabe que el siguiente router no necesita la etiqueta MPLS por estar la red directamente conectada a él o ser el final del circuito virtual. De esta forma, se evita hacer una doble búsqueda en dicho router, tanto en la tabla de LFIB y en la tabla de rutas. Es el modo de funcionamiento por defecto en los routers de Cisco Systems.

**Etiqueta**

La etiqueta MPLS es un identificador de 20 bits encapsulado dentro de la cabecera MPLS de 32 bits. Esta contiene la información necesaria para enrutar un paquete hasta su destino. Las etiquetas se utilizan en los routers para diferenciar entre los distintos FECs (Forward Equivalance Class) y por lo tanto determinar el siguiente salto donde el paquete debe ser enviado.

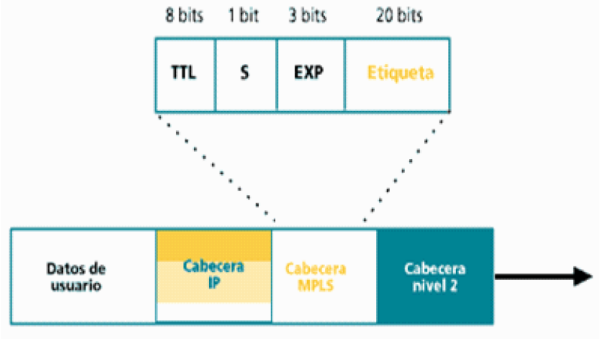



**Figura 2.7:Cabecera MPLS**

La cabecera MPLS se incluye entre las cabeceras de nivel 2 y 3 y contiene los siguientes campos:

**TTL (Time to Live):** es un valor que se decrementa cada vez que el paquete es reenviado por un router de la red MPLS (LSR). Cuando el valor es 0, el paquete se descarta. Su función es evitar que un paquete viaje indefinidamente por la red, provocando tráfico innecesario.

**S (Bottom of stack):** Si su valor es 1 indica que el paquete solo contiene una etiqueta. Si por el contrario vale 0, significa que el paquete posee una pila de etiquetas.

**EXP (Experimental):** Anteriormente era denominado CoS (Class of Service) pero ahora se considera un campo experimental. Se suele usar para proporcionar QoS.

**Etiqueta (label):** Valor local que usa el router para identificar un FEC en el proceso de forwarding, para determinar el próximo salto del paquete o su encapsulación.

### 2.5.6 Funcionamiento de MPLS

Los routers o switches que soportan MPLS trabajan en estos dos planos, específicamente los LER que es el  borde del dominio MPLS cumple dos funciones de encaminamiento y de envío inicial de

los paquetes asignando una cabecera MPLS mientras que los LSR son encargados de la conmutación de las etiquetas.

La operación del MPLS se basa en las componentes funcionales:

- De envío
- Control.

### 2.5.6.1 Funcionamiento del envío de paquetes en MPLS

El Plano de Envío MPLS utiliza la información de las etiquetas para la conmutación local de las mismas y para el envío de los paquetes a sus vecinos dentro del dominio, es decir se encarga de las asignaciones y modificaciones de etiquetas rigiéndose a la información proporcionada por el Plano de Control.

La base del MPLS está en la asignación e intercambio de etiquetas ya expuesto, que permiten el establecimiento de los caminos LSP por la red. Los LSPs son simplex por naturaleza (se establecen para un sentido del tráfico en cada punto de entrada a la red); para el tráfico dúplex requiere dos LSPs, uno en cada sentido. Cada LSP se crea a base de concatenar uno o más saltos (hops) en los que se intercambian las etiquetas, de modo que cada paquete se envía de un "conmutador de etiquetas" (Label-SwichingRouter) a otro, a través del dominio MPLS. Un LSR no es sino un router especializado en el envío de paquetes etiquetados por MPLS.

El paquete conforme avanza dentro de la red MPLS adquiere una nueva etiqueta, el valor de esta etiqueta define el FEC (Forward Equivalence Class) asignado.

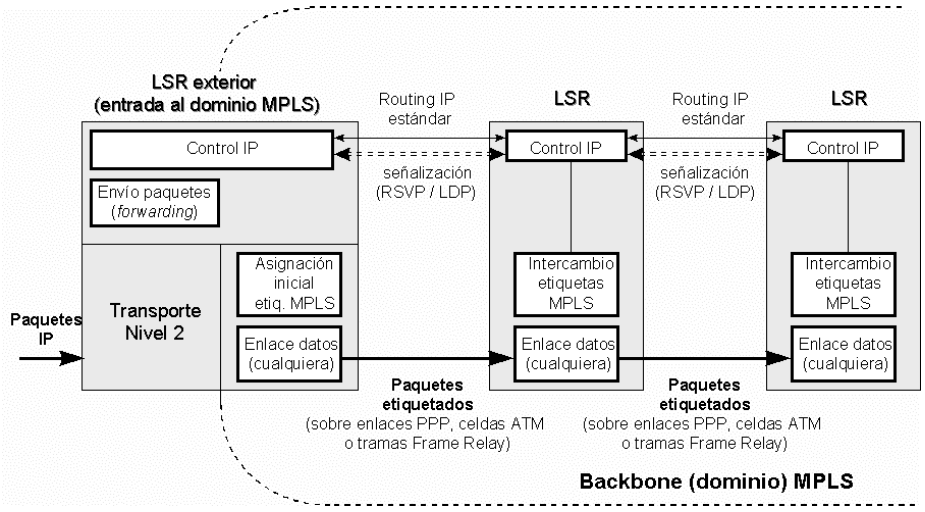



**Figura 2.8:Esquema funcional del MPLS.**

MPLS separa las dos componentes funcionales de control (routing) y de envío (forwarding). Del mismomodo, el envío se implementa mediante el intercambio de etiquetas en los LSPs. MPLS no utiliza ninguno de los protocolos de señalización ni de encaminamiento definidos por el ATM Forum, utiliza el protocolo RSVP o el estándar de señalización (el Label Distribution Protocol, LDP.

Un camino LSP es el circuito virtual que siguen por la red todos los paquetes asignados a la misma FEC. Al primer LSR que interviene en un LSP se le denomina de entrada o de cabecera y al último se le denomina de salida o de cola, los dos están en el exterior del dominio MPLS. El resto, entre ambos, son LSRs interiores del dominio MPLS, el LSR es como un router que funciona a base de intercambiar etiquetas según una tabla de envío. La tabla se construye a partir de la información de encaminamiento que proporciona la componente de control. Cada entrada de la tabla contiene un par de etiquetas entrada/salida correspondientes a cada interfaz de entrada, que se utilizan para acompañar a cada paquete que llega por esa interfaz y con la misma etiqueta (enlos LSR exteriores sólo hay una etiqueta, de salida en el de cabecera y de entrada en el de cola).

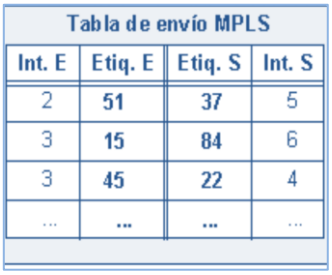


Tabla de envío MPLS

| Int. E | Etiq. E | Etiq. S | Int. S |
|---|---|---|---|
| 2 | 51 | 37 | 5 |
| 3 | 15 | 84 | 6 |
| 3 | 45 | 22 | 4 |
| ... | ... | ... | ... |

**Figura 2.9: Tabla de envío de un LSR.**

A un paquete que llega al LSR por la interfaz 3 de entrada con la etiqueta 45 el LSR le asigna la etiqueta 22 y lo envía por el interfaz 4 de salida al siguiente LSR, de acuerdo con la información de la tabla.

El LSR de entrada recibe un paquete normal (sin etiquetar) cuyadirección de destino es 212.95.193.1. El LSR consulta la tabla de encaminamiento y asigna el paquete a la clase FEC definida por el grupo 212.95/16. Asimismo, este LSR le asigna una etiqueta este caso como ejemplo 5 y envía el paquete al siguiente LSR del LSP. Dentro del dominio MPLS los LSR ignoran la cabecera IP; solamente analizan la etiqueta de entrada, consultan la tabla correspondiente (tabla de conmutación de etiquetas) y la reemplazan por otra nueva, de acuerdo con el algoritmo de intercambio de etiquetas. Al llegar el paquete al LSR de cola, ve que el siguiente salto lo saca de la red MPLS; al consultar ahora la tabla de conmutación de etiquetas quita ésta y envía el paquete por routing convencional.

La identidad del paquete original IP queda enmascarada durante el transporte por la red MPLS, tomando en cuenta las etiquetas que necesita para su envío por los diferentes saltos LSR que configuran los caminos LSP. Las etiquetas se insertan en cabeceras MPLS, entre los niveles 2 y 3.

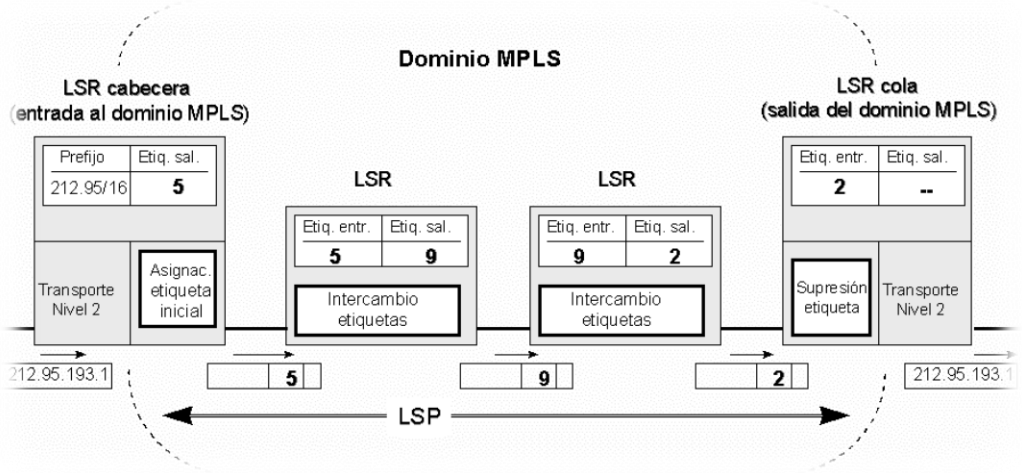



**Figura 2.10: Envío de un paquete por un LSP.**

### 2.5.6.2 Funcionamiento del plano de control

Para el intercambio de información dentro de una red MPLS se usan los protocolos de enrutamiento ya sean de vector distancia o estado de enlace, permitiendo la construcción y mantenimiento de las tablas de enrutamiento que proporcionan las características de la topología, patrón de tráfico o detalles de los enlaces. Manteniendo así la coherencia entre los LER y LSR evitando que a un determinado LSR le llegue un paquete con una etiqueta para el cual no tiene entrada en su tabla.

La difusión de las tablas de enrutamiento a los vecinos es muy importante porque establece los caminos virtuales LSP que los LER indican al inicio para la generación de las tablas de envío utilizando también la señalización que proveen los Protocolos de Distribución de Etiquetas (RSVP, LDP o TDP) y posteriormente el intercambio de etiquetas (Plano de Envío). Al tener la tabla de encaminamiento actualizada se escoge la dirección del próximo salto permitiendo el cálculo de las mejores rutas dentro de la red MPLS y caminos emergentes en caso de fallos.

### 2.5.6.3 Funcionamiento global MPLS

El funcionamiento de MPLS en 5 pasos básicos que son los siguientes:

- Se construyen las tablas de encaminamiento que son proporcionadas como información de los protocolos de enrutamiento interno, tras la difusión de estas tablas de enrutamiento se crean los caminos virtuales LSP que los LER indican.
- Con la utilización de los Protocolos de Distribución de Etiquetas se proporciona la información de las tablas de envío para el intercambio de etiquetas de los LSR.

- Una vez informados los LER/LSR de las tablas de enrutamiento y envío, tras la llegada de un paquete a un LER de entrada, éste le asigna una cabecera MPLS con una etiqueta de acuerdo a su FEC y lo envía dentro del dominio MPLS.
- Una vez dentro del dominio MPLS los LSRs se encargan del intercambio de etiquetas haciendo uso de las tablas de envío, relacionando la etiqueta de entrada y la etiqueta de salida.
- Finalmente al llegar el paquete al LER de salida se elimina la cabecera MPLS del paquete; si su campo S es igual a 1, analiza la etiqueta para saber de qué tipo de red procede y se lo envía por enrutamiento fuera del dominio a su destino final.

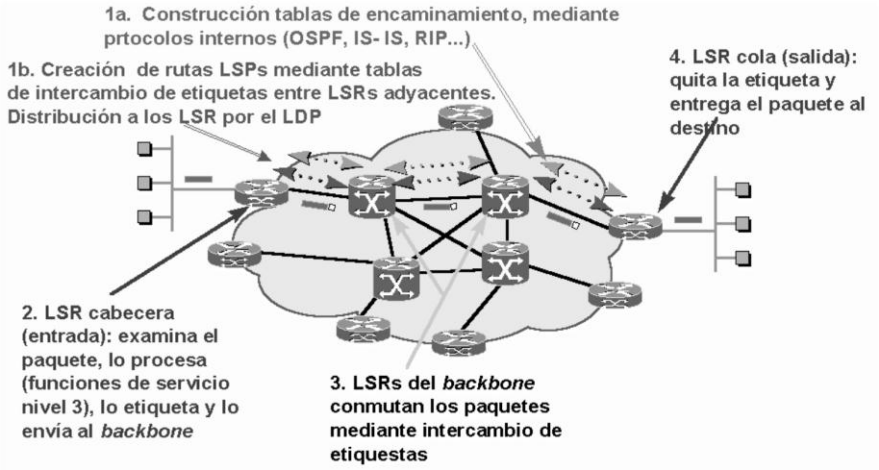



**Figura 2.11:Funcionamiento de una red MPLS.**

## 2.7 Aplicación de MPLS

Las principales aplicaciones que hoy en día tiene MPLS son:

- Ingeniería de tráfico.
- Diferenciación de niveles de servicio mediante clases (CoS).
- Servicio de redes privadas virtuales (VPN).

### 2.7.1 Ingeniería de tráfico

Es una facilidad que ofrece MPLS para adaptar los flujos de tráfico a los recursos físicos de la red, equilibrando de forma óptima la utilización, de manera que no haya algunos que estén supra-utilizados, con posibles puntos calientes y cuellos de botella, mientras otros puedan estar infrautilizados.

La ingeniería de tráfico consiste en trasladar determinados flujos seleccionados por el algoritmo IGP sobre enlaces más congestionados, a otros enlaces más descargados, aunque estén fuera de la ruta más corta (con menos saltos), teniendo el administrador de la red la posibilidad de:

- Establecer rutas explícitas especificando el camino LSP exacto (cobre, fibra óptica, etc.)

- Rutas restringidas para el caso de servicios especiales.
- Calcular la ruta más eficiente en base a los requerimientos y restricciones.
- Obtener informes estadísticos sobre el tráfico que cursa constituyendo una herramienta eficaz para el análisis de la distribución de los recursos de la red y para una planificación futura.

### 2.7.2 Calidad de servicio

La Calidad de Servicio permite controlar algunas de las características que influyen en la transmisión de paquetes como el ancho de banda, latencia, jitter, las pérdidas de los paquetes en la red, retardos, entre otras, garantizando la disponibilidad del servicio.

MPLS está diseñado para poder cursar servicios diferenciados, según el Modelo DiffServ del IETF. Este modelo define una variedad de mecanismos para poder clasificar el tráfico en un reducido número de clases de servicio, con diferentes prioridades.

La etiqueta MPLS tiene el campo EXP para poder propagar la clase de servicio CoS en el correspondiente LSP, logrando transportar distintos tipos de tráfico.

· El tráfico que fluye a través de un determinado LSP se puede asignar a diferentes colas de salida en los diferentes saltos LSR, de acuerdo con la información contenida en los bits del campo EXP.

· Entre cada par de LSR exteriores se pueden provisionar múltiples LSPs, cada uno de ellos con distintas prestaciones y con diferentes garantías de ancho de banda. P. ej., un LSP puede ser para tráfico de máxima prioridad, otro para una prioridad media y un tercero para tráfico best-effort, tres niveles de servicio primero, preferente y turista, que, lógicamente, tendrán distintos precios.

### 2.7.3 Redes Privadas Virtuales (VPN)

Una VPN es un conjunto de sitios que comparten información de ruteo que utiliza una infraestructura de Telecomunicaciones pública y conecta a usuarios de forma remota hacia una red principal, siendo una solución ideal para las empresas, y su objetivo es brindar aplicaciones Intranet y Extranet integrando soluciones multimedia.

El problema que plantea IP VPNs es que son un modelo topológico superpuesto sobre la topología física existente, basados en túneles extremos a extremo (o circuitos virtuales) entre cada par de routers de cliente en cada VPN. Presentando desventajas en cuanto a la poca flexibilidad en la provisión y gestión del servicio, así como en el crecimiento cuando se quieren añadir nuevos emplazamientos. Con MPLS se obvian estos inconvenientes ya que el modelo topológico no se superpone sino que se acopla a la red del proveedor. En el modelo acoplado MPLS, en lugar de conexiones extremo a extremo entre los distintos emplazamientos de una VPN, lo que hay son conexiones IP a una "nube común" en las que solamente pueden entrar los miembros de la misma VPN. Las "nubes" representan las distintas VPNs se implementan mediante los caminos LSPs creados por el mecanismo de intercambio de etiquetas MPLS.

Ventajas que MPLS ofrece para IP VPNs son:

- Proporcionan un modelo en la que la red MPLS "sabe" de la existencia de VPNs.
- Evita la complejidad de los túneles y PVCs.
- Una nueva conexión afecta a un solo router tiene mayores opciones de crecimiento modular.
- Se mantiene garantías QoS extremo a extremo, pudiendo separar flujos de tráfico por aplicaciones en diferentes clases, gracias al vínculo que mantienen el campo EXP de las etiquetas MPLS con las clases definidas a la entrada.

## 2.3SERVICIOS DIFERENCIADOS (DIFFSERV)

### 2.3.1 Introducción

Durante los últimos años, las aplicaciones en los extremos de la red requieren de una mejor calidad de servicio surgiendo varios mecanismos para ofrecer redes de Calidad de Servicio (QoS). Los mecanismos de QoS proporcionan un conjunto de herramientas que el administrador de redes puede utilizar para administrar el uso de recursos de red de una forma controlada y eficaz, ofreciendo disponibilidad de la red y eficiencia en la transmisión, que ayuda a mejorar el servicio a los usuarios de la red, al mismo tiempo reduciendo los costos de ofrecer dichos servicios.

La arquitectura DiffServ satisface requisitos como proporcionar altas prestaciones, escalabilidad, permitir el crecimiento sostenido del tamaño de las redes y su ancho de banda, etc. Se basa en situar el procesado complejo y la gestión de los recursos en los límites de la red, al mismo tiempo que mantiene el reenvío de paquetes en el núcleo de la red de la manera más sencilla posible. En los nodos del núcleo de la red no se mantiene el estado de las conexiones, sino que el tratamiento se basa únicamente en los códigos DS de los paquetes, que designan la clase de calidad que deben recibir.

### 2.3.2 Definición

La arquitectura de DiffServ es definida por el IETF (Internet Engineering Task Force) con el fin de ofrecer distintos servicios y aplicaciones a los usuarios en concordancia a sus necesidades [8].Permite el dividir y el dar prioridad al tráfico de la red mediante el uso de etiquetas en las cabeceras de los paquetes. Es un protocolo de QoS propuesto por IETF [RFC 2475 y RFC 2474] que permite distinguir diferentes clases de servicio marcando los paquetes. Permitiendo a los proveedores de servicios Internet y a usuarios de grandes redes IP corporativas desplegar rápidamente diferentes niveles QoS en la troncal.

La idea principal del modelo DiffServ (Differentiated Services) [9] consiste en clasificar el tráfico entrante en diferentes niveles de servicios, para enseguida aplicarle un comportamiento agregado para todos los flujos de una determinada clase de servicio. Cada comportamiento es identificado

por un único campo de servicio diferenciado (DS field). Ese comportamiento, en general, se denomina PHB (Per Hop Behavior) o comportamiento por salto. En el núcleo de la red, los paquetes son encaminados según el PHB asociado con su DS field.
En [10] se define el formato y el uso del DS field en la cabecera IP con el objetivo de suplantar el campo TOS (8 bits) [11] de la cabecera IPv4 y el octeto de clase de tráfico IPv6 [12]. Actualmente seis de los ocho bits del DS field son usados como codepoint (DSCP) para seleccionar el Per Hop Behavior que decidirá que comportamiento debe ser aplicado al tráfico cuando encaminado a través del núcleo de la red [13].

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
- 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.

### 2.3.4 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 provisionamiento 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.

**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 flujo 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 flujos de tráfico seleccionados por el clasificador.

### 2.3.5 Arquitectura de Servicios Diferenciados

La Arquitectura de Servicios Diferenciados (Diffserv) está basado en un modelo simple de tratamiento del tráfico, utilizado para grandes redes enrutadas. La sofisticada clasificación, marcado de los paquetes, política y operaciones son implementadas en los bordes de la red o en los hosts. El marcado de paquetes se realiza mediante la asignación de un código específico

(DSCP – Diffserv CodePoint), que es todo lo que se necesita para identificar a cada clase de tráfico.

Se 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 realizarlos servicios.
- El valor del campo DS, usado para marcar paquetes para seleccionar el comportamiento en cada salto.
- Los mecanismos de implementación particulares del nodo que realizan un comportamiento por salto.

**Separación del control y envío**

En el envío IP, la conectividad es lograda por la interacción de dos componentes: la parte del envío del paquete y la parte del ruteo. El envío usa la cabecera del paquete para encontrar una tabla de ruteo que determine la interfaz de salida del paquete. El ruteo setea las entradas en esa tabla y puede necesitar reflejar un rango de tránsito y otras políticas así como también el mantener registro de las fallas de ruta.

**Primitivas del camino de envío**

**La clasificación**

Saca información de los paquetes que ingresan al dominio. La política se encarga de asegurar que el comportamiento cumpla con las reglas que gobiernan la información de los paquetes. El marcado propaga información sobre el agregado de corriente abajo. PHB’s de envío son generalmente implementados por colas de paquetes. Y el encolamiento aísla una corriente de tráfico de otra.

**Marcado de DiffServ**

Cada paquete IP lleva un byte llamado octeto de Tipo de Servicio (TOS octet). Es una característica poco utilizada de IP. En la nueva versión 6 de IP de 128 bits, hay un byte equivalente llamado octeto de Clase de Servicio.

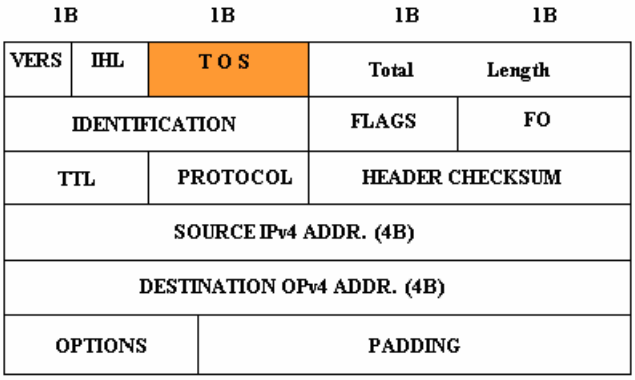



**Figura 2.12: Campo DS - campo TOS de IPv4**

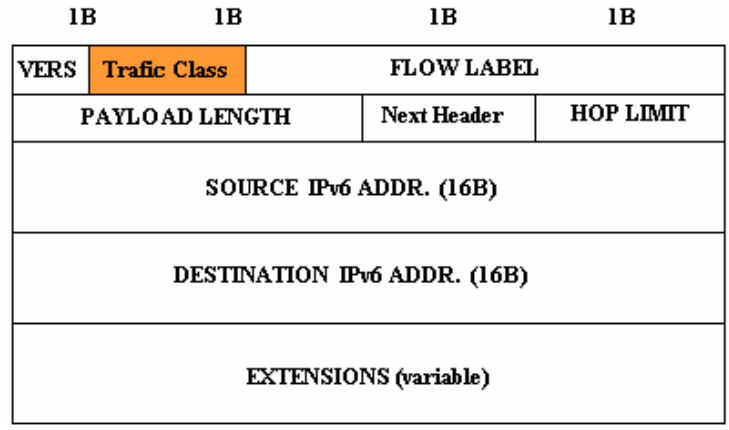



**Figura 2.13: Campo DS – Campo de Clase de Tráfico de IPv6.**

La primera tarea del grupo de DiffServ fue reespecificar este byte. Este campo de 6 bits es conocido como el campo de los Servicios Diferenciados y es marcado con un patrón específico de bits llamado código DS, usado para indicar cómo cada router debe tratar al paquete. Para enfatizar el hecho de que ninguna información de sesión se necesita guardar, este tratamiento es conocido como Per-Hop Behavior (PHB). El octeto luce así:

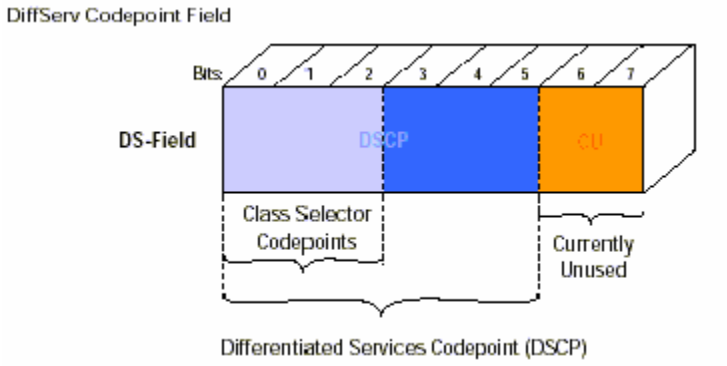



**Figura 2.14:Octeto TOS**

El campo de 6 bits contiene hasta 64 diferentes valores binarios. Los códigos extra restantes dejan espacio para innovación y optimizaciones operacionales locales. El marcado puede ocurrir en dos lugares:

• La fuente original de tráfico, como ser un servidor web, marca el tráfico. Esto tiene la ventaja de que el clasificador puede tener conocimiento explícito de la aplicación en uso y puede por consecuencia marcar paquetes de una manera dependiente de la aplicación.

• Un router, como el primer router que el tráfico encuentra, clasifica y marca el tráfico. Esto tiene la ventaja de que no se necesita ningún cambio a servidores, pero requiere de alguna “inteligencia” extra en los routers.

**Usando la marca**

Cuando un paquete entra en un router, la lógica de ruteo selecciona su puerto de salida y el valor DSCPes usado para conducir el paquete a una cola específica o tratamiento específico en ése puerto. El PHB particular es configurado por un mecanismo administrador de red, seteando la tabla de comportamiento de QoS dentro del router.

**Políticas de control**

**Clasificación y condicionamiento de tráfico**

El SLA puede especificar la clasificación del tráfico y las reglas de re-marcado, así como perfiles de tráfico y acciones a las corrientes de tráfico que son dentro o fuera del perfil (in-out profile). El TCA entre dominios deriva del SLA.

La política de clasificación de paquetes identifica el subconjunto de tráfico que puede llegar a recibir un servicio diferenciado, al ser condicionado y/o mapeado a uno o más BA. El condicionamiento de tráfico reforma la medición, conformación, política y/o remarcado para asegurarse que el tráfico entrante al dominio DS respeta las reglas especificadas en el TCA.

El clasificador de tráfico (classifier trafficc) identifica qué flujo de tráfico llega en el dominio DS basándose en la información del DS field de la cabecera IP. Existen dos tipos de clasificadores: (a) comportamiento agregado (BA - Behavior Aggregate), que clasifica los paquetes por el DS codepoint, y (b) MF (Multi-Field) que selecciona los paquetes por la combinación de uno o más campos de la cabecera IP (dirección origen o destino, DS field, protocolo ID, número de puertos). El clasificador asigna una función de vigilancia que mide la tasa a la que llegan los paquetes para cada flujo (metering).

Un marcador de paquete asociado a la función de vigilancia escribe en la cabecera IP de los paquetes un código que identificará el tipo de servicio que debe ser aplicado en función de un SLA contratado (marking). El tipo de servicio será identificado por el código PHB, que determina un perfil de tráfico soportado por el dominio (shaping). El perfil de tráfico dará un tratamiento prioritario de unos paquetes frente a otros, lo que supone descarte de algunos paquetes en caso de congestión (policing).

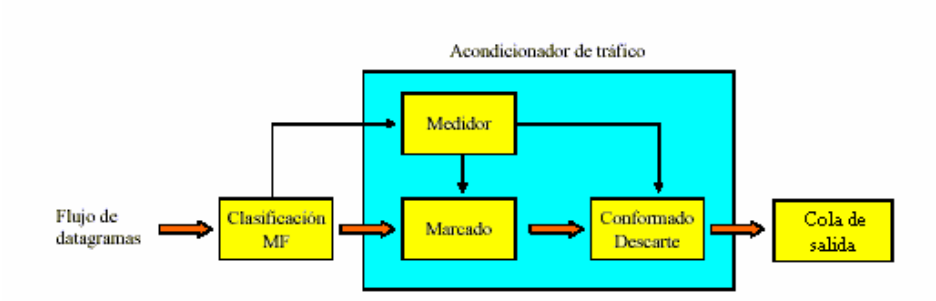



**Figura 2.15:[23] Octeto TOS**

**Perfil de tráfico**

Especifica las propiedades temporales de una corriente de tráfico seleccionada por el clasificador. Provee de reglas para determinar si un paquete está dentro o fuera del perfil. Por ejemplo, un perfil basado en una cubeta con fichas puede parecer así:

codepoint=X, use token-bucket r, b

El perfil anterior indica que todos los paquetes marcados con un código DS X deben ser medidos con medidor de balde con fichas con tasa r y de tamaño b. En este caso los paquetes fuera del perfil son los que arriban cuando hay insuficientes fichas disponibles en el balde.

Diferentes acciones de condicionamiento pueden ser aplicadas a los paquetes dentro y fuera del perfil.

Los paquetes dentro del perfil pueden ser mandados sin ningún otro procesamiento o marcado o remarcado. Los paquetes fuera de perfil pueden ser encolados hasta que estén dentro del perfil (conformados), desechados (política) o remarcados con un código nuevo (re-marcado).

Hay que hacer notar que el perfil de tráfico es un componente opcional de un TCA.

**Acondicionador de tráfico**

Puede contener los siguientes elementos: medidor, marcador, conformador y despachador. Una corriente de tráfico es seleccionada por un clasificador. Un medidor es usado para medir la corriente de tráfico en base a un perfil de tráfico. El estado del medidor respecto a un paquete en particular puede ser usado para afectar el marcado, despacho, o acción de conformación.

Cuando los paquetes salen del acondicionador de tráfico de un nodo DS frontera, el código DS de cada paquete debe setearse a un valor apropiado.

**Componentes del acondicionador**

**Medidor:** miden las propiedades temporales de la corriente de paquetes seleccionada por el clasificador en base a un perfil de tráfico especificado en el TCA. Pasa información de estado a otras funciones de condicionamiento para tomar cierta acción para cada paquete tanto dentro como fuera de perfil.

Marcador: setean el campo DS con un código particular, agregando el paquete marcado a un behavior agregate DS particular. Puede que marque todos los paquetes que son dirigidos a él con un código particular o puede estar configurado para marcar un paquete a un código de un grupo de códigos usados para seleccionar un PHB en un grupo PHB. Cuando el marcador cambia el código en un paquete, se dice haber “remarcado” el paquete.

**Conformador**: Retardan uno o todos los paquetes de una corriente de tráfico de manera de que la corriente cumpla con el perfil de tráfico estipulado. Usualmente tiene un buffer de tamaño finito y los paquetes pueden ser descartados si no hay suficiente espacio de buffer para aguantar a los paquetes retrasados.

**Despachador:** Descartan algunos o todos los paquetes en una corriente de tráfico de manera de que la corriente cumpla con el perfil de tráfico estipulado. Este proceso es conocido como “política”. Un despachador puede ser implementado como un caso especial de un conformador, si ponemos el tamaño del buffer del conformador igual a 0 (o muy pocos) paquetes.

**Adaptación:** Proceso de demorar paquetes para que sean conformes con un perfil de tráfico.

**Política:** Proceso de descarte de paquetes de acuerdo con el estado de un medidor para forzar un perfil de tráfico.

### 2.3.6 Modelo arquitectónico de los Ser vicios Diferenciados

El dominio DiffServ está formado por un conjunto de nodos DS que operan bajo una política de servicio común suministrada por un ISP o un operador de Telefonía IP y un conjunto de grupos PHB implementado en cada nodo, que definen como los paquetes, con el mismo DS field, serán

tratados. El administrador es el responsable por el Acuerdo de Nivel de Servicio (SLAs) firmado entre el usuario y el proveedor del dominio DS.

Básicamente un per-hop behavior PHB es una descripción, con un nivel relativamente alto de abstracción, el comportamiento observable externamente de un conjunto de paquetes denominados Behavior Aggregate (BA), y debe permitir la construcción de servicios predecibles. Los BA son conjuntos de paquetes marcados con un mismo DSCP y enviados en la misma dirección, pudiendo pertenecer a un mismo agregado de paquetes procedentes de múltiples fuentes o aplicaciones. Técnicamente hablando, un PHB denota una combinación de comportamientos de reenvío, clasificación, planificación y descarte en cada salto de paquetes pertenecientes a un mismo BA.

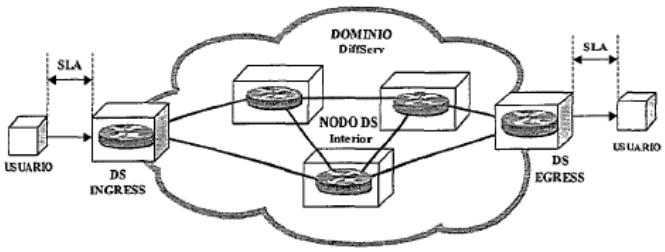



**Figura 2.16: Dominio DiffServ**

El dominio DS está formado por los nodos de borde (exteriores) y los nodos interiores. Los nodos de borde son los responsables de clasificar (Classifier) y acondicionar el tráfico (Traffic Conditioning) de entrada, además, realizan funciones de control de admisión, vigilancia y contabilidad de tráfico. Los nodos interiores, que están localizados en el núcleo, también forman parte del dominio DS y sirven para conectar el interior con la periferia, su función es encaminar el flujo de tráfico según niveles de prioridad. Un dominio DS puede tener una o más redes bajo un mismo administrador, que es el máximo responsable en asegurar que los recursos disponibles son reservados para soportar los SLAs ofrecidos. Los nodos de borde, localizados en los límites del dominio, actúan como nodos de entrada (DS ingress) y salida (DS egress) para diferentes tipos de tráfico. Los nodos de entrada son los responsables de garantizar que el tráfico entrante se comporta conforme a un determinado TCA (Traffic Conditioning Agreement) acordado entre dominios. Mientras que, los nodos de salida pueden realizar funciones del tipo traffic conditioning en el tráfico encaminado hacia otros dominios. El acondicionamiento de tráfico realiza funciones del tipo: vigilar, marcar, modelar y controlar el tráfico según las reglas del TCA de manera particular.

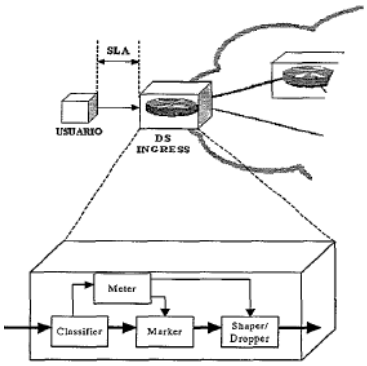



**Figura 2.17: Clasificador y acondicionador de tráfico DiffServ**

Los nodos interiores encaminan el tráfico conforme el DS codepoint, su trabajo es relacionarel DSCP con un PHB soportado por el dominio. En casos de comunicación entre dominios esnecesario identificar un particular PHB a través de un mensaje de negociación de ancho debanda o selección de camino, a codificación de ese mensaje de protocolo está definida en[15]. El PHB [16] está dividido en tres clases de servicio: Expedited Forwarding (EF) [17], Assured Forwarding (AF) [18] y Best-Effort (BE), representadas en Internet por los servicios Premium, Assured y Best-Effort, respectivamente.

Expedited Forwarding (EF) (DSCP=101110) proporciona el mayor nivel de QoS al tráfico agregado. Se utiliza para implementar un servicio extremo a extremo de bajas pérdidas, baja latencia, y baja variación de retardo, a través de dominios DS. Aparece a los extremos de la comunicación como una conexión punto a punto, línea virtual dedicada, o reserva dinámica (RSVP); de modo que cualquier tráfico que exceda el perfil definido es descartado. Se le denomina también Servicio Premium (sólo un nivel de calidad).

Los usuarios que contratan el servicio Premium tienen un SLA contratado con su proveedor o ISP, en este caso, existe un compromiso por parte del usuario de no exceder ellímite de tráfico contratado, caso contrario el mismo será descartado. Debido a suscaracterísticas y garantías de calidad, este servicio es el más indicado para transportar tráfico con rígidos requisitos de tiempo real, como el de Telefonía IP.

Por ejemplo, se utiliza para servicios de tiempo real con un throughput determinado: videoconferencia, voz sobre IP. [22].

El comportamiento EF puede implementarse por diversos mecanismos de colas:

• **Cola con prioridad simple**, siempre que una cola de mayor prioridad no desaloje el tráfico EF por un tiempo mayor que la transmisión de un paquete, a la velocidad configurada (controlable por policía de tráfico tipo cubo goteante)

• **Cola en un grupo de colas servidas por un planificador WRR** (Weighted Round Robin), en el que el porcentaje de ancho de banda de salida asignado al tráfico EF sea igual a la velocidad configurada.
• **Planificador CBQ (Class Based Queue**), que dé la prioridad al tráfico EF, hasta la velocidad configurada.

El Servicio Asegurado (Assured Service) AF define 4 cuatro clases de tráfico: AF1, AF2, AF3, y AF4. Cada clase se asigna a una cantidad específica del espacio del buffer y ancho de banda de la interfaz, de acuerdo con la política establecida. Garantiza una mayor fiabilidad y seguridad para los paquetes con alta prioridad frente a los de baja prioridad de manera predecible. En [18] las clases AF son definidas con diferentes niveles de prioridad, según los recursos disponibles en nodo DiffServ, tales como: espacio en el buffer y ancho de banda.
Los usuarios que contratan el servicio AF tienen un Acuerdo de Nivel de Servicio (SLA) conel ISP o proveedor donde se decide como las aplicaciones irán compartir los recursos de red.
Los SLAs suelen ser estáticos y no requieren señalización con el proveedor. El AF PHB posee cuatro clases con tres niveles de descarte en cada clase. En caso de congestión, el nivel de descarte del paquete determina la relativa importancia del mismo dentro de la clase AF.
Una aplicación típica es una Intranet en la que se requiere una alta probabilidad de llegada de los paquetes siempre que no se exceda la capacidad suscrita [22]. Por ejemplo: [21].

- Oro: El tráfico de esta categoría dispone del 50 % del ancho de banda.
- Plata: reserva el 30% del ancho de banda.
- Bronce: con el 20% del ancho de banda.

En la frontera del dominio DS puede controlarse la cantidad de tráfico en los diferentes niveles de descarte: adaptación, descarte de paquetes, variar el nivel de descarte, y asignar los paquetes a otras clases de AF.
Las implementaciones de AF deben detectar y responder a las congestiones a largo plazo con descarte de paquetes, y manejar las de corto plazo mediante mecanismos de colas.

| | Clase 1 | Clase 2 | Clase 3 | Clase 4 |
|---|---|---|---|---|
| Descarte bajo | 001010 | 010010 | 011010 | 100010 |
| Descarte Medio | 001100 | 010100 | 011100 | 100100 |
| Descarte Alto | 001110 | 010110 | 011110 | 100110 |

**Figura 2.18: Clases de Servicio DiffServ**

El servicio Best-Effort corresponde al tradicional servicio de Internet. En DiffServ seconsidera el perfil del mejor esfuerzo a todos los flujos que no contratan niveles de calidad superiores o a los flujos Assured que no están conformes con los perfiles declarado (contratados) para los flujos.

| Servicio | Características |
|---|---|
| 'Expedited Forwarding' o 'Premium' | • Es el que da más garantías. Equivale a una línea dedicada<br>• Lo garantiza todo: Caudal, tasa de pérdidas, retardo y jitter |
| 'Assured Forwarding' | • Asegura un trato preferente, pero sin fijar garantías (no hay SLA)<br>• Se definen cuatro clases y en cada una tres niveles de descarte de paquetes |
| 'Best Effort' | • Ninguna garantía, obtiene solo las migajas |

**Figura 2.19: Tipos de Servicio.**

Después que los paquetes son marcados con un determinado PHB, los mismos son enviados a una interfaz de salida de un nodo de borde que utiliza dos mecanismos de cola de prioridades. En el caso de EF PHB (Premium) se proporciona un servicio de tiempo real con prioridad absoluta (HOL - Head of Line), lo que garantiza que los paquetes serán enviados primero. Con lo cual, los paquetes no conformes con ese servicio serán automáticamente descartados ya que su transmisión a prioridad más baja degradaría la calidad del servicio y consumiría recursos de red.

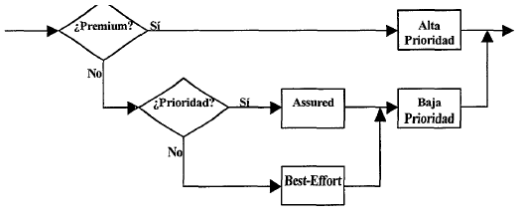



**Figura 2.20: Interfaz del nodo DiffServ**

En el AF PHB se considera la prioridad de descarte de paquetes, lo que determina la calidad extremo a extremo del flujo de tráfico. Es decir, que entre los paquetes tolerantes a retardos algunos están protegidos contra descarte en situación de congestión. Para los paquetes que llevan marcas BE PHB su prioridad de descarte es la más baja y procede de los usuarios que tienen contratado el servicio básico de Internet o de los usuarios del servicio AF PHB que excedan la tasa contratada. Debido a granularidad de clases de servicio el modelo DiffServ se presenta como una solución escalable y de fácil implementación capaz de ofrecer servicio de manera diferenciada a una red en condiciones de congestión. Por lo tanto, es la solución más idónea para ser utilizada como soporte al despliegue de una plataforma de Telefonía IP. Sin embargo, existe un problema que no puede ser resuelto sólo con la utilización de DiffServ en la capa de transporte, que es la

concentración de tráfico de alta prioridad en determinados puntos de la red. Ese problema puede afectar al tráfico de baja prioridad y degradar el tráfico de alta prioridad en los routers de los nodos de borde. Como solución a ese tipo de problema se plantea en [18] una propuesta para soportar DiffServ sobre redes MPLS (Multi- Protocol Label Switching) permitiendo seleccionar los DiffServ Behavior Aggregates (BAs) dentro de un LSP (Label Switched Paths).

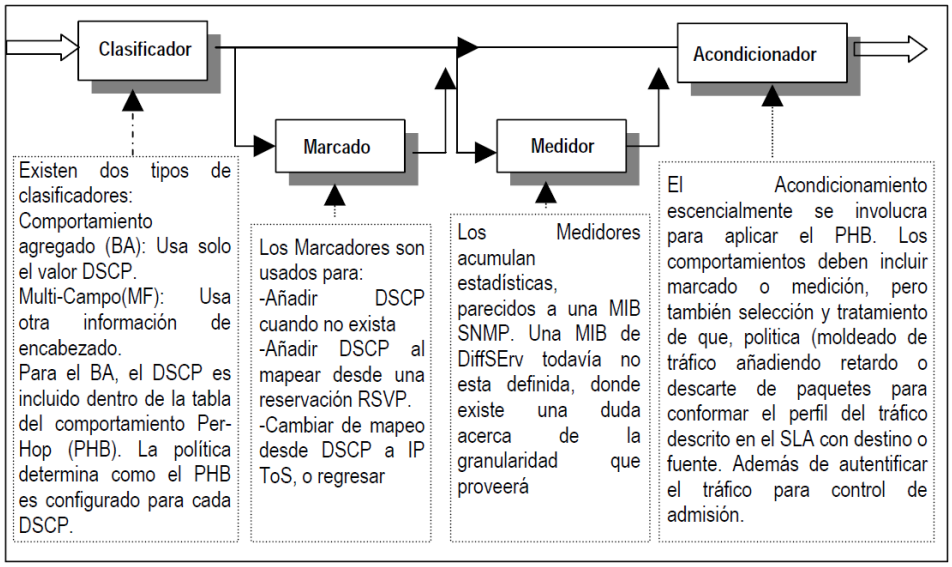



**Figura 2.21: [13] Arquitectura General de los Servicios Diferenciados**

### 2.3.7 Tendencias

En la actualidad los servicios diferenciados están soportados por diversos fabricantes. Los servicios diferenciados es la construcción de una red piloto con el objetivo de probar aplicaciones/servicios que puedan ser implantados en el futuro con garantías de calidad de servicio. Dicha red deberá ser capaz de ofrecer, a manera de ejemplo, video conferencia de alta calidad, servicios de educación a distancia y de telemedicina, además de soportar los servicios tradicionales de datos. Una posible aplicación es implantar una red piloto usando como tecnología del backbone MPLS (MultiProtocol LabelSwitching) y con un modelo de Servicios Diferenciados (DiffServ).

# CAPÍTULO III
# MATERIALES Y MÉTODOS

## 3.1 DISEÑO DE LA INVESTIGACIÓN

La presente investigación se enmarca dentro de un estudio **Cuasi- Experimental**, ya que se trabaja con grupos intactos y además se manipula una variable independiente. Los contenidos a ser enviados en el ambiente de pruebas no serán tomados al azar, sino que se los tendrá definidos antes de realizar dicho ambiente.

Su validez se alcanzará cuando en los diferentes escenarios, uno aplicando "best effort" (BE), MPLS y MPLS con Servicios Diferenciados, realizando comparaciones entre los escenarios midiendo cada uno de los parámetros que determinan la calidad de Servicio (QoS).

## 3.2 TIPO DE INVESTIGACIÓN

Es **investigación experimental** ya que se manipula la variable independiente para analizar las consecuencias que esta manipulación tiene sobre las variables dependientes.

Es **investigación descriptiva** pues se realiza una descripción de las características del objeto de estudio.

## 3.3MÉTODOS, TÉCNICAS E INSTRUMENTOS

### 3.3.1 Métodos

Para este proyecto se ha utilizado los siguientes métodos de investigación:

**Método Científico:** Debido a que para realizar esta investigación, se usa procedimientos establecidos por la comunidad científica a través de su modelo general: planteamiento del problema, formulación de hipótesis, levantamiento de la información, análisis de resultados, comprobación de hipótesis, y difusión de los resultados. Por lo que se ha realizado las siguientes consideraciones:

- Se plantea la investigación basados en el análisis de las redes MPLS aplicando la arquitectura DiffServ
- Se trazan los objetivos de la investigación que permitirán estudiar la arquitectura DiffServ sobre redes MPLS.
- Determinar los parámetros que influyen en la calidad de servicio en tiempo real.
- Se justifican los motivos por los cuales se propone realizar la siguiente investigación.

- Se elabora un marco teórico que ayude a adquirir una visión general de los aspectos más relevantes dentro de esta investigación.
- Se plantea una hipótesis la cual es una posible respuesta al problema planteado y posee una íntima relación entre el problema y el objetivo.
- Se propone la operacionalización de las variables en base a la hipótesis planteada.
- Se realiza la recolección de datos, y se observa el comportamiento de los ambientes de pruebas con el fin de determinar cuál es la solución más adecuada para la virtualización de escritorios y posteriormente comprobar si existe una mejora en la gestión de escritorios de trabajo dentro de una red corporativa al implementar esta solución.
- Se realiza la prueba de la hipótesis con los resultados obtenidos.
- Se elabora las conclusiones y recomendaciones, producto del desarrollo de esta investigación.

**Método Deductivo**

Método que será utilizado en el análisis de la arquitectura DiffServ ya que se estudiará desde una definición y características generales hacia los detalles más particulares.

### 3.3.2 Técnicas

El desarrollo de un trabajo de investigación demanda de una adecuada selección del tema de estudio, de un buen planteamiento del problema a solucionar y de la definición del método que se utilizará para llevarla a cabo. Sumado a esto es muy importante seleccionar las técnicas y herramientas adecuadas que sustenten el desarrollo de la investigación. En este estudio utilizaremos las siguientes técnicas:

- Observación
- Recopilación de información
- Pruebas

### 3.3.3 Instrumentos

El instrumento usado para la medición de la variable dependiente Calidad de Servicio punto a punto en la transmisión de tráfico en tiempo real es:

- D-igt 2.61 GUI 0.92: (Distributed Internet Traffic Generator) el cual permite analizar el tráfico y la Calidad de Servicio QoS que posee la red mediante la obtención jitter, delay y packet loss.

### 3.4 Validación de los instrumentos

La validez del instrumento depende del grado en que se mide el dominio específico de las variables que intervienen en la investigación. El instrumento a utilizarse es una plataforma capaz de producir a nivel de paquetes con gran exactitud replicando apropiadamente procesos estocásticos para ambos IDT (inter Departure Time) y las variables PS (packet Size) aleatorias (Exponencial, uniforme, etc.). D-ITG soporta generación de tráfico IPv4, IPV6 y es capaz de generar tráfico a nivel de red, transporte y aplicación. Ayudará en el análisis de tráfico, mediciones de desempeño y simulación de tráfico de diversos protocolos. Además se obtendrá los datos para los indicadores de la variable dependiente: retardo, tasa de pérdida de paquetes, jitter que se producen en los tres escenarios: Redes IP, MPLS, MPLS y DiffServ.

### 3.5 Población y muestra

Una vez definido el problema a investigar, formulados los objetivos y delimitadas las variables, se hace necesario determinar los elementos o individuos con quienes se va a llevar a cabo esta investigación, para ello se definirá la población y se seleccionará la muestra.

#### 3.5.1 Población

La población es el conjunto de todos los elementos a ser evaluados, en la presente investigación tenemos dos tipos de población: Tráfico en tiempo real y estándares de VoIP.

Los tipos de tráfico son: Voz/fax, datos de transacción, datos de cliente/servidor, mensajería, transferencia de archivos, datos en lote, administración de red y videoconferencia según Lic.Yanina Medina (Teleproceso y Sistemas Distribuidos).

Los estándares de VoIP son: G.711: bit-rate de 56 o 64 Kbps, G.722: bit-rate de 48, 56 o 64 Kbps, G.723: bit-rate de 5,3 o 6,4 Kbps, G.728: bit-rate de 16 Kbps, G.729: bit-rate de 8 o 13 Kbps.

#### 3.5.2 Muestra

De estas poblaciones se seleccionó dos muestras dirigidas basada en los siguientes criterios:

Para el tráfico en tiempo real:

- Los paquetes de VoIP deben recibir un trato especial, son muy sensibles a retardos y necesitan un ancho de banda garantizado. Es necesario ofrecer cierta QoS para disminuir el retardo. Según Adrián Delfino y Sebastián Rivero en su trabajo de Evaluación de Performance en Redes de Telecomunicaciones.
- Los datos son los paquetes que más se envían en la red según Sacha Fuentes publicado en Genbeta.
- En empresas privadas como Serivarsa S.A. se tiene como políticas priorizar el tráfico de VoIP y de datos, al ser los más utilizados e importantes.

- Los datos tienen un tamaño de paquete de 1028 bytes.

Para los estándares de VoIP se tiene:

- Los estándares de VoIP más utilizados según la ITU-T G (Series: Transmission systems and media, digital systems and networks) son: G.711, G.729 y G.723.1.

Según los criterios mencionados anteriormente las muestras dirigidas son:

- Tráfico VoIP y Datos.
- Los estándares de VoIP G.711, G.729 y G.723.1

**3.6 PROCESAMIENTO DE LA INFORMACIÓN**

Para realizar el estudio de las redes MPLS y DiffServ, la información ha sido recopilada de páginas web, revistas especializadas, foros y libros.

Los indicadores se determinaron en base a la información recolectada siendo estos:

- Mecanismos
- Políticas
- Servicios

Para determinar si aplicando la tecnología DiffServ sobre las redes MPLS se tendrá una adecuada QoS punto a punto en la transmisión en tiempo real se realizaron los siguientes escenarios:

- Redes IP: Es la configuración básica de redes IP, donde se configura las ips y protocolo de enrutamiento OSPF. Inyectando tráfico en tiempo real datos y VoIP tomando las mediciones de datos para cada uno de los indicadores.
- Redes MPLS: Es la implementación de MPLS sobre la red IP existente, se medirá cada una de los parámetros de QoS.
- Redes MPLS DiffServ: En este escenario se configurara las políticas de entrada y salida para las interfaces que se aplicara sobre redes MPLS (DiffServ), midiendo los indicadores de QoS.

En cada escenario se tomaron en cuenta los siguientes indicadores de QoS:

- Jitter
- Paquetes perdidos
- Retardo o latencia

Todos estos datos son organizados en tablas de valoración, y posteriormente aplicando métodos estadísticos se comprueba si la implementación de esta solución es la adecuada.

**Ambiente de Pruebas**

Para la implementación del ambiente de pruebas hemos utilizado: 6 router cisco, 3 switch, 2 pcs de escritorio y una portátil

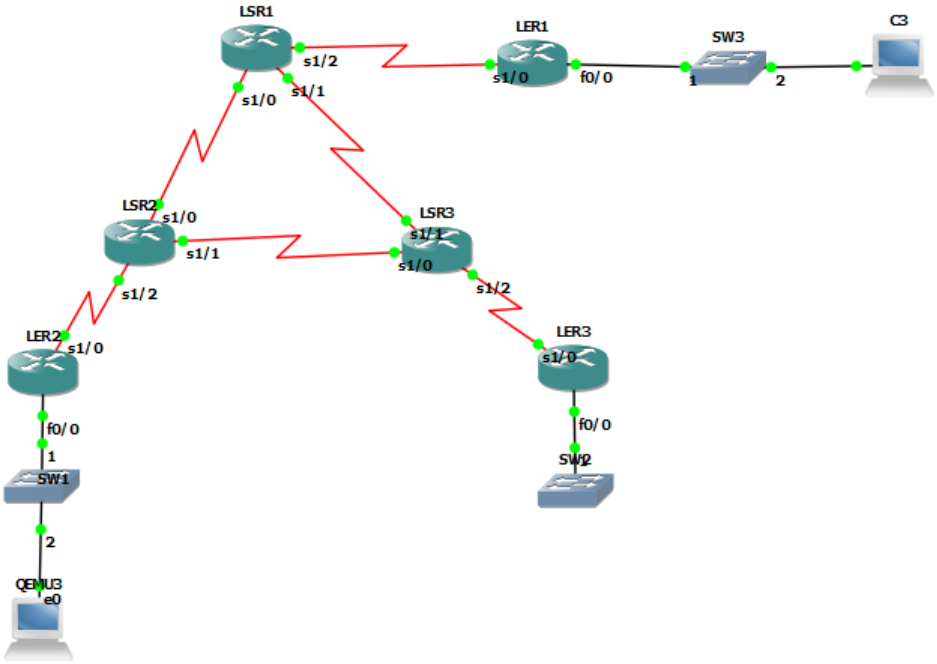



**Figura 3.1: Ambiente de Pruebas**

Las características de cada uno de ellos las describimos a continuación:

**Tabla III.1: Características de los equipos usados en los escenarios de prueba**

| Descripción | Característica |
|---|---|
| Router | Routers cisco 2800<br>Protocolo de interconexión de datos: Ethernet, Fast Ethernet<br>Red / Protocolo de transporte: IPSec<br>Protocolo de gestión remota: SNMP 3<br>Algoritmo de cifrado: DES, Triple DES, AES<br>Características: Cifrado del hardware, soporte de MPLS<br>Memoria RAM: 128 MB (instalados) / 384 MB (máx.), 192 MB (instalados) / 384 MB |

| | |
|---|---|
| | (máx.), 256 MB (instalados) / 384 MB (máx.) - SDRAM, 256 MB (instalados) / 384 MB (máx.)<br>Memoria Flash: 64 MB (instalados) / 128 MB (máx.), 128 MB (instalados) / 128 MB (máx.)<br>Indicadores de estado: Actividad de enlace, alimentación<br>Cumplimiento de normas: IEEE 802.3af<br>Método de autentificación: Secure Shell v.2 (SSH2)<br>Capacidad: Conexiones SSL concurrentes : 10 |
| Switch | Switch 2960<br>Posee 24 puertos Ethernet de 10/100BASE-T y dos enlaces ascendentes de 10/100/1000TX.<br>Proporciona mejoras para Spanning Tree.<br>Configuración de hasta 255 VLANs por puerto.<br>La tasa de envío basada en paquetes de 64bytes es de 6.5Mpps.<br>64 MB de memoria DRAM y 32 MB de memoria Flash.<br>Soporte para MPLS e IPv6. |
| Laptop | Portátil Toshiba Satellite Pro P750 Series |
| Pc | Pc de Escritorio Pentium 4 |

**Elaborado por:** Ing. Pamela Buñay

El software requerido para la implementación de los prototipos que nos servirán como instrumentos para realizar esta investigación son los siguientes:

- Dig-it 2.61 GUI 0.92

### 3.6.1 Escenarios de Prueba

Los escenarios que nos servirán para la obtención de datos son:

**Tabla III.2: Escenarios de pruebas para la toma de datos**

| | |
|---|---|
| **Escenarios de Prueba** | Redes IP con tráfico en tiempo real midiendo los parámetros de QoS |
| | Redes MPLS con tráfico en tiempo real midiendo los parámetros de QoS |

| | |
|---|---|
| | Redes MPLS aplicando la arquitectura DiffServ con tráfico en tiempo real midiendo los parámetros de QoS |

**Elaborado por:** Ing. Pamela Buñay

### 3.6.2 Implementación de Escenarios de Prueba

A continuación se describen como se encuentran implementados los escenarios prototipo que serán el instrumento para la obtención de datos dentro de la investigación.

#### 3.6.2.1 ESCENARIO 1: Redes IP con tráfico en tiempo real

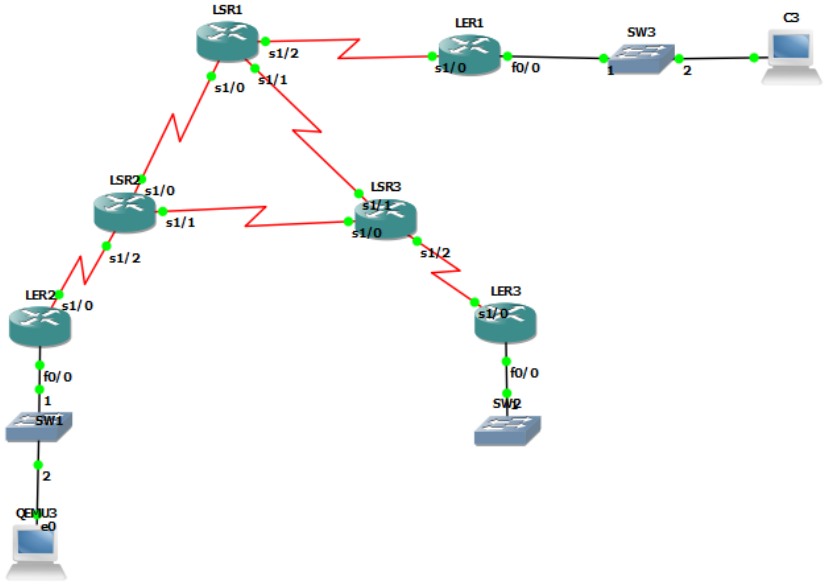



**Figura3.2: Redes IP con tráfico en tiempo real**

En este escenario se configura las IPs y el protocolo OSPF como se detalla a continuación:

- **Configuración básica del router**

Router > enable
Router # configure terminal
Router (config) # hostname LSR1
LSR1 (config) # line con 0
LSR1 (config - con) # password
LSR1 (config - con) # <contraseña de la consola>
LSR1 (config - con) # login
LSR1 (config - con) # exit
LSR1 (config) # enable secret <contraseña enable>
LSR1 (config) # config-register 0x2102

Pulsamos Ctrl+z para salir del modo configuración
LSR1 # write memory
Realizar las mismas configuraciones para los routers LSR2, LSR3, LER1, LER2, LER3.

- **Configuración del direccionamiento IP**

LSR1 (config) # interface s1/0
LSR1 (config-if) # ip address 192.168.2.2 255.255.255.0
LSR1(config-if)# clockrate 64000
LSR1 (config-if) # no shutdown
LSR1 (config) # interface s1/1
LSR1 (config-if) # ip address 192.168.4.1 255.255.255.0
LSR1(config-if)# clockrate 64000
LSR1 (config-if) # no shutdown
LSR1 (config) # interface s1/2
LSR1 (config-if) # ip address 192.168.3.1 255.255.255.0
LSR1 (config-if) # no shutdown

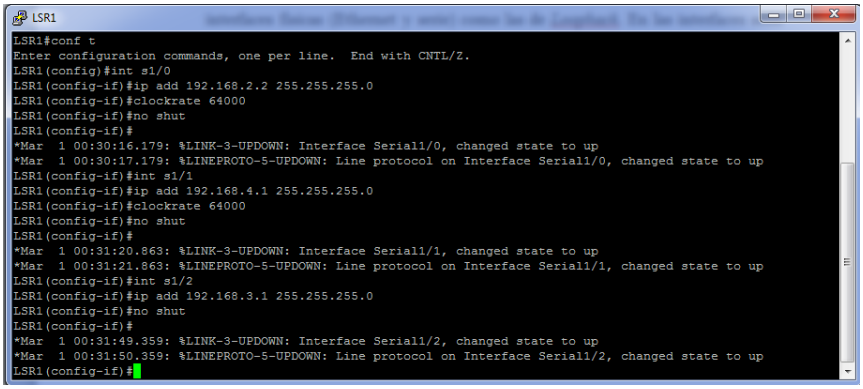



**Figura 3.3: Configuración del direccionamiento IP de LSR1**

Para los otros routers las direcciones ips e interfaces son las siguientes:

**Tabla III.3: Interfaz y direcciones IP de los routers**

| Router | Interfaz y dirección IP |
|---|---|
| **LSR2** | S1/0: 192.168.2.1<br>S1/1: 192.168.6.1<br>S1/2: 192.168.1.2 |
| **LSR3** | S1/0: 192.168.6.2<br>S1/1: 192.168.4.2<br>S1/2:192.168.5.1 |
| **LER1** | S1/0: 192.168.3.2<br>F0/0: 192.168.7.1 |
| **LER2** | S1/0: 192.168.1.1<br>F0/0: 192.168.9.1 |
| **LER3** | S1/0: 192.168.5.2<br>F0/0: 192.168.8.1 |

**Elaborado por: Ing**. Pamela Buñay

- **Configuración de OSPF**

LSR1(config)# router ospf
LSR1(config)# router ospf 1
LSR1(config-router)# network 192.168.2.0 0.0.0.255 area 0
LSR1(config-router)# network 192.168.3.0 0.0.0.255 area 0
LSR1(config-router)# network 192.168.4.0 0.0.0.255 area 0

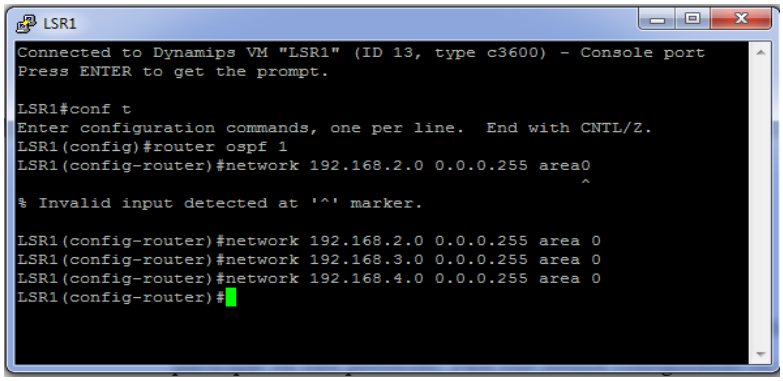



**Figura 3.4:Configuración de OSPF LSR1**

Para los otros equipos considerar las siguientes redes:

**Tabla III.4: Redes para protocolo OSPF**

| Router | Red |
|---|---|
| **LSR2** | 192.168.2.0<br>192.168.6.0<br>192.168.1.0 |
| **LSR3** | 192.168.6.0<br>192.168.4.0<br>192.168.5.0 |
| **LER1** | 192.168.3.0<br>192.168.7.0 |
| **LER2** | 192.168.1.0<br>192.168.9.0 |
| **LER3** | 192.168.5.0<br>192.168.8.0 |

**Elaborado por:** Ing. Pamela Buñay

- **Probar conectividad extremo a extremo**

LSR1# ping 192.168.7.1
LSR1# ping 192.168.9.1
LSR1# ping 192.168.8.1

### 3.6.2.2 ESCENARIO 2: Redes MPLS con tráfico en tiempo real

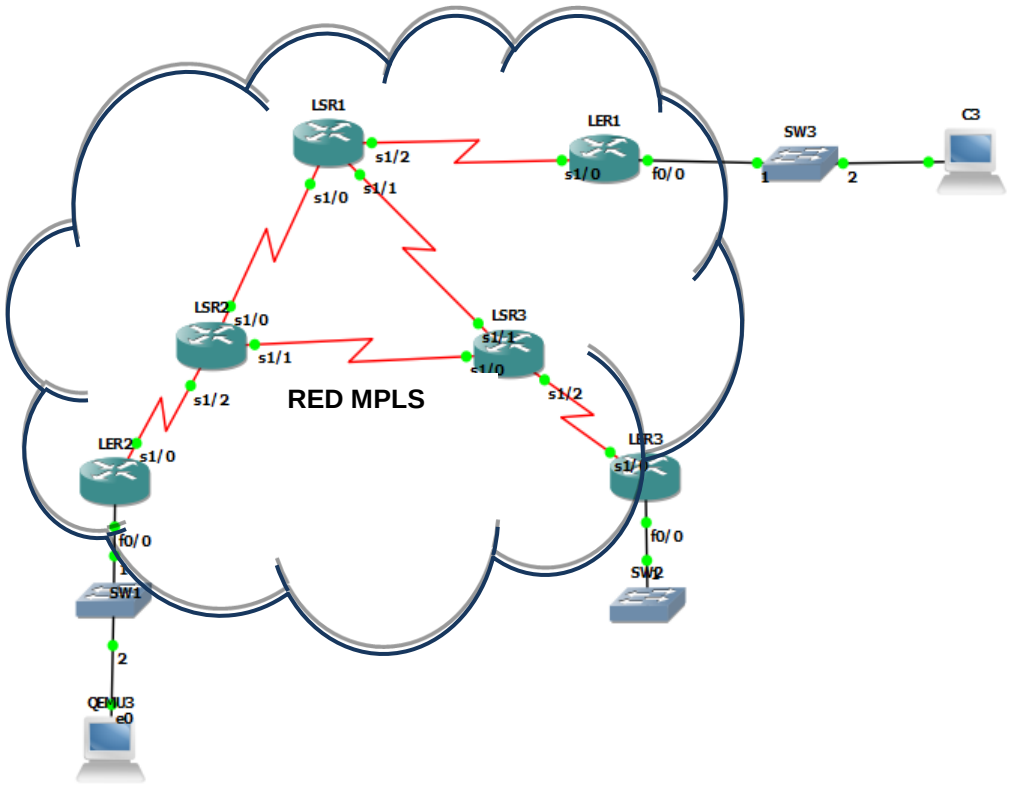



**Figura 3.5:Redes MPLS con tráfico en tiempo real**

Para habilitar MPLS en los routers, hay que indicar qué interfaces del router van a participar en este protocolo. Para ello iremos configurando en dichas interfaces del router el comando "**mpls ip**" de forma que indicamos al router que conmute en entrada y salida las tramas MPLS que reciba o envíe, así como que detecte vecindades de routers MPLS con el protocolo de distribución de etiquetas. Las configuraciones son para LSR y LER.

- **Para activar CEF y poder trabajar en entornos MPLS**

```
LSR1(config)# ip cef
LSR1(config)# # mpls label protocol ldp
LSR1(config)# mpls ip
```

- **Para activar el protocolo de distribución de etiquetas LDP**

Configura MPLS en todas las interfaces.

```
LSR1(config)# interface s1/0
LSR1(config-if)# mpls ip
LSR1(config-if)# mpls label protocol ldp
```

```
LSLSR1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
LSLSR1(config)#ip cef
LSLSR1(config)#mpls label protocol ldp
LSLSR1(config)#mpls ip
LSLSR1(config)#int s1/0
LSLSR1(config-if)#mpls ip
LSLSR1(config-if)#mpls label protocol ldp
LSLSR1(config-if)#int s1/1
LSLSR1(config-if)#mpls ip
LSLSR1(config-if)#mpls label protocol ldp
LSLSR1(config-if)#int s1/2
LSLSR1(config-if)#mpls ip
LSLSR1(config-if)#mpls label protocol ldp
LSLSR1(config-if)#
```

**Figura 3.6:Activar el protocolo de distribución de etiquetas LDP**

Lo mismo se realiza para las interfaces de los routers LSR2, LSR3, LER1, LER2 y LER 3

**Tabla III.5: Interfaz de los routers**

| Router | Interfaz |
|---|---|
| **LSR2** | S1/0<br>S1/1<br>S1/2 |
| **LSR3** | S1/0<br>S1/1<br>S1/2 |
| **LER1** | S1/0<br>F0/0 |
| **LER2** | S1/0<br>F0/0 |
| **LER3** | S1/0<br>F0/0 |

**Elaborado por: Ing**. Pamela Buñay

- **Pruebas de la red con el protocolo de enrutamiento OSPF**

Una vez realizadas las configuraciones se procede a verificar la conectividad.

```
LER1
Connected to Dynamips VM "LER1" (ID 5, type c3600) - Console port
Press ENTER to get the prompt.

LER1#ping 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 60/61/68 ms
LER1#ping 192.168.5.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.5.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 60/70/96 ms
LER1#
```

**Figura 3.7: Conectividad desde LER1 a LER2 y LER3**

- **Verificar la configuración de MPLS**

Para verificar la configuración de MPLS se utiliza el comando **"show mpls forwarding-table"** el cual muestra la asignación de etiquetas por cada ruta.

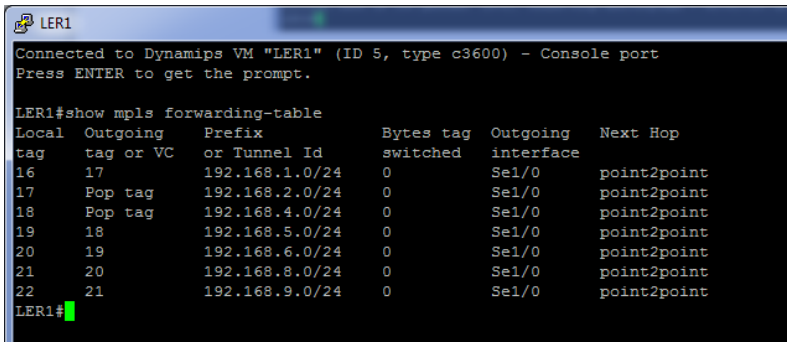



**Figura 3.8: Asignación e intercambio de etiquetas en el LER1**

- Utilizando el comando **"show mpls interfaces"** se puede apreciar que en las interfaces está habilitado el protocolo LDP (Label Distribution Protocol).

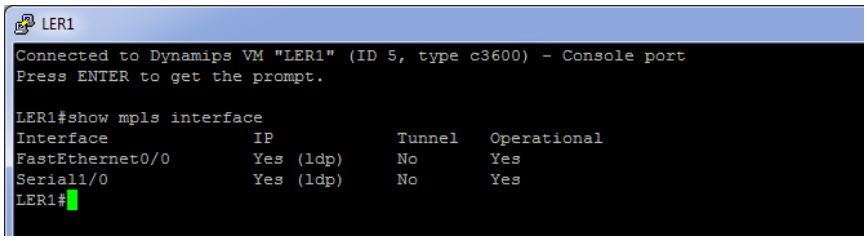



**Figura 3.9: Protocolo LDP habilitado**

- Agregando **"detail"** al anterior comando se puede observar más detalles de la configuración como por ejemplo el tamaño del MTU (Maximum Transmission Unit)

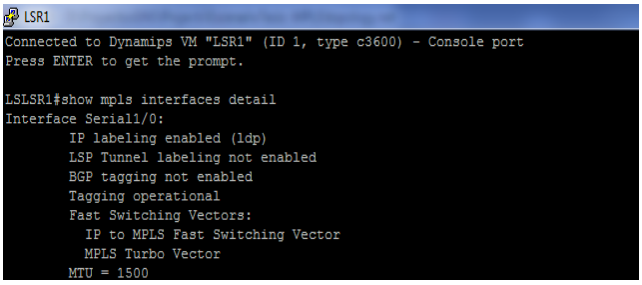



**Figura 3.10: Detalles de las interfaces con MPLS**

- Con el comando **"traceroute"** se verifica los saltos para llegar a una IP de destino y el intercambio en esos saltos del valor de las etiquetas.Además se puede apreciar que el campo EXP por defecto tiene el valor de 0, posteriormente se manipula este campo para ofrecer QoS mediante DiffSev.

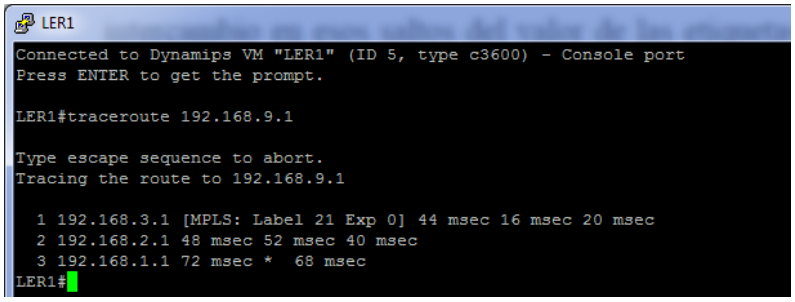



**Figura 3.11: Traceroute a 192.168.9.1**

- **Modificación del tamaño de MTU para MPLS**

Una de las características de MPLS es que permite anidar etiquetas MPLS en función de la aplicación y por tanto puede aumentar el número de cabeceras y para ello hay que informar a las interfaces físicas de dichas eventualidades para evitar el descarte de tramas que superen la MTU. La cabecera MPLS tiene 4 bytes. La MTU por defecto siempre se toma de la propia interfaz, que en el caso de una Ethernet es 1500 bytes.

LSR1(config)# interface s1/0
LSR1(config-if)# mpls mtu 1508
LSR1(config)# interface s1/1
LSR1(config-if)# mpls mtu 1508
LSR1(config)# interface s1/2
LSR1(config-if)# mpls mtu 1508

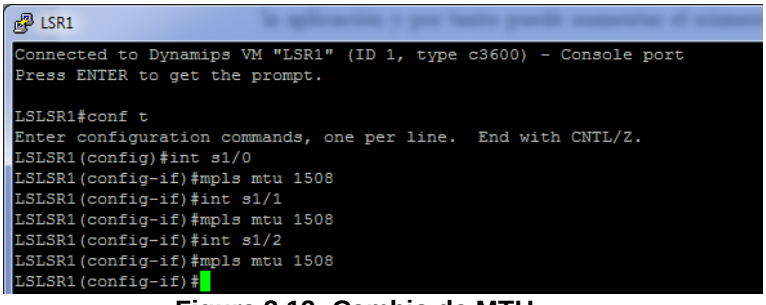



**Figura 3.12: Cambio de MTU**

Realizar lo mismo para los routers LSR2, LSR3, LER1, LER2, LER3.

### 3.6.2.3 ESCENARIO 3: Redes MPLS aplicando la arquitectura DiffServ con tráfico en tiempo real

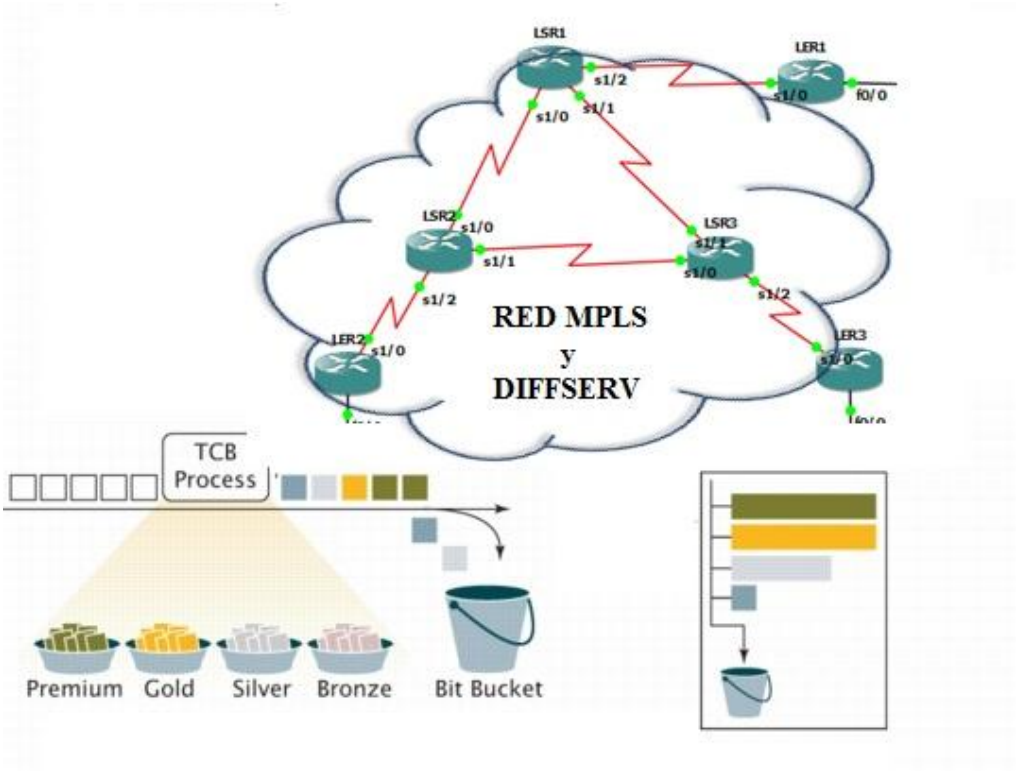



**Figura 3.13: Redes MPLS aplicando la arquitectura DiffServ con tráfico en tiempo real**

Mediante la integración de los modelos MPLS y DiffServ obtenemos una arquitectura en la que MPLS se sitúa en el nivel de red-enlace, y sirve para evitar la congestión de la red, aportando sus características de ingeniería de tráfico. Mientras, DiffServ asegura unos ciertos parámetros de calidad de servicio realizando una distinción y priorización del tráfico.

- **Marcado y clasificación del tráfico**

Crear una clase de tráfico consiste en agrupar un tipo de tráfico de la red bajo un mismo concepto, de tal manera que los paquetes agrupados en esa clase pueden tener un mismo tratamiento cuando llegan al router.

De acuerdo al valor Precedence del paquete IP al ingreso por el router LER1 se lo clasifica dentro de las clases definidas DiffServ que son; AF11, AF12, AF21, AF22, AF31 y AF32 aplicando la siguiente configuración:

```
LER1#configure terminal
LER1(config)#class-map IP-AF11
```

```
LER1(config-cmap)#match ip precedence 0
LER1(config-cmap)#exit
LER1(config)#class-map IP-AF12
LER1(config-cmap)#match ip precedence 1
LER1(config-cmap)#exit
LER1(config)#class-map IP-AF21
LER1(config-cmap)#match ip precedence 2
LER1(config-cmap)#exit
LER1(config)#class-map IP-AF22
LER1(config-cmap)#match ip precedence 3
LER1(config-cmap)#exit
LER1(config)#class-map IP-AF31
LER1(config-cmap)#match ip precedence 4
LER1(config-cmap)#exit
LER1(config)#class-map IP-AF32
LER1(config-cmap)#match ip precedence 5
LER1(config-cmap)#exit
```

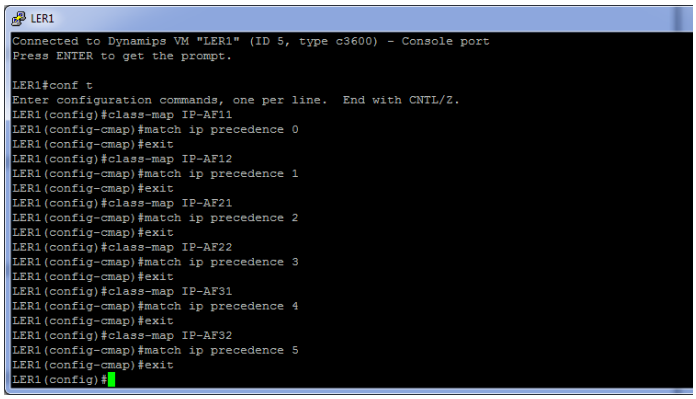



**Figura 3.14: Marcado y clasificación del tráfico**

- **Creación de la política de entrada**

Para configurar una política de tráfico, se usa el comando **policy-map**, en el que se especifica en nombre de la política, y a la que se asocian clases de tráfico, definidas previamente. Todo el tráfico que no se equipara con los criterios de las clases, pertenecen a la clase de tráfico por defecto. Esta también se puede configurar por el usuario, pero no eliminar.

La política configurada en este caso permite primeramente enviar el tráfico a cierta velocidad y copiar el valor Precedence al campo EXP para la transmisión hacia el siguiente salto.

```
LER1(config)#policy-map politica-1
LER1(config-pmap)#class IP-AF11
LER1(config-pmap-c)#police 8000 conform-action set-mpls-exp-imposition-transmit 0 exceed-action
drop
LER1(config-pmap-c)#exit
LER1(config-pmap)#class IP-AF12
```

```
LER1(config-pmap-c)#police 10000 conform-action set-mpls-exp-imposition-transmit 1 exceed-action set-mpls-exp-imposition-transmit 0
LER1(config-pmap-c)#exit
LER1(config-pmap)#class IP-AF21
LER1(config-pmap-c)#police 12000 conform-action set-mpls-exp-imposition-transmit 2 exceed-action set-mpls-exp-imposition-transmit 1
LER1(config-pmap-c)#exit
LER1(config-pmap)#class IP-AF22
LER1(config-pmap-c)#police 12000 conform-action set-mpls-exp-imposition-transmit 3 exceed-action set-mpls-exp-imposition-transmit 2
LER1(config-pmap-c)#exit
LER1(config-pmap)#class IP-AF31
LER1(config-pmap-c)#police 12000 conform-action set-mpls-exp-imposition-transmit 4 exceed-action set-mpls-exp-imposition-transmit 3
LER1(config-pmap-c)#exit
LER1(config-pmap)#class IP-AF32
LER1(config-pmap-c)#police 12000 conform-action set-mpls-exp-imposition-transmit 5 exceed-action set-mpls-exp-imposition-transmit 4
LER1(config-pmap-c)#end
```

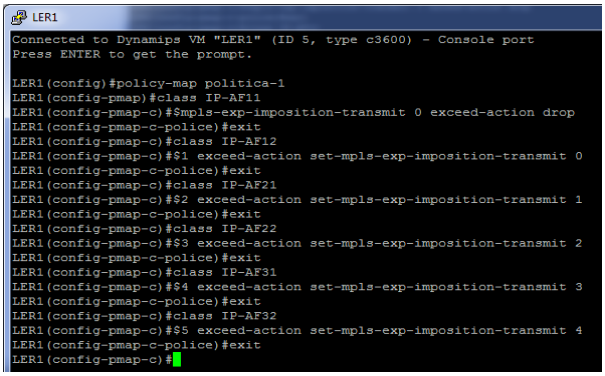



**Figura 3.15: Creación de la política de entrada**

- **Asignación de la política a la interfaz de entrada**

Para asociar una política de tráfico a una interfaz, y especificar la dirección en la cual debe especificarse la política (paquetes entrantes o paquetes salientes), se utiliza el comando **service-policy.**

```
LER1(config)#int f0/0
LER1(config-if)#service-policy input politica-1
LER1(config-if)#exit
```

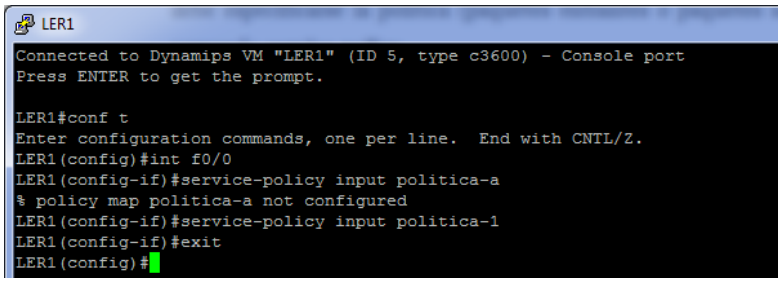



**Figura 3.16: Asignación de la política a la interfaz de entrada**

- **Clasificación de los paquetes en base al campo EXP**

Los paquetes nuevamente son clasificados a la salida del router, para este caso de acuerdo al valor del campo EXP de la etiqueta superior y es colocado en las respectivas clases.

```
LER1#configure terminal
LER1(config)#class-map MPLS-AF11
LER1(config-cmap)#match mpls experimental topmost 0
LER1(config-cmap)#exit
LER1(config)#class-map MPLS-AF12
LER1(config-cmap)#match mpls experimental topmost 1
LER1(config-cmap)#exit
LER1(config)#class-map MPLS-AF21
LER1(config-cmap)#match mpls experimental topmost 2
LER1(config-cmap)#exit
LER1(config)#class-map MPLS-AF22
LER1(config-cmap)#match mpls experimental topmost 3
LER1(config-cmap)#exit
LER1(config)#class-map MPLS-AF31
LER1(config-cmap)#match mpls experimental topmost 4
LER1(config-cmap)#exit
LER1(config)#class-map MPLS-AF32
LER1(config-cmap)#match mpls experimental topmost 5
LER1(config-cmap)#exit
```

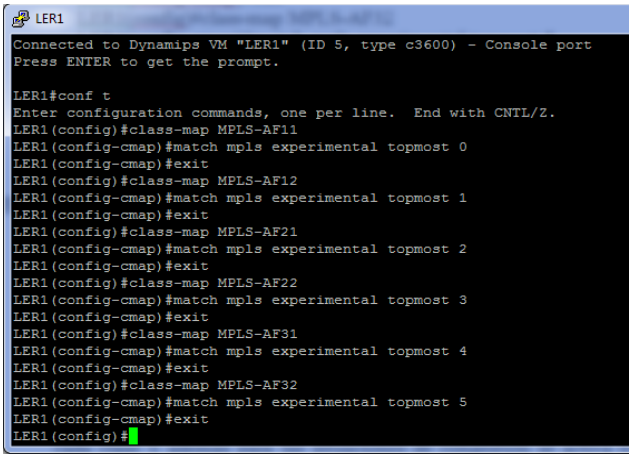



**Figura 3.17: Clasificación de los paquetes en base al campo EXP**

- **Creación de la política a la salida del router**

La política a la salida del router especifica el porcentaje de ancho de banda asignado a cada clase y además para las situaciones de congestión se activa el mecanismo de descarte inteligente para evitar oscilaciones llamado WRED (Weighted Random Early Discard).

```
LER1(config)#policy-map politica-2
LER1(config-pmap)#class MPLS-AF11
LER1(config-pmap-c)#bandwidth percent 5
LER1(config-pmap-c)#random-detect
LER1(config-pmap-c)#exit
LER1(config-pmap)#class MPLS-AF12
LER1(config-pmap-c)#bandwidth percent 10
LER1(config-pmap-c)#random-detect
LER1(config-pmap-c)#exit
LER1(config-pmap)#class MPLS-AF21
LER1(config-pmap-c)#bandwidth percent 10
LER1(config-pmap-c)#random-detect
LER1(config-pmap-c)#exit
LER1(config-pmap)#class MPLS-AF22
LER1(config-pmap-c)#bandwidth percent 15
LER1(config-pmap-c)#random-detect
LER1(config-pmap-c)#exit
LER1(config-pmap)#class MPLS-AF31
LER1(config-pmap-c)#bandwidth percent 15
LER1(config-pmap-c)#random-detect
LER1(config-pmap-c)#exit
LER1(config-pmap)#class MPLS-AF32
LER1(config-pmap-c)#bandwidth percent 20
LER1(config-pmap-c)#random-detect
LER1(config-pmap-c)#end
```

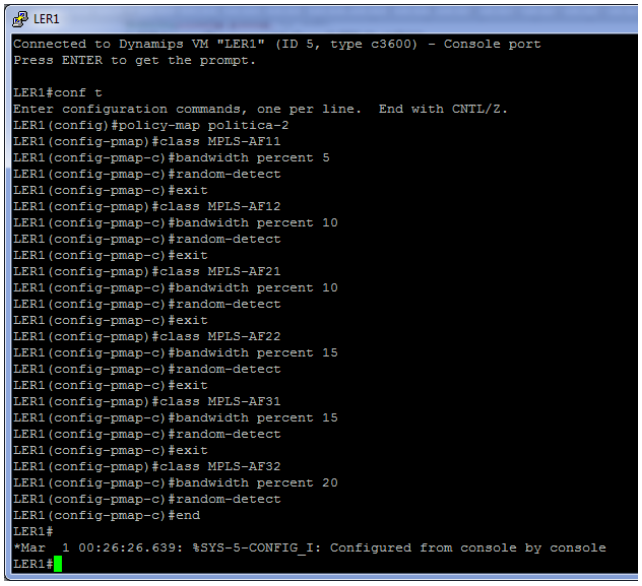



**Figura 3.18: Creación de la política a la salida del router**

- **Asignación de la política en la interfaz de salida**

La política es aplicada a la interfaz de salida del router LER1, en donde los paquetes son clasificados dependiendo del valor del campo EXP y enviados utilizando un ancho de banda de acuerdo a la prioridad.

```
LER1(config)#int s1/0
LER1(config-if)#service-policy output politica-2
LER1(config-if)#exit
```

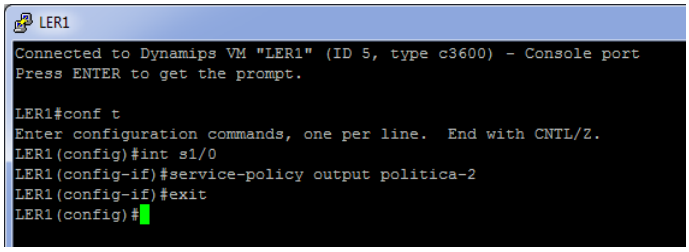



**Figura 3.19: Asignación de la política en la interfaz de salida**

- **Configuración de LRS1**

```
class-map match-all mpls-in
match mpls experimental topmost 3

policy-map mpls-in
class mpls-in
set mpls experimental topmost 2

interface Serial1/2
service-policy input mpls-in
```

- **Configuración de LSR2 y LSR3**

```
class-map match-all MPLS-AF11
match mpls experimental topmost 0
class-map match-all MPLS-AF12
match mpls experimental topmost 1
class-map match-all MPLS-AF21
match mpls experimental topmost 2
class-map match-all MPLS-AF22
match mpls experimental topmost 3
class-map match-all MPLS-AF31
match mpls experimental topmost 4
class-map match-all MPLS-AF32
match mpls experimental topmost 5
class-map match-all qos-group-AF11
match qos-group 0
class-map match-all qos-group-AF12
match qos-group 1
class-map match-all qos-group-AF21
match qos-group 2
class-map match-all qos-group-AF22
match qos-group 3
```

class-map match-all qos-group-AF31
match qos-group 4
class-map match-all qos-group-AF32
match qos-group 5

**policy-map politica-3**
class MPLS-AF11
set qos-group mpls experimental topmost
class MPLS-AF12
set qos-group mpls experimental topmost
class MPLS-AF21
set qos-group mpls experimental topmost
class MPLS-AF22
set qos-group mpls experimental topmost
class MPLS-AF31
set qos-group mpls experimental topmost
class MPLS-AF32
set qos-group mpls experimental topmost

**policy-map politica-4**
class qos-group-AF11
bandwidth percent 5
random-detect
set mpls experimental topmost qos-group
class qos-group-AF12
bandwidth percent 10
random-detect
set mpls experimental topmost qos-group
class qos-group-AF21
bandwidth percent 10
random-detect
set mpls experimental topmost qos-group
class qos-group-AF22
bandwidth percent 15
random-detect
set mpls experimental topmost qos-group
class qos-group-AF31
bandwidth percent 15
random-detect
set mpls experimental topmost qos-group
class qos-group-AF32
bandwidth percent 20
random-detect
set mpls experimental topmost qos-group

**LSR2**
interface Serial1/0
service-policy input politica-3
!
interface Serial1/1
service-policy output politica-4

**LSR3**
interface Serial1/1

service-policy input politica-3
!
interface Serial1/0
service-policy output politica-4

- **Configuración de LER2, LER3**

class-map match-all MPLS-AF11
match mpls experimental topmost 0
class-map match-all MPLS-AF12
match mpls experimental topmost 1
class-map match-all MPLS-AF21
match mpls experimental topmost 2
class-map match-all MPLS-AF22
match mpls experimental topmost 3
class-map match-all MPLS-AF31
match mpls experimental topmost 4
class-map match-all MPLS-AF32
match mpls experimental topmost 5
class-map match-all qos-group-AF11
match qos-group 0
class-map match-all qos-group-AF12
match qos-group 1
class-map match-all qos-group-AF21
match qos-group 2
class-map match-all qos-group-AF22
match qos-group 3
class-map match-all qos-group-AF31
match qos-group 4
class-map match-all qos-group-AF32
match qos-group 5
!
**policy-map politica-5**
class MPLS-AF11
set qos-group mpls experimental topmost
class MPLS-AF12
set qos-group mpls experimental topmost
class MPLS-AF21
set qos-group mpls experimental topmost
class MPLS-AF22
set qos-group mpls experimental topmost
class MPLS-AF31
set qos-group mpls experimental topmost
class MPLS-AF32
set qos-group mpls experimental topmost

**policy-map politica-6**
class qos-group-AF11
bandwidth percent 5
random-detect
set precedence qos-group
class qos-group-AF12
bandwidth percent 10
random-detect
set precedence qos-group

```
class qos-group-AF21
bandwidth percent 10
random-detect
set precedence qos-group
class qos-group-AF22
bandwidth percent 15
random-detect
set precedence qos-group
class qos-group-AF31
bandwidth percent 15
random-detect
set precedence qos-group
class qos-group-AF32
bandwidth percent 20
random-detect
set precedence qos-group
!
interface Serial1/0
service-policy input politica-5
!
interface FastEthernet0/0
service-policy output politica-6
```

Ver Anexo I, los archivos de configuración de los routers.

## 3.7 PLANTEAMIENTO DE LA HIPÓTESIS

La aplicación de la arquitectura ¨ Servicios Diferenciados ¨ sobre redes MPLS proveerá una adecuada Calidad de Servicio punto a punto en la transmisión de tráfico en tiempo real.

### 3.8 OPERACIONALIZACIÓN DE LAS VARIABLES

#### 3.8.1 Operacionalización Conceptual

**Tabla III.6: Operacionalización de las variables**

| Variable | Tipo | Concepto |
|---|---|---|
| Arquitectura ¨ Servicios Diferenciados ¨ sobre redes MPLS | Independiente | Proporciona un método que intenta garantizar la calidad de servicio en redes. Da soporte a varios servicios con diferentes requerimientos en QoS. |
| Calidad de Servicio punto a punto en la transmisión de tráfico en tiempo real. | Dependiente | Son las tecnologías que garantizan la transmisión de cierta cantidad de información en un tiempo dado. |

**Elaborado por:** Ing. Pamela Buñay

### 3.8.2 Operacionalización Metodológica

**Tabla III.7: Operacionalización Metodológica**

| Variable | Indicadores | Técnica | Fuentes de Verificación/ Instrumentos |
|---|---|---|---|
| Arquitectura ¨ Servicios Diferenciados ¨ sobre redes MPLS | Políticas<br>Mecanismo<br>Servicios | Observación<br>Recopilación de información | Internet<br>Libros<br>Revistas<br>Documentos<br>Textos<br>RFC´s |
| Calidad de Servicio punto a punto en la transmisión de tráfico en tiempo real. | Latencia o Retardo<br>Jitter<br>Pérdida de paquetes | Observación<br>Pruebas | D-igt 2.61 GUI 0.92 |

**Elaborado por:** Ing. Pamela Buñay

# CAPÍTULO IV
# RESULTADOS Y DISCUSIÓN

Una vez que se ha efectuado un análisis de los aspectos más relevantes de la Redes MPLS y arquitectura DiffServ, y se ha definido la metodología a seguir para esta investigación, el propósito de este capítulo es realizar una evaluación de los indicadores de las variables dependientes e independientes, para posteriormente comprobar si la implementación de redes MPLS con arquitectura DiffServ provee de una adecuada QoS punto a punto en la transmisión de tráfico en tiempo real. El análisis de estos resultados estará apoyado en pruebas realizadas en prototipos implementados y que fueron descritos anteriormente.

## 4.1 DETERMINACIÓN DE PARÁMETROS

Para realizar el análisis dentro de esta investigación se determina ciertos parámetros (indicadores), que permitirán evaluar cada uno de los escenarios permitiendo determinar si se puede obtener una QoS adecuada para el tráfico en tiempo real.

Estos parámetros han sido determinados en base a información de relevancia publicada en investigaciones similares, revistas especializadas, estudios de tesis, foros de internet.

Los indicadores determinados para la variable independiente son los siguientes:

- Políticas
- Mecanismo
- Tipo de Servicio

Indicadores definidos para la variable dependiente son:

- Latencia o Retardo
- Jitter
- Pérdida de paquetes

### 4.2 ANÁLISIS Y PRESENTACIÓN DE RESULTADOS

#### 4.2.1 Análisis de la Variable Independiente

**Variable independiente:** Arquitectura ¨ Servicios Diferenciados ¨ sobre redes MPLS.

En este proceso de análisis de la variable independiente se realiza el estudio técnico de las redes IP y redes MPLS con DiffServ, se elaborarán cuadros comparativos, seguidos estos de una interpretación, estos cuadros comparativos serán elaborados de acuerdo a los parámetros de comparación definidos anteriormente.

**INDICADOR No.1. Políticas**

El objetivo de este indicador es determinar las políticas que se usan en redes MPLS aplicando la arquitectura de Servicios Diferenciados.
Se basa en el marcado de paquetes únicamente. No hay reserva de recursos por flujo, no hay protocolo de señalización, no hay información de estado en los routers. En vez de distinguir flujos individuales clasifica los paquetes en categorías (según el tipo de servicio solicitado).

**Tabla IV.1: Políticas**

| Característica | Redes IP | Redes MPLS con DiffServ |
|---|---|---|
| **Ingeniería de tráfico** | No | Si |
| **Prioriza aplicaciones en la Red** | No | Si |
| **Políticas de diferenciación de tráfico** | No | Si |

**Elaborado por:** Ing. Pamela Buñay
**Fuente:**Santiago Felici, Sidnei De Oliveira Guerra

**Interpretación:**

La utilización de MPLS para aplicar ingeniería de tráfico promete proporcionar QoS mientras se optimizan los recursos de la red, existiendo en la actualidad un buen número de propuestas en esta línea. Sin embargo, MPLS por sí solo no puede proporcionar diferenciación de tráfico, siendo este requisito imprescindible para la provisión de garantías QoS. Por ello,puede complementarse con DiffServ para aplicar esta diferenciación.
En las redes IP no se realiza una diferenciación de tráfico por este motivo se presenta complicaciones para la prestación de servicios que requieren la transmisión de datos en tiempo real, la llegada de datos desordenados o la pérdida de información puede ser crítica. En DiffServ se divide el tráfico en clases permitiendo controlar la cantidad de tráfico que cada cliente envía a la red de cada clase de tráfico y asegurar requerimientos de QoS utilizando en cada enlace políticas de scheduling y dropping. En MPLS con DiffServ existen varias clases de servicios a la que pertenece cada paquete, se utiliza el soporte de MPLS para DiffServ, donde se redefine la

cabecera EXP de MPLS para la especificación de dicha clase de servicio. El campo EXP es de tres bits, por lo que cada paquete puede pertenecer a una de las 23 = 8 clases posibles. Sin embargo en DiffServ se definen de servicio (EF o Expedited Forwarding, BE o Best Effort y AF o Assured Forwarding) como se muestra a continuación:

**Tabla IV.2: Políticas en Servicios Diferenciados**

| CLASES | DIFFSERV | TRÁFICO | TIPO | EXP |
|---|---|---|---|---|
| ***Best Effort*** | AF11 | Aplicaciones que no reciben ninguna garantía de QoS | ICMP | 0 |
| ***Bronce*** | AF12 | Protocolos y aplicaciones para administrar la red | SNMP,TELNET | 1 |
| ***Plata*** | AF21-AF22 | Aplicaciones empresariales | Bases de datos, transacciones web | 2, 3 |
| ***Oro*** | AF31 | Videoconferencia y streaming | HTTP | 4 |
| ***Premium*** | AF32 | VoIP | TCP, UDP | 5 |

**Elaborado por: Ing**. Pamela Buñay
**Fuente:** Sandra Karina Narváez Pupiales.

MPLS se basa en el etiquetado de los paquetes en base a criterios de prioridad y/o calidad (QoS).MPLS con DiffServ clasifica el tráfico entrante en diferentes niveles de servicio para enseguida aplicarle un comportamiento agregado para todos los flujos de una determinada clase de servicio. Los servicios son:

**Tabla IV.3: Servicios Diferenciados**

| Servicio | Características |
|---|---|
| **'Expedited Forwarding' o 'Premium'** | •Es el que da más garantías. Equivale a una línea dedicada<br>•Garantiza Caudal, tasa de pérdidas, retardo y jitter<br>•Valor 101110 en DSCP |
| **'Assured Forwarding'** | •Asegura un trato preferente, pero sin fijar garantías (no hay SLA)<br>•Se definen cuatro clases y en cada una tres niveles de descarte de paquetes |
| **'Best Effort' con prioridad** | •Sin garantías, pero obtendrá trato preferente frente a 'best effort sin prioridad' |
| **'Best Effort' sin prioridad** | •Ninguna garantía, obtiene solo las migajas |

**Elaborado por: Ing**. Pamela Buñay

**Fuente:**Rogelio Montañana, Departamento de Informática Universidad de Valencia

**INDICADOR No.2. Mecanismos**

**Tabla IV.4: Mecanismos**

| Característica | Redes IP | Redes MPLS con DiffServ |
|---|---|---|
| **Multicapa** | No | Si |
| **Dotación de QoS** | No | Si |
| **Multiprotocolo** | No | Si |

**Elaborado por:** Ing. Pamela Buñay

**Fuente:**Janneth Espinosa, Liliana Enciso, Audrey Romero, Hugo Zamora

**Interpretación:**

Las redes IP no son multicapa, solo actúan en la capa de Red mientras que las Redes MPLS operan al nivel de enlace-red proporcionando un método de envío rápido por su conmutación de etiquetas y sus caminos LSP; DiffServ realiza la diferenciación y priorización del tráfico necesaria para dotar a IP de QoS. IP no asegura QoS debido a que hace uso del mecanismo Best Effort.

El utilizar el etiquetado por debajo de capa 3, permite que MPLS pueda funcionar independientemente del protocolo de capa 3 utilizado, de ahí lo de "multiprotocol" se puede usar: IP, ATM, Frame Relay. No sucede lo mismo con las redes IP puesto que se usa el protocolo IP.

**INDICADOR No.3. Servicios**

**Tabla IV.5: Servicios**

| Característica | Redes IP | Redes MPLS con DiffServ |
|---|---|---|
| **Circuitos Virtuales** | No | Si |
| **Conmutación de Paquetes** | Si | Si |
| **Orientado a la conexión** | No | Si |
| | | |

**Elaborado por: Ing**. Pamela Buñay

**Fuente:**Adrián Delfino, Sebastián Rivero, Pablo Belzarena, Rogelio Alvez

**Interpretación:**

Las redes basadas en IP utilizan la tecnología de conmutación de paquetes, que usa la capacidad disponible de una forma mucho más eficiente y que minimiza el riesgo de posibles problemas como la desconexión.

MPLS realiza la conmutación de los paquetes o datagramas en función de las etiquetas añadidas en capa 2 y etiqueta dichos paquetes según la clasificación establecida por la QoS. Para poder crear los circuitos virtuales como en ATM, utiliza etiquetas añadidas a los paquetes. Estas etiquetas definen el circuito virtual por toda la red, intentando conseguir las ventajas de ATM, pero sin sus inconvenientes. Puede ser utilizado para transportar diferentes tipos de tráfico, incluyendo tráfico de voz y de paquetes IP.

El protocolo de red IP no asegura que se pueda transmitir información en un orden o un tiempo determinado, sino que la red hará lo posible por transmitir la información de la manera más rápida y fiable posible. MPLS con DiffServ ofrece un servicio orientado a conexión porque mantiene un estado de la comunicación entre dos nodos y mantiene circuitos virtuales.

### 4.2.2 Análisis de la Variable Dependiente

**Variable dependiente:** Calidad de Servicio punto a punto en la transmisión de tráfico en tiempo real.

El tráfico que se analizará la variable dependiente es el de VoIP y Datos, pues son los más utilizados en las redes. Se propone utilizar los estándares de VoIP G.711, G.729 y G.723.1, pues están entre los más utilizados según lo especifica la ITU-T G (Series: Transmission systems and media, digital systems and networks).

**G.711:** es un estándar de la ITU-T para la compresión de audio. Este estándar es usado principalmente en telefonía, y fue liberado para su uso en el año 1972**.**Representa señales de audio con frecuencias de la voz humana, mediante muestras comprimidas de una señal de audio digital con una tasa de muestreo de 8000 muestras por segundo.

**G723.1:** es un estándar ITU-T de codec de voz de banda ancha. Esta es una extensión de acuerdo a la recomendación G.721 adaptiva del pulso diferencial del código de modulación de 24 y 40 kbit/s para equipos de aplicaciones de multiplicación de circuitos digitales.

**G.729:** es un algoritmo de compresión de datos de audio para voz que comprime audio de voz en trozos de 10 milisegundos. Se usa mayoritariamente en aplicaciones de Voz sobre IP VoIP por sus

bajos requerimientos en ancho de banda. El estándar G.729 opera a una tasa de bits de 8 kbit/s, pero existen extensiones, las cuales suministran también tasas de 6.4 kbit/s y de 11.8 kbit/s para peor o mejor calidad en la conversación respectivamente.

Para los datos se inyectará un paquete de tamaño 1028 bytes, pues los paquetes no son tan sensibles como los paquetes de VoIP.

Se utilizará D-ITG 2.61 (Anexo II) que es un generador de tráfico para mediciones. A continuación se muestra la configuración para VoIP y datos. Para el caso de VoIP debemos seleccionar los códec G.711, G.729 y G.723.1 e ir realizando las mediciones en los tres escenarios.

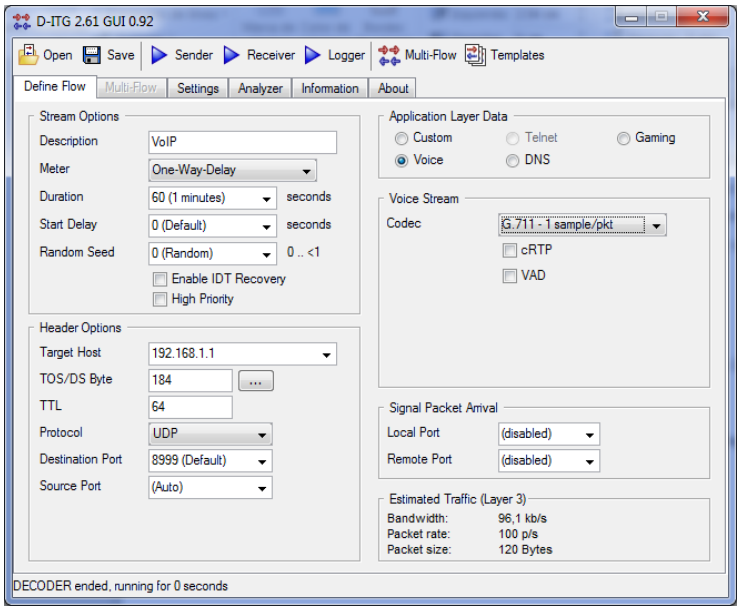



**Figura 4.1: Configuración para tráfico VoIP**

Configuración de datos como se muestra a continuación:

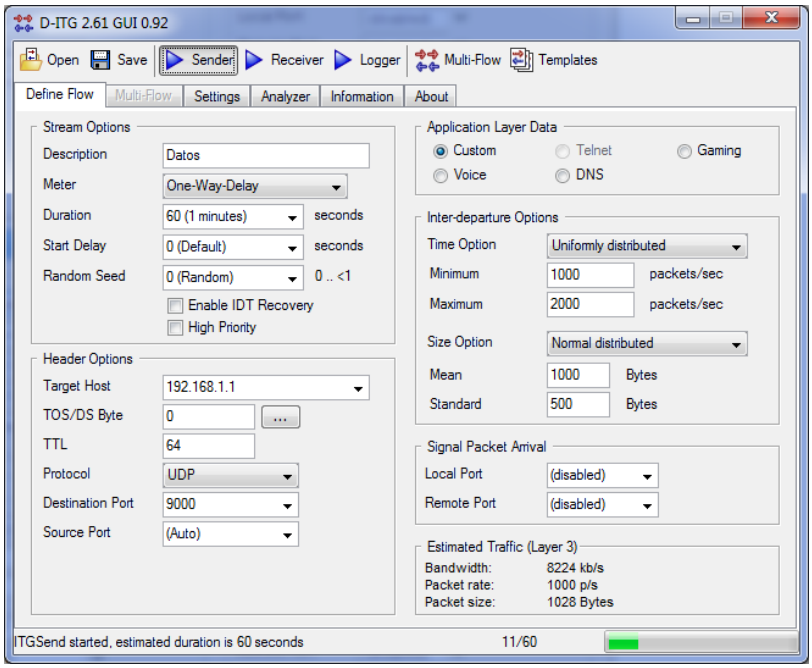



**Figura 4.2:Configuración para tráfico Datos**

### 4.2.2.1 Valoración de los Indicadores de la Variable Dependiente

**INDICADOR N. 1. Retardo**

Este es un parámetro que indica el retraso en la llegada de los flujos de datos a su destino en la red. Sus efectos son distintos de acuerdo a si el protocolo decapa transporte que se usa es TCP o UDP.

Tomando como referencia las recomendaciones UIT-T G.1010, Y.1541 y la IEEE 802.1p, se establecen los umbrales máximos de retardo:

**Tabla IV.6: Retardo**

| Clase de calidad de servicio | Descripción | Umbral de retardo internacional (ms) |
|---|---|---|
| 0 | Tiempo real, alta interacción, sensibles al retardo. (Voz y video en tiempo real) | 100 |
| 1 | Tiempo real, interactivos, sensibles al retardo (Voz y video en tiempo real de menor calidad) | 150 |
| 2 | Datos de alta prioridad (transaccionales, altamente interactivos) | 200 |
| 3 | Datos de mediana prioridad (Datos transaccionales interactivos) | 225 |
| 4 | Datos de baja prioridad (transacciones cortas, datos en grandes cantidades, flujo continuo de video *streaming*) | 250 |
| 5 | Datos de mejor esfuerzo | 300 |

**Fuente:**UIT-T G.1010, Y.1541 y la IEEE 802.1p

En base a los valores anteriores y como aporte personal, se obtiene la siguiente tabla:

**Tabla IV.7: Valoración cualitativa de índices del indicador 1 variable dependiente**

| Tráfico | Adecuado | Conforme | No Adecuado |
|---|---|---|---|
| VoIP | <100ms | >100ms y <150ms | >150ms |
| Datos | <250ms | >250ms y <300ms | >300ms |

**Elaborado por: Ing**. Pamela Buñay

**Tabla IV.8:Datos tomados en escenarios de prueba para indicador 1 de la variable dependiente**

| Tráfico | Redes IP | MPLS | MPLS y DiffServ |
|---|---|---|---|
| G.711 | 206.553 ms | 85.1053ms | 53.02202ms |
| G723.1 | 307.971ms | 90.1401ms | 62.98705ms |
| G.729 | 388.938ms | 95.0146ms | 62.98701ms |
| Datos | 395.892ms | 229.861ms | 105.3307ms |

**Elaborado por:** Ing. Pamela Buñay

**Tabla IV.9:Retardo**

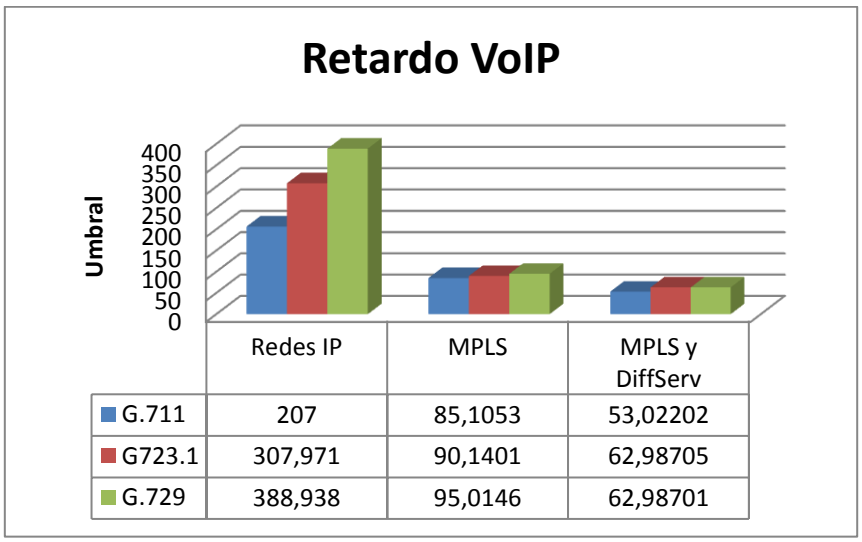
| Tráfico | Redes IP | MPLS | MPLS y DiffServ |
|---|---|---|---|
| G.711 | No Adecuado | Adecuado | Adecuado |
| G723.1 | No Adecuado | Adecuado | Adecuado |
| G.729 | No Adecuado | Adecuado | Adecuado |
| Datos | No Adecuado | Adecuado | Adecuado |

**Elaborado por:** Ing. Pamela Buñay

**Interpretación**

Los paquetes de VoIP deben recibir un trato especial ya que es muy sensible a retardos y necesita un ancho de banda garantizado. Entonces es necesario ofrecer cierta QoS de manera de poder disminuir el retardo.

Los datos tomados en los tres escenarios para VoIP, se ve que hay diferentes retardos para Redes IP, redes MPLS y MPLS con DiffServ.




| | Redes IP | MPLS | MPLS y DiffServ |
|---|---|---|---|
| G.711 | 207 | 85,1053 | 53,02202 |
| G723.1 | 307,971 | 90,1401 | 62,98705 |
| G.729 | 388,938 | 95,0146 | 62,98701 |

**Figura 4.3: Valor de retardo en VoIP**

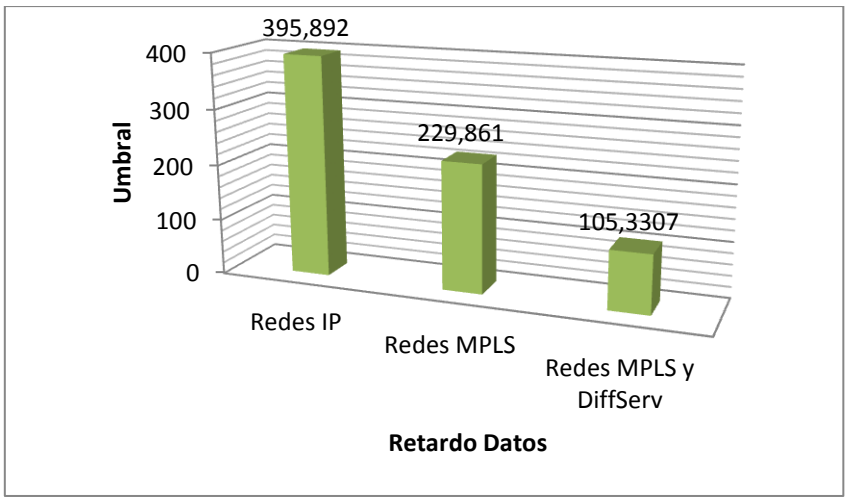



**Figura 4.4: Valor de retardo en datos**

En redes MPLS el retardo baja notablemente tanto en VoIP como Datos, debido a que se tiene una conmutación por etiquetas caminos cortos en vez de direcciones. Asigna a los datagramas de cada

flujo una etiqueta única que permite una conmutación rápida en los routers intermedios (solo se mira la etiqueta, no la dirección de destino).

Con MPLS y DiffServ se mejora más aun, puesto que los paquetes tanto de VoIP como Datos tienen un trato diferenciado como alta prioridad la VoIP y de menor prioridad los datos. Los tiempos bajan considerablemente en ambos casos, puesto que también se toma en cuenta la rapidez de MPLS en donde se utilizan etiquetas. Con esto se tiene como resultado que en Redes MPLS y la combinación de Redes MPLS y DiffServ es Adecuado el retardo mientras que en IP no se tiene ninguna garantía de Calidad de Servicio debido a que el retardo es un indicador de Calidad.

Sacando el promedio de retardos de VoIP para los tres escenarios se tiene: Redes IP 301.151ms, Redes MPLS 90.087 ms, MPLS y DiffServ59.665ms. Con estos resultados se nota que en Redes IP se tiene un alto retardo que sufren los paquetes tanto VoIP como Datos como consecuencia al trato no diferenciado.

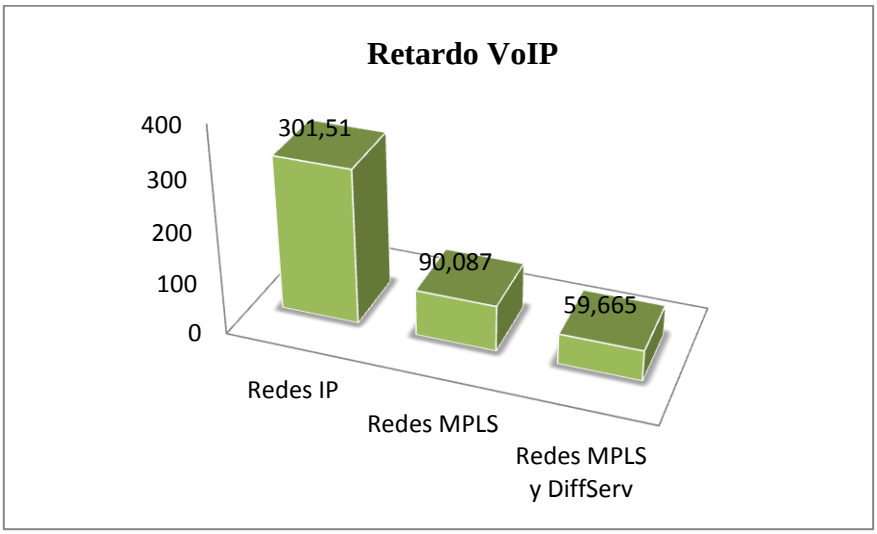



**Figura 4.5: Valor Promedio de Retardo en VoIP**

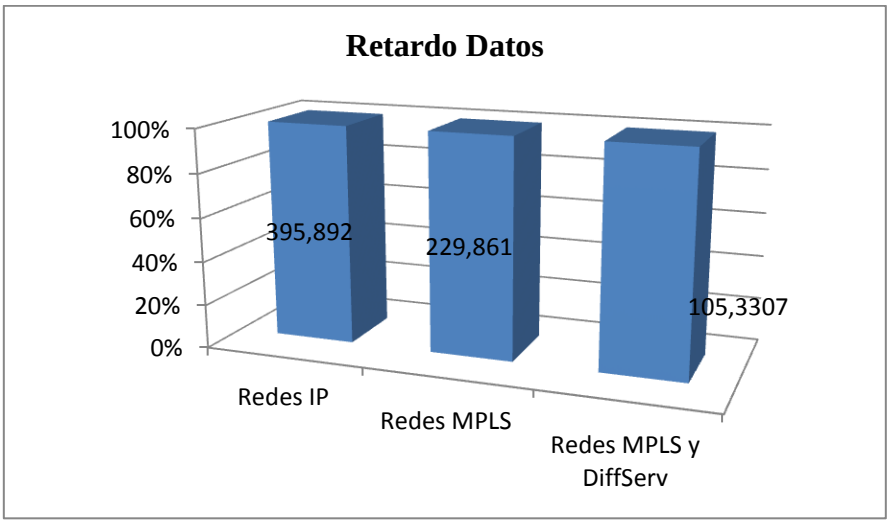



**Figura 4.6: Valor Promedio de Retardo en Datos**

**INDICADOR N. 2. Jitter**

Desgraciadamente el retardo que se produce en los flujos de datos no es constante, y esta variación en el retardo provoca un nuevo problema al que se conoce como jitter. Este puede aparecer por congestión en la red, pero principalmente debido a una incorrecta sincronización de bit entre los elementos de red, y se presenta como un estrechamiento y alargamiento del ancho de los pulsos en el receptor.

Tomando como referencia las recomendaciones UIT-T G.1010, Y.1541 y la IEEE 802.1p, se establecen los umbrales máximos de Jitter:

**Tabla IV.10:Jitter**

| Clase de calidad de servicio | Descripción | Umbral de jitter internacional (ms) |
|---|---|---|
| 0 | Tiempo real, alta interacción, sensibles al retardo. (Voz y video en tiempo real) | 45 |
| 1 | Tiempo real, interactivos, sensibles al retardo (Voz y video en tiempo real de menor calidad) | 50 |
| 2 | Datos de alta prioridad (transaccionales, altamente interactivos) | 55 |
| 3 | Datos de mediana prioridad (Datos transaccionales interactivos) | N/A |
| 4 | Datos de baja prioridad (transacciones cortas, datos en grandes cantidades, flujo continuo de video *streaming*) | N/A |
| 5 | Datos de mejor esfuerzo | N/A |

**Fuente:**UIT-T G.1010, Y.1541 y la IEEE 802.1p

Con referencia a la tabla anterior y como aporte personal se obtiene la siguiente tabla:

**Tabla IV.11: Valoración cualitativa de índices del indicador 2 variable dependiente**

| Tráfico | Adecuado | Conforme | No Adecuado |
|---|---|---|---|
| VoIP | <40ms | >40ms y <50ms | >50ms |
| Datos | <55ms | >55ms y 70<ms | >70ms |

**Elaborado por: Ing**. Pamela Buñay

**Tabla IV.12: Datos tomados en escenarios de prueba para indicador 2 de la variable dependiente**

| Tráfico | Redes IP | MPLS | MPLS y DiffServ |
|---|---|---|---|
| G.711 | 48.1443ms | 31.6313ms | 20.0706ms |
| G723.1 | 54.2689ms | 36.2561ms | 21.2723ms |
| G.729 | 65.8277ms | 37.9215ms | 21.8723ms |
| Datos | 79.5689ms | 45.2658ms | 30.0706ms |

**Elaborado por: Ing**. Pamela Buñay

**Tabla IV.13:Jitter**

| Tráfico | Redes IP | MPLS | MPLS y DiffServ |
|---|---|---|---|
| G.711 | No Adecuado | Adecuado | Adecuado |
| G723.1 | No Adecuado | Adecuado | Adecuado |
| G.729 | No Adecuado | Adecuado | Adecuado |
| Datos | No Adecuado | Adecuado | Adecuado |

**Elaborado por: Ing**. Pamela Buñay

**Interpretación**

Las comunicaciones en tiempo real (como VoIP) son especialmente sensibles a este efecto. En general, es un problema frecuente en enlaces lentos o congestionados.
La solución más adecuada consiste básicamente en asignar una pequeña cola o almacén para ir recibiendo los paquetes y sirviéndolos con un pequeño retraso. Si algún paquete no está (se perdió o no ha llegado todavía) cuando sea necesario se descarta.

Como se puede apreciar en la figura el jitter en Redes IP son altos con relación a MPLS, MPLs y DiffServ. Esto se debe generalmente porque los paquetes de voz son descartados por el receptor

cuando este no recibe los paquetes a tiempo, en la práctica los usuarios perciben este problema como un entrecortado en la voz.

En MPLS baja el jitter porque gracias al label switching, técnica usada en MPLS para enrutar paquetes, conseguimos hacer el enrutado a más velocidad, a la vez que disminuimos el retardo y el jitter.

En redes MPLS y Diffserv tiene una notoria reducción porque a más de tener más velocidad se tiene alta prioridad por los paquetes VoIP, al crear las políticas tanto de entrada como salida en las interfaces de los routers y al ser considerado como un servicio Premium en donde el jitter, retardo y pérdida de paquetes son garantizadas.

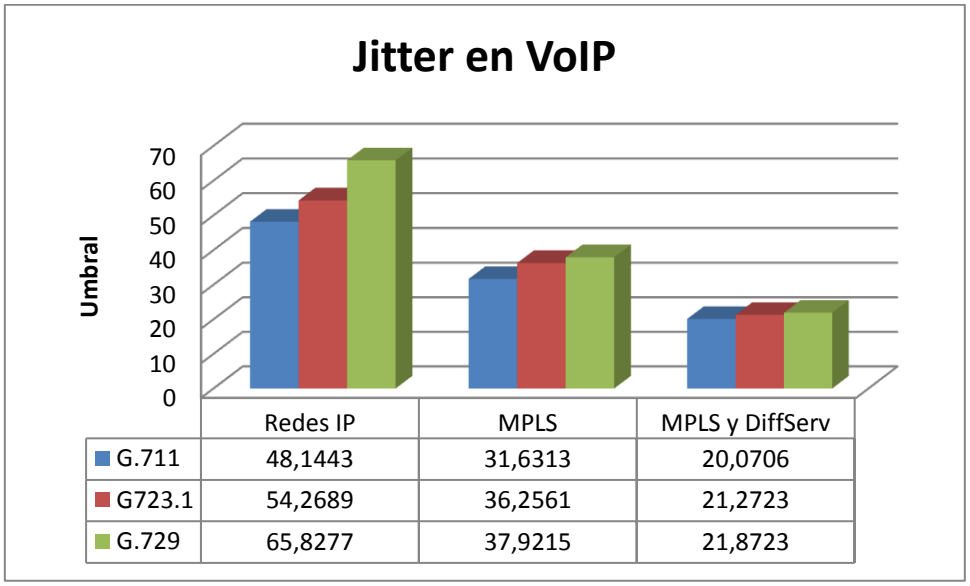



**Figura 4.7: Jitter en VoIP**

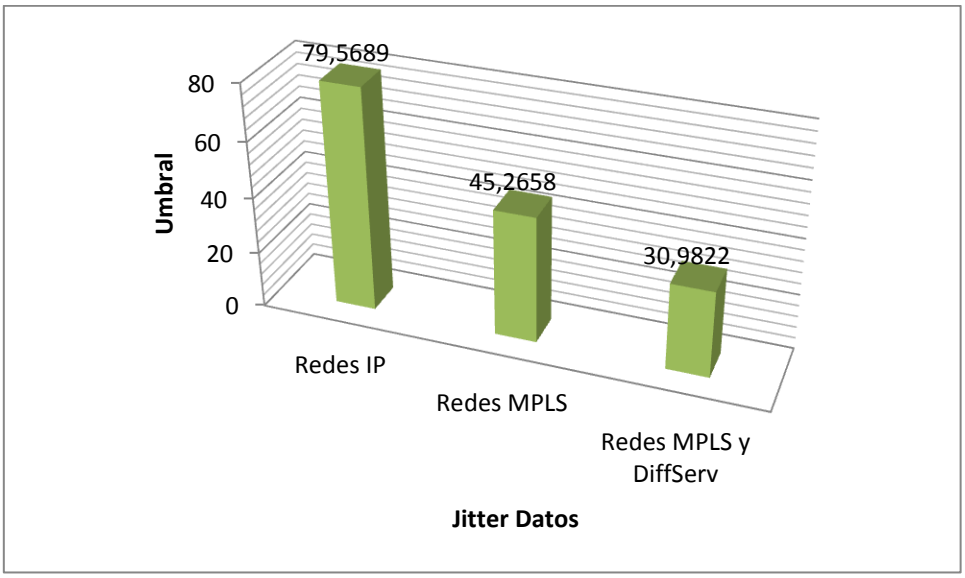



**Figura 4.8: Jitter en Datos**

Para los tres escenarios Redes IP, MPLS, MPLS y DiffServ se tiene un promedio para VoIP de: 56.0803ms, 35.2696ms, 11.0717 ms respectivamente. Para datos se tiene 79.5689ms para Redes IP, 45.2658ms para MPLS y 20.9822ms para MPLS y DiffServ

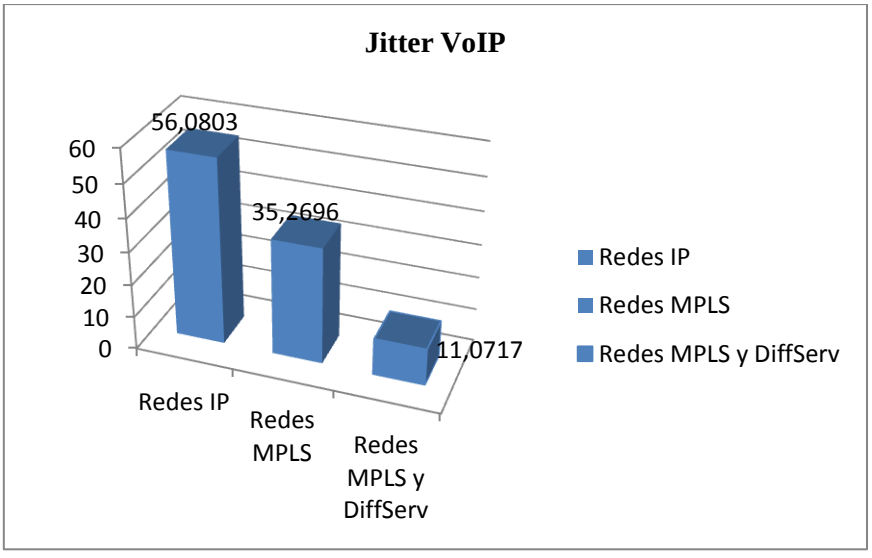



**Figura 4.9:Valor Promedio de Jitter en VoIP**

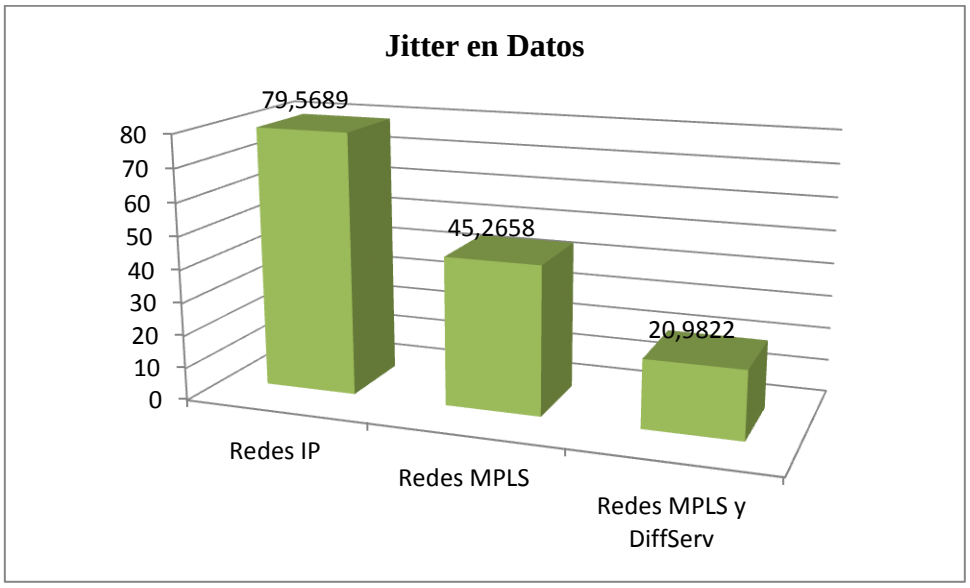



**Figura 4.10: Valor Promedio de Jitter en Datos**

**INDICADOR N. 3. Pérdida de paquetes**

Las comunicaciones en tiempo real están basadas en el protocolo UDP. Este protocolo no está orientado a conexión y si se produce una pérdida de paquetes no se reenvían. Además la pérdida de paquetes también se produce por descartes de paquetes que no llegan a tiempo al receptor.Tomando como referencia las recomendaciones UIT-T G.1010, Y.1541 y la IEEE 802.1p, se establecen los umbrales máximos de Pérdida de paquetes:

**Tabla IV.14:Pérdida de paquetes**

| Clase de calidad de servicio | Descripción | Umbral de pérdida de paquetes local (%) |
|---|---|---|
| 0 | Tiempo real, alta interacción, sensibles al retardo. (Voz y video en tiempo real) | 1% |
| 1 | Tiempo real, interactivos, sensibles al retardo (Voz y video en tiempo real de menor calidad) | 3% |
| 2 | Datos de alta prioridad (transaccionales, altamente interactivos) | 3% |
| 3 | Datos de mediana prioridad (Datos transaccionales interactivos) | 5% |
| 4 | Datos de baja prioridad (transacciones cortas, datos en grandes cantidades, flujo continuo de video *streaming*) | 5% |
| 5 | Datos de mejor esfuerzo | 5% |

**Fuente:** UIT-T G.1010, Y.1541 y la IEEE 802.1p

Con referencia a la tabla anterior y como aporte personal se obtiene la siguiente tabla:

**Tabla IV.15: Valoración cualitativa de índices del indicador 3 variable dependiente**

| Tráfico | Adecuado | Conforme | No Adecuado |
|---|---|---|---|
| VoIP | <1% | >1% y <3% | >3% |
| Datos | <3% | >3% y <5% | >5% |

**Elaborado por: Ing**. Pamela Buñay

**Tabla IV.16: Datos tomados en escenarios de prueba para indicador 3 de la variable dependiente**

| Tráfico | Redes IP | MPLS | MPLS y DiffServ |
|---|---|---|---|
| G.711 | 3.892% | 1.281% | 0.515% |
| G723.1 | 4.186% | 1.877% | 0.6179% |
| G.729 | 4.554% | 2.091% | 0.7179% |
| Datos | 9.898% | 4.653% | 2.698% |

**Elaborado por:** Ing. Pamela Buñay

**Tabla IV.17:Pérdida de paquetes**

| Tráfico | Redes IP | MPLS | MPLS y DiffServ |
|---|---|---|---|
| G.711 | No Adecuado | Conforme | Adecuado |
| G723.1 | No Adecuado | Conforme | Adecuado |
| G.729 | No Adecuado | Conforme | Adecuado |
| Datos | No Adecuado | Conforme | Adecuado |

**Elaborado por: Ing**. Pamela Buñay

**Interpretación**

Como se puede visualizar existe más paquetes perdidos en las redes IP, esto se debe a que no se tiene ninguna garantía en el envío y recepción de paquetes y debidos al encolamiento que el tráfico experimenta mientras transita la red. Con las redes MPLS baja considerablemente el porcentaje de pérdida, esto es porque tiene el mecanismo de creación de circuito virtual entre los routers (orientado a la conexión).

En MPLS y DiffServ se observa como disminuye los paquetes perdidos debido a que el etiquetado de los paquetes se realiza en base a criterios de prioridad y/o calidad (QoS).Una manera de

disminuir la cantidad de paquetes perdidos sin disminuir el retardo de VoIP es la de mantener las prioridades.

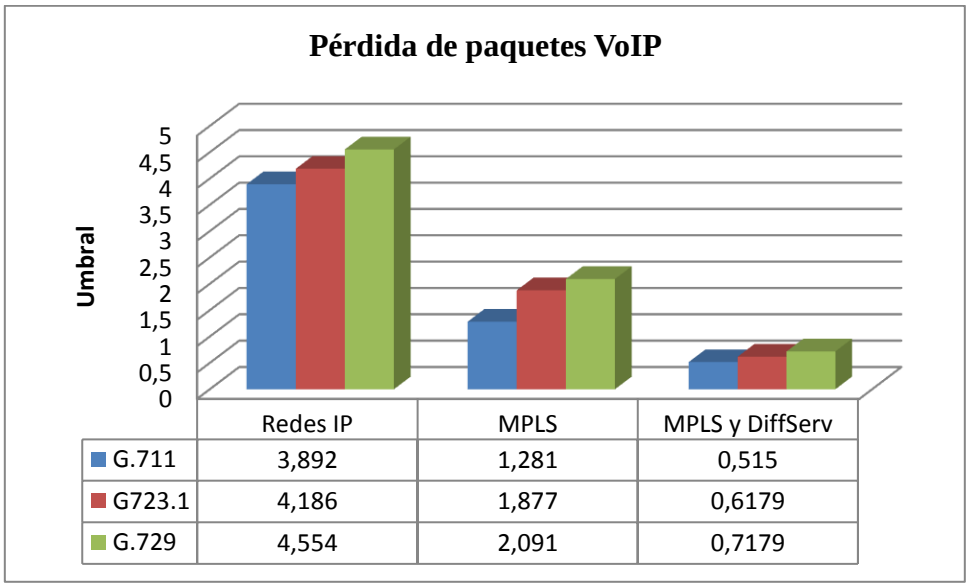



**Figura 4.11: Paquetes perdidos en VoIP**

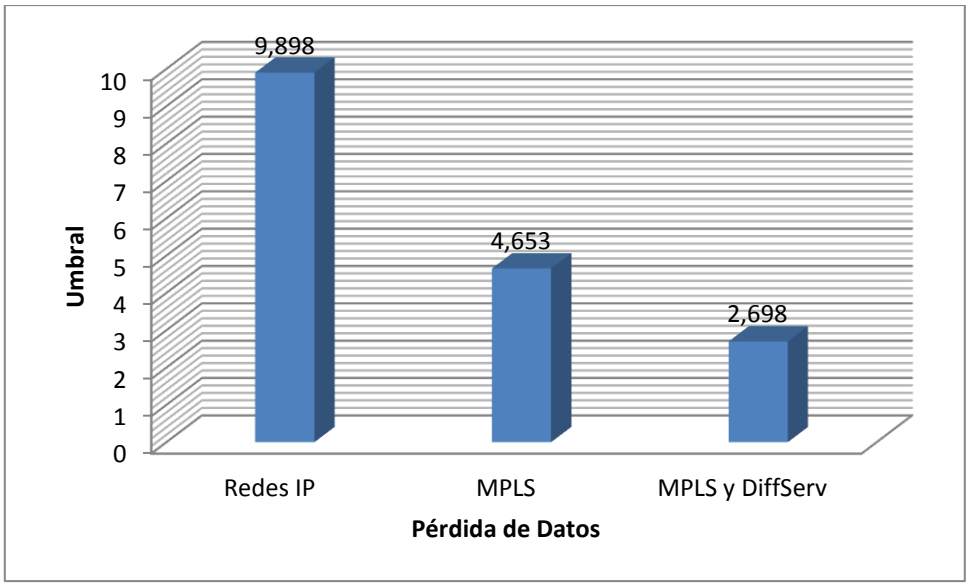



**Figura 4.12: Paquetes perdidos en datos**

Sacando un promedio de los tres escenarios redes IP, MPLS, MPLS y Diffserv se tiene: 4.2107%, 1.7497% y 0.6169% respectivamente.

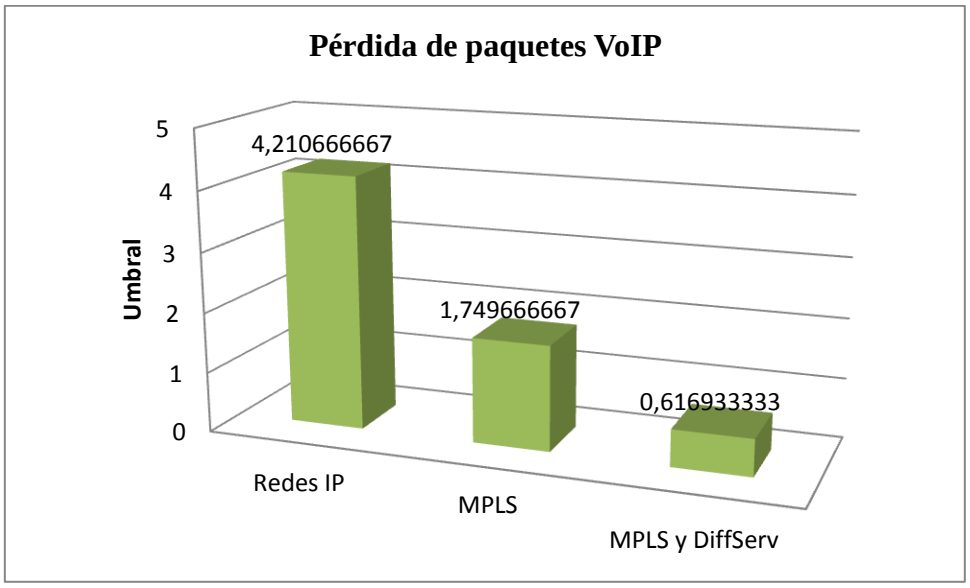



**Figura 4.13: Valor promedio de paquetes perdidos en VoIP**

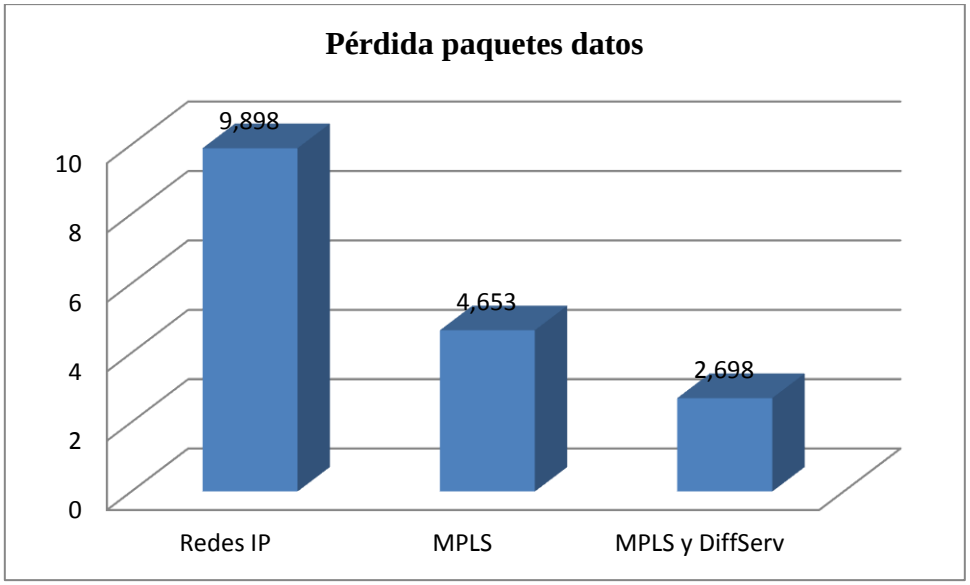



**Figura 4.14: Valor promedio de paquetes perdidos en datos**

### 4.2.3 Resumen de la valoración de los indicadores

**Tabla IV.18:Resultados cualitativos variable dependiente**

| Indicador | Redes IP | MPLS | MPLS y DiffServ |
|---|---|---|---|

| | | | |
|---|---|---|---|
| Retardo | No Adecuado | Adecuado | Adecuado |
| Jitter | No Adecuado | Adecuado | Adecuado |
| Pérdidas de paquetes | No Adecuado | Conforme | Adecuado |

**Elaborado por:** Ing. Pamela Buñay

Los valores cuantitativos que se les asignarán a los valores obtenidos en el análisis de la variable dependiente son:

**Tabla IV.19:Valores Cualitativos y cuantitativos**

| Valor Cualitativo | Valor Cuantitativo |
|---|---|
| Adecuado | 3 |
| Conforme | 2 |
| No adecuado | 1 |

**Elaborado por:** Ing. Pamela Buñay

**Tabla IV.20:Resultados cualitativos variable dependiente**

| V.I / V.D | Redes IP | MPLS | MPLS y DiffServ |
|---|---|---|---|
| **Retardo** | 1 | 3 | 3 |
| **Jitter** | 1 | 3 | 3 |
| **Pérdidas de paquetes** | 1 | 2 | 3 |
| | **3/9** | **8/9** | **9/9** |

**Elaborado por:** Ing. Pamela Buñay

### 4.2.4 Comprobación de la Hipótesis de la Investigación

**Hi:** La aplicación de la arquitectura ¨ Servicios Diferenciados ¨ sobre redes MPLS proveerá una adecuada Calidad de Servicio punto a punto en la transmisión de tráfico en tiempo real.
**Ho:** La aplicación de la arquitectura ¨ Servicios Diferenciados ¨ sobre redes MPLS no proveerá una adecuada Calidad de Servicio punto a punto en la transmisión de tráfico en tiempo real.

**Tabla IV.21:Presentación de resultados**

| V.I | Indicador | Redes IP | MPLS | MPLS y |
|---|---|---|---|---|

| V.D | | | | DiffServ |
|---|---|---|---|---|
| Proveerá una adecuada Calidad de Servicio punto a punto en la transmisión de tráfico en tiempo real. | Retardo | 0 | 3 | 3 |
| | Jitter | 0 | 3 | 3 |
| | Pérdidas de paquetes | 0 | 2 | 3 |
| | **Total indicadores** | 0 | 8 | 9 |
| No proveerá una adecuada Calidad de Servicio punto a punto en la transmisión de tráfico en tiempo real. | Retardo | 1 | 0 | 0 |
| | Jitter | 1 | 0 | 0 |
| | Pérdidas de paquetes | 1 | 0 | 0 |
| | **Total indicadores** | 1 | 0 | 0 |

**Elaborado por:** Ing. Pamela Buñay

La tabla de contingencia creada para el cálculo de la ji cuadrada, la variable dependiente con sus indicadores. En la Tabla IV.22 se anotan las frecuencias observadas en la investigación.

**Tabla IV.22:Tabla de contingencia con las frecuencias observadas**

| Redes IP | Redes IP | MPLS | MPLS y DiffServ | |
|---|---|---|---|---|
| Proveerá una adecuada Calidad de Servicio punto a punto en la transmisión de tráfico en tiempo real | 0 | 8 | 9 | 17 |
| No proveerá una adecuada Calidad de Servicio punto a punto en la transmisión de tráfico en tiempo real. | 1 | 0 | 0 | 1 |
| **Total** | 1 | 8 | 9 | **18** |

**Elaborado por:** Ing. Pamela Buñay

La Tabla IV.23 contiene las frecuencias esperadas, la cual constituye los valores que esperaríamos encontrar si las variables no estuvieran relacionadas. La chi cuadrada partirá del supuesto de “no relación entre las variables” y se evaluará si es cierto o no, analizando si las frecuencias observadas son diferentes de lo que pudiera esperarse en caso de ausencia de correlación.
La frecuencia esperada de cada celda, se calcula mediante la siguiente fórmula aplicada a la tabla de frecuencias observadas.

$$fe = \frac{(total\_de\_fila)(total\_de\_columna)}{N}$$

Donde ***N*** es el número total de frecuencias observadas

Para la primera celda la frecuencia esperada sería:

$$fe = \frac{(17)(1)}{18} = 0.944$$

**Tabla IV.23:Tabla de frecuencias esperadas.**

| | **Redes IP** | **MPLS** | **MPLS y DiffServ** | **Total** |
|---|---|---|---|---|
| Proveerá una adecuada Calidad de Servicio punto a punto en la transmisión de tráfico en tiempo real | 0.9444 | 7.5556 | 8.5 | 17 |
| No proveerá una adecuada Calidad de Servicio punto a punto en la transmisión de tráfico en tiempo real. | 0.0556 | 0.4444 | 0.5 | 1 |
| **Total** | 1 | 8 | 9 | **18** |

**Elaborado por:** Ing. Pamela Buñay

Una vez obtenidas las frecuencias esperadas, se aplica la siguiente fórmula de ji cuadrada:

$$\chi^2 = \sum \frac{(O-E)^2}{E}$$

Dónde:

***O*** es la frecuencia observada en cada celda

***E*** es la frecuencia esperada en cada celda

**Tabla IV.24:Cálculo de χ2**

| Celda | Observadas | Esperadas | O-E | $(O-E)^2$ | $(O-E)^2/E$ |
|---|---|---|---|---|---|
| Proveerá una adecuada Calidad de Servicio punto a punto en la transmisión de tráfico en tiempo real con redes IP | 0 | 0.9444 | -0.9444 | 0.89189136 | 0.9444 |
| Proveerá una adecuada Calidad de Servicio punto a punto en la transmisión de tráfico en tiempo real con MPLS | 8 | 7.5556 | 0.4444 | 0.19749136 | 0.02614 |
| Proveerá una adecuada Calidad de Servicio punto a punto en la transmisión de tráfico en tiempo real con MPLS y DiffServ | 9 | 8.5 | 0.5 | 0.25 | 0.02941 |
| No proveerá una adecuada Calidad de Servicio punto a punto en la transmisión de tráfico en tiempo real con redes IP | 1 | 0.0556 | 0.9444 | 0.83629136 | 15.0412 |
| No proveerá adecuada Calidad de Servicio punto a punto en la transmisión de tráfico en tiempo real con MPLS | 0 | 0.4444 | -0.444 | 0.197136 | 0.4444 |
| No proveerá adecuada Calidad de Servicio punto a punto en la transmisión de tráfico en tiempo real con MPLS y DiffServ | 0 | 0.5 | -0.5 | 0.25 | 0.5 |
| | | | | **TOTAL DE LA TABLA X2 =** | **16.98555** |

**Elaborado por:** Ing. Pamela Buñay

**INTERPRETACIÓN:**

Para saber si el valor de $\chi^2$ es o no significativo, se debe determinar los grados de libertad mediante la siguiente fórmula:

$$Gl = (r-1)(c-1)$$

Dónde:

r es el número de filas de la tabla de contingencia

c es el número de columnas de la tabla de contingencia

Por lo tanto:

$$Gl = (3-1)(2-1) = 2$$

De la tabla de distribución del χ2 que se encuentra en el Anexo III y eligiendo como nivel de confianza α = 0,05 se obtiene: χ2 = 5.99. El valor de χ2 calculado en esta investigación es de **16.98555**que es muy superior al de la tabla de distribución; por lo que el valor χ2 está en la zona de rechazo de la hipótesis Nula por lo cual se acepta la hipótesis de investigación.

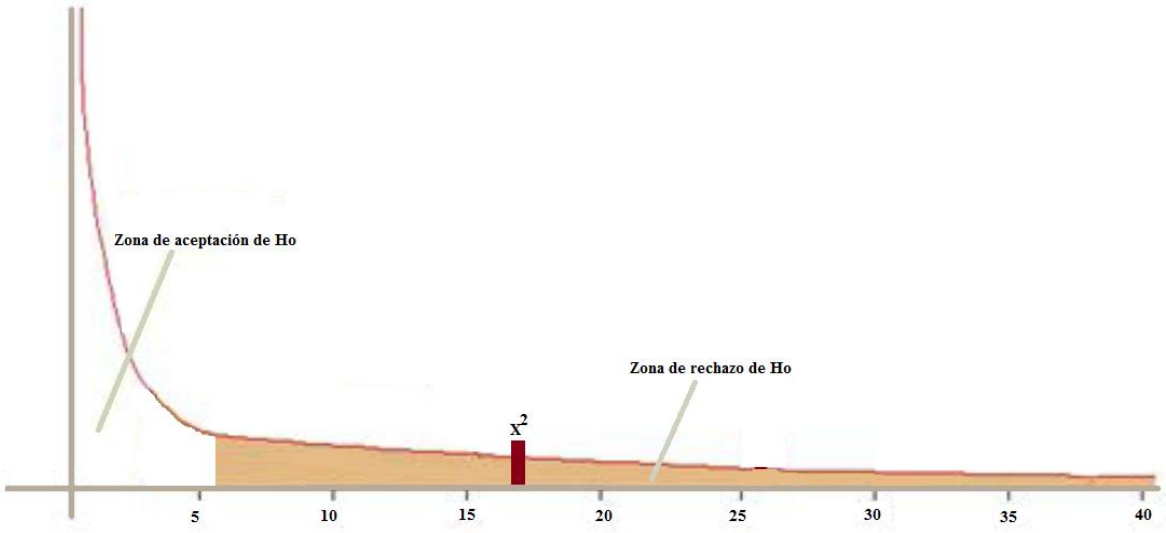


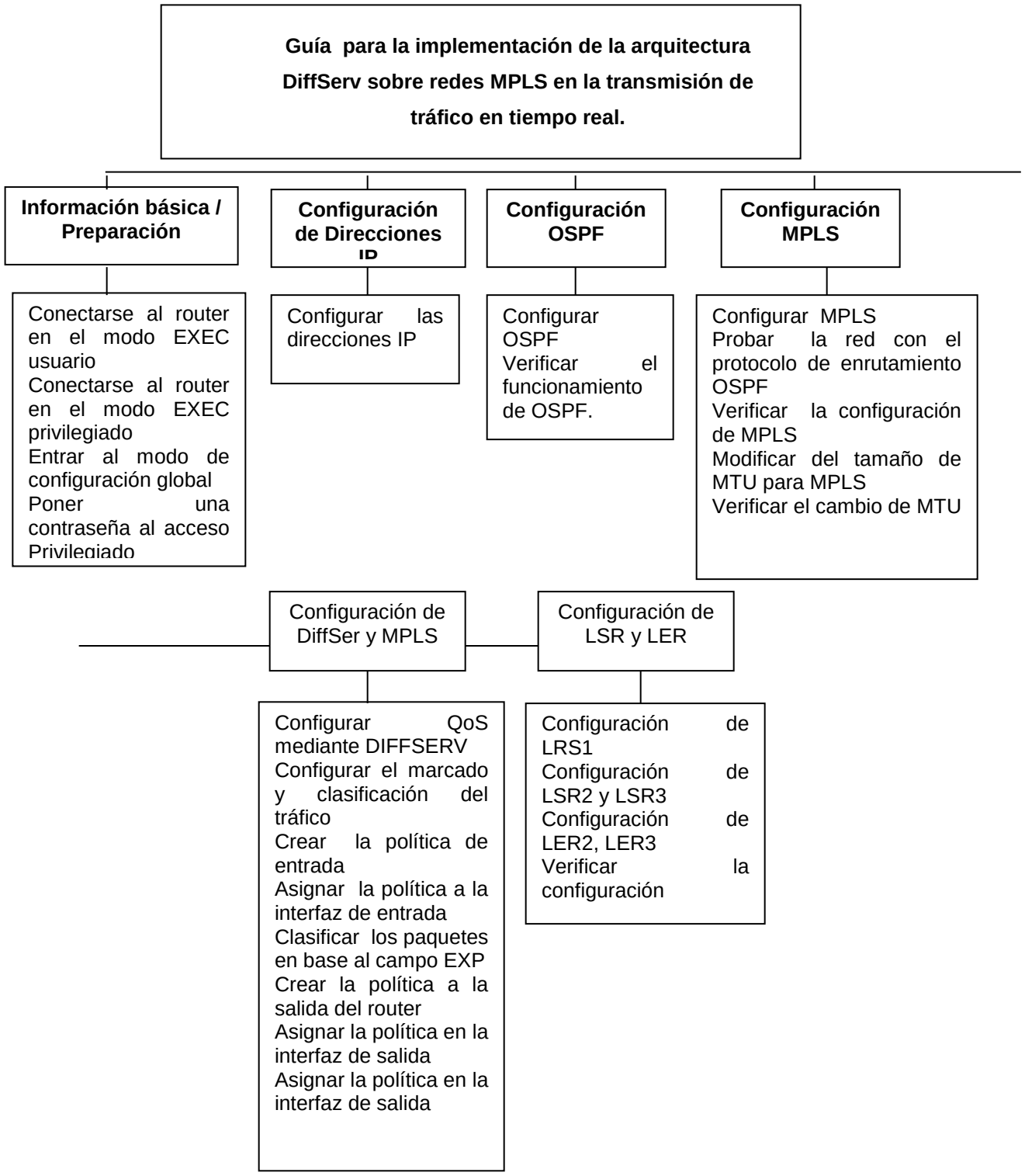

**Figura 4.15: Gráfica de la Función χ2**

### 4.2.5 Guía de usuario

**Guía para la implementación de la arquitectura DiffServ sobre redes MPLS en la transmisión de tráfico en tiempo real.**

Se describe la guía referencial para la implementación de la arquitectura DiffServ sobre redes MPLS en la transmisión de tráfico en tiempo real.

**Esquema de la Guía de usuario**



## Información básica / Preparación

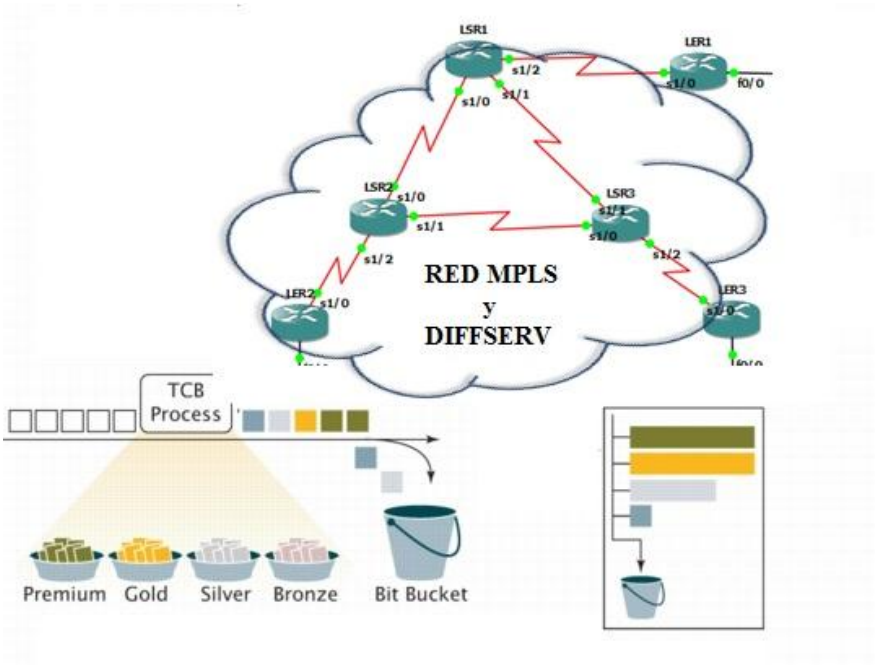



**Figura 4.16:Escenario**

Este sistema puede ser administrado en línea de comandos, propios a los equipos de Cisco Systems.

Los diferentes modos de usuarios:

- **Modo usuario:** Permite consultar toda la información relacionada al router sin poder modificarla. El shell es el siguiente:

  Router >

- **Usuario privilegiado:** Permite visualizar el estado del router e importar o exportar imágenes de IOS. El shell es el siguiente:

  Router #

- **Modo de configuración global:** Permite utilizar los comandos de configuración generales del router. El shell es el siguiente:

  Router (config) #

- **Modo de configuración de interfaces:** Permite utilizar comandos de configuración de interfaces (Direcciones IP, máscaras, etc.). El shell es el siguiente:

  Router (config-if) #

- **Modo de configuración de línea:** Permite configurar una línea (ejemplo: acceso al router por Telnet). El shell es el siguiente:

  Router (config-line) #

- **Modo espacial:** RXBoot Modo de mantenimiento que puede servir, especialmente, para reinicializar las contraseñas del router. El shell es el siguiente:

  rommon >

**Paso 1 Conectarse al router en el modo EXEC usuario**

Router >

**Paso 2 Conectarse al router en el modo EXEC privilegiado**

Router>enable

**Paso 3 Entrar al modo de configuración global**

Router#configure terminal

**Paso 4 Poner una contraseña al acceso Privilegiado**

Router (config) #
Router (config) # enable password contraseña

La próxima vez que un usuario intente conectarse en modo usuario privilegiado, le será solicitada una contraseña.

**Paso 5 Guardar la configuración Del Router**

Se recomienda guardar regularmente la configuración utilizando el siguiente comando (en modo privilegiado)
Router#copy running-config startup-config

**Paso 6 Asignar un nombre al router**

Router > enable
Router # configure terminal
Router (config) # hostname LSR1

**Paso 7 Configurar la contraseña de consola en el router**
LSR1 (config) # line con 0
LSR1 (config - con) # password
LSR1 (config - con) # <contraseña de la consola> //Definimos el password de la consola
LSR1 (config - con) # login
LSR1 (config - con) # exit

**Paso 8 Configurar el password de enable**
LSR1 (config) # enable secret <contraseña enable> //Habilitamos el password del enable
LSR1 (config) # config-register 0x2102

**Paso 9 Guardar la configuración Del Router**
Pulsamos Ctrl+z para salir del modo configuración
LSR1 # write memory // Guardamos los cambios

## Configuración dirección IP

### Paso 10 Configurar las direcciones IP

Configurar las interfaces de todos los routers con las direcciones IP, tanto las interfaces físicas (Ethernet y serie) como las de *Loopback*. En las interfaces serie además se añade el comando "*clock rate*" y habilitar con el comando "**no shutdown".**

LSR1 (config) # interface s1/0 // Accedemos al interfaz s1/0
LSR1 (config-if) # ip address 192.168.2.2 255.255.255.0 //Le asignamos una IP y mascara de subred

LSR1(config-if)# clockrate 64000 //Asignamos el clock rate
LSR1 (config-if) # no shutdown // Habilitamos el interfaz
LSR1 (config) # interface s1/1 // Accedemos al interfaz s1/1
LSR1 (config-if) # ip address 192.168.4.1 255.255.255.0 //Le asignamos una IP y mascara de subred

LSR1(config-if)# clockrate 64000 //Asignamos el clock rate
LSR1 (config-if) # no shutdown // Habilitamos el interfaz
LSR1 (config) # interface s1/2 // Accedemos al interfaz s1/2
LSR1 (config-if) # ip address 192.168.3.1 255.255.255.0 //Le asignamos una IP y mascara de subred

LSR1 (config-if) # no shutdown // Habilitamos el interfaz

Las mismas configuraciones para los otros routers con las siguientes direcciones:

**Tabla IV.25: Interfaz y direcciones IP de los routers**

| Router | Interfaz y dirección IP |
|---|---|
| **LSR2** | S1/0: 192.168.2.1<br>S1/1: 192.168.6.1<br>S1/2: 192.168.1.2 |
| **LSR3** | S1/0: 192.168.6.2<br>S1/1: 192.168.4.2<br>S1/2: 192.168.5.1 |
| **LER1** | S1/0: 192.168.3.2<br>F0/0: 192.168.7.1 |
| **LER2** | S1/0: 192.168.1.1<br>F0/0: 192.168.9.1 |
| **LER3** | S1/0: 192.168.5.2<br>F0/0: 192.168.8.1 |

**Elaborado por: Ing**. Pamela Buñay

**Configuración OSPF**

**Paso 11 Configurar OSPF**

Para configurar el algoritmo de encaminamiento OSPF en un área (por ejemplo el área 0), los pasos a seguir son los siguientes:

Usamos el comando “router ospf process-id” para crear un proceso OSPF en el router. “processid” es un identificador del proceso OSPF para el caso de que haya múltiples procesos OSPF ejecutándose en el router y es un número escogido por el administrador del sistema. Para indicar las redes que se deben anunciar se usa el comando “network NetID WildcardMask area area-id”. El comando “network” indica las interfaces que van a enviar o procesar mensajes de encaminamiento.

```
LSR1(config)# router ospf < identificador del proceso OSPF >
LSR1(config)# router ospf 1
LSR1(config-router)# network <dirección IP>< wildcard-mask> area <area-id>
LSR1(config-router)# network 192.168.2.0 0.0.0.255 area 0
LSR1(config-router)# network 192.168.3.0 0.0.0.255 area 0
LSR1(config-router)# network 192.168.4.0 0.0.0.255 area 0
```

Para configurar los demás router considerar lo siguiente:

**Tabla IV.26: Redes para protocolo OSPF**

| Router | Red |
|---|---|
| **LSR2** | 192.168.2.0<br>192.168.6.0<br>192.168.1.0 |
| **LSR3** | 192.168.6.0<br>192.168.4.0<br>192.168.5.0 |
| **LER1** | 192.168.3.0<br>192.168.7.0 |
| **LER2** | 192.168.1.0<br>192.168.9.0 |
| **LER3** | 192.168.5.0<br>192.168.8.0 |

**Elaborado por: Ing**. Pamela Buñay

**Paso 12 Verificar el funcionamiento de OSPF.**

Usar los siguientes comandos:

**show ip protocols**

Permite ver que protocolos de encaminamiento hay activos listando parámetros tales como temporizadotes, métricas, filtros, etc

**show ip route**

Permite ver la tabla de encaminamiento

**show ip route ospf**

Permite ver la tabla de encaminamiento solo para entradas OSPF

**show ip ospf interface**

Lista información relacionada con una interfaz que usa OSPF. Permite comprobar si las interfaz pertenecen al área a la que se suponen deberían pertenecer. Tambien permite averiguar si una interfaz es DR, BDR o DROTHER (no es ni DR ni BDR), su prioridad y si la red es de tipo BMA o NBMA.

**show ip ospf**

Lista el número de veces que el algoritmo SPF (Short-First Path) se ha ejecutado
Verificación del funcionamiento de OSPF. Usar los siguientes comandos:

**show ip protocols**

Permite ver que protocolos de encaminamiento hay activos listando parámetros tales como temporizadotes, métricas, filtros, etc

**show ip route**

Permite ver la tabla de encaminamiento

**show ip route ospf**

Permite ver la tabla de encaminamiento solo para entradas OSPF

**show ip ospf interface**

Lista información relacionada con una interfaz que usa OSPF. Permite comprobar si las interfaz pertenecen al área a la que se suponen deberían pertenecer. También permite averiguar si una interfaz es DR, BDR o DROTHER (no es ni DR ni BDR), su prioridad y si la red es de tipo BMA o NBMA.

**show ip ospf**

Lista el número de veces que el algoritmo SPF (Short-First Path) se ha ejecutado

**show ip ospf neighbor**

Lista información acerca de los vecinos OSPF por cada interfaz

**show ip ospf neighbor**

Lista información detallada acerca de los vecinos OSPF por cada interfaz

**show ip ospf database**

Lista los contenidos de la DB topológica

**debug ip ospf "op"**

Donde "op" son distintas opciones permiten debuguear la distintas operaciones que ejecuta OSPF (adjacency, events, etc)lista información acerca de los vecinos OSPF por cada interfaz

**show ip ospf neighbor**

Lista información detallada acerca de los vecinos OSPF por cada interfaz

**show ip ospf database**

Lista los contenidos de la DB topológica debug ip ospf "op": donde "op" son distintas opciones permiten debuguear las distintas operaciones que ejecuta OSPF (adjacency, events, etc)

**Configuración MPLS**

**Paso 13 Configurar MPLS**

Para habilitar MPLS en los routers, hay que indicar qué interfaces del router van a participar en este protocolo. Para ello iremos configurando en dichas interfaces del router el comando "**mpls ip**" de forma que indicamos al router que conmute en entrada y salida las tramas MPLS que reciba o envíe, así como que detecte vecindades de routers MPLS con el protocolo de distribución de etiquetas. Las configuraciones son para LSR y LER.

**Para activar CEF y poder trabajar en entornos MPLS**

LSR1(config)# ip cef //Habilitamos el protocolo CEF en el router
LSR1(config)# # mpls label protocol ldp //Definimos el protocolo LDP como protocolo para la distribución de las etiquetas
LSR1(config)# mpls ip //Habilitamos MPLS a nivel global

**Para activar el protocolo de distribución de etiquetas LDP**

**LDP** (Label Distribution Protocol): un protocolo para la distribución de etiquetas MPLS entre los equipos de la red.

Configura MPLS en todas las interfaces.

LSR1(config)# interface serial<nombre de la interfaz>
LSR1(config)# interface s1/0
LSR1(config-if)# mpls ip
LSR1(config-if)# mpls label protocol ldp
Realizar las mismas configuraciones para las demás interfaces.

**Tabla III.27: Interfaz de los routers**

| Router | Interfaz |
|---|---|
| **LSR2** | S1/0<br>S1/1<br>S1/2 |
| **LSR3** | S1/0<br>S1/1<br>S1/2 |
| **LER1** | S1/0<br>F0/0 |
| **LER2** | S1/0<br>F0/0 |
| **LER3** | S1/0<br>F0/0 |

**Elaborado por: Ing**. Pamela Buñay

**Paso 14 Probar la red con el protocolo de enrutamiento OSPF**

Utilizar el comando **ping** para verificar si existe comunicación entre todos los routers del escenario.

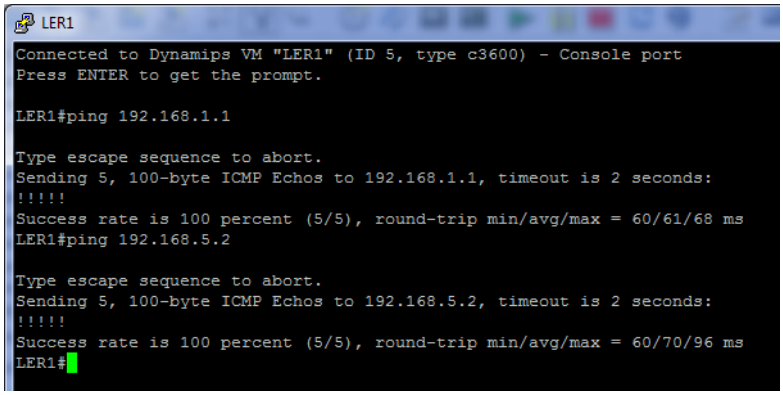



**Figura 4.17: Conectividad desde LER1 a LER2 y LER3**

**Paso 15 Verificar la configuración de MPLS**

Para ver de qué comandos disponemos utilizamos el comando "?"

LER1#show mpls ?

atm-ldp ATM LDP Protocol information
forwarding-table Show the Label Forwarding Information Base (LFIB)
interfaces Per-interface MPLS forwarding information
ip MPLS IP information
label Label information
ldp Label Distribution Protocol information
traffic-eng Traffic engineering information

En primer lugar para ver rápidamente las interfaces trabajando con MPLS ejecutaremos “show mpls interfaces” y saber qué protocolo de intercambio de etiquetas usan.

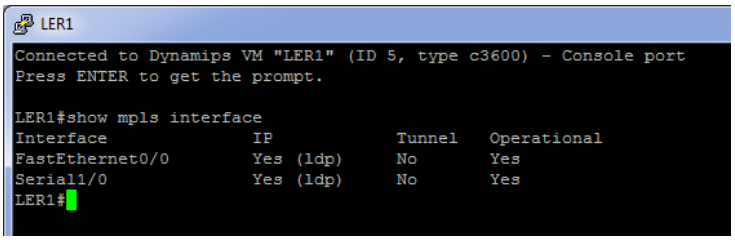



**Figura 4.18:Protocolo LDP habilitado en la red**

Para verificar la configuración de MPLS se utiliza el comando “show mpls forwarding-table” el cual muestra la asignación de etiquetas por cada ruta.

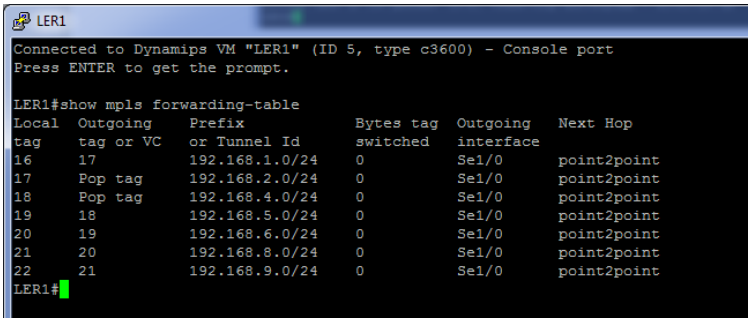



**Figura 4.19:Asignación e intercambio de etiquetas en el LER1**

Para los otros equipos usar las siguientes interfaces:

**Tabla III.28: Interfaz de los routers**

| Router | Interfaz |
|---|---|
| **LSR2** | S1/0<br>S1/1<br>S1/2 |
| **LSR3** | S1/0<br>S1/1<br>S1/2 |
| **LER1** | S1/0<br>F0/0 |
| **LER2** | S1/0 |

| | |
|---|---|
| | F0/0 |
| **LER3** | S1/0<br>F0/0 |

**Elaborado por: Ing**. Pamela Buñay

Agregando “detail” al anterior comando se puede observar más detalles de la configuración como por ejemplo el tamaño del MTU (Maximum Transmission Unit) y que otras opciones de MPLS están desabilitadas para la interfaz como BGP y los túnelrs LSP (Label Switches Path).

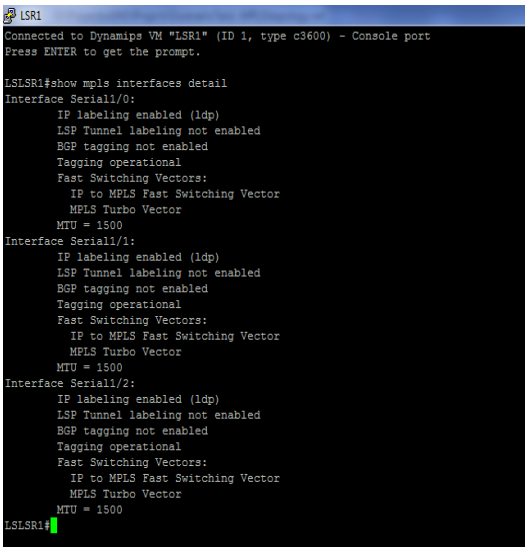



**Figura 4.20:Detalles de las interfaces con MPLS**

Con el comando “traceroute” se verifica los saltos para llegar a una IP de destino y el intercambio en esos saltos del valor de las etiquetas.

Además se puede apreciar que el campo EXP por defecto tiene el valor de 0, posteriormente se manipula este campo para ofrecer QoS mediante DiffSev.

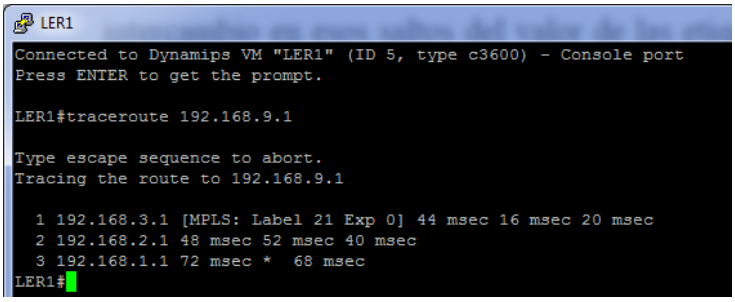



**Figura 4.21:Traceroute a 192.168.9.1**

**Otros comandos a utilizar son:**

- show mpls ldp discovery para observar información de TDP (o LDP), como el identificativo del router MPLS y los vecinos.
- show mpls ldp neighbor para detección de las adyacencias de TDP (oLDP) y el estado de las conexiones establecidas.

**Paso 16 Modificar del tamaño de MTU para MPLS**

Una de las características de MPLS es que permite anidar etiquetas MPLS en función de la aplicación y por tanto puede aumentar el número de cabeceras y para ello hay que informar a las interfaces físicas de dichas eventualidades para evitar el descarte de tramas que superen la MTU. La cabecera MPLS tiene 4 bytes. La MTU por defecto siempre se toma de la propia interfaz, que en el caso de una Ethernet es 1500 bytes.

```
LSR1(config)# interface s1/0
LSR1(config-if)# mpls mtu 1508
LSR1(config)# interface s1/1
LSR1(config-if)# mpls mtu 1508
LSR1(config)# interface s1/2
LSR1(config-if)# mpls mtu 1508
```

**Paso 17 Verificar el cambio de MTU**

Para verificar el cambio de MTU ingresamos **show mpls interfaces detail**

**Configuración de DiffServ**

**Paso 18 Configurar QoS mediante DIFFSERV**

Mediante la integración de los modelos MPLS y DiffServ obtenemos una arquitectura en la que MPLS se sitúa en el nivel de red-enlace, y sirve para evitar la congestión de la red, aportando sus características de ingeniería de tráfico. Mientras, DiffServ asegura unos ciertos parámetros de calidad de servicio realizando una distinción y priorización del tráfico.

**Paso 19Configurar el marcado y clasificación del tráfico**

Se utiliza con el comando **class-map,** y para eliminar la clase de tráfico, se hace con no class-map. **class-map** [**match-any** | **match-all**] *class-name* no **class-map** [**match-any** | **match-all**] *class-name*

| Comando | Propósito |
|---|---|
| `Router(config)# class-map class-map-name` | Especifica el nombre definido por el usuario para la clase de tráfico. Los nombres pueden tener un máximo de 40 caracteres. En este caso, el tráfico debe cuadrar con todos los criterios de clasificación del tráfico. |
| `Router(config)# class-map match-all class-map-name` | Especifica que todos los criterios de equiparación deben darse en el tráfico entrante para poder ser clasificado como parte del tráfico de la clase. |
| `Router(config)# class-map match-any class-map-name` | Especifica que uno de los criterios de clasificación debe darse para poder clasificar el tráfico entrante como tráfico de la clase. |
| `Router(config-cmap)# match any` | Especifica que todos los paquetes serán equiparados |
| `Router config-cmap)# match class-map class-name` | Especifica el nombre de la clase de tráfico que sera usada como criterio de equiparación |
| `Router(config-cmap)# match ip dscp ip-dscp-value` | Specifies up to eight differentiated services code point (DSCP) values used as match criteria. The value of each service code point is from 0 to 63. |
| `Router (config-cmap)# match protocol protocol` | Specifies the name of the protocol used as a match criterion against which packets are checked to determine if they belong to the class. |

**Figura 4.22:Clases de tráfico**

```
LER1#configure terminal
LER1(config)#class-map IP-AF11
LER1(config-cmap)#match ip precedence 0
LER1(config-cmap)#exit
LER1(config)#class-map IP-AF12
LER1(config-cmap)#match ip precedence 1
LER1(config-cmap)#exit
LER1(config)#class-map IP-AF21
LER1(config-cmap)#match ip precedence 2
LER1(config-cmap)#exit
LER1(config)#class-map IP-AF22
LER1(config-cmap)#match ip precedence 3
LER1(config-cmap)#exit
LER1(config)#class-map IP-AF31
LER1(config-cmap)#match ip precedence 4
LER1(config-cmap)#exit
LER1(config)#class-map IP-AF32
LER1(config-cmap)#match ip precedence 5
LER1(config-cmap)#exit
```

**Paso 20 Crear la política de entrada**

Para configurar una política de tráfico, se usa el comando **policy-map**, en el que se especifica en nombre de la política, y a la que se asocian clases de tráfico, definidas previamente. Todo el tráfico

que no se equipara con los criterios de las clases, pertenecen a la clase de tráfico por defecto. Esta también se puede configurar por el usuario, pero no eliminar.

| Command | Purpose |
|---|---|
| `Router (config)# policy-map policy-name` | Especifica el nombre de la política de tráfico a configurar (máximo 40 caracteres). |
| `Router (config-pmap)# class class-name` | Especifica que esa clase de tráfico (puede ser predefinida), que fue configurada con el commando class-map, se usa para clasificar el tráfico en la política de tráfico. |
| `Router (config-pmap)# class class-default` | Para utilizar en la política la clase por defecto, se utiliza este commando. |
| `Router (config-pmap-c)# bandwidth {bandwidth-kbps \| percent percent}` | Especifica un ancho de banda mínimo que se garantiza a una clase de tráfico en periodos de congestion. |
| `Router (config-pmap-c)# default command` | Establece cualquier commando a su valor por defecto |
| `Router (config-pmap-c)# fair-queue number-of-queues` | Especifica en número de colas reservadas para una clase de tráfico |
| `Router (config-pmap-c)# police bps burst-normal burst-max conform-action action exceed-action action violate-action action` | Especifica un ancho de banda máximo utilizable por una clase de tráfico usando el algoritmo token bucket.. |
| `Router (config-pmap-c)# queue-limit packets` | Especifica el máximo número de paquetes encolados para una clase de tráfico (en ausencia del commando random-detect) |
| `Router (config-pmap-c)# random-detect` | Habilita la política WRED (Weighted Random Early Detection) para una clase de tráfico que tiene un ancho de banda garantizado.. |
| `Router (config-pmap-c)# set ip dscp ip-dscp-value` | Especifica el valor IP DSCP de paquetes dentro de una clase de tráfico. El valor IP DSCP está en el rango 0 a 63. |
| `Router (config-pmap-c)# service-policy policy-map-name` | Especifica el nombre de una política de tráfico que se usará como criterio de equiparación. |

**Figura 4.23:Política de entrada**

La política configurada en este caso permite primeramente enviar el tráfico a cierta velocidad y copiar el valor Precedence al campo EXP para la transmisión hacia el siguiente salto.

LER1(config)#policy-map politica-1
LER1(config-pmap)#class IP-AF11
LER1(config-pmap-c)#police 8000 conform-action set-mpls-exp-imposition-transmit 0 exceed-action drop
LER1(config-pmap-c)#exit

```
LER1(config-pmap)#class IP-AF12
LER1(config-pmap-c)#police 10000 conform-action set-mpls-exp-imposition-transmit 1 exceed-
action set-mpls-exp-imposition-transmit 0
LER1(config-pmap-c)#exit

LER1(config-pmap)#class IP-AF21
LER1(config-pmap-c)#police 12000 conform-action set-mpls-exp-imposition-transmit 2 exceed-
action set-mpls-exp-imposition-transmit 1
LER1(config-pmap-c)#exit

LER1(config-pmap)#class IP-AF22
LER1(config-pmap-c)#police 12000 conform-action set-mpls-exp-imposition-transmit 3 exceed-
action set-mpls-exp-imposition-transmit 2
LER1(config-pmap-c)#exit

LER1(config-pmap)#class IP-AF31
LER1(config-pmap-c)#police 12000 conform-action set-mpls-exp-imposition-transmit 4 exceed-
action set-mpls-exp-imposition-transmit 3
LER1(config-pmap-c)#exit

LER1(config-pmap)#class IP-AF32
LER1(config-pmap-c)#police 12000 conform-action set-mpls-exp-imposition-transmit 5 exceed-
action set-mpls-exp-imposition-transmit 4
LER1(config-pmap-c)#end
```

**Paso 21 Asignar la política a la interfaz de entrada**

Para asociar una política de tráfico a una interfaz, y especificar la dirección en la cual debe especificarse la política (paquetes entrantes o paquetes salientes), se utiliza el comando **service-policy.**

**service-policy** {**input** | **output**} *policy-map-name*

**no service-policy** {**input** | **output**} *policy-map-name*

| Comando | Propósito |
|---|---|
| `Router(config-if)# service-policy output policy-map-name` | Especifica el nombre de la política de tráfico que se asociará en una interfaz en la dirección de salida. La política evalua todo el tráfico que abandona la interfaz. |
| `Router(config-if)# service-policy input policy-map-name` | Especifica el nombre de la política de tráfico que se asocia en la dirección de entrada de una interfaz. La política evalua todo el tráfico que entra en esa interfaz. |

**Figura 4.24:Asignación de la política a la interfaz de entrada**

```
LER1(config)#int f0/0
LER1(config-if)#service-policy input politica-1
LER1(config-if)#exit
```

**Paso 22 Clasificar los paquetes en base al campo EXP**

Los paquetes nuevamente son clasificados a la salida del router, para este caso de acuerdo al valor del campo EXP de la etiqueta superior y es colocado en las respectivas clases.

```
LER1#configure terminal
LER1(config)#class-map MPLS-AF11
LER1(config-cmap)#match mpls experimental topmost 0
LER1(config-cmap)#exit
LER1(config)#class-map MPLS-AF12
LER1(config-cmap)#match mpls experimental topmost 1
LER1(config-cmap)#exit
LER1(config)#class-map MPLS-AF21
LER1(config-cmap)#match mpls experimental topmost 2
LER1(config-cmap)#exit
LER1(config)#class-map MPLS-AF22
LER1(config-cmap)#match mpls experimental topmost 3
LER1(config-cmap)#exit
LER1(config)#class-map MPLS-AF31
LER1(config-cmap)#match mpls experimental topmost 4
LER1(config-cmap)#exit
LER1(config)#class-map MPLS-AF32
LER1(config-cmap)#match mpls experimental topmost 5
LER1(config-cmap)#exit
```

**Paso 23 Crear la política a la salida del router**

La política a la salida del router especifica el porcentaje de ancho de banda asignado a cada clase y además para las situaciones de congestión se activa el mecanismo de descarte inteligente para evitar oscilaciones llamado WRED (Weighted Random Early Discard).

```
LER1(config)#policy-map politica-2
LER1(config-pmap)#class MPLS-AF11
LER1(config-pmap-c)#bandwidth percent 5
LER1(config-pmap-c)#random-detect
LER1(config-pmap-c)#exit

LER1(config-pmap)#class MPLS-AF12
LER1(config-pmap-c)#bandwidth percent 10
LER1(config-pmap-c)#random-detect
LER1(config-pmap-c)#exit

LER1(config-pmap)#class MPLS-AF21
LER1(config-pmap-c)#bandwidth percent 10
LER1(config-pmap-c)#random-detect
LER1(config-pmap-c)#exit

LER1(config-pmap)#class MPLS-AF22
LER1(config-pmap-c)#bandwidth percent 15
LER1(config-pmap-c)#random-detect
LER1(config-pmap-c)#exit
```

```
LER1(config-pmap)#class MPLS-AF31
LER1(config-pmap-c)#bandwidth percent 15
LER1(config-pmap-c)#random-detect
LER1(config-pmap-c)#exit

LER1(config-pmap)#class MPLS-AF32
LER1(config-pmap-c)#bandwidth percent 20
LER1(config-pmap-c)#random-detect
LER1(config-pmap-c)#end
```

**Paso 24 Asignar la política en la interfaz de salida**

La política es aplicada a la interfaz de salida del router LER1, en donde los paquetes son clasificados dependiendo del valor del campo EXP y enviados utilizando un ancho de banda de acuerdo a la prioridad.

```
LER1(config)#int s1/0
LER1(config-if)#service-policy output politica-2
LER1(config-if)#exit
```

**Paso 25 Asignar la política en la interfaz de salida**

```
LER1(config)#interface FastEthernet1/0
LER1(config-if)#service-policy input mpls-in
```

**Paso 26 Configuración LRS1**

```
LRS1 (config )#class-map match-all mpls-in
LRS1 (config-cmap)#match mpls experimental topmost 3
LRS1 (config-cmap)#exit
    LRS1 (config )#policy-map mpls-in
    LRS1 (config )#class mpls-in
    LRS1 (config-cmap)#set mpls experimental topmost 2
    LRS1 (config-cmap)#exit
```

**Paso 27 Configuración de LSR2 y LSR3**

```
LSR2 #configure terminal
LSR2 (config)#class-map match-all MPLS-AF11
LSR2 (config-cmap)#match mpls experimental topmost 0
LSR2 (config-cmap)#exit

LSR2 (config)#class-map match-all MPLS-AF12
LSR2 (config-cmap)#match mpls experimental topmost 1
LSR2 (config-cmap)#exit

LSR2 (config)#class-map match-all MPLS-AF21
LSR2 (config-cmap)#match mpls experimental topmost 2
LSR2 (config-cmap)#exit

LSR2 (config)#class-map match-all MPLS-AF22
```

```
LSR2 (config-cmap)#match mpls experimental topmost 3
LSR2 (config-cmap)#exit

LSR2 (config)#class-map match-all MPLS-AF31
LSR2 (config-cmap)#match mpls experimental topmost 4
LSR2 (config-cmap)#exit

LSR2 (config)#class-map match-all MPLS-AF32
LSR2 (config-cmap)#match mpls experimental topmost 5
LSR2 (config-cmap)#exit

LSR2 (config)#class-map match-all qos-group-AF11
LSR2 (config-cmap)#match qos-group 0
LSR2 (config-cmap)#exit

LSR2 (config)#class-map match-all qos-group-AF12
LSR2 (config-cmap)#match qos-group 1
LSR2 (config-cmap)#exit

LSR2 (config)#class-map match-all qos-group-AF21
LSR2 (config-cmap)#match qos-group 2
LSR2 (config-cmap)#exit

LSR2 (config)#class-map match-all qos-group-AF22
LSR2 (config-cmap)#match qos-group 3
LSR2 (config-cmap)#exit

LSR2 (config)#class-map match-all qos-group-AF31
LSR2 (config-cmap)#match qos-group 4
LSR2 (config-cmap)#exit

LSR2 (config)#class-map match-all qos-group-AF32
LSR2 (config-cmap)#match qos-group 5
LSR2 (config-cmap)#exit

LSR2 (config)#policy-map politica-3
LSR2 (config-pmap)#class MPLS-AF11
LSR2 (config-cmap)#set qos-group mpls experimental topmost

LSR2 (config-pmap)#class MPLS-AF12
LSR2 (config-pmap-c)#set qos-group mpls experimental topmost

LSR2 (config-pmap)#class MPLS-AF21
LSR2 (config-pmap-c)#set qos-group mpls experimental topmost

LSR2 (config-pmap)#class MPLS-AF22
LSR2 (config-pmap-c)#set qos-group mpls experimental topmost
LSR2 (config-pmap)#class MPLS-AF31
LSR2 (config-pmap-c)#set qos-group mpls experimental topmost

LSR2 (config-pmap)#class MPLS-AF32
LSR2 (config-pmap-c)#set qos-group mpls experimental topmost

LSR2 (config)#policy-map politica-4
```

```
LSR2 (config-pmap)#class qos-group-AF11
LSR2 (config-pmap-c)#bandwidth percent 5
LSR2 (config-pmap-c)#random-detect
LSR2 (config-pmap-c)#set mpls experimental topmost qos-group

LSR2 (config-pmap)#class qos-group-AF12
LSR2 (config-pmap-c)#bandwidth percent 10
LSR2 (config-pmap-c)#random-detect
LSR2 (config-pmap-c)#set mpls experimental topmost qos-group

LSR2 (config-pmap)#class qos-group-AF21
LSR2 (config-pmap-c)#bandwidth percent 10
LSR2 (config-pmap-c)#random-detect
LSR2 (config-pmap-c)#set mpls experimental topmost qos-group

LSR2 (config-pmap)#class qos-group-AF22
LSR2 (config-pmap-c)#bandwidth percent 15
LSR2 (config-pmap-c)#random-detect
LSR2 (config-pmap-c)#set mpls experimental topmost qos-group

LSR2 (config-pmap)#class qos-group-AF31
LSR2 (config-pmap-c)#bandwidth percent 15
LSR2 (config-pmap-c)#random-detect
LSR2 (config-pmap-c)#set mpls experimental topmost qos-group

LSR2 (config-pmap)#class qos-group-AF32
LSR2 (config-pmap-c)#bandwidth percent 20
LSR2 (config-pmap-c)#random-detect
LSR2 (config-pmap-c)#set mpls experimental topmost qos-group

LER1(config)#interface FastEthernet1/0
LER1(config-if)#service-policy input mpls-in

LSR2 (config)#interface Serial1/0
LSR2 (config-if)#service-policy input politica-3

LSR2 (config)#interface Serial1/1
LSR2 (config-if)#service-policy output politica-4
```

Para LSR3 las mismas configuraciones cambian las interfaces de entrada y salida

```
LSR3 (config)#interface Serial1/1
LSR3 (config-if)#service-policy input politica-3

LSR3 (config)#interface Serial1/0
LSR3 (config-if)#service-policy output politica-4
```

**Paso 28 Configuración de LER2, LER3**

```
LER2 (config)#class-map match-all MPLS-AF11
LER2 (config-cmap)#match mpls experimental topmost 0

LER2 (config)#class-map match-all MPLS-AF12
match mpls experimental topmost 1
```

```
LER2 (config)#class-map match-all MPLS-AF21
match mpls experimental topmost 2

LER2 (config)#class-map match-all MPLS-AF22
match mpls experimental topmost 3

LER2 (config)#class-map match-all MPLS-AF31
match mpls experimental topmost 4

LER2 (config)#class-map match-all MPLS-AF32
match mpls experimental topmost 5

LER2 (config)#class-map match-all qos-group-AF11
match qos-group 0

LER2 (config)#class-map match-all qos-group-AF12
match qos-group 1

LER2 (config)#class-map match-all qos-group-AF21
match qos-group 2

LER2 (config)#class-map match-all qos-group-AF22
match qos-group 3

LER2 (config)#class-map match-all qos-group-AF31
match qos-group 4

LER2 (config)#class-map match-all qos-group-AF32
match qos-group 5

LER2 (config)#policy-map politica-5
LER2 (config-pmap)#class MPLS-AF11
LER2 (config-cmap)#set qos-group mpls experimental topmost

LER2 (config-pmap)#class MPLS-AF12
LER2 (config-cmap)#set qos-group mpls experimental topmost

LER2 (config-pmap)#class MPLS-AF21
LER2 (config-cmap)#set qos-group mpls experimental topmost
LER2 (config-pmap)#class MPLS-AF22
LER2 (config-cmap)#set qos-group mpls experimental topmost

LER2 (config-pmap)#class MPLS-AF31
LER2 (config-cmap)#set qos-group mpls experimental topmost

LER2 (config-pmap)#class MPLS-AF32
LER2 (config-cmap)#set qos-group mpls experimental topmost


LER2 (config)#policy-map politica-6
LER2 (config-pmap)#class qos-group-AF11
LER2 (config-pmap)#bandwidth percent 5
LER2 (config-pmap)#random-detect
```

```
LER2 (config-pmap)#set precedence qos-group

LER2 (config-pmap)#class qos-group-AF12
LER2 (config-pmap)#bandwidth percent 10
LER2 (config-pmap)#random-detect
LER2 (config-pmap)#set precedence qos-group

LER2 (config-pmap)#class qos-group-AF21
LER2 (config-pmap)#bandwidth percent 10
LER2 (config-pmap)#random-detect
LER2 (config-pmap)#set precedence qos-group

LER2 (config-pmap)#class qos-group-AF22
LER2 (config-pmap)#bandwidth percent 15
LER2 (config-pmap)#random-detect
LER2 (config-pmap)#set precedence qos-group

LER2 (config-pmap)#class qos-group-AF31
LER2 (config-pmap)#bandwidth percent 15
LER2 (config-pmap)#random-detect
LER2 (config-pmap)#set precedence qos-group

LER2 (config-pmap)#class qos-group-AF32
LER2 (config-pmap)#bandwidth percent 20
LER2 (config-pmap)#random-detect
LER2 (config-pmap)#set precedence qos-group

LER2 (config)#interface Serial1/0
LER2 (config-if)#service-policy input politica-5

LER2 (config)#interface FastEthernet0/0
LER2 (config-if)#service-policy output politica-6
```

Para LER3 realizar las mismas configuraciones.

**Paso 29 Verificar la configuración**

Para verificar y mostrar la información referente a una política o a una clase, se pueden utilizar los siguientes comandos:

| Comando | Propósito |
|---|---|
| `Router# show class-map` | Muestra toda la información de las clases de tráfico |
| `Router# show class-map class-name` | Muestra la información de la clase especificada |
| `Router# show policy-map` | Muestra todas las políticas |
| `Router# show policy-map policy-map-name` | Muestra la política especificada |
| `Router# show policy-map interface` | Muestra configuraciones y estadísticas de todas las entradas y salidas de las políticas asociadas a la interfaz |
| `Router# show policy-map interface interface-spec` | Muestra configuraciones y estadísticas de todas las entradas y salidas de las políticas asociadas a una interfaz concreta |
| `Router# show policy-map interface interface-spec input` | Muestra configuración y estadísticas de las políticas de entrada asociadas a la interfaz |
| `Router# show policy-map interface interface-spec output` | Muestra configuración y estadísticas de las |

**Figura 4.25:Verificar la configuración**

# CONCLUSIONES

- Se aplicó MPLS (MultiProtocol LabelSwitching) como backbone y con un modelo de Servicios Diferenciados (DiffServ) para ofrecer garantías de calidad de servicio a aplicaciones/servicios para VoIP y Datos.

- DiffServ divide el tráfico en clases permitiendo controlar la cantidad de tráfico que cada cliente envía a la red de cada clase de tráfico asegurando requerimientos de QoS, utilizando en cada enlace políticas de clasificación y condicionamiento de tráfico.

- Al analizar el indicador Retardo en los tres escenarios, las Redes IP obtuvo un valor promedio de 301.151ms en VoIP y 395.892ms en datos; frente a 90.087 ms VoIP y 229.861ms de MPLS; DiffServ con un valor de 59.665ms VoIP y 105.3307ms en datos. Dando como resultado que las redes IP no son adecuadas para proveer calidad de servicio, mientras que MPLS es adecuada pero es más adecuada si se implementa con DiffServ.

- En el indicador Jitter se obtuvo para los tres escenarios Redes IP, MPLS, MPLS y DiffServ un promedio para VoIP de: 56.0803ms, 35.2696ms, 11.0717 ms respectivamente. Para datos se tiene 79.5689ms para Redes IP, 45.2658ms para MPLS y 20.9822ms para MPLS y DiffServ. En Redes IP es muy alto el Jitter en relación a MPLS, MPLs y DiffServ. Generalmente los paquetes de voz son descartados por el receptor cuando este no recibe los paquetes a tiempo, convirtiéndole en no Adecuada para la provisión de Calidad de Servicio. En MPLS conseguimos hacer el enrutado a más velocidad, a la vez que disminuimos el retardo y el jitter, convirtiéndole en Adecuada. En redes MPLS y DiffServ disminuye más aún porque a más de tener más velocidad se tiene alta prioridad por los paquetes VoIP, al crear las políticas tanto de entrada como salida en las interfaces.

- Al analizar el indicador de paquetes perdidos en los tres escenarios redes IP, MPLS, MPLS y DiffServ, existe más paquetes perdidos en las redes IP con 4.2107% en VoIP y 9.898%Datos, redes MPLS 1.7497% VoIP y 4.653%, redes MPLS y DiffServ 0.6169 % y 2.698%. En las redes IP existe mayor de pérdida de paquetes debido a que no se tiene ninguna garantía en él envió y recepción de paquetes convirtiéndola en no Adecuada para la provisión de Calidad de Servicio. Con redes MPLS baja considerablemente el porcentaje de pérdida, generalmente porque tiene el mecanismo de creación de circuito virtual entre los routers (orientado a la conexión) siendo Conforme. En MPLS y DiffServ baja aún más el porcentaje de paquetes

perdidos por el etiquetado de los paquetes que se realiza en base a criterios de prioridad y/o calidad (QoS), convirtiéndole en Adecuado para la provisión de Calidad de Servicio.

- La implementación de MPLS y DiffServ resultó ser adecuada para la provisión de Calidad de Servicio punto a punto en la transmisión de tráfico en tiempo real, pues es adecuada para los tres indicadores Retardo, Latencia y Perdida de paquetes que fueron analizados. MPLS es mediamente adecuada, debido a que en la pérdida de paquetes se obtuvo el resultado de Conforme. Las redes IP no son Adecuadas para ninguno de los indicadores estudiados pues se obtuvieron valores que no satisfacían los umbrales mínimos para proveer Calidad de Servicio.

# RECOMENDACIONES

- Para seleccionar los indicadores deben ser seleccionados cuidadosamente, estos deben estar orientados a características que ayuden a comprobar la hipótesis.

- Para asegurar Calidad de Servicio punto a punto se recomienda implementar MPLS como backbone y con un modelo de Servicios Diferenciados (DiffServ).

- Las políticas tanto de entrada como de salida de las interfaces deben ser configuradas según los criterios de los administradores de la red, para satisfacer las necesidades de los usuarios.

- Antes de implementar MPLS y DiffServ se debe diseñar escenarios que permitan realizar todas las mediciones para obtener los resultados deseados.

- Se recomienda a los administradores hacer un análisis de las aplicaciones utilizadas con mayor frecuencia por los usuarios para diseñar las políticas a aplicarse.

- Se debe realizar un monitoreo de la red para ver si las políticas implementadas dan los resultados esperados.

# BIBLIOGRAFÍA

1. **FERGUSON, P., y HUDSON, G.**, Quality of Service: Delivering QoS in the internet and Corporate Network., Wiley Computer Book., New York., EEUU., 1998., Pp., 210-260.

**TESIS**

1. **CHICAIZA, J.**, "Soporte de MPLS para Servicios Diferenciados" Arquitectura y Diseño de Redes de Alta Calidad,. Programa de Doctorado conjunto en Informática. Universidad Técnica Particular de Loja Universidad Politécnica de Madrid., Madrid., España., **TESIS**., 1998., Pp., 155-167.

2. **DOMINGO, C.,** Tesis Doctoral: Diferenciación en servicios y mejora de la supervivencia en redes ad hoc., Departamento de Ingeniería Telemática., Universidad Politécnica de Cataluña., Cataluña., España., **TESIS**., 2009,. Pp., 155-167.

**RFC**

1. **ALMQUIST, P**., Type of service in the internet protocol suite., RFC 1349., 1992., Pp., 18-36.

2. **AWDUCHE, D., y MALCOLM, J.,** Requirements for traffic engineering over MPLS., RFC2702., IETF., 1999., Pp., 19-25.
3. **BLACK, D.,** Per Hop Behavior Identification Cedes., RFC 3140., 2001., Pp., 9-18.

4. **BLAKE, S.,** An Architecture for Differentiated Services., RFC 2475., 1998., Pp., 11-25.

5. **DEERING, S., y HINDEN, R.,** Internet Protocol, Versión 6 (IPv6) Specification., RFC 2460., 1998., Pp., 9-15.

6. **HEINANEN, J.,** Assured Forwarding PHB Group., RFC 2597., 1999., Pp., 8-26.

7. **JACOBSON, V.,** An Expedited Forwarding PHB., RFC 2598. 1999., Pp., 24-33.

8. **ROSEN, E.,** Multiprotocol Label Switching Architecture., IETF., MPLS., 1999., Pp., 33-42.

9. **ROSEN, E., y VISWANATHAN, A**., Multiprotocol label switching Architecture., RFC3031., IETF., 2001,. Pp., 10-20.

**BIBLIOGRAFÍA DE INTERNET**

**1. CALIDAD DE SERVICIO.,**
http://www.imaginar.org/ngn/manuales/P_calidad_servicio.pdf
2011-09-26.

**2. SERVICIOS DIFERENCIADOS.,**
www.info-ab.uclm.es/asignaturas/42650/PDFs/Tema3b.pdf.
2012-10-04.

3. **REVELO, E**., "Calidad de servicio en redes"., http://slideshare.net/prestonj_jag/calidad-de-servicio-en-redes/
2012-04-06.

# ANEXO I

## Archivos de configuración de los routers

### LER1

```
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname LER1
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
ip subnet-zero
!
!
no ip domain lookup
ip domain name lab.local
!
ip cef
ip audit po max-events 100
mpls label protocol ldp
!
!
class-map match-all MPLS-AF11
  match mpls experimental topmost 0
class-map match-all MPLS-AF22
  match mpls experimental topmost 3
class-map match-all MPLS-AF32
  match mpls experimental topmost 5
class-map match-all MPLS-AF31
  match mpls experimental topmost 4
class-map match-all MPLS-AF12
  match mpls experimental topmost 1
class-map match-all MPLS-AF21
  match mpls experimental topmost 2
class-map match-all IP-AF12
  match ip precedence 1
class-map match-all IP-AF21
  match ip precedence 2
class-map match-all IP-AF31
  match ip precedence 4
class-map match-all IP-AF32
  match ip precedence 5
class-map match-all IP-AF11
  match ip precedence 0
class-map match-all IP-AF22
```

```
  match ip precedence 3
!
!
policy-map politica-1
  class IP-AF11
   police 8000 conform-action set-mpls-exp-imposition-transmit 0 exceed-action
drop
  class IP-AF12
   police 10000 conform-action set-mpls-exp-imposition-transmit 1 exceed-action
set-mpls-exp-imposition-transmit 0
  class IP-AF21
   police 12000 conform-action set-mpls-exp-imposition-transmit 2 exceed-action
set-mpls-exp-imposition-transmit 1
  class IP-AF22
   police 12000 conform-action set-mpls-exp-imposition-transmit 3 exceed-action
set-mpls-exp-imposition-transmit 2
  class IP-AF31
   police 12000 conform-action set-mpls-exp-imposition-transmit 4 exceed-action
set-mpls-exp-imposition-transmit 3
  class IP-AF32
   police 12000 conform-action set-mpls-exp-imposition-transmit 5 exceed-action
set-mpls-exp-imposition-transmit 4
policy-map politica-2
  class MPLS-AF11
   bandwidth percent 5
   random-detect
  class MPLS-AF12
   bandwidth percent 10
   random-detect
  class MPLS-AF21
   bandwidth percent 10
   random-detect
  class MPLS-AF22
   bandwidth percent 15
   random-detect
  class MPLS-AF31
   bandwidth percent 15
   random-detect
  class MPLS-AF32
   bandwidth percent 20
   random-detect
!
interface FastEthernet0/0
 ip address 192.168.7.1 255.255.255.0
 duplex auto
 speed auto
 mpls label protocol ldp
 tag-switching mtu 1508
 tag-switching ip
 service-policy input politica-1
!
```

```
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial1/0
 ip address 192.168.3.2 255.255.255.0
 mpls label protocol ldp
 tag-switching mtu 1508
 tag-switching ip
 serial restart-delay 0
 service-policy output politica-2
!
interface Serial1/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
router ospf 1
 log-adjacency-changes
 network 192.168.3.0 0.0.0.255 area 0
 network 192.168.7.0 0.0.0.255 area 0
!
no ip http server
no ip http secure-server
ip classless
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
!
!
```

```
end
```

**LER2 y LER3**

```
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname LER2
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
ip subnet-zero
!
!
no ip domain lookup
ip domain name lab.local
!
ip cef
ip audit po max-events 100
mpls label protocol ldp
!
class-map match-all qos-group-AF32
  match qos-group 5
class-map match-all qos-group-AF11
  match qos-group 0
class-map match-all qos-group-AF22
  match qos-group 3
class-map match-all qos-group-AF12
  match qos-group 1
class-map match-all qos-group-AF21
  match qos-group 2
class-map match-all qos-group-AF31
  match qos-group 4
class-map match-all MPLS-AF11
  match mpls experimental topmost 0
class-map match-all MPLS-AF22
  match mpls experimental topmost 3
class-map match-all MPLS-AF32
  match mpls experimental topmost 5
class-map match-all MPLS-AF31
  match mpls experimental topmost 4
class-map match-all MPLS-AF12
  match mpls experimental topmost 1
class-map match-all MPLS-AF21
  match mpls experimental topmost 2
!
```

```
!
policy-map politica-5
  class MPLS-AF11
   set qos-group mpls experimental topmost
  class MPLS-AF12
   set qos-group mpls experimental topmost
  class MPLS-AF21
   set qos-group mpls experimental topmost
  class MPLS-AF22
   set qos-group mpls experimental topmost
  class MPLS-AF31
   set qos-group mpls experimental topmost
  class MPLS-AF32
   set qos-group mpls experimental topmost
policy-map politica-6
  class qos-group-AF11
   bandwidth percent 5
   random-detect
   set precedence qos-group
  class qos-group-AF12
   bandwidth percent 10
   random-detect
   set precedence qos-group
  class qos-group-AF21
   bandwidth percent 10
   random-detect
   set precedence qos-group
  class qos-group-AF22
   bandwidth percent 15
   random-detect
   set precedence qos-group
  class qos-group-AF31
   bandwidth percent 15
   random-detect
   set precedence qos-group
  class qos-group-AF32
   bandwidth percent 20
   random-detect
   set precedence qos-group

!
interface FastEthernet0/0
 ip address 192.168.9.1 255.255.255.0
 duplex auto
 speed auto
 mpls label protocol ldp
 tag-switching mtu 1508
 tag-switching ip
 service-policy output politica-6
!
interface FastEthernet0/1
```

```
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial1/0
 ip address 192.168.1.1 255.255.255.0
 mpls label protocol ldp
 tag-switching mtu 1508
 tag-switching ip
 serial restart-delay 0
 service-policy input politica-5
!
interface Serial1/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
router ospf 1
 log-adjacency-changes
 network 192.168.1.0 0.0.0.255 area 0
network 192.168.9.0 0.0.0.255 area 0
!
no ip http server
no ip http secure-server
ip classless
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
!
!
end
```

**LSR1**

```
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname LSLSR1
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
ip subnet-zero
!
!
no ip domain lookup
ip domain name lab.local
!
ip cef
ip audit po max-events 100
mpls label protocol ldp
!
class-map match-all mpls-in
  match mpls experimental topmost 3
!
!
policy-map mpls-in
  class mpls-in
   set mpls experimental topmost 2
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial1/0
 ip address 192.168.2.2 255.255.255.0
 mpls label protocol ldp
 tag-switching mtu 1508
 tag-switching ip
 serial restart-delay 0
 clock rate 64000
!
```

```
interface Serial1/1
 ip address 192.168.4.1 255.255.255.0
 mpls label protocol ldp
 tag-switching mtu 1508
 tag-switching ip
 serial restart-delay 0
 clock rate 64000
!
interface Serial1/2
 ip address 192.168.3.1 255.255.255.0
 mpls label protocol ldp
 tag-switching mtu 1508
 tag-switching ip
 serial restart-delay 0
 clock rate 64000
 service-policy input mpls-in
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
router ospf 1
 log-adjacency-changes
 network 192.168.2.0 0.0.0.255 area 0
 network 192.168.3.0 0.0.0.255 area 0
 network 192.168.4.0 0.0.0.255 area 0
!
no ip http server
no ip http secure-server
ip classless
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
login
!
!
end
```

**LSR2 y LSR3**

```
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
```

```
no service password-encryption
!
hostname LSR2
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
ip subnet-zero
!
!
no ip domain lookup
ip domain name lab.local
!
ip cef
ip audit po max-events 100
mpls label protocol ldp
!
class-map match-all qos-group-AF32
  match qos-group 5
class-map match-all qos-group-AF11
  match qos-group 0
class-map match-all qos-group-AF22
  match qos-group 3
class-map match-all qos-group-AF12
  match qos-group 1
class-map match-all qos-group-AF21
  match qos-group 2
class-map match-all qos-group-AF31
  match qos-group 4
class-map match-all MPLS-AF11
  match mpls experimental topmost 0
class-map match-all MPLS-AF22
  match mpls experimental topmost 3
class-map match-all MPLS-AF32
  match mpls experimental topmost 5
  match qos-group 0
class-map match-all MPLS-AF31
  match mpls experimental topmost 4
class-map match-all MPLS-AF12
  match mpls experimental topmost 1
class-map match-all MPLS-AF21
  match mpls experimental topmost 2
!
!
policy-map politica-3
  class MPLS-AF11
   set qos-group mpls experimental topmost
  class MPLS-AF12
   set qos-group mpls experimental topmost
```

```
  class MPLS-AF21
   set qos-group mpls experimental topmost
  class MPLS-AF22
   set qos-group mpls experimental topmost
  class MPLS-AF31
   set qos-group mpls experimental topmost
  class MPLS-AF32
   set qos-group mpls experimental topmost
policy-map politica-4
  class qos-group-AF11
   bandwidth percent 5
   random-detect
   set mpls experimental topmost qos-group
  class qos-group-AF12
   bandwidth percent 10
   random-detect
   set mpls experimental topmost qos-group
  class qos-group-AF21
   bandwidth percent 10
   random-detect
   set mpls experimental topmost qos-group
  class qos-group-AF22
   bandwidth percent 15
   random-detect
   set mpls experimental topmost qos-group
  class qos-group-AF31
   bandwidth percent 15
   random-detect
   set mpls experimental topmost qos-group
  class qos-group-AF32
   bandwidth percent 20
   random-detect
   set mpls experimental topmost qos-group
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial1/0
 ip address 192.168.2.1 255.255.255.0
 mpls label protocol ldp
 tag-switching mtu 1508
 tag-switching ip
```

```
 serial restart-delay 0
 service-policy input politica-3
!
interface Serial1/1
 ip address 192.168.6.1 255.255.255.0
 mpls label protocol ldp
 tag-switching mtu 1508
 tag-switching ip
 serial restart-delay 0
 clock rate 64000
 service-policy output politica-4
!
interface Serial1/2
 ip address 192.168.1.2 255.255.255.0
 mpls label protocol ldp
 tag-switching mtu 1508
 tag-switching ip
 serial restart-delay 0
 clock rate 64000
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
router ospf 1
 log-adjacency-changes
 network 192.168.1.0 0.0.0.255 area 0
 network 192.168.2.0 0.0.0.255 area 0
 network 192.168.6.0 0.0.0.255 area 0
!
no ip http server
no ip http secure-server
ip classless
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
login
!
!
end
```

# ANEXO II

## Instalación de D-itg

D-ITG es el software más utilizado y recomendado para realizar mediciones de desempeño y simular tráfico de diversos protocolos. D-ITG es una plataforma de código abierto para la generación de tráfico, capaz de producir tráfico IPv4 e IPv6 para paquetes con tamaño determinado y es capaz de calcular el retardo de ida (OWD - One Way Delay) y de ida y vuelta (RTT - Round Trip Time). D-ITG sigue el modelo cliente-servidor. Hay cuatro ejecutables básicos que implementanlos componentes de la plataforma: ITGSend, ITGRecv, ITGLog, e ITGDec[28].

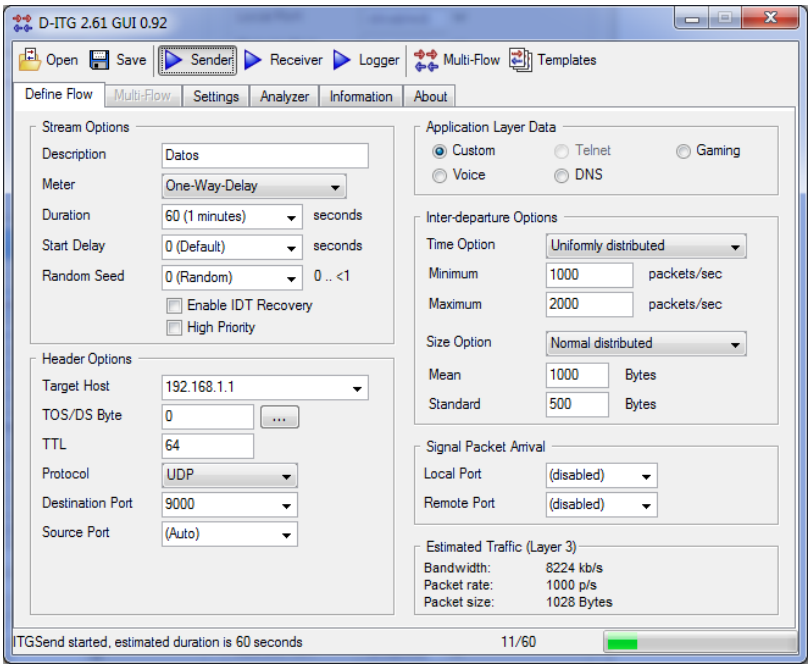
Descargar el inyector y la interfaz gráfica de usuario para DITG, localizadas en las siguientes páginas web:

- Inyector D-ITG-2.7.0-Beta: http://www.grid.unina.it/software/ITG/download.php
- Interfaz itggui-0911: http://www.semken.com/projekte/index.html

La instalación se realizará en Ubuntu.

1. Crear una carpeta dentro del usuario llamada DITG.
2. Descargar el inyector y su interfaz gráfica descomprimida.
3. Abrir un terminal que está dentro de la opción accesorios en la pestaña aplicación.
4. Desde la carpeta src del inyector compilamos
   cd /home/user/DITG/src
   make
5. Esto permite obtener los binarios en el directorio /home/user/DITG/bin, y los copiamos en /usr/local
   cp /home/usuario/DITG/bin/ITG* /usr/local/bin
   cp /home/usuario/DITG/bin/lib* /usr/local/lib
6. Crear el directorio mediante mkdir /home/Pame/DITG/logs, o manualmente. En esta carpeta se guardaran los logs, cuando se haga uso del inyector.
7. Para desplegar la interfaz gráfica se ejecuta desde el directorio /home/user/DITG:

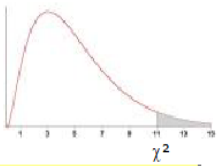
   $ java-jar ITGGUI.jar


D-ITG 2.61 GUI 0.92
Open
Save
Sender
Receiver
Logger
Multi-Flow
Templates
Define Flow
Multi-Flow
Settings
Analyzer
Information
About
Stream Options
Description
Datos
Meter
One-Way-Delay
Duration
60 (1 minutes)
seconds
Start Delay
0 (Default)
seconds
Random Seed
0 (Random)
0 .. <1
Enable IDT Recovery
High Priority
Header Options
Target Host
192.168.1.1
TOS/DS Byte
0
...
TTL
64
Protocol
UDP
Destination Port
9000
Source Port
(Auto)
Application Layer Data
Custom
Telnet
Gaming
Voice
DNS
Inter-departure Options
Time Option
Uniformly distributed
Minimum
1000
packets/sec
Maximum
2000
packets/sec
Size Option
Normal distributed
Mean
1000
Bytes
Standard
500
Bytes
Signal Packet Arrival
Local Port
(disabled)
Remote Port
(disabled)
Estimated Traffic (Layer 3)
Bandwidth:
8224 kb/s
Packet rate:
1000 p/s
Packet size:
1028 Bytes
ITGSend started, estimated duration is 60 seconds
11/60

# ANEXO III

Cátedra: Probabilidad y Estadística
Facultad Regional Mendoza
UTN

**Tabla D.7: VALORES CRÍTICOS DE LA DISTRIBUCIÓN JI CUADRADA**




| g.d.l | 0,001 | 0,005 | 0,01 | 0,02 | 0,025 | 0,03 | 0,04 | 0,05 | 0,10 | 0,15 | 0,20 | 0,25 | 0,30 | 0,35 | 0,40 | g.d.l |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 10,828 | 7,879 | 6,635 | 5,412 | 5,024 | 4,709 | 4,218 | 3,841 | 2,706 | 2,072 | 1,642 | 1,323 | 1,074 | 0,873 | 0,708 | 1 |
| 2 | 13,816 | 10,597 | 9,210 | 7,824 | 7,378 | 7,013 | 6,438 | 5,991 | 4,605 | 3,794 | 3,219 | 2,773 | 2,408 | 2,100 | 1,833 | 2 |
| 3 | 16,266 | 12,838 | 11,345 | 9,837 | 9,348 | 8,947 | 8,311 | 7,815 | 6,251 | 5,317 | 4,642 | 4,108 | 3,665 | 3,283 | 2,946 | 3 |
| 4 | 18,467 | 14,860 | 13,277 | 11,668 | 11,143 | 10,712 | 10,026 | 9,488 | 7,779 | 6,745 | 5,989 | 5,385 | 4,878 | 4,438 | 4,045 | 4 |
| 5 | 20,515 | 16,750 | 15,086 | 13,388 | 12,833 | 12,375 | 11,644 | 11,070 | 9,236 | 8,115 | 7,289 | 6,626 | 6,064 | 5,573 | 5,132 | 5 |
| 6 | 22,458 | 18,548 | 16,812 | 15,033 | 14,449 | 13,968 | 13,198 | 12,592 | 10,645 | 9,446 | 8,558 | 7,841 | 7,231 | 6,695 | 6,211 | 6 |
| 7 | 24,322 | 20,278 | 18,475 | 16,622 | 16,013 | 15,509 | 14,703 | 14,067 | 12,017 | 10,748 | 9,803 | 9,037 | 8,383 | 7,806 | 7,283 | 7 |
| 8 | 26,124 | 21,955 | 20,090 | 18,168 | 17,535 | 17,010 | 16,171 | 15,507 | 13,362 | 12,027 | 11,030 | 10,219 | 9,524 | 8,909 | 8,351 | 8 |
| 9 | 27,877 | 23,589 | 21,666 | 19,679 | 19,023 | 18,480 | 17,608 | 16,919 | 14,684 | 13,288 | 12,242 | 11,389 | 10,656 | 10,006 | 9,414 | 9 |
| 10 | 29,588 | 25,188 | 23,209 | 21,161 | 20,483 | 19,922 | 19,021 | 18,307 | 15,987 | 14,534 | 13,442 | 12,549 | 11,781 | 11,097 | 10,473 | 10 |
| 11 | 31,264 | 26,757 | 24,725 | 22,618 | 21,920 | 21,342 | 20,412 | 19,675 | 17,275 | 15,767 | 14,631 | 13,701 | 12,899 | 12,184 | 11,530 | 11 |
| 12 | 32,909 | 28,300 | 26,217 | 24,054 | 23,337 | 22,742 | 21,785 | 21,026 | 18,549 | 16,989 | 15,812 | 14,845 | 14,011 | 13,266 | 12,584 | 12 |
| 13 | 34,528 | 29,819 | 27,688 | 25,472 | 24,736 | 24,125 | 23,142 | 22,362 | 19,812 | 18,202 | 16,985 | 15,984 | 15,119 | 14,345 | 13,636 | 13 |
| 14 | 36,123 | 31,319 | 29,141 | 26,873 | 26,119 | 25,493 | 24,485 | 23,685 | 21,064 | 19,406 | 18,151 | 17,117 | 16,222 | 15,421 | 14,685 | 14 |
| 15 | 37,697 | 32,801 | 30,578 | 28,259 | 27,488 | 26,848 | 25,816 | 24,996 | 22,307 | 20,603 | 19,311 | 18,245 | 17,322 | 16,494 | 15,733 | 15 |
| 16 | 39,252 | 34,267 | 32,000 | 29,633 | 28,845 | 28,191 | 27,136 | 26,296 | 23,542 | 21,793 | 20,465 | 19,369 | 18,418 | 17,565 | 16,780 | 16 |
| 17 | 40,790 | 35,718 | 33,409 | 30,995 | 30,191 | 29,523 | 28,445 | 27,587 | 24,769 | 22,977 | 21,615 | 20,489 | 19,511 | 18,633 | 17,824 | 17 |
| 18 | 42,312 | 37,156 | 34,805 | 32,346 | 31,526 | 30,845 | 29,745 | 28,869 | 25,989 | 24,155 | 22,760 | 21,605 | 20,601 | 19,699 | 18,868 | 18 |
| 19 | 43,820 | 38,582 | 36,191 | 33,687 | 32,852 | 32,158 | 31,037 | 30,144 | 27,204 | 25,329 | 23,900 | 22,718 | 21,689 | 20,764 | 19,910 | 19 |
| 20 | 45,315 | 39,997 | 37,566 | 35,020 | 34,170 | 33,462 | 32,321 | 31,410 | 28,412 | 26,498 | 25,038 | 23,828 | 22,775 | 21,826 | 20,951 | 20 |
| 21 | 46,797 | 41,401 | 38,932 | 36,343 | 35,479 | 34,759 | 33,597 | 32,671 | 29,615 | 27,662 | 26,171 | 24,935 | 23,858 | 22,888 | 21,991 | 21 |
| 22 | 48,268 | 42,796 | 40,289 | 37,659 | 36,781 | 36,049 | 34,867 | 33,924 | 30,813 | 28,822 | 27,301 | 26,039 | 24,939 | 23,947 | 23,031 | 22 |
| 23 | 49,728 | 44,181 | 41,638 | 38,968 | 38,076 | 37,332 | 36,131 | 35,172 | 32,007 | 29,979 | 28,429 | 27,141 | 26,018 | 25,006 | 24,069 | 23 |
| 24 | 51,179 | 45,559 | 42,980 | 40,270 | 39,364 | 38,609 | 37,389 | 36,415 | 33,196 | 31,132 | 29,553 | 28,241 | 27,096 | 26,063 | 25,106 | 24 |
| 25 | 52,620 | 46,928 | 44,314 | 41,566 | 40,646 | 39,880 | 38,642 | 37,652 | 34,382 | 32,282 | 30,675 | 29,339 | 28,172 | 27,118 | 26,143 | 25 |
| 26 | 54,052 | 48,290 | 45,642 | 42,856 | 41,923 | 41,146 | 39,889 | 38,885 | 35,563 | 33,429 | 31,795 | 30,435 | 29,246 | 28,173 | 27,179 | 26 |
| 27 | 55,476 | 49,645 | 46,963 | 44,140 | 43,195 | 42,407 | 41,132 | 40,113 | 36,741 | 34,574 | 32,912 | 31,528 | 30,319 | 29,227 | 28,214 | 27 |
| 28 | 56,892 | 50,993 | 48,278 | 45,419 | 44,461 | 43,662 | 42,370 | 41,337 | 37,916 | 35,715 | 34,027 | 32,620 | 31,391 | 30,279 | 29,249 | 28 |
| 29 | 58,301 | 52,336 | 49,588 | 46,693 | 45,722 | 44,913 | 43,604 | 42,557 | 39,087 | 36,854 | 35,139 | 33,711 | 32,461 | 31,331 | 30,283 | 29 |
| 30 | 59,703 | 53,672 | 50,892 | 47,962 | 46,979 | 46,160 | 44,834 | 43,773 | 40,256 | 37,990 | 36,250 | 34,800 | 33,530 | 32,382 | 31,316 | 30 |
| 31 | 61,098 | 55,003 | 52,191 | 49,226 | 48,232 | 47,402 | 46,059 | 44,985 | 41,422 | 39,124 | 37,359 | 35,887 | 34,598 | 33,431 | 32,349 | 31 |
| 32 | 62,487 | 56,328 | 53,486 | 50,487 | 49,480 | 48,641 | 47,282 | 46,194 | 42,585 | 40,256 | 38,466 | 36,973 | 35,665 | 34,480 | 33,381 | 32 |
| 33 | 63,870 | 57,648 | 54,776 | 51,743 | 50,725 | 49,876 | 48,500 | 47,400 | 43,745 | 41,386 | 39,572 | 38,058 | 36,731 | 35,529 | 34,413 | 33 |
| 34 | 65,247 | 58,964 | 56,061 | 52,995 | 51,966 | 51,107 | 49,716 | 48,602 | 44,903 | 42,514 | 40,676 | 39,141 | 37,795 | 36,576 | 35,444 | 34 |
| 35 | 66,619 | 60,275 | 57,342 | 54,244 | 53,203 | 52,335 | 50,928 | 49,802 | 46,059 | 43,640 | 41,778 | 40,223 | 38,859 | 37,623 | 36,475 | 35 |
| 40 | 73,402 | 66,766 | 63,691 | 60,436 | 59,342 | 58,428 | 56,946 | 55,758 | 51,805 | 49,244 | 47,269 | 45,616 | 44,165 | 42,848 | 41,622 | 40 |
| 60 | 99,607 | 91,952 | 88,379 | 84,580 | 83,298 | 82,225 | 80,482 | 79,082 | 74,397 | 71,341 | 68,972 | 66,981 | 65,227 | 63,628 | 62,135 | 60 |
| 80 | 124,839 | 116,321 | 112,329 | 108,069 | 106,629 | 105,422 | 103,459 | 101,879 | 96,578 | 93,106 | 90,405 | 88,130 | 86,120 | 84,284 | 82,566 | 80 |
| 90 | 137,208 | 128,299 | 124,116 | 119,648 | 118,136 | 116,869 | 114,806 | 113,145 | 107,565 | 103,904 | 101,054 | 98,650 | 96,524 | 94,581 | 92,761 | 90 |
| 100 | 149,449 | 140,169 | 135,807 | 131,142 | 129,561 | 128,237 | 126,079 | 124,342 | 118,498 | 114,659 | 111,667 | 109,141 | 106,906 | 104,862 | 102,946 | 100 |
| 120 | 173,617 | 163,648 | 158,950 | 153,918 | 152,211 | 150,780 | 148,447 | 146,567 | 140,233 | 136,062 | 132,806 | 130,055 | 127,616 | 125,383 | 123,289 | 120 |
| 140 | 197,451 | 186,847 | 181,840 | 176,471 | 174,648 | 173,118 | 170,624 | 168,613 | 161,827 | 157,352 | 153,854 | 150,894 | 148,269 | 145,863 | 143,604 | 140 |



**Tabla de ji cuadrada**