PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO – CHILE
ESCUELA DE INGENIERÍA ELÉCTRICA

CALIDAD DE SERVICIO SOBRE REDES MULTI-PROTOCOLO DE CONMUTACIÓN DE ETIQUETAS

José Ignacio Varela Caneo

INFORME FINAL DEL PROYECTO PRESENTADO EN CUMPLIMIENTO DE LOS REQUISITOS PARA OPTAR AL TÍTULO PROFESIONAL DE **INGENIERO CIVIL ELECTRÓNICO**

MARZO DE 2017

CALIDAD DE SERVICIO EN REDES MULTI-PROTOCOLO DE CONMUTACIÓN DE ETIQUETAS

INFORME FINAL

Presentado en cumplimiento de los requisitos
para optar al título profesional de

INGENIERO CIVIL ELECTRÓNICO

otorgado por la

ESCUELA DE INGENIERÍA ELÉCTRICA

de la

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO

José Ignacio Varela Caneo

Profesor Guía: Sr. Francisco Alonso Villalobos
Profesor Correferente: Sr. Jorge Zazópulos Del Fierro

MARZO DE 2017

ACTA DE APROBACIÓN

La Comisión Calificadora designada por la Escuela de Ingeniería Eléctrica ha aprobado el texto del Informe Final de Proyecto de Titulación, desarrollado entre el Segundo Semestre de 2012 y el Primer Semestre de 2013 y denominado

CALIDAD DE SERVICIO EN REDES MULTI-PROTOCOLO DE CONMUTACIÓN DE ETIQUETAS

Presentado por el Señor
José Ignacio Varela Caneo

Sr. Francisco Alonso Villalobos
Profesor Guía

Sr. Jorge Zazópulos Del Fierro
Segundo Revisor

Sr. Héctor Vargas Oyarzún
Secretario Académico

Valparaíso, MARZO DE 2017

Por su infinita paciencia y comprensión.
Por hacer aun más allá de su alcance y dar todo de sí durante todos estos años.
“No importa cuán duro el invierno, hay una primavera por delante”. PJ
A mis padres, José María Varela SM. y Olga Beatriz Caneo M.

AGRADECIMIENTOS

Agradezco a mis padres, José María Varela SM. y Olga Beatriz Caneo M. a quienes debo mi vida y gran parte de quién soy. Por su sacrificio incondicional en tiempos prósperos y adversos. Gracias al esfuerzo conjunto logramos estos días alcanzar un noble sueño, entendiendo este gran paso como un medio y no como un fin.

Agradecer a mi tía Celinda Rodríguez (Q.E.P.D) y tío Raúl Hernández por brindarme su casa en tiempos de transición, lo que significó un gran apoyo para poder estudiar en Viña del Mar.

También quiero agradecer a mi tío Guillermo Rivera, quien me ha apoyado desinteresadamente siempre que lo he necesitado en proyectos laborales y personales.

A María Cecilia Vargas, que, avanzado el tiempo de mi egreso, con amor y preocupación apoyó el término de este proceso para obtención de este título.

Gracias a mi grupo de estudio, por su paciencia y el apoyo que nos otorgamos mutuamente, por creer en mí en los tiempos en que dirigimos el Centro de Alumnos, por compartir más allá de lo académico.

Un especial agradecimiento a quienes han sido mis compañeros en la Rama Estudiantil IEEE, por permitirme dirigirlos muchas veces de manera enérgica, esperando que sigan viendo en mí y nuestros compañeros, un respaldo para futuras generaciones.

Mis más sinceras palabras de agradecimiento a ustedes.

CALIDAD DE SERVICIO EN REDES MULTI-PROTOCOLO DE CONMUTACIÓN DE ETIQUETAS

JOSÉ IGNACIO VARELA CANEO

Profesor Guía Sr. Francisco Alonso Villalobos

RESUMEN

Este trabajo corresponde a un estudio de metodologías existentes para aplicar Calidad de Servicio sobre redes MPLS, se presentan las características de las redes MPLS y cómo éstas por si mismas representan una solución a los problemas de congestión de tráfico que surgieron en la expansión de las redes a nivel mundial con la masificación de Internet. Se presentan los modelos de Calidad de Servicio, principalmente utilizando Servicios Diferenciados que corresponde a uno de los modelos más recientes y que actualmente es el que otorga las soluciones a la congestión en el transporte de datos.

A modo de pruebas, se lleva a cabo el análisis de un escenario montado primero de manera virtual a través de un simulador, y luego en laboratorio para realizar las mediciones de los parámetros que permitieron identificar y determinar los niveles de Calidad de Servicio deseables en un ambiente de red que representa el tráfico *end to end* de un cliente pasando a través de la red MPLS de un proveedor de servicio con altos niveles de congestión.

Para la implementación de Calidad de Servicio sobre MPLS, primero se realizó un análisis de las variables que se deben tener en cuenta para la comprensión técnica. Finalmente, para la comprobación del estudio se utilizó la herramienta IP SLA que incorporan los equipos Cisco utilizados en el escenario de laboratorio que permite medir de manera eficaz y rápida, el nivel de Calidad de Servicio deseado para cumplir un Acuerdo Legal de Servicios establecido entre las partes interesadas (servidor - cliente).

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

GLOSARIO DE TÉRMINOS

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

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

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

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

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

QoS: Calidad de servicio, del inglés *Quality of service*

MPLS: *Multirpotocol Label Switching*

PDR: *Peak Data Rate*

CDR: *Commited Data Rate*

PBS: *Peak Burst Size*

CBS: *Commited Burst Size*

EBS: *Excess Burst Size*

SLA: *Service Level Agreement*

ACL: *Access Control List*

SO: Sistema Operativo

DS: *Differentiated Services*

TTL: *Time to Live*

MOS: *Mean Opinion Scores*

ICPIF: *Calculated Planning Implement Factor*

ITU: *International Telecommunication Union*

PCM: *Pulse Code Modulation*

FIFO: *First in – First out*

IETF: *Internet Engineering Task Force*

FR: *Frame Relay*

RFC: *Request for Comments*

TCP: *Transmission Control Protocol*

UDP: *User Datagram Protocol*

TTL: *Time to live*

PING: *Packet Internet Groper*

NBAR: *Network Based Application Recognition*

VC:	*Virtual Circuit*
AS o ASN:	*Autonomous System Number*
FITCE:	*Federation of Telecommunications Engineers of the European Community*
ETSI:	*European Telecommunications Standards Institute*
ETR:	*ETSI Technical Report*

INTRODUCCIÓN

La actual demanda de aplicaciones relacionadas con información multimedia, como son la video-conferencia, audio-conferencia, video bajo demanda (VoD) o sistemas cooperativos (pizarras compartidas, teletrabajo, telemedicina y otros) y su coexistencia con aplicaciones más clásicas (bases de datos, transferencias de ficheros, www, entre otras), requieren tecnologías de comunicaciones capaces de ofrecer elevadas prestaciones.

Hace pocos años, debido básicamente a la baja capacidad de las redes, la posibilidad de llevar a cabo cualquiera de las aplicaciones referenciadas anteriormente era prácticamente impensable, pero en estos momentos es una realidad. Se ha avanzado mucho en compresión de audio y vídeo, y en tecnologías de redes. Aun así, quizás el mayor avance haya sido el auge de Internet y la capacidad de conectarse desde casa utilizando únicamente un ordenador personal y un módem.

Afortunadamente, en la actualidad se están implantando nuevas tecnologías de fibra óptica que proporcionan el gran ancho de banda requerido por las aplicaciones anteriores, pero no basta sólo con el aumento del mismo, es necesario gestionarlo de manera eficiente: utilizarlo en un porcentaje elevado asegurando una calidad determinada. Esto es lo que se conoce como calidad de servicio (QoS).

Hasta hace poco el término QoS no era importante en la mayoría de los sistemas. Para comprobarlo tan sólo se debe pensar en los algoritmos que se usan actualmente en la transmisión de paquetes por la red (pensar en el sistema *Best Effort* utilizado en Internet), estos algoritmos suelen garantizar la llegada de todos los paquetes, pero no dan ninguna cota respecto al límite de su llegada a destino. Esta forma de transmisión es buena para muchas aplicaciones, como por ejemplo la transmisión de ficheros (FTP), la navegación Web, el correo electrónico, donde lo importante es que los datos lleguen correctamente. Para el tráfico en tiempo real, en cambio, los datos necesitan llegar a su destino en un tiempo determinado, ya que tardar un poco más implica que la aplicación se detendría por falta de datos, lo cual sería inadmisible.

Durante los últimos años, *Multiprotocol Label Switching* MPLS ha sido desarrollado e implementado de manera exitosa por las compañías de *Carrier* más grandes del mundo, la madurez de internet y el nacimiento de las aplicaciones actuales han traído consigo mayores requerimientos en Calidad de Servicio. MPLS provee características poderosas que han resultado ser esenciales para otorgar QoS.

La mayoría de las redes de ordenadores, a excepción de ATM, no han sido diseñadas para proporcionar implícitamente unos niveles de calidad de servicio necesarios para la transmisión del tráfico multimedia. IP y Ethernet ofrecen un servicio *Best Effort* (mejor esfuerzo) inadecuado para la excesiva carga de las aplicaciones actuales, por lo tanto para poder soportar este tipo de tráfico es necesario utilizar distintos protocolos, así como una serie de políticas para la gestión de los diferentes recursos de la red, intentando obtener una calidad de servicio extremo a extremo y garantizando la compatibilidad de las distintas técnicas a causa de la heterogeneidad de las redes.

Debido a que los distintos tipos de tráfico de datos requieren diferente trato, regulaciones y servicios garantizados, estos tráficos se han vuelto más susceptibles a

varios obstáculos en la red, entre los cuales se tiene carencia de ancho de banda, retardo, latencia y pérdida de datos.

Las herramientas de QoS actuales, se han desarrollado como una alternativa al simple aumento de ancho de banda. Estos mecanismos de QoS están diseñados para proporcionar a las aplicaciones de manera específica un servicio garantizado o consistente en la ausencia de las condiciones óptimas de ancho de banda. [1]

El trabajo que se presenta en este informe, conjuga las características de MPLS como la tecnología que trajo consigo las redes orientadas a conexión sin utilización de IP junto a las características de QoS. Además, los servicios de redes y aplicaciones pueden explotar todas las ventajas de IP, mientras hacen uso de redes funcionales, confiables y predecibles.

CAPÍTULO 1

ANTECEDENTES DEL PROYECTO

1.1 Objetivos del Proyecto

A continuación se detallan los objetivos planteados para este proyecto de titulación.

1.1.1 Objetivos Generales

Este trabajo mostrará el desarrollo y avance de las redes y su problemática en cuanto al enrutamiento de los distintos tipos de tráficos que circulan por ella, realizando un estudio de las técnicas que permiten otorgar calidad de servicio de un extremo a otro a través de redes que contemplan en su trayecto el protocolo MPLS.

El enfoque principal, se centrará en conocer las características de MPLS y QoS, Calidad de Servicio, de su sigla en inglés para *Quality of Service*; y de qué manera ambas herramientas interactúan para brindar mejoras significativas al transporte de datos, que permiten la sustentación de las redes de comunicación actuales.

1.1.2 Objetivos Específicos

El estudio contempla las diferentes herramientas que permiten otorgar calidad de servicio a través de priorización de tráfico. Tras definir una estructura de red, se verificará el uso de determinadas configuraciones en un entorno de simulación que luego será trasladado al laboratorio. Así, finalmente demostrar a través de mediciones concretas, el comportamiento y la efectividad de las tecnologías actuales que permiten garantizar calidad de servicio.

Se utilizará para este propósito tecnología CISCO disponible en el Laboratorio de Redes de la Escuela de Ingeniería Eléctrica de la Pontificia Universidad Católica de Valparaíso (PUCV).

1.1.3 Objetivos Académicos

Con este proyecto se comprenderá el funcionamiento de la tecnología MPLS, su utilidad en las redes actuales, su adaptabilidad y capacidad de prestar funciones que otorgan calidad de servicio a las redes complejas que hoy se conforman tanto a nivel interno en las organizaciones y externo, desde la perspectiva de los proveedores de servicio.

La documentación de las implementaciones tanto simuladas como de laboratorio, será de gran utilidad a la hora de requerir entender conceptos claves de los mecanismos y configuraciones que permiten otorgar Calidad de Servicio.

1.2 Descripción de la problemática y su estado actual

1.2.1 Estado del arte

Internet es una red de computadores que se desarrolló siguiendo el modelo de datagrama, es decir, provee un servicio no confiable no orientado a la conexión, donde cada paquete se enruta de forma independiente. Se la utilizó principalmente para transferencia de datos y compartición de información, por lo que tradicionalmente las aplicaciones más utilizadas fueron de acceso remoto, transferencia de archivos y correo electrónico.

Hoy en día las aplicaciones que se están desplegando, tales como la telefonía sobre IP o video-conferencia, tienen fuertes requerimientos de anchos de banda, retardos acotados, o de tasa de pérdidas de paquetes extremadamente bajas. Debido a los cambios de uso de la Internet, se está trabajando en herramientas que permitan asegurar a las aplicaciones un cierto nivel de desempeño de la red, de modo de satisfacer los requerimientos de tiempo real.

También se trabaja en diferenciar el tráfico, pues distintas aplicaciones tienen distintas necesidades de recursos. Puesto que en Internet todos los paquetes son tratados de la misma forma, y en caso de congestión todos los flujos sufren igual baja de desempeño (disminución de ancho de banda disponible, aumento de pérdidas de paquetes, entre otras), se dice que la Internet sólo da un tipo de servicio, conocido como *best-effort* (se obtiene lo que se puede con el mejor esfuerzo).

Asegurar los recursos de las aplicaciones, y diferenciar servicios, se conoce como Calidad de Servicio (QoS, *Quality of Service*). Por ello, la IETF ha trabajado en cuatro tópicos relacionados con la entrega de QoS. Para la distribución de recursos, se desarrollaron las arquitecturas Servicios Integrados (*IntServ*) y Servicios Diferenciados (DiffServ). Para optimizar el uso de estos recursos se trabaja en la arquitectura *Multiprotocol Label Switching* (MPLS) y en técnicas y herramientas de Ingeniería de Tráfico.

1.2.2 Distribución de recursos o diferenciación de servicios

La Internet consiste en un conjunto de recursos finitos compartidos (ancho de banda, colas o *buffers*, ente otros). En caso de congestión, dichos recursos se vuelven un bien escaso y valioso, y muchos paquetes pueden ser delegados en una cola, o aún descartados. Si la red pretende dar QoS, debe decidir activamente quién dispondrá de los recursos y en qué cantidad. Sin embargo, con el modelo clásico de colas FIFO (*First In-*

First Out) sin discriminación de flujo, todos los paquetes reciben el mismo tratamiento y el mismo desempeño.

Las arquitecturas de distribución de recursos desarrolladas proveen:

- Marcos para distribuir recursos que soporten el aseguramiento de recursos y diferenciación de servicios.
- Nuevos modelos de servicios, además del *best-effort*.
- Lenguaje para describir el aseguramiento de recursos y los requerimientos de los mismos.
- Mecanismos para forzar la distribución de recursos.

1.2.3 Servicios Integrados

Fue la primera arquitectura de QoS desarrolladas por la IETF a principios de los 90s. En el diseño se consideró que las aplicaciones de tiempo real serían las más importantes y sensibles de las aplicaciones necesitando QoS.

Esta arquitectura está basada en reservación de recursos por flujo. Para que se le aseguren recursos, una aplicación debe hacer una reservación antes de transmitir. Primero se caracteriza la fuente del flujo y sus requerimientos; la red luego utiliza un algoritmo de ruteo para elegir la ruta que cumpla con dichos requerimientos; y por último un protocolo de reservación establece el estado de reserva a lo largo de la ruta (cada nodo debe chequear si hay recursos disponibles antes de aceptar la reserva).

Se definen dos modelos de servicio nuevos a elegir por los usuarios: Servicio Garantizado (*Guaranteed Service*), y Servicio de Carga Controlada (*Controlled Load Service*). El primero da una cota de *delay* máximo determinístico, a través de control de admisión. El segundo modelo no da garantías y sólo simula una red débilmente cargada. El protocolo RSVP (protocolo de reserva de recursos) se estandarizó para señalizar los requerimientos de la aplicación a la red, y establecer la reservación a lo largo de la ruta.

Esta arquitectura presenta problemas de escalabilidad, pues se requiere una reserva explícita para cada flujo de datos (y en la Internet estos pueden ser varios miles, cada uno con una entrada en una tabla de admisión). Además se traduce en un *overhead* innecesariamente grande para aplicaciones web. Por esto, Servicios Integrados parece tener futuro principalmente en redes corporativas, donde desaparecen los problemas de escalabilidad, y hay demanda de telefonía IP y video-conferencia en las *intranets*. Para la Internet, sin embargo, se hizo patente que se requería otra arquitectura que fuese más simple y escalable, y que pudiera proveer un servicio mejor al *best-effort*; ahí surge Servicios Diferenciados.

1.2.4 Arquitectura *IntServ*

Esta arquitectura consiste en un conjunto de mecanismos y protocolos usados para hacer reservas explícitas de recursos en Internet. Para que la red asegure recursos a una aplicación, esta debe establecer una reservación de recursos a lo largo de la ruta antes de enviar paquetes. El transmisor comienza describiendo las características del flujo y los requerimientos de recursos a la red; la red acepta la petición si puede satisfacer la demanda; finalmente la reserva se establece en la ruta, y el transmisor puede enviar paquetes.

La arquitectura asume que los contratos de calidad de servicio se refieren principalmente a cotas de *delay* máximo. Este parámetro se consideró como el más importante para aplicaciones de tiempo real.

1.2.5 Servicios Diferenciados (DS)

Como se mencionó anteriormente, DiffServ surge como una alternativa a IntServ más simple y escalable, que da un mejor servicio que *best-effort*. En vez de una reserva explícita por flujo, se usa políticas de borde (*edge policing*), aprovisionamiento, y priorización de tráfico para brindar diferenciación de tráfico.

El tráfico de los usuarios se divide en un número pequeño de Clases de Transferencia (*Forwarding Classes*). Para cada una de estas clases, la cantidad de tráfico que se puede inyectar a la red está limitada por el borde de la red (*Edge Policing*). Este borde también mapea los paquetes entrantes en una de las clases, en función de un Acuerdo de Nivel de Servicio (SLA, *Service Level Agreements*). La clase se codifica en el paquete, y da cuenta de prioridad, de descarte o de cola, entre otras.

DiffServ se apoya en aprovisionamiento de recursos para asegurar recursos a las aplicaciones. Como no hay reservas explícitas, se debe cuidar que la red tenga suficientes recursos para satisfacer la demanda.

El IETF ha proporcionado una nueva definición del campo TOS en el RFC 2474, llamada ahora DSCP. DS es usado como un indicador de un cierto perfil de QoS que debe aplicarse a ese paquete de datos. La figura 1-1 muestra el campo DSCP en la cabecera IP.

1.2.6 Arquitectura *DiffServ*

La arquitectura de servicios diferenciados nace como respuesta a la necesidad de otorgar diferentes niveles de servicio de una manera más simple y gruesa en la Internet. En vez de otorgar recursos a flujos individuales, se otorgan a un número pequeño de clases que agrupan varios flujos. No se requiere reservación previa, ni procesos de clasificación complejos al interior de la red, por lo que mejora la escalabilidad.

1.2.7 Optimización de desempeño

La optimización de desempeño es la forma de organizar los recursos de la forma más eficiente de modo de maximizar la probabilidad de proveer los requerimientos y minimizar el costo de esto. La ingeniería de tráfico se preocupa de lograr estos objetivos mediante aprovisionamiento eficiente de recursos, y control de los flujos de la red. El control de flujo es fundamental, pues en la red IP clásica, con ruteo basado en direcciones de destino, típicamente algunos enlaces están fuertemente saturados mientras otros tienen poca carga, lo que se traduce en un uso deficiente de la red. Por ello, es crucial el ser capaces de elegir explícitamente la ruta de un flujo, de modo de distribuirlos eficientemente por la red. La herramienta que surgió para esta labor es MPLS, que es un protocolo que permite levantar Circuitos Virtuales, que son rutas explícitamente definidas en la red; la elección de la ruta se puede hacer mediante algoritmos de ruteo clásicos (que no se traducen en uso eficiente de la red), o mediante algoritmos de ruteo explícito con restricciones (que sí dan uso eficiente de la red).

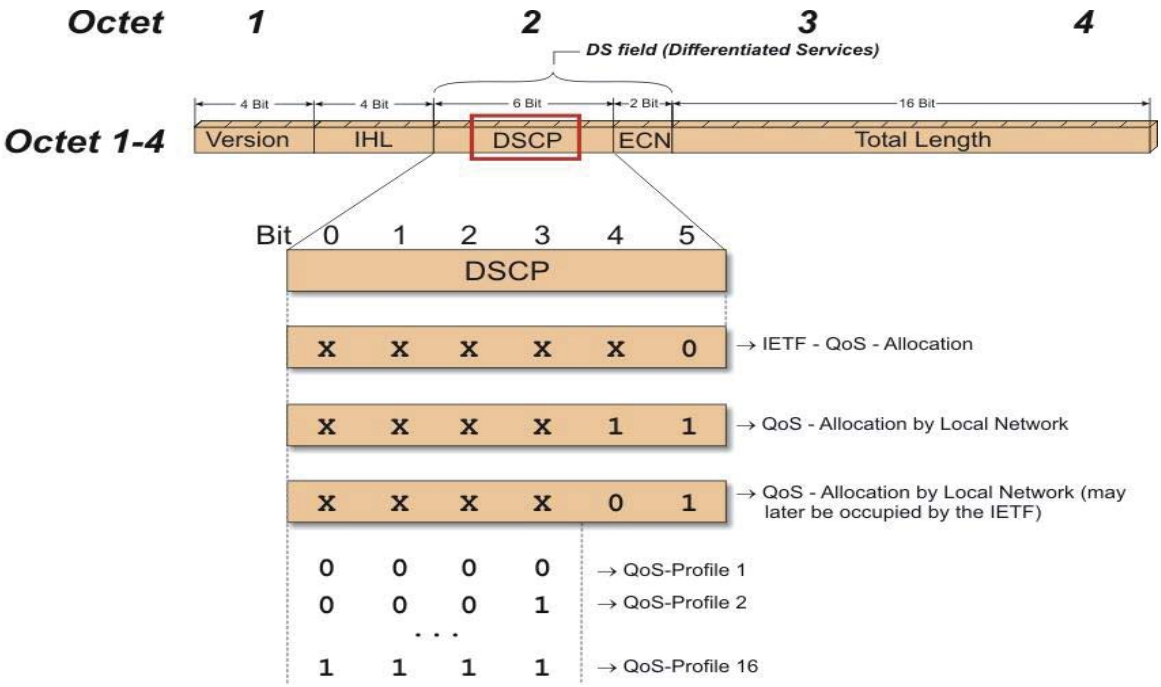


Fig. 1-1 Campo DSCP [2]

1.2.8 *Multiprotocol Label Switching*

MPLS surge a fines de los 90s como una forma de integrar ATM e IP en los *backbones* (troncales de la red).

Hace uso de una técnica conocida como *Label Switching*, que codifica una etiqueta en la cabecera del paquete que se usa para retransmitir el mismo (*forwarding*). Cuando un *router* que soporta MPLS recibe un paquete, decide la interfaz de transmisión según la etiqueta, en vez de hacerlo mediante un algoritmo de ruteo.

1.2.9 Ingeniería de Tráfico

Trata de distribuir los flujos de modo de usar eficientemente los recursos de la red, distribuyendo el tráfico, lo cual disminuye la congestión en la misma.

Como los algoritmos de ruteo clásicos no permiten la optimización, se hace uso de algoritmos de ruteo basados en restricciones (*constraint-based routing*). También se hace necesario conocer la topología de la red, la demanda de tráfico, entre otros.

1.2.10 Modelos de servicio

Los modelos de servicio describen como la red distribuye los recursos a los usuarios. Indica qué servicios pueden pedir los usuarios a la red y qué contratos puede ofrecer ésta última.

Para hacer una reservación, una aplicación debe primero especificar los requerimientos del flujo (*flow specification*). Representa un contrato de servicio donde la aplicación se compromete a enviar un tipo de tráfico, y la red satisfará las demandas. La aplicación deberá mantener lo estipulado en su especificación de modo de obtener el servicio acordado, pues si inyecta más paquetes, la red ya no tendrá suficientes recursos para cumplir; por ello se realiza usualmente algún tipo de tratamiento del flujo en el nodo de ingreso a la red (*policing*).

Por otra parte, los parámetros para describir requerimientos de calidad de servicio son:

- Ancho de Banda Mínimo (*Minimum Bandwidth*): El mínimo ancho de banda requerido por el flujo de una aplicación. Este se garantiza mediante algoritmos de despacho (WFQ).
- *Delay*: Puede ser *delay* medio o el *delay* de peor caso. Tiene 3 componentes: *delay* de propagación (velocidad de la luz en el medio), *delay* de transmisión (ancho de banda finito provoca serialización del paquete), y *delay* en cola. Estos dos últimos se pueden convertir a requerimientos de ancho de banda.
- Dispersión de *Delay* (*Delay Jitter*): Es la máxima diferencia entre el *delay* mínimo y el *delay* máximo.
- Tasa de pérdida (*Loss Rate*): Es la tasa de paquetes que se pierden. Usualmente es debido a congestión.

1.3 Resultados esperados

Este trabajo apunta a buscar una manera práctica de realizar pruebas de tráfico de distintos tipos de datos, estableciendo los requerimientos para dichas pruebas en un laboratorio de redes. Esto, con la finalidad de contar con este proyecto de título como una herramientas simplificada que permita la comprensión y rápida asimilación de los conceptos básicos tanto de Calidad de Servicio, como de redes MPLS. De esta manera, poder desarrollar distintas pruebas con el objetivo de medir diferentes instancias de redes y la aplicabilidad de métodos específicos que otorguen Calidad de Servicio.

CAPÍTULO 2

CARACTERÍSTICAS GENERALES DE MPLS

2.1 Inicios de MPLS

MPLS es el esfuerzo de llevar características de circuitos virtuales al mundo IP. En la red IP tradicional, se tiene el modelo de datagrama, que consiste en que cada paquete se rutea en forma independiente, pudiendo los paquetes seguir rutas distintas y llegar en desorden. En redes orientadas a la conexión, como ATM y *Frame Relay*, se establece un circuito virtual antes de enviar datos, de modo que los paquetes siguen siempre el mismo camino y en orden. MPLS usa una etiqueta corta que se inserta en la cabecera de los paquetes para conmutar los mismos (elegir la interfaz de salida), de manera análoga a ATM o Frame Relay. Cada etiqueta está asociada a un circuito virtual.

El multiprotocolo de conmutación de etiquetas (MPLS) reduce significativamente el procesamiento de paquetes que se requiere cada vez que un paquete ingresa a un enrutador en la red, esto mejora el desempeño de dichos dispositivos y del desempeño de la red en general. Dicho protocolo se puede considerar en desarrollo constante ya que en los últimos años la demanda de esta tecnología ha ido creciendo. Las capacidades más relevantes de dicho protocolo son cuatro: Soporte de Calidad sobre servicio (QoS), Ingeniería de tráfico, soporte para Redes Privadas Virtuales (VPNs) y soporte multiprotocolo.

La motivación original era integrar IP y ATM en los *backbones*, de modo de simplificar la administración de las redes. Con MPLS, el plano de control IP (agentes de ruteo) administran la implementación de los circuitos virtuales de la capa inferior.

La implementación del circuito se logra mediante un protocolo de señalización, tal como LDP, CR-LDP, o RSVP-TE.

2.1.1 Vista General

En MPLS, un circuito virtual se llama *Label Switched Path* (LSP), un *router* que re-envía tráfico basado en etiquetas MPLS se denomina *Label Switched Router* (LSR), y la etiqueta se llama *Label* (Etiqueta). Un LSR decide la interfaz y el próximo salto (*hop*) para reenviar un paquete, consultando una tabla (*label switching table*) con la entrada de la etiqueta del paquete. Esta tabla entrega además la etiqueta con que se debe reenviar el paquete al próximo salto, pues las etiquetas tienen significado local a la interfaz.

El ejemplo de la figura 2-1 muestra 4 sitios de clientes (1, 2, 3, 4), conectados por un *backbone* MPLS compuestos por LSRs (A, B, C, D, E). Hay dos LSPs establecidos. Uno conecta al cliente 1 con 3, usando la ruta A->C->E, con etiqueta 23 y 42. Otro conecta al cliente 2 con 4, mediante la ruta A->B->D->E, con etiquetas 12, 96, 24. Si el cliente 1 envía tráfico a 3, los paquetes llegan a LSR A (el LSR de ingreso, el LSR de inicio de un LSP), el cual añade la etiqueta 23 a los mismos. Cuando el LSR 23 recibe los paquetes, utiliza la etiqueta 23 como entrada a la tabla de *forwarding*, y encuentra que se

debe reenviar el paquete al nodo E, reemplazando la etiqueta con el valor 42. Cuando E recibe a su vez el paquete, consulta la tabla de *forwarding*, se percata que él termina el LSP (es decir, es un LSR de egreso), remueve la etiqueta, y reenvía el paquete a 3.

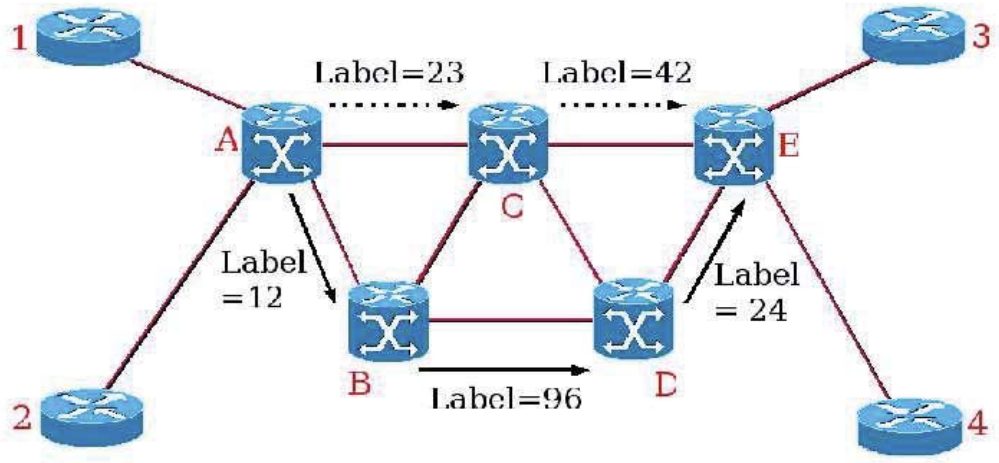


Fig. 2-1 Cuatro clientes conectados a un *Backbone* MPLS

La figura 2-2 también es un ejemplo de cómo se genera el intercambio de paquetes y asignación de etiquetas a través de una red MPLS.

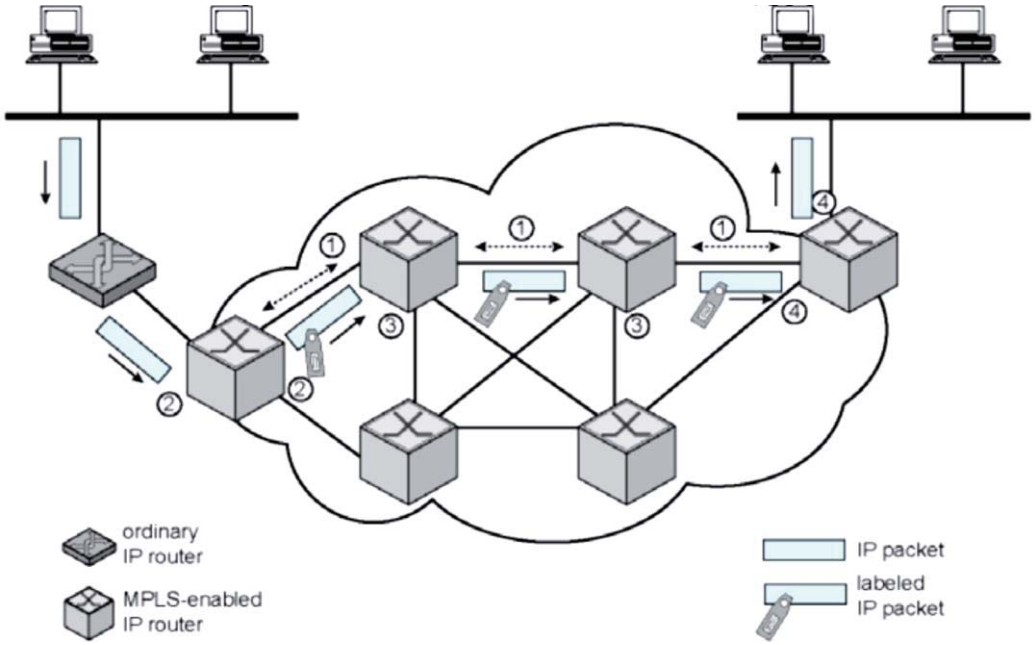


Fig. 2-2 Intercambio de paquetes en una red MPLS [3]

2.1.2 Ingeniería de Tráfico

Es la habilidad de definir rutas dinámicamente y planear la asignación de recursos con base en la demanda, así como optimizar el uso de la red. MPLS facilita la asignación de recursos en las redes para balancear la carga dependiendo de la demanda y proporciona diferentes niveles de soporte dependiendo de las demandas de tráfico de los usuarios. El protocolo IP provee una forma primitiva de Ingeniería de Tráfico al igual que el protocolo del Camino Más Corto Primero (OSPF) que permite a los enrutadores cambiar la ruta de los paquetes cuando sea necesario para balancear la carga. Sin embargo, esto no es suficiente ya que este tipo de ruteo dinámico puede llevar a congestionar la red y no soporta QoS.

Todo tráfico entre dos puntos finales (*endpoints*) sigue la misma ruta y puede ser cambiada si ocurriera congestión, sin embargo este cambio sólo ocurre cuando hay congestión, que es algo que siempre se trata de evitar.

En MPLS a diferencia de OSPF no se ve paquete por paquete sino flujos de paquetes con su respectivo QoS y demanda tráfico predecible. Con este protocolo es posible predecir rutas en base a flujos individuales, pudiendo haber diferentes flujos entre canales similares pero dirigiéndose a diferentes enrutadores.

Si llegase a amenazar congestión en la red, las rutas MPLS pueden ser re-ruteadas inteligentemente, de esta manera se pueden cambiar las rutas de flujo de paquetes dinámicamente conforme a las demandas de tráfico de cada flujo.

2.1.3 Beneficios principales de Ingeniería de Tráfico MPLS

- Permite al eje troncal (*backbone*) expandirse sobre las capacidades de la ingeniería de tráfico de las redes de Modo de Transferencia Asíncrona (ATM) y *Frame Relay* (FR) de Capa 2.
- La ingeniería de tráfico es esencial para los ejes troncales de proveedores de servicios. Dichos ejes deben soportar un uso elevado de su capacidad de transmisión.
- Utilizando MPLS las capacidades de ingeniería de tráfico son integradas a la Capa 3 (OSI), lo que optimiza el ruteo de trafico IP gracias a las pautas establecidas por la topología y las capacidades de la red troncal.
- La ingeniería de trafico MPLS rutea el flujo de tráfico a lo largo de la red basándose en los recursos que dicho flujo requiere y en los recursos disponibles en toda la red.
- MPLS emplea la ruta más corta que cumpla con los requisitos del flujo de tráfico, que incluye: requisitos de ancho de banda, de medios y de prioridades sobre otros flujos.

2.1.4 ¿Cómo funciona la Ingeniería de Trafico MPLS?

MPLS es básicamente una integración de tecnologías de Capa 2 a Capa 3, dicha integración permite manejar el tráfico a conveniencia. De esta manera, el flujo de paquetes viaja a través de un túnel de datos en el eje troncal creado por el Protocolo de Reserva de Recursos (RSVP), la ruta de dicho túnel está dada por los requisitos de recursos del túnel y de la red (*constraint-based routing*). El Protocolo de Enrutamiento Interno (IGP) rutea el tráfico a dichos túneles.

Con un buen manejo del tráfico en las redes, se pueden evitar congestionamientos, mejorar el desempeño general y reducir la latencia y el descarte de paquetes. En pocas palabras se maximiza la capacidad de la red y se minimizan los costos.

2.1.5 Soporte de Redes Virtuales Privadas (VPN)

MPLS provee un mecanismo eficiente para el manejo de redes privadas virtuales. De esta manera, el tráfico de una red privada “atraviesa” la Internet eficazmente y de manera transparente para el usuario, eliminando cualquier tráfico externo y protegiendo la información.

Las VPN creadas con tecnología MPLS tienen una mayor capacidad de expansión y son más flexibles en cualquier red, principalmente IP. MPLS se encarga de reenviar (*forward*) paquetes a través de túneles privados utilizando etiquetas que actúan como códigos postales. Dicha etiqueta tiene un identificador que la aísla a esa VPN.

Las ventajas principales de implementar MPLS en VPN son:

- Maximizar la capacidad de ampliación.
- Actualización transparente para el usuario.
- Utilización óptima de los recursos de la red.
- Diferenciación entre servicios.
- Reducción de costos mediante consolidación de servicios.
- Seguridad y rapidez de transmisión de información.
- Uso de tecnología de vanguardia.

2.1.6 Soporte Multiprotocolo

MPLS puede ser utilizando con diversas tecnologías, es decir no es necesario actualizar los *routers* IP existentes. Los *routers* MPLS pueden trabajar con *routers* IP a la par, lo que facilita la introducción de dicha tecnología a redes existentes ya que está

diseñada para trabajar con redes ATM y *Frame Relay*. Al igual que los *routers*, los *switches* MPLS pueden trabajar con *switches* normales.

Esta tecnología puede trabajar con tecnologías puras como son IP Internet, ATM y *Frame Relay*. Todo esto con la ventaja de tener redes mixtas añadiendo QoS para optimizar y expandir los recursos.

Estas características hacen de MPLS una tecnología innovadora que se puede aplicar a redes nuevas así como a redes ya existentes.

La figura 2-3 ilustra los protocolos que soporta MPLS.

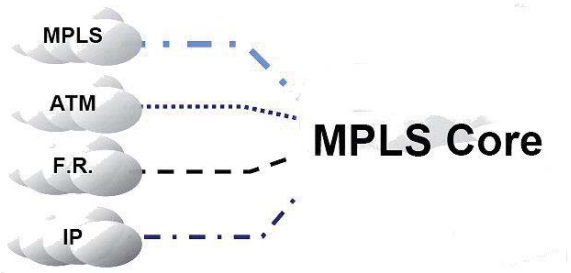


Fig. 2-3 Soporte Multiprotocolo de MPLS [4]

2.2 Ventajas de MPLS sobre otras tecnologías

Las soluciones más comunes para implementar redes privadas son las siguientes:

- Frame Relay
- Circuitos ATM
- Túneles tradicionales (IP-IP y GRE)
- IPSec
- L2F
- PPTP
- L2TP
- MPLS (a manera de comparación)

2.2.1 Arquitectura

La figura 2-4 muestra una red MPLS sencilla. Un *router* que inicia un LSP se denomina de ingreso (*ingress* LSR), como el LSR A. Un *router* que termina un LSP se denomina de egreso (*egress* LSR), como el LSR C. El *router* al interior del LSP se llama intermediario (*intermediate* LSR). Cada uno realiza una operación diferente de los otros

tipos. Además los LSP son direccionales; para un par de *routers*, el LSR que envía tráfico se llama *upstream* LSR, y el que recibe se denomina *downstream* LSR.

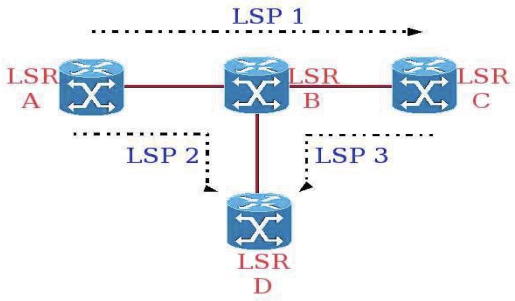


Fig. 2-4 Etiquetas de rutas en MPLS

2.2.2 Etiqueta (*Label*)

Un *label* es una etiqueta pequeña, de tamaño fijo, y de significado local, usada para la conmutación de etiquetas. Un paquete se denomina paquete etiquetado (*labeled packet*), si la etiqueta está codificada en el paquete. El enfoque más común es un encapsulamiento MPLS que lleva la etiqueta e información adicional, añadiendo una capa fina entre la capa de enlace de datos y la capa de red (IP).

Un LSR debe ser capaz de mapear una etiqueta con un LSP, por lo que la etiqueta debe ser la mayoría de las veces única en un LSR particular.

Además, cada etiqueta está asociada con una *Clase equivalente de envío* (*Forwarding Equivalent Class*, FEC), que corresponde a un grupo de flujos IP que viajan por un mismo LSP y reciben igual trato de envío.

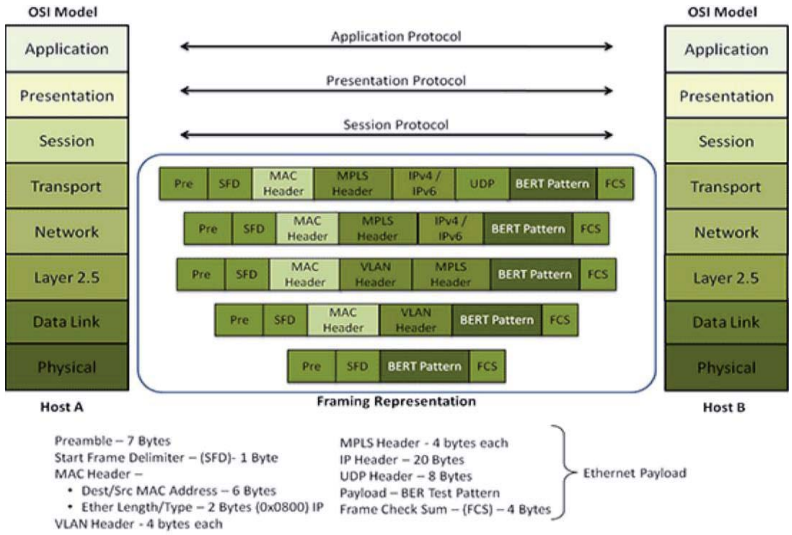


Fig. 2-5 Etiquetas de MPLS y posición en modelo OSI

En la figura 2-5, se observa el modelo de capas incluyendo la capa 2.5 intermedia y su relación directa con las etiquetas utilizadas en MPLS.

2.2.3 Encabezado (*Shim-Header*)

MPLS incluye un encabezado antes del paquete de capa 3, conocido como *shim-header*. Éste contiene el etiquetado o *label*, 3 bits experimentales usados actualmente para mapear Clases de Servicio (CoS), un bit para indicar si hay más *labels* (*shim-headers*) en la pila, y 8 bits para TTL. La figura 2-6 a continuación, ilustra la estructura del encabezado MPLS y cómo se distribuyen los bits para cada una de sus partes.

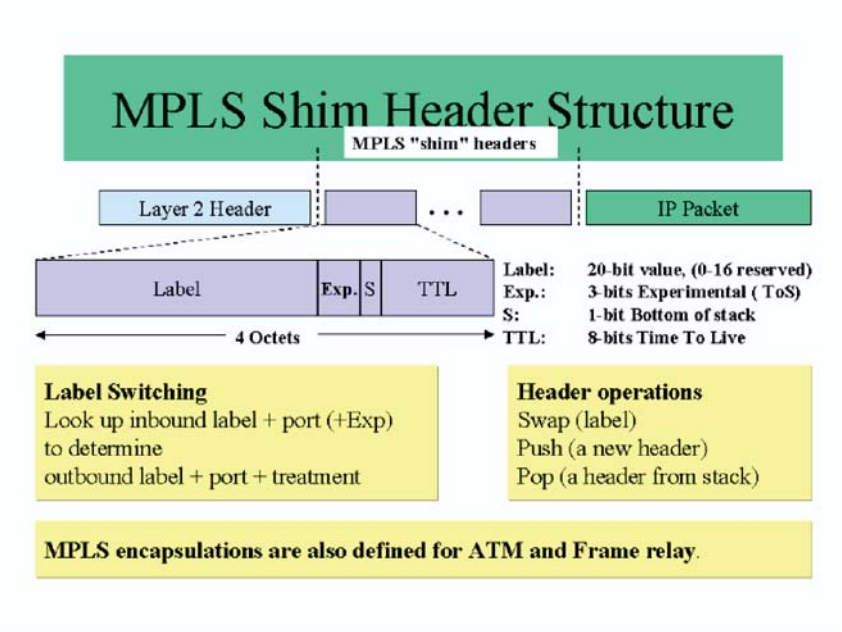


Fig. 2-6 Estructura de encabezado MPLS

2.2.4 Pila Jerárquica de Etiquetas

MPLS permite codificar más de una etiqueta en el paquete, lo cual se conoce como pila de etiquetas (*label stack*). Esto permite dar soporte a túneles anidados (Un LSP que viaja en el interior de otro LSP).

2.2.5 Funcionamiento básico de MPLS

Una red MPLS consiste de un conjunto de Enrutadores de Conmutación de Etiquetas (LSR) que tienen la capacidad de conmutar y rutear paquetes en base a la etiqueta que se ha añadido a cada paquete. Cada etiqueta define un flujo de paquetes entre dos puntos finales. Cada flujo es diferente y es llamado Clase de Equivalencia de Reenvío (FEC), así como también cada flujo tiene un camino específico a través de los LSR de la red, es por eso que se dice que la tecnología MPLS es “orientada a conexión”. Cada FEC, además de la ruta de los paquetes contiene una serie de caracteres que define los requerimientos de QoS del flujo. Los *routers* de la red MPLS no necesitan examinar ni procesar el encabezado IP, sólo es necesario reenviar cada paquete dependiendo el valor de su etiqueta. Esta es una de las ventajas que tienen los *routers* MPLS sobre los *routers* IP, en donde el proceso de reenvío es más complejo.

En un *router* IP cada vez que se recibe un paquete se analiza su encabezado IP para compararlo con la tabla de enrutamiento (*routing table*) y ver cuál es el siguiente salto (*next hop*). El hecho de examinar estos paquetes en cada uno de los puntos de tránsito que deberán recorrer para llegar a su destino final, significa un mayor tiempo de procesamiento en cada nodo y por lo tanto, una mayor duración en el recorrido.

2.2.6 Ventajas específicas de MPLS

En este momento ya es posible identificar algunas de las ventajas internas más importantes que MPLS presenta:

- Un domino MPLS, consiste de una serie de *routers* habilitados con MPLS continuos y contiguos. El tráfico puede entrar por un punto final físicamente conectado a la red, o por otro *router* que no sea MPLS y que esté conectado a una red de computadoras sin conexión directa a la nube MPLS.
- Se puede definir un Comportamiento por Salto (PHB) diferente en cada *router* de la FEC. El PHB define la prioridad en la cola y las políticas de desechado de los paquetes.
- Para determinar el FEC se pueden utilizar varios parámetros que define el administrador de la red.

a) Dirección IP fuente o destino y/o las direcciones IP de la red.

b) Utilizar el ID del protocolo IP.

c) Etiqueta de flujo IPv6.

d) Numero de puerto de la fuente o del destino.

e) El punto de código (*codepoint*) de los servicios diferenciados (DSCP).

- El reenvío de la información se lleva a cabo mediante una búsqueda simple (*lookup*) en una tabla predefinida que enlaza los valores de las etiquetas con las direcciones del siguiente salto (*next hop*).

- Los paquetes enviados de mismos *endpoints* pueden tener diferente FEC, por lo que las etiquetas serán diferentes y tendrán un PHB distinto en cada LSR. Esto puede generar diferentes flujos en la misma red.

2.2.7 Enrutamiento en los bordes y *Switching* en el centro

La estructura general MPLS se basa en que los enrutadores de los extremos de la nube son los que realizan el mayor trabajo. Los Enrutadores de Etiquetas Frontera (LER) que realizan las labores de ruteo de paquetes con funciones de decisión de rutas manejan muchísima información, extremadamente complicada, que puede estar basada en la interfaz de entrada, en los valores del encabezado de red, en la red a la que pertenece, en el tipo de tráfico, entre otros. Después se tienen los dispositivos internos LSR que hacen conmutación de paquetes a velocidades impresionantes y con eficacia incomparable, ya que solo es necesario leer la etiqueta del encabezado MPLS que define hacia dónde va y de donde viene. Como se muestra en la figura 2-7 los LER de los extremos añaden y quitan etiquetas y asignan FEC, los conmutadores trabajan basándose en las etiquetas MPLS.

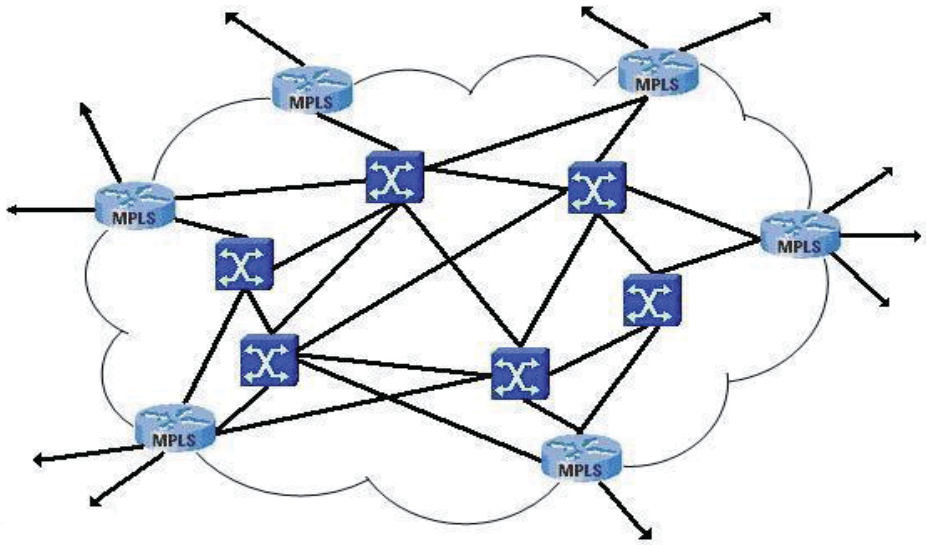


Fig. 2-7 Dominio MPLS

CAPÍTULO 3

CALIDAD DE SERVICIO

3.1 QoS (*Quality of Service*)

En el ámbito de las telecomunicaciones, en 1984 el documento E-800 de la UIT, define el término QoS como: “el efecto colectivo del rendimiento de un servicio que determina el grado de satisfacción del usuario de dicho servicio”. Es una definición comúnmente aceptada, que no deja ninguna duda de que se trata de una percepción del usuario, pues es éste quién, al final, establece unos requerimientos mínimos para cualificar.

En el ámbito de la telemática, QoS es la capacidad de un elemento de red (bien una aplicación, un servidor, un enrutador, un conmutador, entre otros) de asegurar que su tráfico y los requisitos del servicio previamente establecidos puedan ser satisfechos. Habilitarla requiere además la cooperación de todas las capas de la red, así como de cada elemento de la misma.

Desde este punto de vista, la QoS también suele 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 la capacidad. En este punto es necesario prestar atención especial al hecho de que la QoS no crea ancho de banda.

La QoS tiene, básicamente, cuatro variantes estrechamente relacionadas: la QoS que el usuario desea, la que el proveedor ofrece, la que el proveedor consigue realmente y la que, finalmente, percibe el usuario. En cualquiera de ellas existen algunos parámetros que están muy condicionados por las características técnicas de la red soporte, y por eso el primer Informe Técnico que publicó, en 1994, el ETSI fue la ETR-003, “*General Aspects of Quality of Service (QoS) and Network Performance (NP)*”, atendiendo a las inquietudes surgidas en el seno de FITCE, que tuvieron su reflejo oficial en los acuerdos de la reunión de Estrasburgo, de 1991, poniendo en marcha los estudios que permitiesen definir los parámetros técnicos de la red, a partir de los requisitos de los usuarios. La metodología resultante es la que se refleja en el documento de ETSI, antes citado.

3.1.1 Soporte QoS

QoS permite a los administradores de redes, el uso eficiente de los recursos de sus redes con la ventaja de garantizar que se asignaran más recursos a aplicaciones que así lo necesiten, sin arriesgar el desempeño de las demás aplicaciones. En otras palabras el uso de QoS le da al administrador un mayor control sobre su red, lo que significa menores costos y mayor satisfacción del cliente o usuario final.

3.1.2 ¿Por qué la importancia de QoS?

En los últimos años el tráfico de redes ha aumentado considerablemente, la necesidad de transmitir cada vez más información en menos tiempo, como video y audio en tiempo real (*streaming media*). La solución no es solo aumentar el ancho de banda (*bandwidth*) cada vez más, ya que en la mayoría de los casos esto no es posible y además es limitado. Es aquí donde la administración efectiva de recursos que provee QoS entra a relucir.

3.1.3 Beneficios principales de QoS

QoS trabaja a lo largo de la red y se encarga de asignar recursos a las aplicaciones que lo requieran, dichos recursos se refieren principalmente al ancho de banda. Para asignar estos recursos QoS se basa en prioridades, algunas aplicaciones podrán tener más prioridades que otras, sin embargo se garantiza que todas las aplicaciones tendrán los recursos necesarios para completar sus transacciones en un periodo de tiempo aceptable.

En resumen, QoS otorga mayor control a los administradores sobre sus redes, mejora la interacción del usuario con el sistema y reduce costos al asignar recursos con mayor eficiencia (*bandwidth*). Mejora el control sobre la latencia (*Latency* y *jitter*) para asegurar la capacidad de transmisión de voz sin interrupciones y por último disminuye el porcentaje de paquetes desechados por los enrutadores: confiabilidad (*Reliability*). MPLS impone un marco de trabajo orientado a conexión en un ambiente de Internet basado en IP y facilita el uso de SLA de tráfico exigente.

3.1.4 Cuándo aplicar QoS

Sería muy fácil dar Calidad de Servicio si las redes nunca se congestionaran. Para ello, habría que sobredimensionar todos los enlaces, cosa no siempre posible o deseable.

Para dar QoS cuando existe congestión, es preciso tener mecanismos que permitan dar un trato distinto al tráfico preferente y cumplir el SLA (*Service Level Agreement*).

El SLA suele ser estático y definido en el momento de negociación del contrato con el proveedor de servicio o ISP (*Internet Service Provider*).

La figura 3-1 grafica en qué momentos de la congestión es posible aplicar QoS.

Es sabido que incluso desde una perspectiva de optimizar el uso global de los recursos no es deseable una excesiva carga en los enlaces. Cuando la carga aumenta, el tiempo de servicio crece de forma exponencial y como consecuencia de esto, las aplicaciones no pueden funcionar o retransmiten la información que creían perdida. Por tanto, a partir de un cierto nivel de carga, no sólo crece el tiempo de servicio, sino que disminuye el rendimiento obtenido del enlace debido a las retransmisiones.

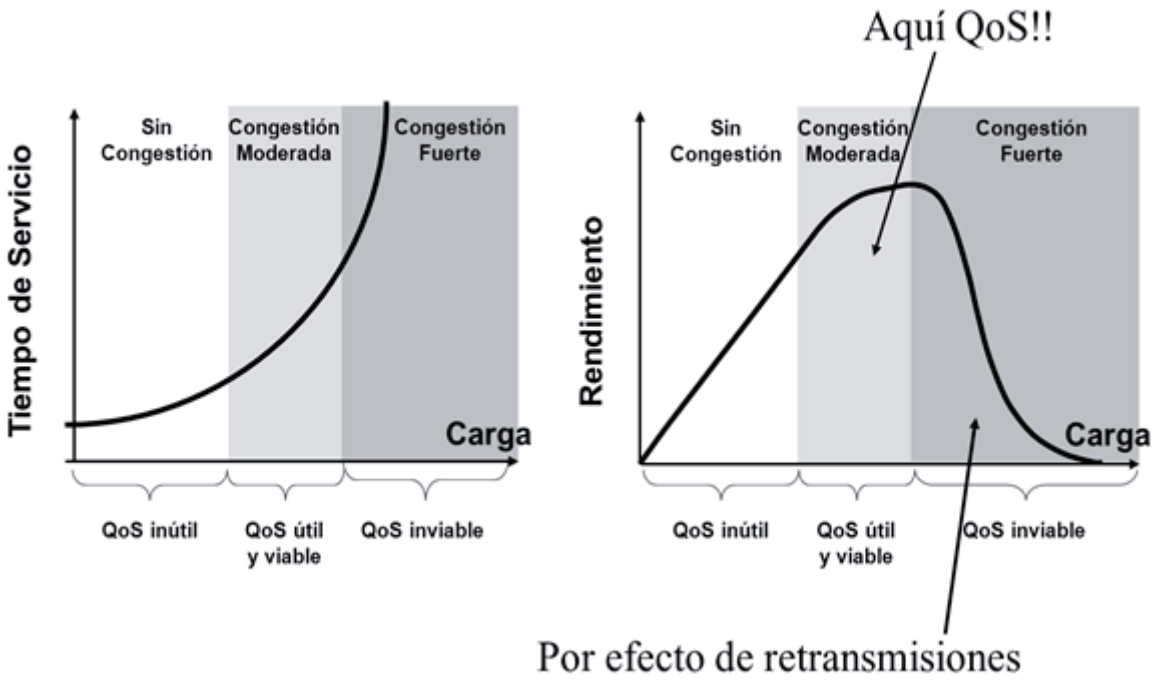


Fig. 3-1 Efectos de la congestión en el tiempo de servicio y el rendimiento

El objetivo de la Calidad de Servicio es asegurar que en casos de carga relativamente elevada (la zona marcada como de 'congestión moderada' en la figura 3-1) las aplicaciones que lo requieran podrán disfrutar de un tiempo de servicio reducido. Si la red tiene siempre niveles de carga inferiores, el funcionamiento se complica y no se obtiene beneficio al aplicar mecanismos de Calidad de Servicio. Si la red tiene normalmente niveles fuertes de congestión, los mecanismos de Calidad de Servicio difícilmente serán capaces de asegurar el nivel de calidad pedido a las aplicaciones que así lo requieran.

3.2 Parámetros de QoS

Son varios los acrónimos terminados en "*oS*" que hacen referencia a la obtención de calidad de servicio en redes, llevando en ocasiones a situaciones equívocas por el mal uso de los mismos. Si bien QoS es el único que se refiere completamente a la Calidad de Servicio, englobando todas las técnicas que se encuentran en torno a ella, CoS (clase de servicio) y ToS (tipo de servicio) son, sencillamente, dos de las técnicas utilizadas para su obtención.

Calidad de Servicio: Definido en el apartado anterior, recoge varios parámetros o atributos que describen un servicio, tales como se muestran en la tabla 3-1:

Tabla 3-1 Parámetros de QoS [5]

Parámetro	Unidades	Significado
Retardo (*delay*) o Latencia (*latency*)	ms	Tiempo medio que tardan en llegar los paquetes
Fluctuación (*Jitter*)	ms	Fluctuación que se puede producir en el retardo
Tasa de pérdida (*Loss Rate*)	%	Proporción de paquetes perdidos respecto de los enviados
Ancho de Banda (*Bandwidth*)	Kb/s	Indica el caudal máximo que se puede transmitir

De la tabla 3-1, los parámetros que definen QoS varían según la aplicación, y la identificación de los mismos es lo que permitirá en primera instancia clasificar o determinar la prioridad de algunas aplicaciones sobre otras.

A continuación, se muestran las fórmulas de cálculo para los parámetros que intervienen en la medición de QoS en una red LAN; estas fórmulas no toman en cuenta modelos de QoS:

- Retardo, parámetro que se determina como el tiempo que usan los paquetes en la transmisión. Su fórmula describe la relación de la longitud del paquete o del flujo de paquetes y la tasa de transmisión que posee para la comunicación.

$$Dprom = \frac{L}{C}$$

donde,

$Dprom$ =Retardo promedio [ms]

L = Longitud del paquete [bits]

C =Tasa de transmisión del paquete [bps]

Esta fórmula se aplica en forma general para el cálculo del retardo por transmisión en una red.

- Variación del retardo (*Jitter*), cuyo valor determina el efecto del retardo en la comunicación ya que produce fluctuación en el canal por la diferencia entre varios retardos de paquetes en un mismo flujo. La fórmula se extrae desde el Manual de la Herramienta de Inyección de Tráfico D-ITG. [6]

$$Jprom = \frac{\sum_i^n |Di|}{n}$$

donde,
$Jprom$ =*Jitter* promedio [ms]
Di =Retardo promedio [ms]
n =Número de retardos

- Pérdida de paquetes (*Packet Loss Rate*), determina una tasa de paquetes que no han sido transmitidos exitosamente, es decir una proporción de los paquetes recibidos sobre los paquetes enviados en la comunicación. [7]

$$Tp = \frac{Ps - Pr}{Ps} * 100\%$$

donde,
Tp =Tasa de pérdida de paquetes [%]
Ps = Paquetes enviados
Pr =Paquetes recibidos

Relacionado a QoS, la tasa de pérdida de paquetes es el caso donde las colas se encuentran llenas en el momento que el paquete llega para ser enviado por el canal, por lo tanto, es eliminado.

- Ancho de banda, parámetro que indica la cantidad reservada del caudal máximo del canal o enlace; es decir la tasa a la cual se transmiten los datos hacia el receptor.

$$AB = \frac{Ptx}{t_{tx}} * Tfr$$

donde,
AB =Ancho de banda [Mbps]
Ptx =Total de Paquetes transmitidos
t_{tx} =Tiempo total de transmisión [s]
Tfr =Tamaño de trama [bits]

Este parámetro se considera un rendimiento (*throughput*) por el hecho que es el ancho de banda que necesita el flujo de paquetes (provenientes de alguna aplicación) para poder transmitir sus paquetes hacia el receptor.

A continuación en la tabla 3-2 se muestra una lista de diferentes aplicaciones junto con sus respectivos parámetros de QoS:

Tabla 3-2 Requerimientos de QoS según tipo de aplicación [5]

Tipo de aplicación	Ancho de Banda	Retardo	Jitter	Tasa de Pérdidas
Interactivo (telnet, www)	Bajo	Bajo	Medio/Alto	Media
Batch (e-mail, ftp)	Alto	Alto	Alto	Alta
Telefonía	Bajo	Bajo	Bajo	Baja
Video Interactivo	Alto	Medio/Alto	Bajo	Baja
Video Unidireccional	Alto	Medio/Alto	Bajo	Baja
Frágil (ej.:emulación de circuitos)	Bajo	Bajo	Medio/Alto	Nula

Según la tabla 3-2, las aplicaciones interactivas requieren de una tasa de pérdida media, retardo y ancho de banda bajos pero de un *jitter* medio puesto que no requiere de variaciones de latencia bajos para transmitir correctamente, en cambio videoconferencia requiere de una tasa de pérdida, retardo y *jitter* bajos pero su demanda de ancho de banda es alto porque son aplicaciones de tiempo real; comparando los parámetros de las dos aplicaciones, la videoconferencia usa más recursos por lo que se la puede categorizar como una aplicación con mayor prioridad.

La arquitectura de QoS está compuesta de tres campos fundamentales que se muestran en la figura 3-2.

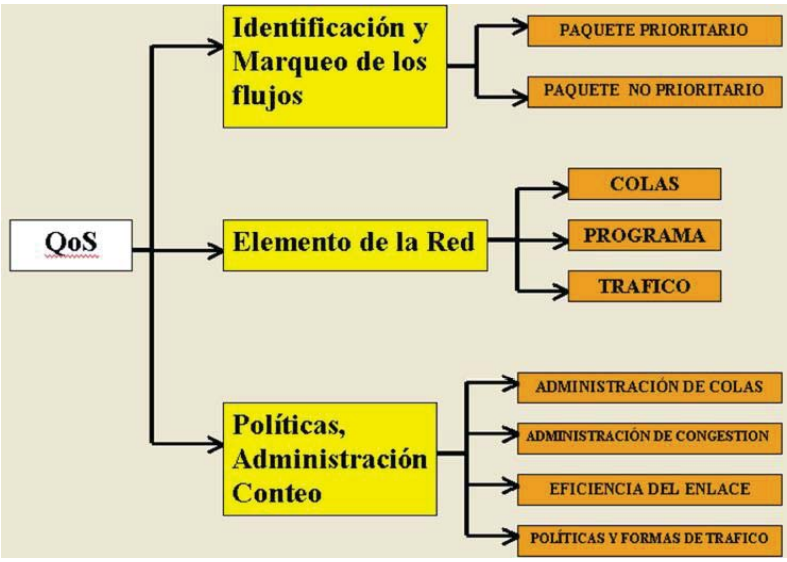


Fig. 3-2 Arquitectura QoS

3.2.1 CoS (*Class of Service*)

Clase de Servicio: Este término implica, a su vez, dos procedimientos: en primer lugar la priorización de los distintos tipos de tráfico claramente definidos a través de la

red y, en segundo lugar, la definición de un pequeño número de clases de servicio a las que aplicarla.

Priorizar es importante en los puntos de congestión de la red, donde las decisiones de priorización pueden ser realizadas por *switches* y *routers*.

Las aplicaciones que requieren distinguir clases de servicio incluyen procesos transaccionales, el vídeo y cualquier otro tráfico sensible al tiempo.

No se debe confundir CoS con QoS, pues, a diferencia de QoS, CoS no garantiza ancho de banda o latencia, en cambio permite a los administradores de red solicitar prioridad para el tráfico basándose en la importancia de éste.

Independientemente de la diferenciación, tanto CoS como QoS categorizan el tráfico para asegurar que el tráfico considerado crítico siempre fluya por la red, a pesar del ancho de banda demandado o de las aplicaciones de menor importancia.

Existen muchas posibles definiciones de tipos de *Calidad de Servicio*, pero la mayoría de las empresas definen las clases de tráfico por tipo de aplicación, tipo de dispositivo o por tipo de usuario. Hoy es además posible definir clases separadamente en *routers* o *switches* individuales, pero suele ser poco práctico.

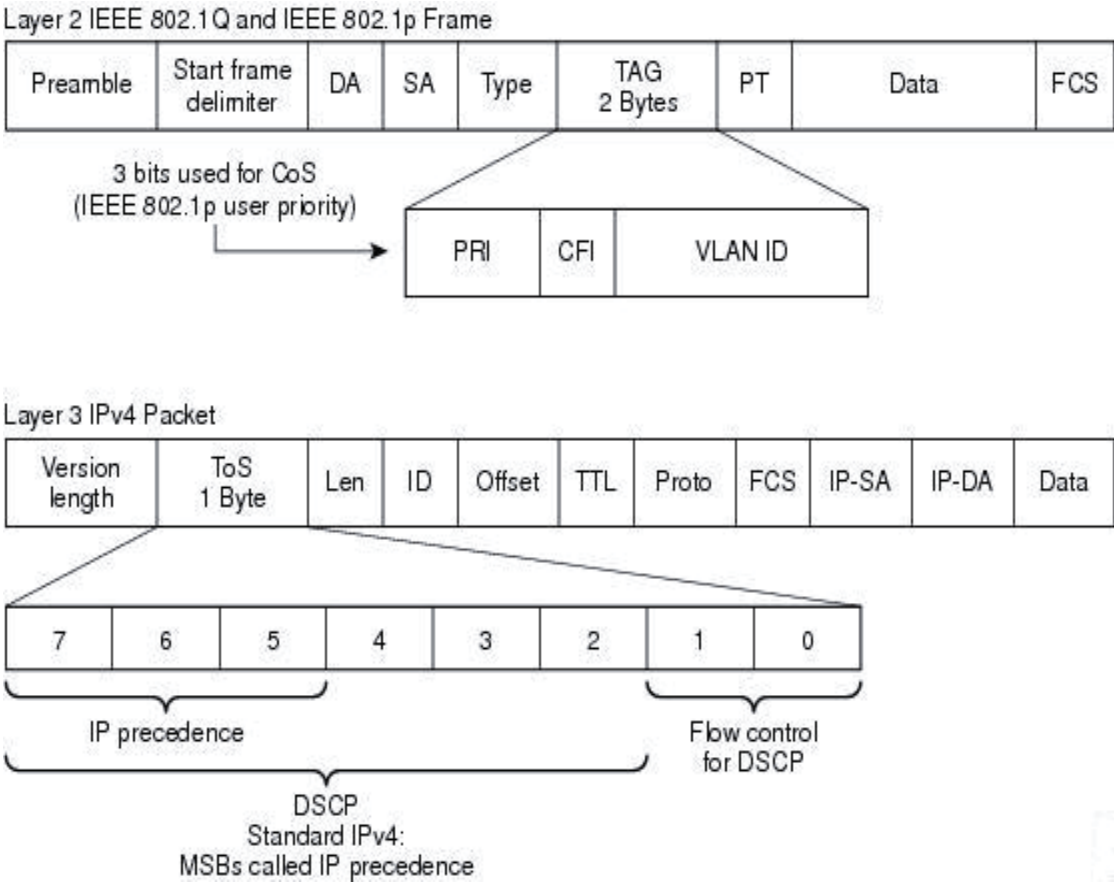


Fig. 3-3 Clasificación por capas de tramas y paquetes

Como se apreció en la figura 3-3, la norma IEEE 802.1p incluye un campo de 3 bits donde especificar la clase de servicio, definiendo de esta manera las siguientes combinaciones posibles para asignar las prioridades (ver figura 3-4).

Combinación	CoS	Prioridad
111	Network Critical	7
110	Interactive Invoice	6
101	Interactive Multimedia	5
100	Streaming Multimedia	4
011	Bussiness Critical	3
010	Standard	2
001	Background	1
000	Best Effort	0

Fig. 3-4 Modelo de 3 bits de Clases de Servicio

Un ejemplo de tecnología que usa CoS es el estándar IEEE 802.1p/Q, representado en la figura 3-5.

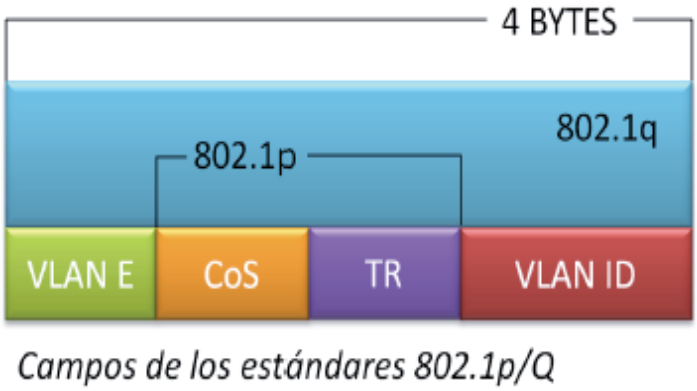


Fig. 3-5 Estándar IEEE 802.1p/Q

Para conocer dónde aplicar CoS, se presenta el esquema de la figura 3-6.

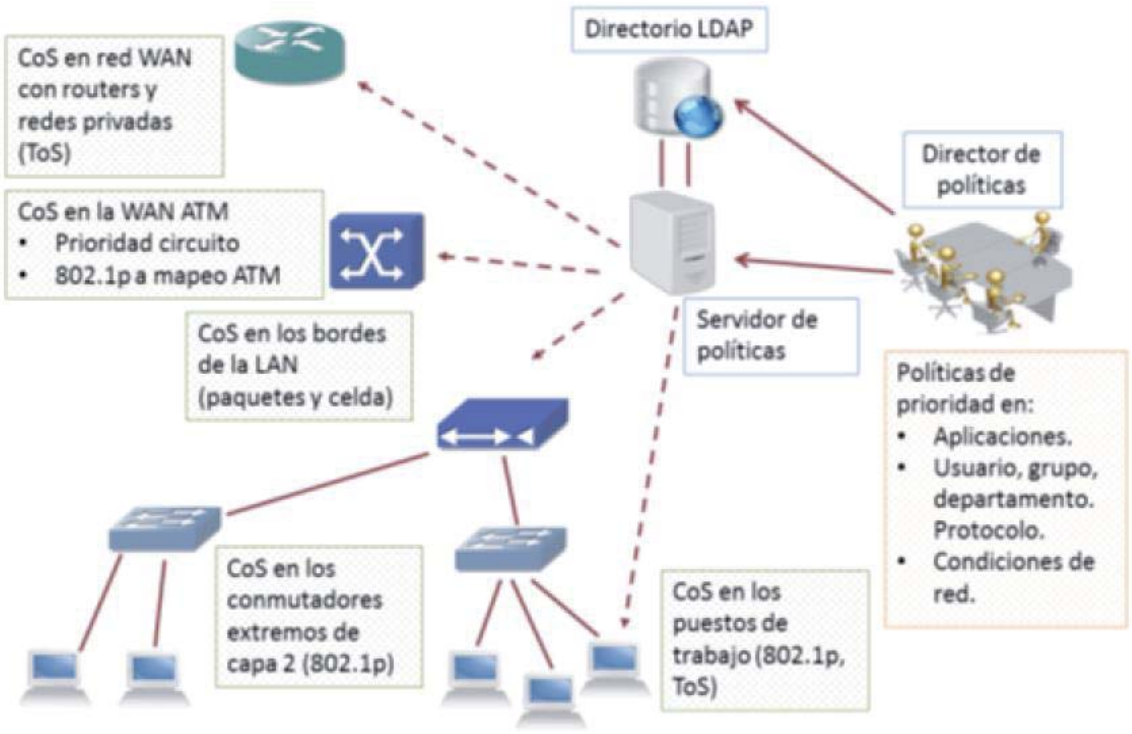


Fig. 3-6 Dónde aplicar CoS

3.2.2 ToS (*Type of Service*)

Tipo de Servicio es equivalente a un carril destinado a los autos de uso compartido. Se reserva ancho de banda con antelación y después se asigna el tráfico que necesite preferencia, como el de voz o un CoS con prioridad, de modo que este tráfico pueda utilizar el ancho de banda reservado. ToS no implica, por lo tanto, ningún tipo de garantías.

ToS está incluido como uno de los campos en la tecnología de QoS denominada *DiffServ* (servicios diferenciados), donde también es conocido como *DiffServ codepoint* (DSCP o punto de código *Diffserv*). Es un campo de 8 bits, estando los dos últimos reservados. Con los otros 6 bits restantes es posible obtener 64 combinaciones o '*codepoint'*, de ellas, 48 son utilizadas para direccionar el espacio global y 16 son para uso local.

Parte del protocolo IP Versión 4 (figura 3-7) reserva un campo en el paquete IP para el tipo de servicio (IP TOS).

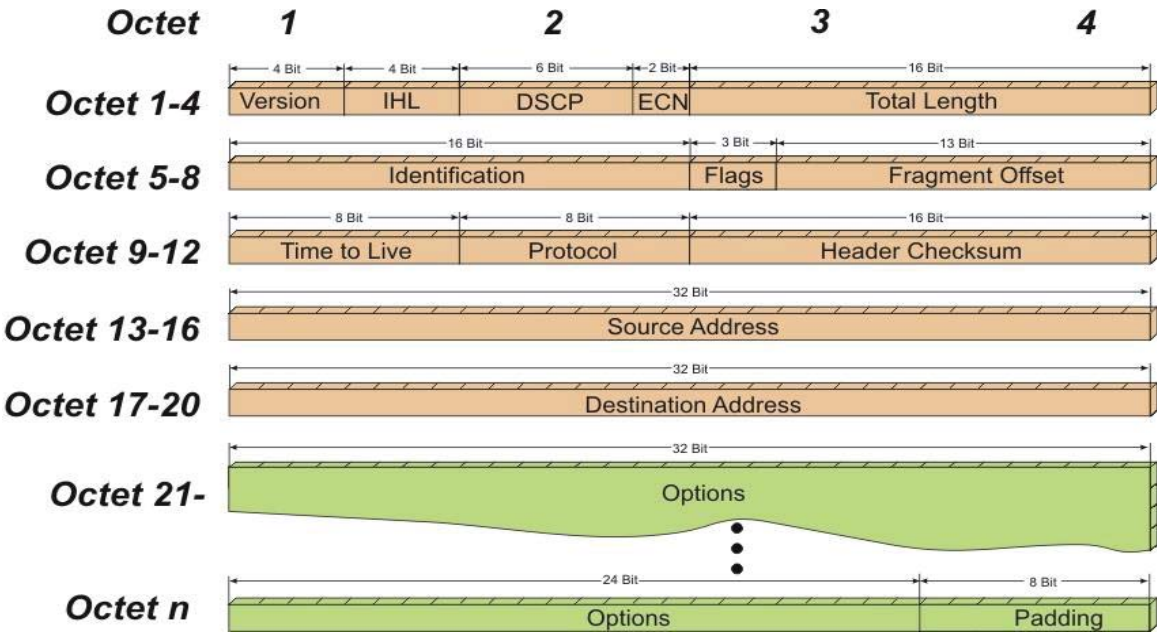


Fig. 3-7 Formato de encabezado IPV4

En el campo TOS se pueden especificar los atributos de fiabilidad, capacidad de procesamiento y retardos del servicio, tal y como se ve en la figura 3-8.

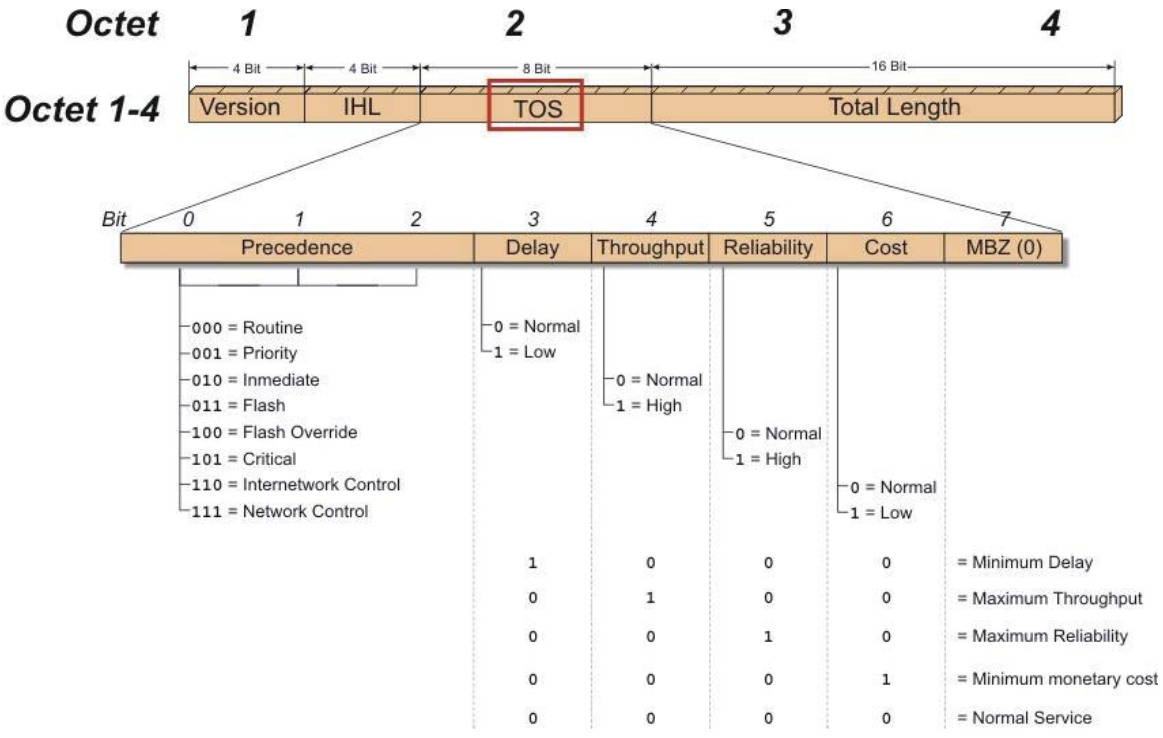
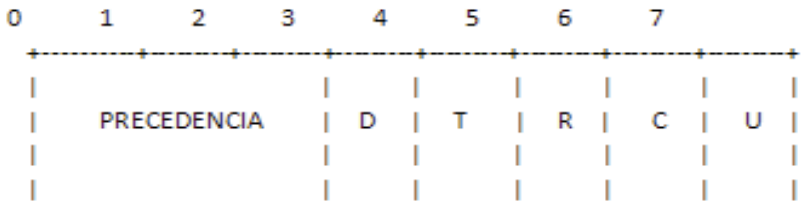


Fig. 3-8 Campo ToS en IPv4

El tipo de servicio proporciona una calidad de servicio deseada, a través de parámetros que determinan la prioridad de un servicio como la precedencia, máximo *throughput*, mínimo retardo, máxima fiabilidad, y mínimo costo. La precedencia tiene 8 niveles ordenados como lo muestra la figura 3-8.



Precedencia (ocho niveles)

111 - Control de Red
110 - Control Entre Redes
101 - CRITICO/ECP
100 - Muy Urgente (Flash Override)
011 - Urgente (Flash)
010 - Inmediato
001 - Prioridad
000 - Rutina

D, T, R, C: flags para indicar la ruta que se quiere utilizar

D: Delay (mínimo retardo)
T: Throughput (máximo rendimiento)
R: Reliability (máxima fiabilidad)
C: Cost (mínimo costo)
U: bit reservado

Fig. 3-9 Niveles de ToS [8]

3.3 Clasificaciones de QoS

Es posible realizar una clasificación de QoS bajo distintas especificaciones, así se podría diferenciar según el tipo de tráfico, dónde aplicarla, la reserva de recursos de la red y otros parámetros, tal y como se indica a continuación:

3.3.1 Según la sensibilidad del tráfico

Teniendo en cuenta la variedad de tráfico existente y los requerimientos de retardo, latencia y ancho de banda para cada tipo, se encuentra que con:

- QoS muy sensible al retardo: Un ejemplo de este tipo es para el tráfico de vídeo comprimido. Para este caso es necesario garantizar la disponibilidad de una

determinada y gran cantidad de ancho de banda reservado para este tráfico y un valor de retardo mínimo que asegure la correcta transmisión del mismo. Para conseguirlo, será necesario utilizar mecanismos de prioridad, definidos posteriormente en el capítulo de protocolos y arquitecturas, así como encolar adecuadamente los flujos de datos.

- QoS algo sensible al retardo: Como la resultante de la aplicación de la emulación de circuito. Al igual que en el caso anterior se garantiza hasta un cierto nivel de ancho de banda, aunque en menor valor. De la misma manera, será necesario asignar prioridades para la transmisión de los datos.

- QoS muy sensible a pérdidas: Como sucede con el tráfico tradicional. Si se garantiza un nivel de pérdidas de valor cero entonces nunca se descartarán paquetes ni se desbordarán los *buffers* de almacenamiento del flujo, lo que facilitará el control de transmisión, por otra parte, esta garantía se hace a nivel de acceso al medio (MAC) o en capas superiores, pero nunca a nivel físico.

- QoS nada sensible: Por ejemplo el tráfico de servicios de noticias. La filosofía de este tipo de QoS es usar cualquier oportunidad de transmisión restante y asumir que la capacidad de los *buffers* posteriores es suficiente para llevarla a cabo, asignándole a este tipo de tráfico la prioridad más baja. A este tipo responden los algoritmos *Best Effort* o al mejor esfuerzo, utilizado en Internet.

3.3.2 Según quién solicite el nivel de QoS

Teniendo en cuenta que la petición de QoS puede ser realizada por el usuario final o por los conmutadores de la red, se tienen los siguientes dos casos:

- QoS Implícita: En este tipo el *router* o conmutador asigna automáticamente los niveles de calidad servicio en función del criterio especificado por el administrador, como el tipo de aplicación, protocolo o dirección de origen. Hoy en día todos los *routers*, y algunos conmutadores, ofrecen este tipo de QoS. El proceso es el siguiente:

i) Estaciones finales: Las estaciones finales transmiten los paquetes.

ii) Conmutador o *router*: Le llegan los paquetes, realiza un estudio de los datos entrantes y los prioriza, repartiéndolos en diferentes colas según la prioridad asignada. Estos datos vuelven a ser transmitidos hacia el siguiente conmutador o *router*, donde se repite el proceso.

- QoS explícita: Permite al usuario o aplicación solicitar directamente un determinado nivel de servicio que han de respetar los conmutadores y *routers*. El proceso sería:

iii) Estaciones finales: En este caso las estaciones finales transmiten una petición RSVP, si ésta es aceptada, los paquetes son transmitidos.

iv) Conmutador o *router*: Los datos entrantes son priorizados de acuerdo a instrucciones del nodo de destino, pasando al próximo conmutador o *router*, donde se repetirá el proceso.

3.3.3 Según las garantías

En esta clasificación se va a tener en cuenta la reserva de recursos del sistema para proporcionar los servicios.

- QoS garantizada (*Hard* QoS): Es aquella en la que se produce una reserva absoluta de los recursos de la red para un tráfico determinado, asegurándose así unos niveles máximos de garantías para este tráfico.
- QoS no garantizadas (*Lack* of QoS): Es una calidad de servicio sin garantías. El tráfico es transmitido por la red a expensas de lo que en ella pueda sucederle. Es el tipo de QoS correspondiente a los servicios *Best Effort*.
- QoS Servicios diferenciados (*Soft* QoS): Es el punto medio entre los dos tipos anteriores. Para este tipo se realiza una diferenciación de tráfico, siendo tratados algunos mejor que el resto (expedición más rápida, más ancho de banda promedio, menos tasa de error promedio).

3.3.4 Según el lugar de aplicación

Es posible aplicar calidad de servicio en los extremos y en los bordes de la red, por lo tanto, se tiene:

- QoS extremo a extremo (*end-to-end*): Es la aplicación de las políticas entre los extremos de la red. Con este tipo de QoS, la función de los conmutadores se reduce a observar la marca de los paquetes (802.1p), sin tener que calcular la clase de servicio de cada paquete.
- QoS borde a borde (*edge-to-edge*): Es la aplicación de las políticas entre dos puntos cualesquiera de la red.

CAPÍTULO 4

MÉTODOS DE PRIORIZACIÓN DE TRÁFICO

4.1 Modelos y mecanismos de priorización de tráfico

Este capítulo está dedicado al estudio e investigación de los diversos métodos y mecanismos de priorización de tráfico empleados por las tecnologías de equipos de red para la implementación de calidad de servicio en una red LAN. La finalidad de este capítulo es profundizar en las técnicas para manejar diferentes tráficos, considerando los cuatro parámetros que deben permitir una calidad de servicio en la red, y se establecen algunos modelos que permiten priorizar un tráfico.

4.1.1 Modelo de servicio “Mejor Esfuerzo”

Se llama modelo de mejor esfuerzo al servicio que provee la red cuando hace todo lo posible para intentar entregar el paquete a su destino, a pesar que no hay garantía de su recepción. Una aplicación enviará datos en cualquier cantidad, cuando lo necesite, sin pedir permiso o notificar a la red. Éste es el modelo utilizado por las aplicaciones de FTP y HTTP [9].

Pero el servicio de mejor esfuerzo no es un modelo apropiado para aplicaciones sensibles al retardo o variaciones de ancho de banda, las cuales necesitan de un tratamiento especial; tal es el caso de aplicaciones de VoIP y videoconferencias.

4.1.2 Modelo de Servicios Integrados (*IntServ*)

El modelo de servicios integrados (*IntServ*) provee a las aplicaciones un nivel garantizado de recursos negociando parámetros de red, de extremo a extremo [9]. La aplicación solicita el nivel de servicio necesario con el fin de operar apropiadamente, y se basa en QoS mediante señalización explícita para reserva de los recursos de red necesarios antes de que la aplicación comience a transmitir. Estas reservaciones se mantienen en pie hasta que la aplicación termina o hasta que el ancho de banda requerido por ésta sobrepase el límite asignado para dicha aplicación.

La idea básica del modelo es hacer reservas de recursos por flujos y su objetivo es preservar el modelo de datagramas de IP, pero al mismo tiempo soportar aplicaciones en tiempos reales. El modelo *IntServ* exige la clasificación de las aplicaciones de acuerdo a su flexibilidad ante pérdidas, cuya información se ve reflejada en la tabla 4-1:

Tabla 4-1 Aplicaciones vs. Flexibilidad de pérdidas

Aplicaciones	Flexible a pérdidas	No flexible a pérdidas
Elásticas	Datos UDP: DNS, SNMP, NTP, entre otros	Datos sobre TCP: FTP, Web, *e-mail*, entre otros
Tiempo real	Flujos multimedia, *streaming*, videoconferencia, VoIP	Emulación de circuitos (simulación de líneas dedicadas)

Para la implementación del modelo se deben habilitar otros servicios puesto que requiere de altos recursos para que el flujo esté siempre activo. La arquitectura del modelo define 3 tipos de servicios, tal como se muestra en la figura 4-1:

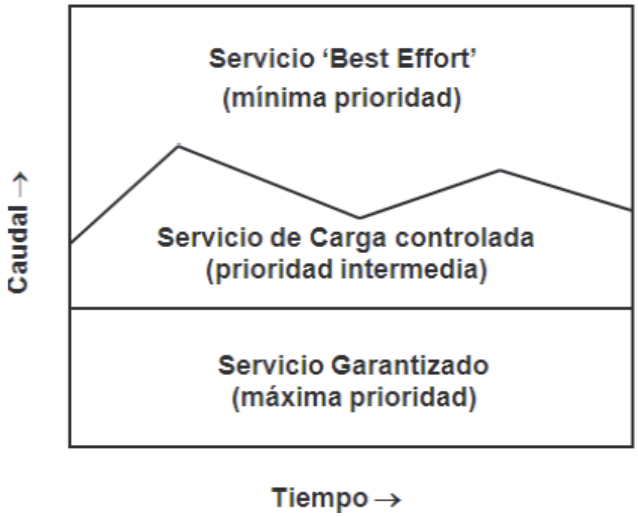


Fig. 4-1 Reparto de recursos en *IntServ*

- Servicio Garantizado: Mantiene un *throughput* mínimo y retardo máximo pero supone que cada *router* del trayecto debe dar garantías al tráfico; limitada por el medio físico.
- Servicio Carga Controlada: Calidad a una red con poca carga, mantiene un retardo bajo pero sin ofrecer garantías en la red.
- Servicio Mejor Esfuerzo: No presta garantías al flujo que circula por la red (sin QoS).

La metodología que usa la aplicación para reservar los recursos sucede antes de iniciar el flujo de paquetes y el proceso se resume en los siguientes pasos:

a) La fuente inicia el establecimiento de una reserva describiendo primero a la red las características del flujo y los requerimientos de los recursos.

b) La red puede aceptar este nuevo flujo de la aplicación sólo si hay suficientes recursos para comprometerse con los recursos solicitados.

c) Ya establecida la reserva, la aplicación envía sus paquetes a lo largo del canal reservado y la red obtendrá un flujo continuo de tráfico.

El modelo de servicio integrado, comúnmente conocido como servicio "garantizado", se fundamenta en dos componentes claves: plano de control y plano de datos.

- El plano de control establece la reserva de recursos,
- El plano de datos envía los paquetes de datos basado en el estado de la reserva.

La figura 4-2 muestra el modelo de referencia de servicios integrados junto a sus componentes:

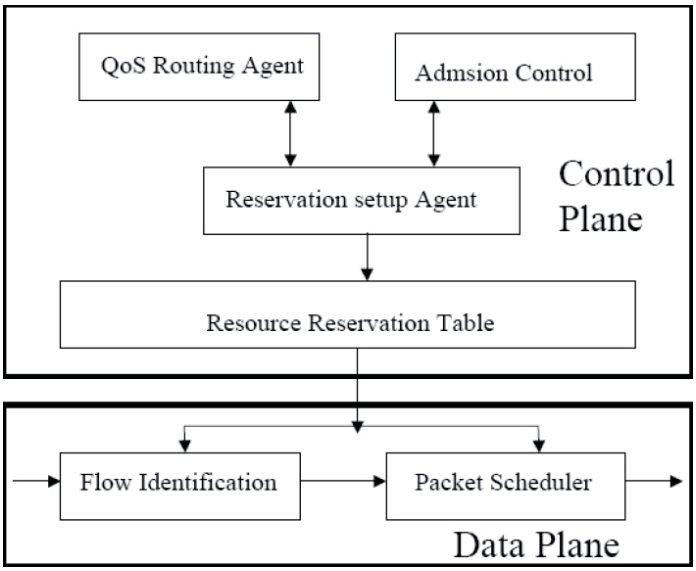


Fig. 4-2 Modelo de referencia de servicios integrados.

Como se indicó anteriormente, el plano de control debe establecer la reserva de recursos y este proceso empieza con la especificación del flujo que debe realizar una aplicación para dar a conocer a la red su flujo de tráfico y los requerimientos de QoS; así, la solicitud de establecimiento de recursos es enviada a la red. Cuando el *router* de la red recibe la solicitud, realiza dos tareas: utiliza un protocolo para enrutar la solicitud de reserva para el siguiente salto, y posteriormente coordinar con el control de admisión para decidir si hay suficientes recursos en la red para comprometerse con los recursos solicitados.

El modelo *IntServ* se basa en el Protocolo de Reservación de Recursos (RSVP) para señalizar y reservar la QoS deseada para cada flujo en la red. El protocolo transporta

información sobre las características del tráfico y requerimientos de los recursos; RSVP debe salto por salto intentar hacer la reserva solicitada debido a que la información de estados para cada reserva necesita ser mantenida por cada enrutador a lo largo de la ruta; se convierte en una desventaja de escalabilidad para miles de flujos a través de una red central.

Ahora, el plano de datos encargado de enviar los paquetes de acuerdo a la información de reserva de recursos a la red, también pone en funcionamiento la identificación de flujos que selecciona los paquetes de los flujos reservados y son transmitidos en las colas apropiadas; el planificador de paquetes, por su parte asigna los recursos a los flujos basándose en la información de las reservas. El modelo *IntServ* hace uso del algoritmo *Weighed Fair Queuing* WFQ (RFC 2212) para gestionar el flujo de paquetes.

4.1.3 Modelo de Servicios Diferenciados (*DiffServ*)

Este modelo incluye un conjunto de herramientas de clasificación y mecanismos de encolamiento de paquetes, que proveen a ciertas aplicaciones o protocolos con determinadas prioridades sobre el resto del tráfico en la red. El tráfico de red puede ser clasificado por dirección de red, protocolo, puertos, interfaz de ingreso o cualquier tipo de clasificación que pueda ser alcanzada mediante el uso de listas de acceso, en su variante para la implementación de QoS.

El protocolo IPv4 posee el campo ToS para gestionar el marcado de paquetes con el nivel de servicio requerido. Esta definición no se utilizó mayormente debido a la ambigüedad de su significado, por lo que más tarde se convirtió en el denominado campo DSCP (Servicios Diferenciados por Código de Punto). Este campo sí tuvo una aceptación global y se asumió una interpretación estándar que permite a las redes planificar metodologías basándose en ésta. Antes de que apareciera el modelo *DiffServ* hubo una compatibilidad entre DSCP y el valor de ToS para el proceso de marcado del tráfico por lo que este código se denominó Selector de Clase (CS); conformándose posteriormente un servicio más de *DiffServ*.

Una vez que existe la capacidad de marcar los paquetes utilizando DSCP, es necesario proveer del tratamiento apropiado para cada una de estas clases. La colección de paquetes con el mismo valor DSCP circulando hacia una dirección determinada, es llamada Comportamiento Agregado (BA). Es así como múltiples aplicaciones/fuentes pueden pertenecer al mismo BA.

Para permitir el seguimiento del comportamiento de cada tráfico que ingresa a los nodos se utiliza el protocolo de comportamiento por salto (PHB). El PHB se refiere a la programación, encolamiento, limitación y modelamiento del comportamiento de un nodo, basado en el BA perteneciente del paquete. Los PHB’s son implementados en nodos por medio de algunos mecanismos de manejo de *buffer* y despachado de paquetes.

Un PHB es seleccionado en un nodo por medio del mapeado del código *DiffServ* (DS) en un paquete recibido. Los PHB’s estándares tienen un código recomendado. Los

PHB's estándares se especifican por las características de QoS que brinda el modelo *DiffServ*, donde específicamente son cuatro PHBs [10]:

- PHB Default: El valor DSCP es cero (000000) y el servicio esperado es exactamente el servicio por defecto de la Internet de hoy.
- PHB CS: Son siete valores DSCP que funcionan desde el 001000 al 111000 y son específicos para seleccionar hasta siete comportamientos, cada uno de los cuales tiene una mayor probabilidad de un envío de tiempo que su predecesor (CS7-CS1).
- PHB EF: El reenvío prioritario (EF) PHB tiene un valor recomendado DSCP de 101110 porque la tasa de inicio del tráfico EF debe igualar o superar una tasa configurable; éste PHB permite la creación de servicios en tiempo real con una tasa de *throughput* configurable. El objetivo de este PHB es que el flujo agregado vea siempre o casi siempre, la cola vacía. El EF PHB puede ser usado para construir un servicio punta a punta de bajas pérdidas, baja latencia, bajo *jitter* (colas muy pequeñas) y/o ancho de banda asegurado.
- PHB AF: El reenvío seguro (AF) PHB es el más utilizado en la arquitectura *DiffServ*. Dentro de esta PHB los 4 grupos AF (llamados clase AF1, AF2, AF3 y AF4 o clases Cisco) son divididos en 3 grupos "olímpicos": oro, plata y bronce, representando la tendencia a descartar paquetes. Cada paquete será entregado a una clase de servicio mientras se apegue a un perfil de tráfico. Cualquier exceso de tráfico será aceptado por la red, pero tendrá mayor probabilidad de ser descartado según la clase de servicio y grupo.

Se definen N clases AF tal que a cada clase AF se le reservan ciertos recursos (*buffer* y ancho de banda) en cada uno de los nodos *DiffServ* (DS), de forma que los retardos y/o pérdidas de una clase sean siempre inferiores a los de una clase de menor prioridad.

Dentro de cada clase, los paquetes se pueden clasificar a su vez en M categorías de preferencia de descarte (dependiendo del marcado en la frontera). En caso de congestión la preferencia de descarte determina la importancia relativa del paquete dentro de la clase. Actualmente N=4 y M=3 son definidos para uso general. Un paquete que pertenezca a una clase AFi y tenga una preferencia de descarte j es marcado con el código AFij. Los códigos recomendados por la IETF en su recomendación RFC 2597 se indican en la figura 4-3.

The RECOMMENDED values of the AF codepoints are as follows: AF11 = '001010', AF12 = '001100', AF13 = '001110', AF21 = '010010', AF22 = '010100', AF23 = '010110', AF31 = '011010', AF32 = '011100', AF33 = '011110', AF41 = '100010', AF42 = '100100', and AF43 = '100110'. The table below summarizes the recommended AF codepoint values.

	Class 1	Class 2	Class 3	Class 4
Low Drop Prec	001010	010010	011010	100010
Medium Drop Prec	001100	010100	011100	100100
High Drop Prec	001110	010110	011110	100110

Fig. 4-3 Códigos del PHB AF – RFC 2597

El modelo arquitectónico de *DiffServ* se constituye por lo general en el funcionamiento de una nube compuesta de muchas redes IP, donde se considera la nube como una región uniforme donde es posible agregar diferentes tipos de tráfico con un distinto trato de envío; por tal razón el modelo permite que el tráfico entrante sea clasificado y condicionado en los bordes de la red para ser asignado a diferentes BA's. La figura 4-4 ejemplifica la arquitectura para la que está diseñado el modelo *DiffServ*.

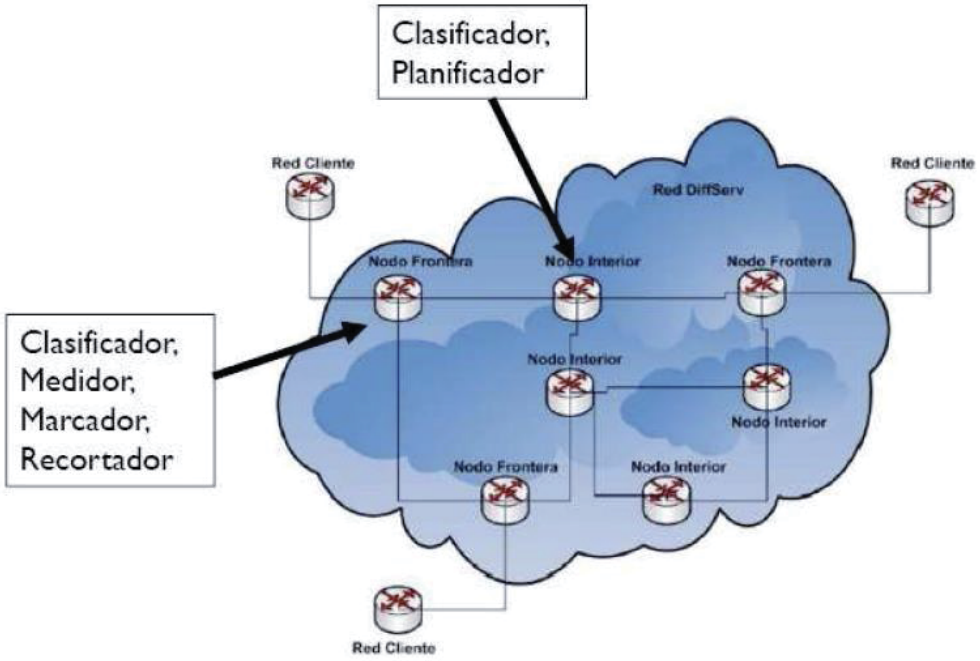


Fig. 4-4 Arquitectura modelo DiffServ

De acuerdo a la figura 4-4, se encuentra un concepto denominado dominio *DiffServ* (DS) que es un conjunto de nodos que operan con una política de provisionamiento de servicios común y con un grupo de PHB's implementados en cada nodo. Está formado por nodos *DiffServ* de frontera que clasifican y condicionan el tráfico entrante para asegurarse que los paquetes que transitan el dominio estén apropiadamente marcados para seleccionar un PHB de los grupos PHB que son soportados dentro del dominio [12]. Los nodos seleccionan el comportamiento de envío basándose en su código DS, y lo hacen asociando este valor a unos de los PHB soportados.

A continuación, se citan algunas características del modelo que hace referencia a su arquitectura de priorización y suministro de tráfico.

- Sólo los nodos frontera clasifican el tráfico y marcan paquetes,
- Los nodos interiores usan las clases codificadas en la cabecera del paquete para determinar el tratamiento de los paquetes.

Las funciones de un nodo frontera que se consideran las más importantes dentro del modelo se encargan de la clasificación de paquetes y acondicionamiento del tráfico (marcador, medidor, recortador, desechador); cada parámetro se resume en la tabla 4-2.

Tabla 4-2 Componentes de un nodo frontera.

Elemento	Función
Clasificador	Divide el flujo de paquetes entrantes en múltiples grupos basándose en reglas predefinidas.
Medidor (*Meter*)	Compara el flujo de tráfico de un cliente con su perfil de tráfico. Los paquetes que cumplen el perfil se dejan ingresar directo a la red. Los paquetes que no cumplen deben pasar por el acondicionamiento (*marking, shaping, dropping*).
Marcador (*Marker*)	Fija el campo DSCP (*codepoint*) a un valor particular. Así se incluye el paquete de retransmisión. Los paquetes marcados como no conformes podrían ser desechados por la red ante congestion.
Recortador (*Shaper*)	Un recortador no permite que el paquete pase hacia la red hasta que cumpla con el perfil de tráfico (retarda los paquetes).
Desechador (*Dropper*)	Desecha los paquetes no cumplientes con el perfil de tráfico.

Están definidos dos tipos de clasificadores: el clasificador BA clasifica paquetes basado en el código DS solamente; para que esto funcione se requiere que los paquetes sean marcados antes de ingresar al clasificador. El clasificador MF (Multi-Campo)

selecciona paquetes basado en el valor de una combinación de uno o más campos de cabecera, como IP origen, IP destino, campo DS, protocolo ID, puertos origen y destino, entre otra información. El clasificador debe autentificar la información que usa para clasificar el paquete y el marcado de paquetes con base en los tipos de aplicaciones, con base en direcciones particulares de origen, destino o prefijos de red.

Se definen dos modelos *DiffServ* con garantía de QoS y un servicio sin QoS, que fundamentan la repartición de recursos del modelo, que se muestra en la figura 4-5.

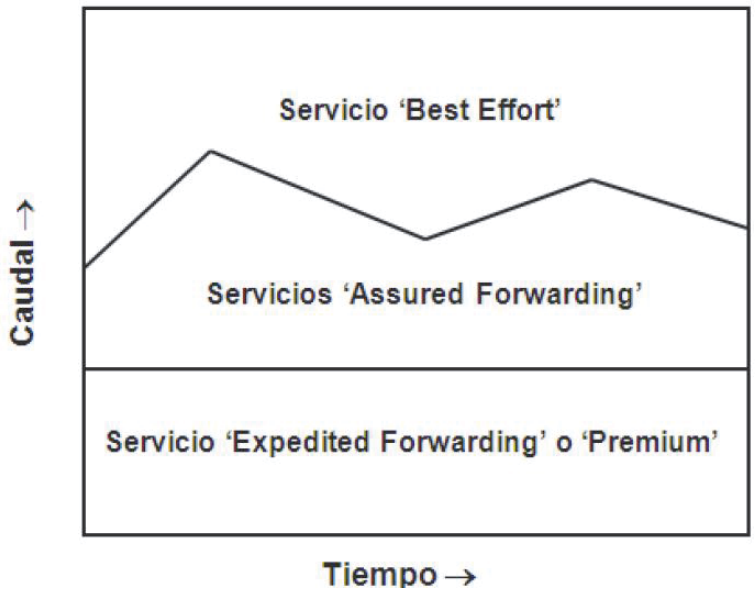


Fig. 4-5 Reparto de recursos en el modelo DiffServ

a) Servicio Reenvío Prioritario (EF): Este tipo de *DiffServ* minimiza el retardo y la variación del retardo y provee el más alto nivel de QoS posible. Especifica el tráfico de mayor prioridad, equivale a una línea dedicada.

b) Servicio Reenvío Seguro (AF): Los paquetes se marcan con alta prioridad aunque en este caso no se garantiza un ancho de banda. Se definen cuatro clases posibles pudiéndose asignar a cada clase una cantidad de recursos en los *routers* como ancho de banda, espacio en *buffers*, entre otros.

c) Servicio Mejor Esfuerzo: Este servicio se caracteriza por tener a cero los tres primeros bits del DSCP. En este caso los dos bits restantes pueden utilizarse para marcar una prioridad dentro del grupo *best effort*. Este servicio no ofrece ningún tipo de garantía.

En resumen, la tabla 4-3 especifica las técnicas de QoS para cada clase de servicio o aplicación que gestiona el *DiffServ*.

Tabla 4-3 Resumen de técnicas de QoS para cada Clase de Servicio

Clase de servicio	DSCP	Valor DSCP	Acondicionamiento en nodo frontera	PHB Utilizado	Tipo de Cola
Telefonia	EF	101110	Usando políticas (única tasa y tamaño de ráfaga)	RFC3246	SP
Señalización	CS5	101000	Usando políticas (única tasa y tamaño de ráfaga)	RFC2474	WRR/WFQ
Tiempo Real Interactivo	CS4	100000	Usando políticas (única tasa y tamaño de ráfaga)	RFC2474	WRR/WFQ
Streaming Multimedia	AF31 AF32 AF33	011010 011100 011110	Usando baja tasa, con marcador de 3-colores (como en el RFC2698)	RFC2597	WRR/WFQ
Broadcast de video	CS3	011000	Usando políticas (única tasa y tamaño de ráfaga)	RFC2474	WRR/WFQ
Estándar	DF	000000	No aplica	RFC2474	WRR/WFQ

A continuación, la figura 4-6 es un ejemplo claro de cómo actuaría el modelo *DiffServ* en un entorno real de red, ya que muestra el tipo de clasificador que usa para una aplicación que requiere establecer entre una empresa X y una empresa Y. Lo importante es analizar los mecanismos de acondicionamiento que usa cada clasificador

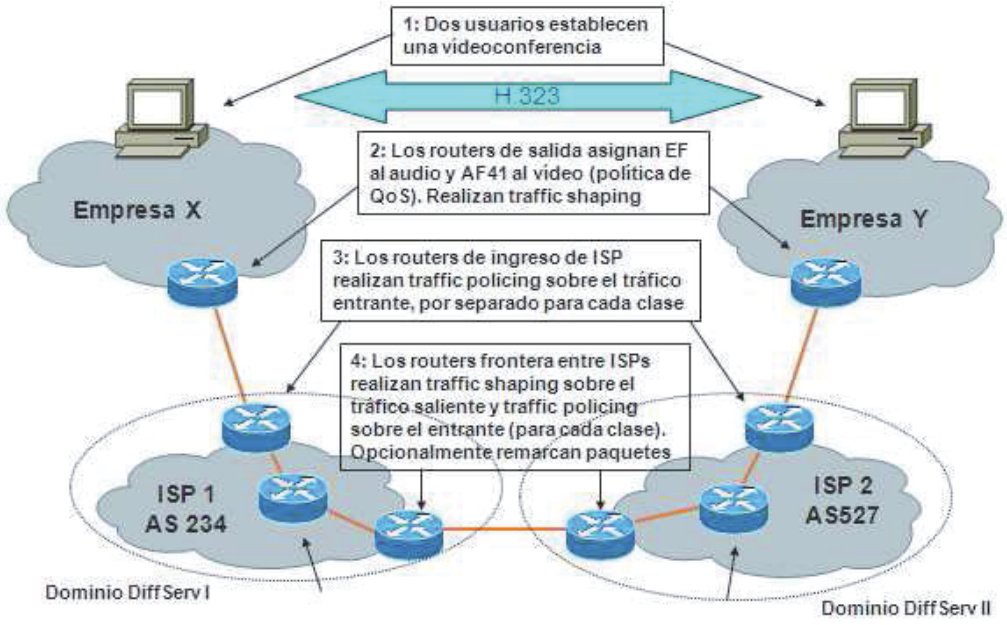


Fig. 4-6 Funcionamiento de *DiffServ* en Internet

CAPÍTULO 5

PRUEBAS DE EMULACIÓN DE UNA RED MPLS

5.1 Revisión del Emulador GNS3

5.1.1 Arquitectura del emulador

Dynamips es el motor de emulación que permite emular diferentes plataformas *hardware* usando imágenes de sistemas operativos de CISCO en un mismo *host*. Entre dichas plataformas se encuentran los *Routers* 1700, 2600, 3600, 3700 y 7200. Por otro lado, puede emular *switches Ethernet, Frame-Relay* y ATM con funcionalidades básicas.

En cuanto a la emulación de *switches*, Dynamips no es capaz de emular *switches* Catalyst sino que provee una versión limitada de un *switch* virtual, cuyas limitaciones pueden ser resueltas usando métodos alternativos como la emulación de NM-16ESW que el emulador sí soporta. Por otro lado, Dynamips tampoco es capaz de emular *Firewalls* PIX, para ello se usa el emulador PEMU a través de Dynagen [11].

Inicialmente Dynamips consume grandes cantidades de CPU del PC emulador, esto se debe principalmente a que realiza la emulación de los *routers* instrucción por instrucción y a que no puede saber cuándo un *router* virtual está inactivo, de modo que ejecuta instrucciones como si la imagen del IOS estuviera realizando algún trabajo útil.

Para resolver el problema del excesivo uso de CPU, se crea un proceso llamado IDLEPC. Por otro lado, Dynamips también consume memoria RAM del PC emulador, ya que, en teoría cada *router* virtual debe tener a su disposición, como mínimo, toda la cantidad de memoria RAM que necesita para poder trabajar, por lo tanto, esta cantidad se hace impráctica si se requieren emular redes con varios *routers*. Para resolver el problema del excesivo uso de memoria del PC emulador se usan herramientas que permiten compartir la memoria del mismo entre varios *routers* emulados con la misma IOS, y herramientas que usan el disco en vez de la memoria del emulador [11].

5.1.2 IDLE-PC

Se trata de una herramienta que realiza un análisis en el código de una imagen IOS para determinar los puntos más probables que representen un bucle de inactividad, de modo que, cuando se detecten, haga que los *routers* virtuales “duerman” durante ese instante. Es decir, IDLE-PC ayuda a Dynamips a emular el estado inactivo de la CPU virtual de un *router*.

Algunas características adicionales de este proceso son las siguientes:

- La aplicación de un mal valor de IDLE-PC hace que la PC del emulador trabaje entre 60% - 100% cuando emula un solo un *router*, mientras que un buen valor hace

que sólo trabaje entre 1% -10% de la capacidad. Estos valores dependen de lo potente que sea el emulador usado.

- Está ligado a la versión de Dynamips que se usa, si se cambia de versión, es muy probable que se necesita cambiar de valor de IDLE-PC.

- Será diferente para IOS de diferentes versiones y por supuesto para diferentes plataformas. Se aplicará a un *router* virtual cada vez que use esa IOS.

- No son exclusivos de un PC o sistema operativo, por lo tanto, los archivos "dynagenidledb.ini" pueden ser copiados y compartidos y el valor de IDLE-PC seguirá siendo bueno.

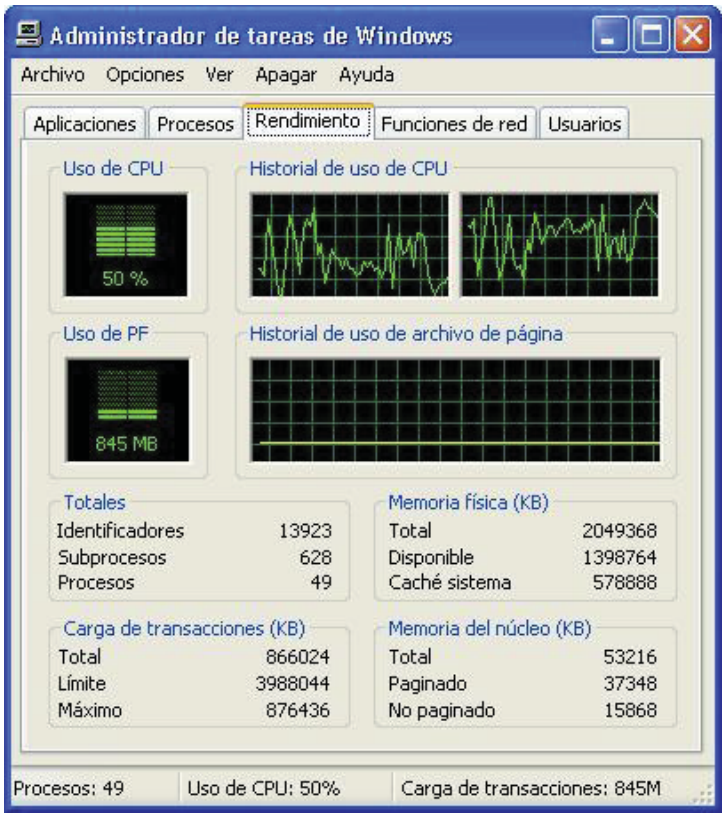


Fig. 5-1 CPU sin IDLE-PC Intel Core2Duo 6420 @2.13Ghz/3.25 Gb de RAM

La figura 5-1 muestra que el uso del procesador alcanza niveles del 50%, lo cual reduce considerablemente los recursos del computador que se utiliza en la simulación. En cambio, en la siguiente figura 5-2, se aprecia cómo se reduce considerablemente a un uso cercano al 1% del procesador. Debe considerarse por otro lado que este procedimiento se midió a partir del "encendido" de un solo *router* en el software en cuestión.

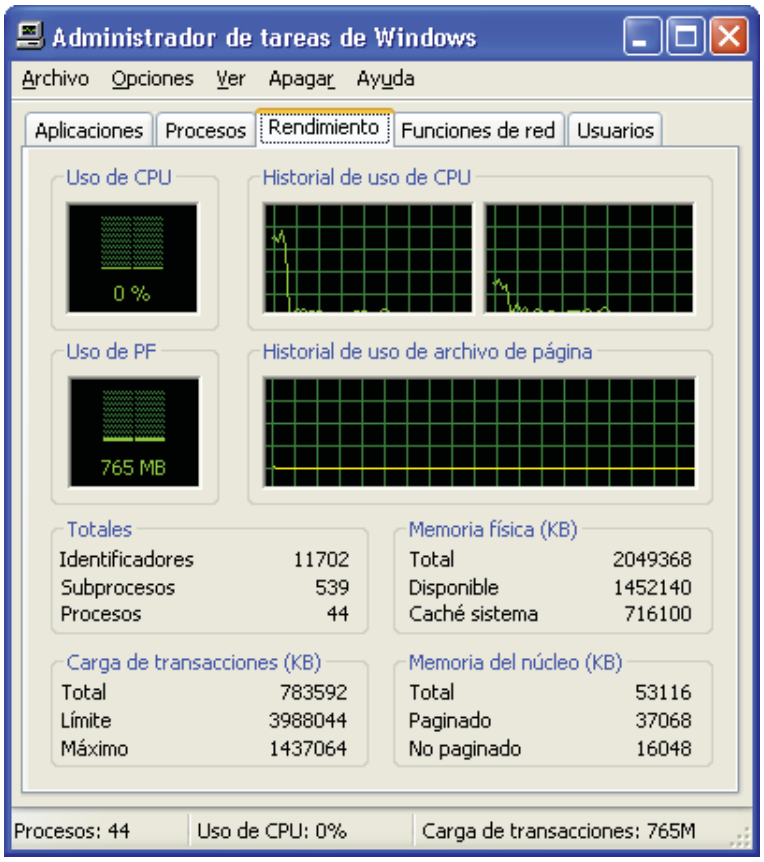


Fig. 5-2 CPU con IDLE-PC Intel Core2Duo 6420 @2.13Ghz/3.25 Gb de RAM

5.1.3 Herramientas de optimización del uso de memoria

Las herramientas con las que cuenta Dynamips (y que vienen incluidas en el *software*) para optimizar el uso de memoria, tanto real como virtual, del host emulador son las siguientes:

- Ghostios: Se encarga de reducir la cantidad de memoria real que se necesita del emulador para crear topologías con *routers* que corran a la vez, es decir, permite que el emulador comparta una parte de su memoria entre todos los *routers* que usen una misma imagen IOS de modo que cada *router* emulado no tenga que almacenar una copia idéntica de un mismo IOS en su memoria virtual. El resultado, en este caso, es un archivo que contiene la región de memoria compartida ubicado en el directorio "*working*", llamado c2600-i-mz.123-3h.image.ghost.

- Sparsemem: Se encarga de reducir la cantidad de memoria virtual que usa un router emulado, es decir, la memoria necesaria para la ejecución de una IOS, ya que sólo asigna la cantidad de memoria que la IOS va a usar en un momento determinado y no toda la memoria RAM configurada, lo que permite crear más *routers* virtuales por proceso Dynamips. Esta herramienta no está habilitada por defecto.

- Mmap: Realiza la correspondencia de archivos temporales del disco con la memoria virtual configurada en los *routers* emulados, para que cuando se requiera leer estos archivos, el sistema operativo ponga en caché sólo las secciones de los mismos que están siendo utilizados. Estos archivos tienen la extensión "ram", el tamaño de la memoria RAM configurada y se encuentran en el directorio "*working*" creado por GNS3 en cada simulación.

5.1.4 Dynagen

Dynagen es una interfaz escrita en Python que provee la gestión, mediante línea de comando (CLI), de las plataformas emuladas por Dynamips haciendo más fácil su uso. Usa el modo "Hypervisor" para comunicarse con Dynamips y ambas pueden correr en la misma o en diferente PC. También simplifica la gestión de las redes virtuales ya que implementa comandos para listar, iniciar, parar, reiniciar, suspender, reanudar los diferentes dispositivos emulados, además determina los valores de IDLEPC y realiza capturas de paquetes. [11]

A partir de sus últimas versiones, Dynagen es capaz de trabajar con el emulador de *firewalls* PEMU, el cual viene integrado en GNS3 dotando al emulador de capacidad de añadir *Firewalls* CISCO en las topologías. Además, es capaz de conectar de forma transparente a Dynamips los diferentes dispositivos virtuales como *switches* Ethernet, Frame-Relay y ATM soportados por Dynamips.

Dynagen usa un archivo de texto de fácil interpretación llamado "*Network File*", con extensión ".net", para conocer todas las características de *hardware* de los dispositivos de red a emular y realizar las interconexiones entre ellos.

5.1.5 *Network File*

Se trata de un archivo, escrito usando sintaxis INI (INI *file syntax*), que almacena la configuración de todos los dispositivos de red de la topología virtual a simular, como son los *routers*, *switches* y las interconexiones entre ellos. Este archivo puede especificar valores tan concretos como los descriptores de los adaptadores de red (NIO) que se encargan de la conexión con equipos reales o los puertos en los que trabajan dichos adaptadores de red, entre otros. [11]

La figura 5-3 muestra la plataforma base que conforman el motor del software simulador.

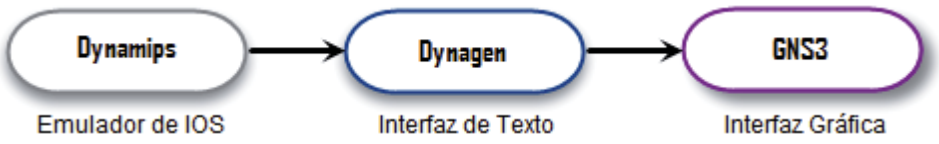


Fig. 5-3 Plataforma base de GNS3

5.1.6 Ventajas en la utilización de GNS3

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

La unión de Dynamips-Dynagen-GNS3, como se observa en la figura 5-3, crea una plataforma que permite el fácil diseño de topologías de red complejas ya que se realizan tan sólo arrastrando los componentes y dibujando líneas entre *routers* de forma intuitiva. Por lo tanto, GNS3 es idóneo para el entrenamiento de estudiantes que desean familiarizarse con dispositivos de red.

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

- Se encuentra disponible de forma gratuita en la red.
- Es fácil de instalar ya que todos los programas que necesita para funcionar se encuentran en un solo paquete de instalación.
- Está en constante actualización y periódicamente se puede encontrar versiones de la aplicación más robustas y con nuevas funcionalidades.
- Permite la conexión Telnet a la consola de un *router* virtual, de forma fácil directamente desde la interfaz gráfica.
- Alternativamente, también permite trabajar directamente desde consola de gestión de Dynagen.
- Permite la comunicación entre redes virtuales con redes del mundo real.
- Es apropiado para simular redes de grandes tamaños ya que permite que un cliente GNS3 pueda correr en una máquina diferente a la que contiene al emulador Dynamips, repartiendo el procesamiento entre diferentes PCs.
- Puede capturar los paquetes que pasan por enlaces virtuales y escribir los resultados de la captura en archivos que pueden ser interpretados por aplicaciones como Wireshark o tcpdumps.
- Los foros de Internet evidencian que es una aplicación ampliamente utilizada.

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

5.1.7 Requerimientos del Sistema en Windows XP

- Memoria RAM: Dynamips asigna por defecto 16MB de memoria RAM al compilador JIT para que realice la compilación del código del simulador en sistema Windows. Además, cada imagen IOS de un *router* real requiere una cantidad determinada de memoria RAM para funcionar, por lo tanto, inicialmente, la suma de los valores anteriores sería la cantidad de memoria RAM real necesaria para la simulación de un *router*. En la práctica este valor es mucho menor debido a que Dynamips implementa herramientas que permiten una optimización del uso de la memoria del simulador, las cuales ya fueron explicadas anteriormente [11].

- CPU: En un principio, Dynamips usará mucha cantidad de CPU porque no sabe cuándo la CPU virtual del *router* está inactiva, por lo tanto, ejecuta todas las instrucciones de las rutinas de inactividad de la IOS como si fueran instrucciones que realizan un trabajo real. El cálculo del valor de IDLE-PC hará que el consumo de CPU del emulador baje drásticamente. Si se elige un buen valor, la utilización de CPU por cada *router* será baja, con lo cual el funcionamiento del emulador será óptimo.

- Disco: Se necesita 39,65 MB de espacio de disco para almacenar a la aplicación GNS3 y a sus dependencias y emuladores asociados. Además, se necesita 0,19 MB para almacenar WinPCAP, lo que hace un aproximado de 40 MB de disco necesario. Este parámetro no es determinante a la hora de escoger un buen host donde montar la red virtual debido a que para la mayoría de PCs estos valores resultan fácilmente alcanzables.

Para poder estimar las capacidades recomendables para el buen funcionamiento de un equipo emulador, se hará un análisis comparativo donde se refleje el consumo de recursos cuando se emulen ciertos números de *routers* en equipos con características de procesamiento diferentes. Las figuras 5-4 y 5-5 muestran las características del escenario usado para la prueba y los resultados finales obtenidos en 2 emuladores diferentes.

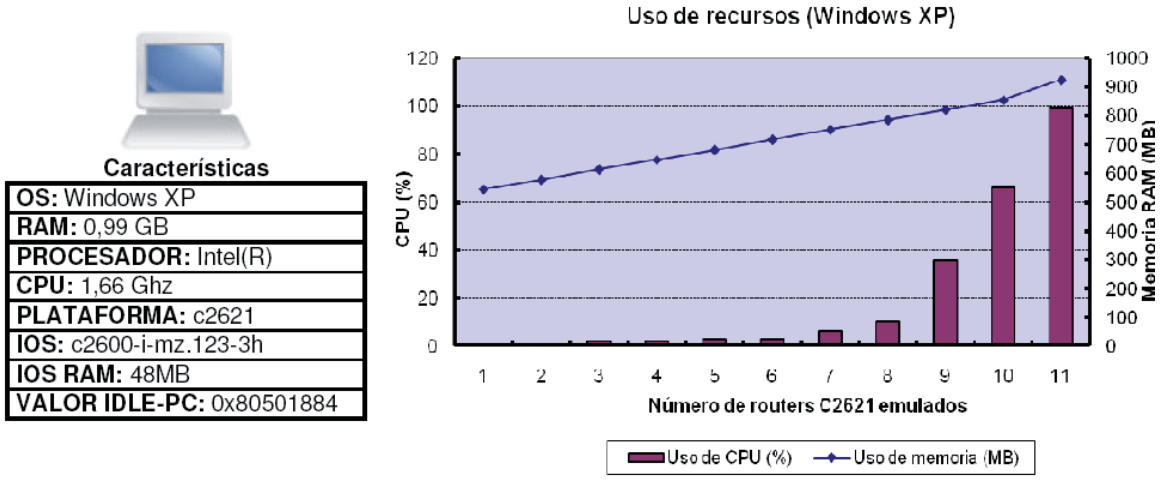


Fig. 5-4 Escenario I Windows y resultados de pruebas de uso de recursos

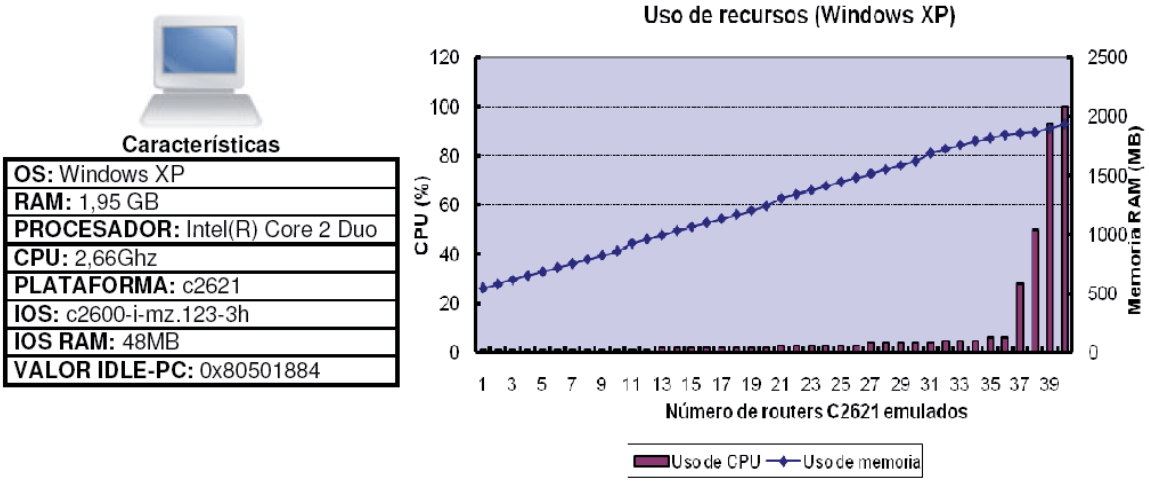


Fig. 5-5 Escenario II Windows y resultados de pruebas de uso de recursos

Las figuras anteriores comprueban que el número de *routers* que puede soportar un emulador es directamente proporcional a sus capacidades, así un PC con procesador de 1,6 Ghz. podría emular como máximo 11 *routers* y otro de 2 procesadores de aproximadamente 2,6 Ghz podría emular 40 *routers*, siempre y cuando se emule la misma plataforma, el mismo IOS con la misma RAM y se use el mismo valor de IDLEPC.

En la prueba se observa que la RAM usada aumenta de forma progresiva a pasos de 35MB por *router* aproximadamente, este valor obtenido es inferior al configurado en los *routers* (48MB) con lo cual se confirma la eficiencia de las herramientas de optimación de recursos de Dynamips. Por otro lado, el hecho de que el uso de la CPU aumente de forma drástica cuando se ha usado casi toda la memoria RAM disponible, indica que se está realizando "*swapping*", es decir, se empieza a reemplazar la memoria RAM por espacio en el disco.

Con respecto al comportamiento de GNS3 durante la prueba es importante resaltar que, pese a que la utilización de CPU parece que ha sido mínima durante gran parte de la prueba, esto no ha sido así, cada vez que se añadía un *router* o se abría una ventana de Telnet, la utilización de CPU subía de forma abrupta y fluctuaba por unos instantes, después se estabilizaba y descendía a valores mínimos nuevamente.

Cabe resaltar que todas las pruebas fueron realizadas cuando el emulador sólo tenía corriendo la aplicación GNS3 y el monitor de sistema de Windows para efectuar las mediciones. Además, los *routers* no tenían configuración alguna.

5.1.8 Requerimientos del sistema en Linux (Ubuntu 9.4)

- Memoria RAM: En Linux, la memoria RAM requerida teórica para la emulación de un *router* sería la que Dynamips asigna por defecto al compilador JIT (64MB) y la cantidad de RAM que cada imagen IOS requiere para funcionar en un equipo real, aunque, como ya se ha explicado, en la práctica se necesitan valores inferiores.

- CPU: En Linux también se toma en cuenta el valor de IDLE-PC para estimar los requerimientos de CPU del emulador.

- Disco: El espacio total necesario en disco para la instalación de GNS3 en Linux es de aproximadamente 117,2MB, valor que es mayor al requerido en Windows debido a la necesidad de instalación adicional de dependencias. Este parámetro no es determinante a la hora de escoger un buen *host* de trabajo.

Se repiten las mismas pruebas realizadas en el apartado anterior con dos equipos de características de procesamiento diferentes. Las siguientes figuras 5-6 y 5-7, muestran los resultados obtenidos.

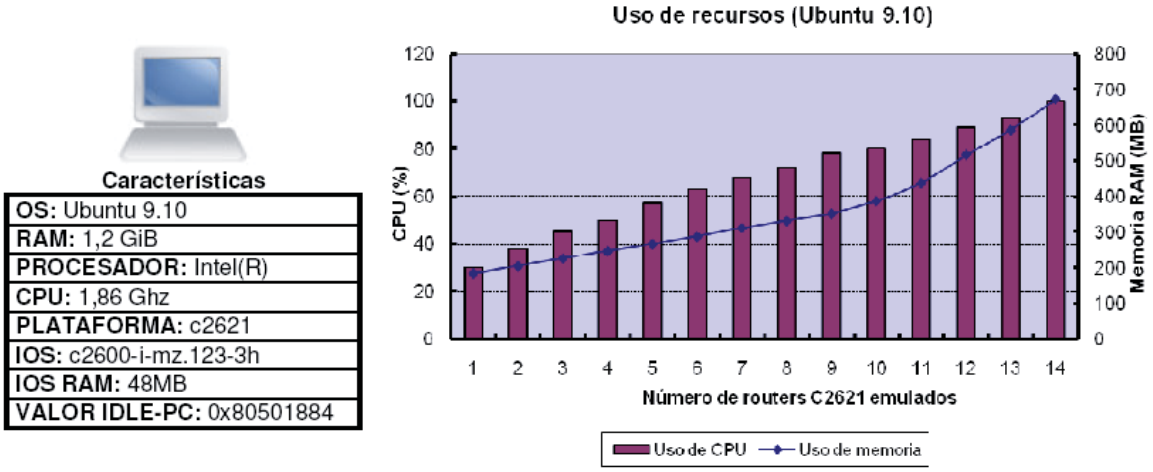


Fig. 5-6 Escenario I Ubuntu y resultados de pruebas de uso de recursos

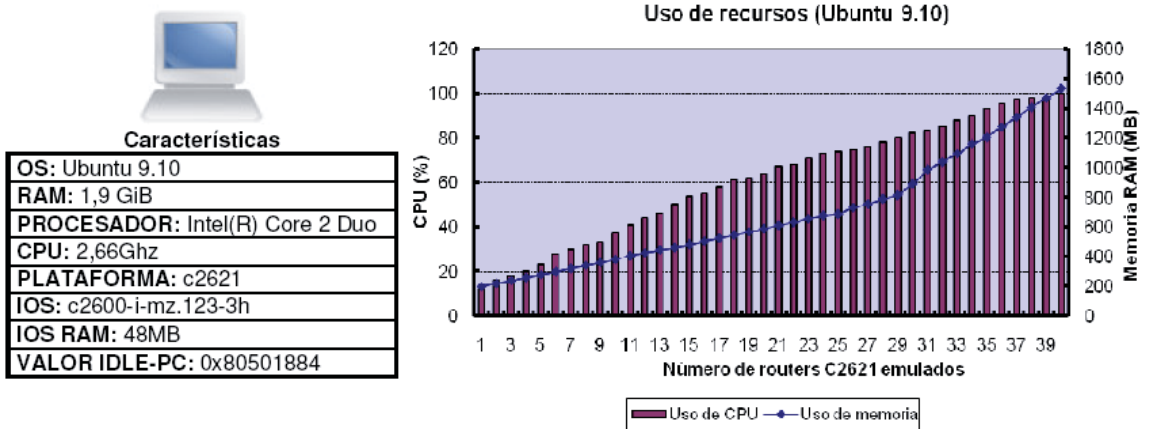


Fig. 5-7 Escenario II Ubuntu y resultados de pruebas de uso de recursos

Los resultados obtenidos en Linux muestran un aumento progresivo del uso de la CPU cuando se añade un nuevo *router*, de modo que, se puede notar más claramente la estrecha relación entre la velocidad de la CPU de un emulador y la cantidad de *routers* que puede emular. Un emulador con algo menos de 2Ghz de CPU puede emular como máximo 14 *routers* y otro con 2 procesadores de 2Ghz cada uno aproximadamente podría emular 40 *routers*, siempre y cuando se emule la misma plataforma, el mismo IOS con la misma RAM y se use el mismo valor de IDLE-PC [11].

En esta prueba también se comprueba la eficiencia del uso de las herramientas de optimación de recursos de Dynamips, ya que al agregar un *router* a la topología, se aumenta aproximadamente en 22 MB la RAM usada, en vez de los 48 MB configurados. Cabe resaltar que cuando la utilización de la CPU del emulador es aproximadamente 80%, la memoria RAM empieza a subir más rápidamente, por lo cual, éste podría ser un valor limitador de rendimiento. Este sistema es más estable puesto que no se observaron cambios bruscos en el uso de la CPU cuando se añadían nuevos elementos, sino que su aumento fue progresivo.

Cabe resaltar, otra vez, que todas las pruebas fueron realizadas cuando el emulador sólo tenía corriendo la aplicación GNS3 y el monitor de sistema de Linux para efectuar las mediciones. Además, los *routers* no tenían configuración alguna.

5.1.9 Observaciones y Recomendaciones

Un emulador con sistema operativo Windows requerirá más memoria RAM (~35MB por router) que un emulador con sistema operativo Linux (~22MB por router) para un mismo número de *routers* emulados. Por otro lado, teniendo en cuenta estas pruebas, no es fácil estimar el número de *routers* que puede soportar eficientemente un emulador, con determinadas capacidades de procesamiento, que usa Windows. Sin embargo, para Linux se puede decir que un emulador con procesador Intel de 2 Ghz y 1 GB de RAM puede emular 10 *routers* y otro con procesador Intel Core 2 Duo de 2,6 Ghz y 2 GB de RAM puede emular 28 *routers*, para tener como máximo un 80% de CPU ocupada [11].

El PC que se debe elegir para la instalación del simulador de red GNS3 debe tener un buen balance entre CPU y Memoria. Algunas recomendaciones adicionales serían:

- Elegir un buen valor de IDLE-PC. Un valor aceptable es aquel que hace que el CPU use entre 0% y 50% para un solo *router* y un buen valor es aquel que sólo usa 10%.
- Usar una IOS que requiera las más bajas cantidades de RAM para funcionar pero que cuente con las funcionalidades que se necesita.
- No crear topologías que usan más del 80% del CPU del emulador, ya que el uso de capacidades de los *routers*, como MPLS, OSPF ó IS-IS, hará que este valor aumente.

- Usar el sistema operativo Linux en los casos en los que se requieran probar "*Bancos de prueba*" que necesiten estar activos por mucho tiempo, ya que es más estable.

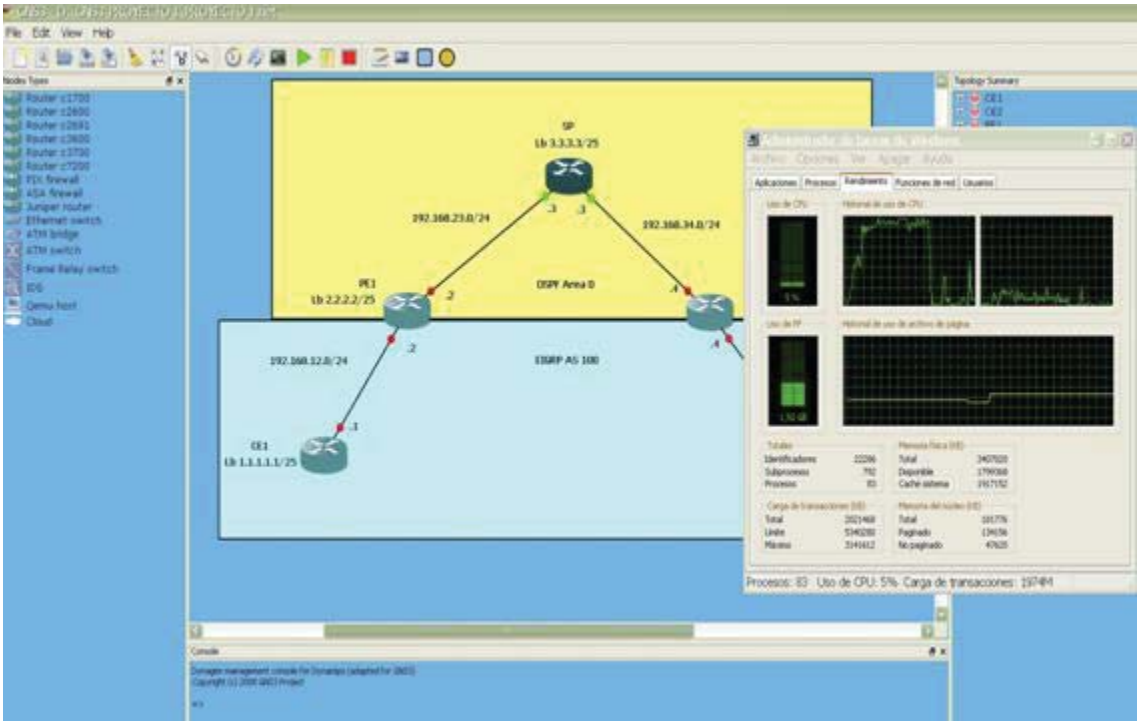


Fig. 5-8 Configuración de IDLE PC en GNS3

En la figura 5-8 se puede apreciar la disminución de uso de recursos de la CPU, una vez que se aplica la configuración adecuada de IDLE PC. Cabe señalar que la plataforma se montó en Windows XP por la compatibilidad de los *softwares* que se encontraban a disposición.

5.1.10 Emulación de *Routers* CISCO

Como ya se ha indicado anteriormente, para emular *routers* CISCO reales se necesita una imagen CISCO IOS perteneciente al *router* que contiene las características que se quieren "clonar". En este sentido, el emulador habilitará un número de ranuras o "slots" dependiendo del tipo de plataforma que se emula y en cada una de esas ranuras se podrán colocar solo ciertos tipos de adaptadores de interfaces. Por lo tanto, si se quiere añadir "capacidades de *hardware*" en un *router* virtual, se debe seleccionar el tipo de adaptador de red que éste puede soportan en la configuración del *router* virtual.

En cuanto a la configuración de cada uno de los routers utilizados en la emulación, los adaptadores que se pueden utilizar de acuerdo al *software* GNS3 se listan a continuación en la figura 5-9.

	Adaptadores de Interfaces Disponibles	
Routers	**Nombre**	**Descripción**
1700s	WIC-1T	1 puerto serie
	WIC-2T	2 puertos serie
	WIC-1ENET	1 puerto Ethernet
2600s	NM-1E	1 puerto Ethernet
	NM-4E	4 puertos Ethernet
	NM-1FE-TX	1 puerto FastEthernet
	NM-16ESW	Módulo de switch Ethernet (16 puertos)
	NM-NAM	Conecta el router virtual a un PC virtual
	NM-IDS	Conecta el router virtual a un PC virtual
	WIC-1T	1 puerto serie
	WIC-2T	2 puertos serie
3600s	NM-1E	1 puerto Ethernet
	NM-4E	4 puertos Ethernet
	NM-1FE-TX	1 puerto FastEthernet
	NM-16ESW	Módulo de switch Ethernet (16 puertos)
	NM-4T	4 puertos serie
	Leopard-2FE	Puerto automático FastEthernet en slot 0
3700s	NM-1FE-TX (FastEthernet, 1 port)	1 puerto FastEthernet
	NM-4T	4 puertos serie
	NM-16ESW	Módulo de switch Ethernet (16 puertos)
	GT96100-FE	2 puertos integrados
	NM-NAM	Conecta el router virtual a un PC virtual
	NM-IDS	Conecta el router virtual a un PC virtual
	WIC-1T	1 puerto sere
	WIC-2T	2 puertos serie
7200s	C7200-IO-FE	Solo puerto FastEthernet en slot 0
	C7200-IO-2FE	2 puertos FastEthernet en slot 0
	C7200-IO-GE-E	Sólo Puerto GigabitEthernet en slot 0
	PA-FE-TX	1 puerto FastEthernet
	PA-2FE-TX	2 puertos FastEthernet
	PA-4E	4 puertos Ethernet
	PA-8E	8 puertos Ethernet
	PA-4T+	4 puertos serie
	PA-8T	8 puertos serie
	PA-A1	Puerto ATM
	PA-POS-OC3	Puerto POS
	PA-GE	1 puerto GigabitEthernet

Fig. 5-9 Lista de adaptadores de interfaz en GNS3 [11]

CAPÍTULO 6

SIMULACIÓN DE LA RED MPLS

6.1 Determinando la topología

Para determinar la composición de la topología a implementar, se realizó una verificación de los distintos puertos o interfaces con las que se contaba en los dispositivos actuales del Laboratorio de Redes de la Escuela de Ingeniería Eléctrica de la Pontificia Universidad Católica de Valparaíso (PUCV). Con esta revisión de los *routers* recientemente adquiridos y equipos anteriores se obtuvo el siguiente listado que se observa en la Tabla 6-1:

Tabla 6-1 Equipos Laboratorio de Redes Escuela Ingeniería Eléctrica PUCV

Router	Cantidad	Módulos Libres	Módulos/Puertos
Cisco 1841	5	2 WIC c/u	2 Fast Ethernet 2 Serial
Cisco 2811	1	4 HWIC	
Cisco 2621	1	1 EN / 2 WIC	2 Ethernet
Cisco 2621	1	1 EN / 1 WIC	2 Ethernet 2 Serial
Cisco 2801	1	1 WIC	2 Fast Ethernet

Debido al tipo de puertos y el número de estos, se plantea la posibilidad de una red básica MPLS con redundancia en conexiones seriales y un *Backbone* principal conectado mediante *Fast Ethernet*, aunque se realizarán configuraciones para poder determinar un mismo ancho de banda a toda la red MPLS, limitándolo a 200Kbps.

En esta etapa de simulación, se realizó una configuración básica de lo que corresponde al *Backbone* principal de la red MPLS del proveedor con conexión a un cliente con un *Router* considerado como matriz o principal y otro equipo que será considerado como un segundo lugar geográfico del cliente.

En la red montada para el proveedor, se utilizaron 3 *routers* 7200 que son los que más se acercan a las características de los Cisco 1841 que se utilizan luego como PE y el Cisco 2801 a utilizar como *router* central (SP en la simulación).

6.1.1 Topología de red escogida

La topología escogida, se presenta a continuación en la figura 6-1, y está conformada por dos *routers* de borde, llamados PE1 y PE2, un *router* central del ISP, en este caso llamado SP y finalmente dos *router* que representan dos sitios de un cliente geográficamente separados, llamados CE1 y CE2.

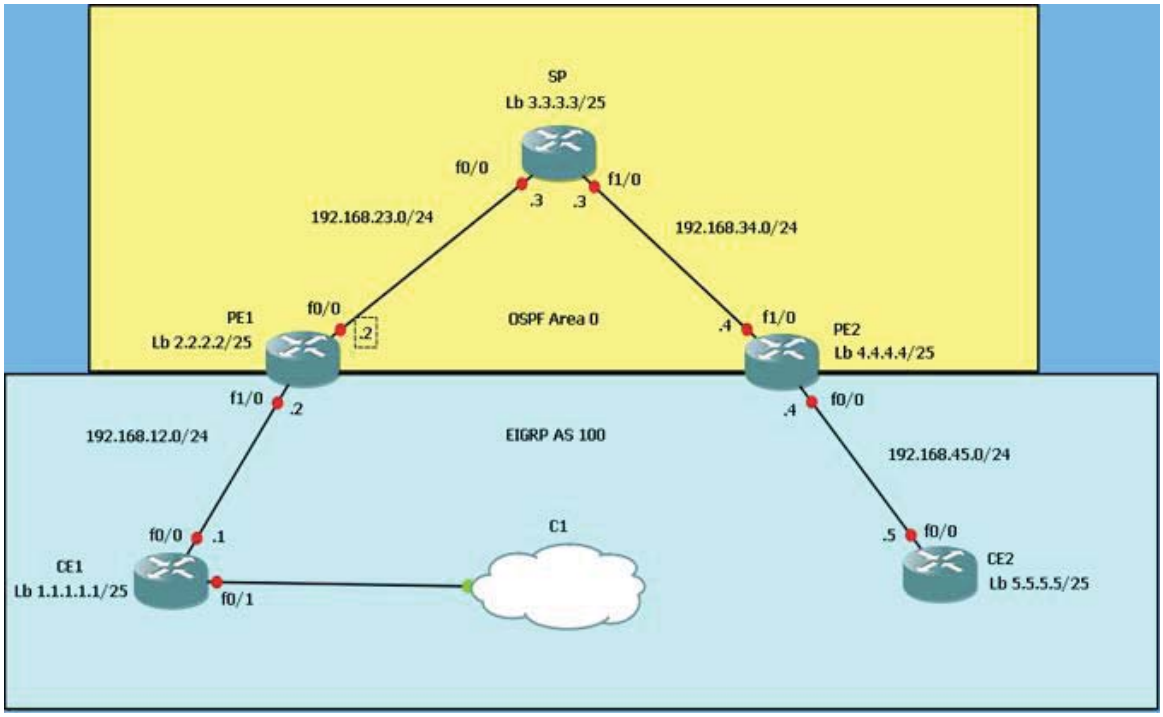


Fig. 6-1 Topología creada en GNS3

6.1.2 Metas por Configuraciones

- Configurar todas las direcciones IP especificadas en el cuadro de topología.
- Configurar una interfaz de loopback0 en cada *router*:
  CE: 1.1.1.1 / 25
  PE1: 2.2.2.2 / 25
  SP: 3.3.3.3 / 25
  PE2: 4.4.4.4 / 25
  CE: 5.5.5.5 / 25
- Configurar el área OSPF 0 en el lado del proveedor (*Router* PE1, PE2 y SP).
- Publicación de las interfaces *Loopback* en la configuración OSPF.
- Aseguramiento de plena accesibilidad en el dominio OSPF.

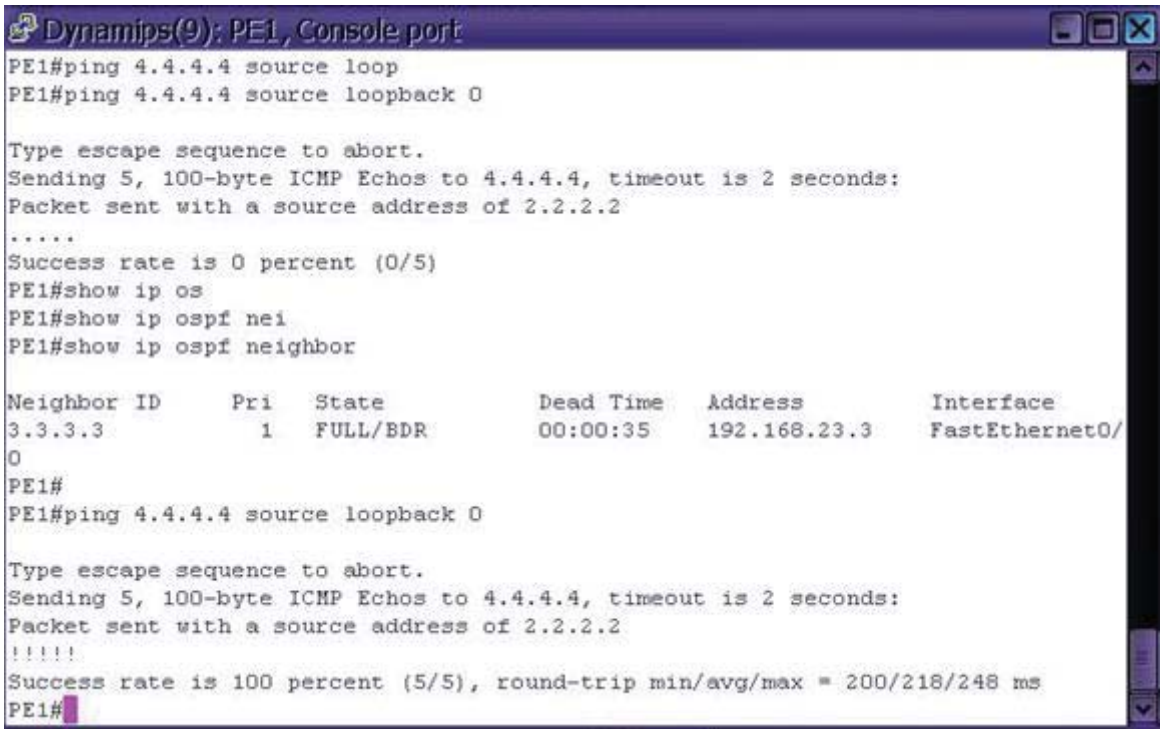


Fig. 6-2 Comprobación Ping desde *Loopback* 2.2.2.2 a 4.4.4.4

La figura 6-2 muestra la comprobación de conectividad entre los *router* PE1 y PE2, cuyo tráfico pasa a través de SP (el router central en la red del proveedor).

- Configuración de MPLS en todas las interfaces físicas en el dominio de proveedor de servicios, sin configurar MPLS sobre interfaces físicas que apuntan hacia el cliente.
- Configurar VRF "CUSTOMER" en PE1 y PE2 de la siguiente forma:
  RD 100:1
  Route-target both 1:100

- En el *router* PE1 y PE2 agregar las interfaces que apuntan hacia el cliente a la VRF ya creada.
- Asegurar conectividad haciendo ping desde la VRF, desde PE1:
  ping vrf customer 192.168.12.1

Las siguientes figuras 6-3 y 6-4 muestran una comprobación de conectividad a través del comando ping entre el ISP y el cliente; la primera, entre PE1 y CE1 y la segunda, entre PE2 y CE2. Esto demuestra la conexión de cada uno de los sitios del cliente y el proveedor de servicios.

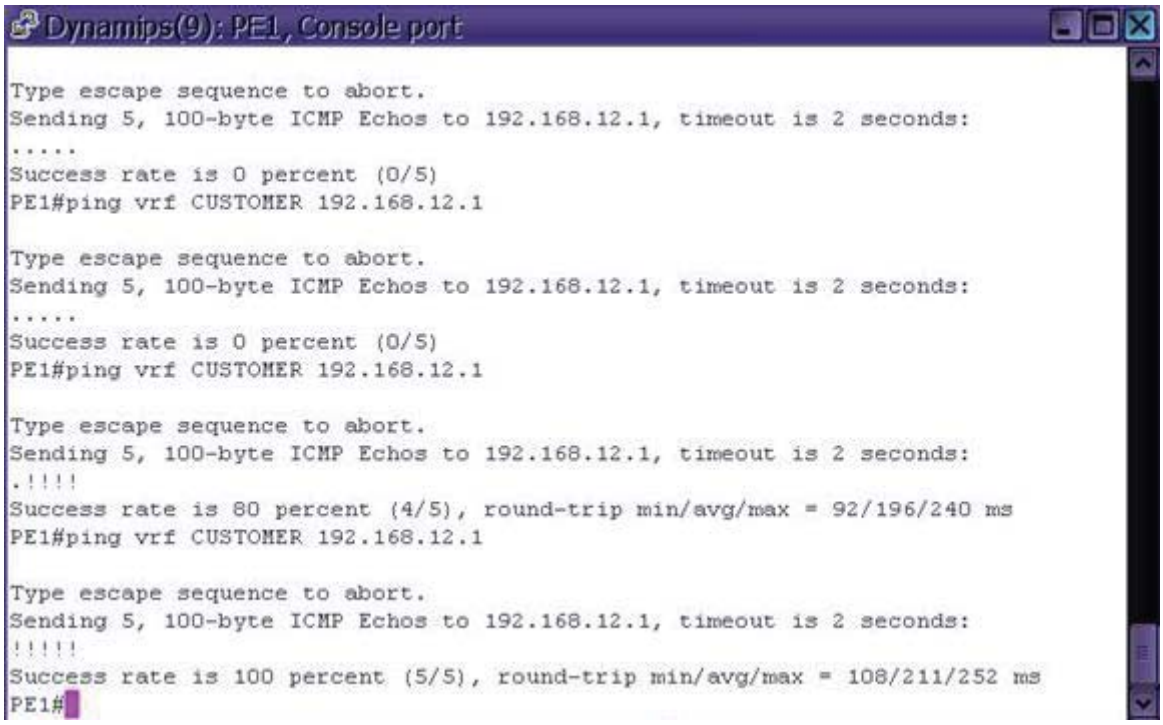


Fig. 6-3 Comprobación Ping desde PE1 a CE1

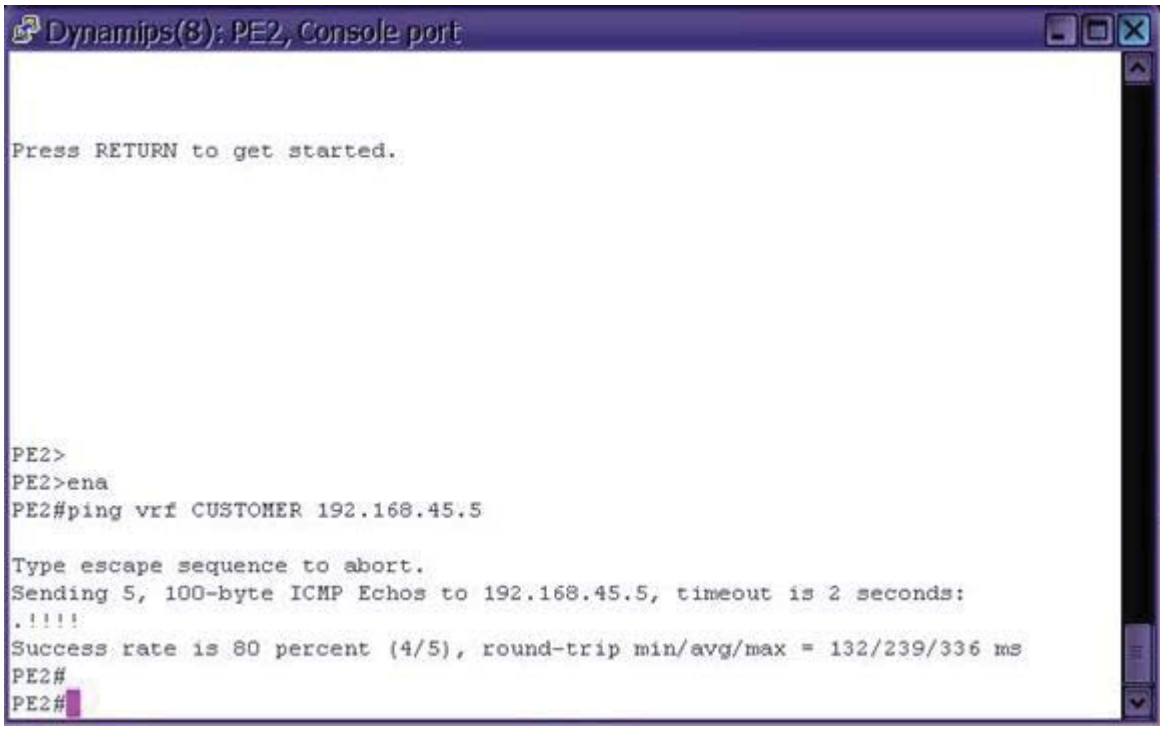


Fig. 6-4 Comprobación Ping desde PE2 a CE2

- Configurar EIGRP AS100 en el *router* CE1 y CE2. Publicar las *loopbacks*.
- Configurar EIGRP en el *router* PE1 y PE2 para el VRF correcta "CUSTOMER".
- Asegurarse de que se ha establecido una relación entre el vecino EIGRP *router* CE1 y el PE1, y entre el PE2 y el CE2.
- Ver si han aprendido las rutas mediante el uso de "show *ip route vrf* cliente".

Para comprobar la correcta relación entre la red EIGRP del cliente y la VRF CUSTOMER por el lado del ISP, la figura 6-5 muestra una vista del comando *show ip eigrp vrf CUSTOMER neighbors* aplicado en el equipo PE2, que arroja como resultado la IP del equipo CE2 (192.168.45.5).

Dynamips(8): PE2, Console port

```
PE2#show ip ei
PE2#show ip eigrp ne
PE2#show ip eigrp 1 ne
PE2#show ip eigrp 1 neighbors
IP-EIGRP neighbors for process 1
PE2#show ip eigrp vr
PE2#show ip eigrp vrf CUSTOMER ne
PE2#show ip eigrp vrf CUSTOMER neighbors
IP-EIGRP neighbors for process 100
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   192.168.45.5            Fa0/0             11 00:06:07  473  4257  0  3
PE2#
PE2#
PE2#
PE2#show ip rou
PE2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     2.0.0.0/25 is subnetted, 1 subnets
O       2.2.2.0 [110/3] via 192.168.34.3, 13:34:54, FastEthernet1/0
     3.0.0.0/25 is subnetted, 1 subnets
O       3.3.3.0 [110/2] via 192.168.34.3, 13:34:54, FastEthernet1/0
     4.0.0.0/25 is subnetted, 1 subnets
C       4.4.4.0 is directly connected, Loopback0
O    192.168.23.0/24 [110/2] via 192.168.34.3, 13:34:54, FastEthernet1/0
C    192.168.34.0/24 is directly connected, FastEthernet1/0
PE2#
PE2#
```

Fig. 6-5 Comando *Show ip eigrp vrf CUSTOMER neighbors* en PE2

Configurar BGP 1 entre el *Router* PE1 y PE2, verificar que las actualizaciones usan como fuentes las interfaces de las *loopbacks*.

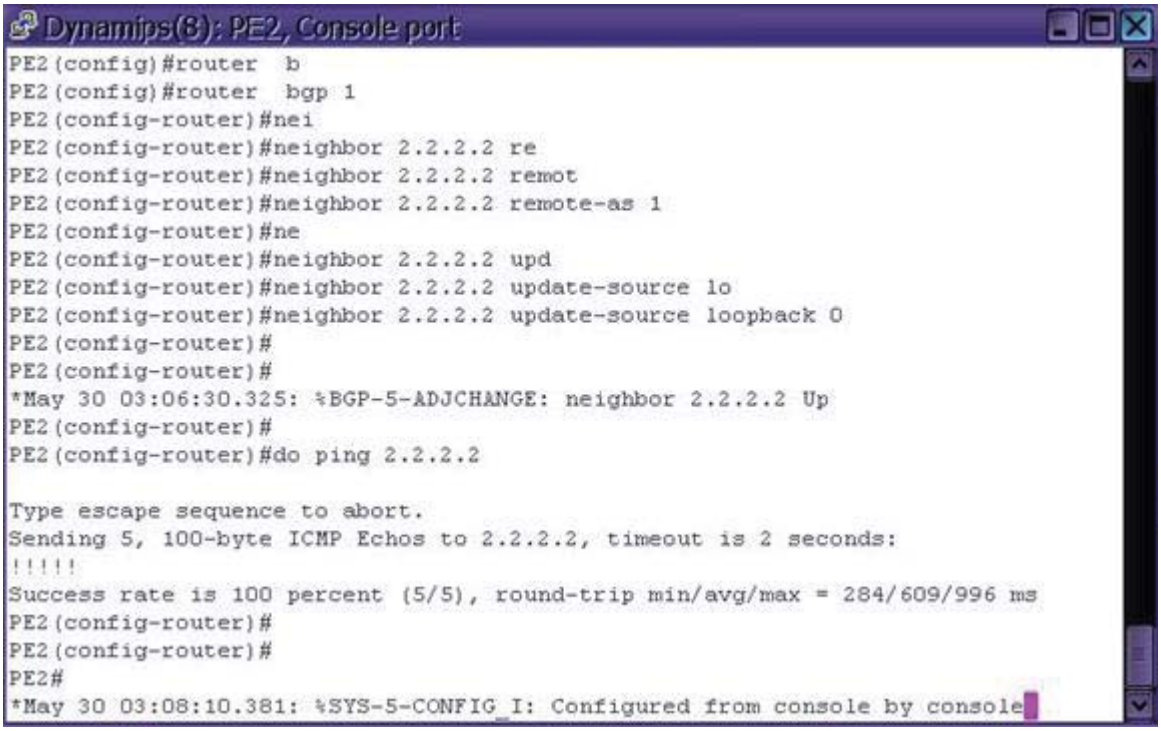


Fig. 6-6 Comprobación de BGP con Ping desde PE2 a Loopback 2.2.2.2

La figura 6-6 muestra la conectividad entre la red EIGRP del cliente y la red OSPF del ISP a través de la redistribución en BGP

- Se configuran las familias BGP correspondientes, direcciones y se asegura que la información de comunidades se envían entre equipos vecinos.
- Redistribuir EIGRP en BGP, usando las correctas direcciones de familias para el VRF "CUSTOMER".
- Redistribuir la información de BGP de nuevo en EIGRP, usando las siguientes métricas:

a) Ancho de banda: 200 kbps

b) Demora: 1000

c) Fiabilidad: 255

d) Carga: 1

e) MTU: 1500

- Asegúrese de que tiene plena conectividad entre el *router* CE1 y CE2. Se debe ver cada una de las rutas EIGRP que se han realizado sobre el *Backbone* del proveedor de servicios MPLS.

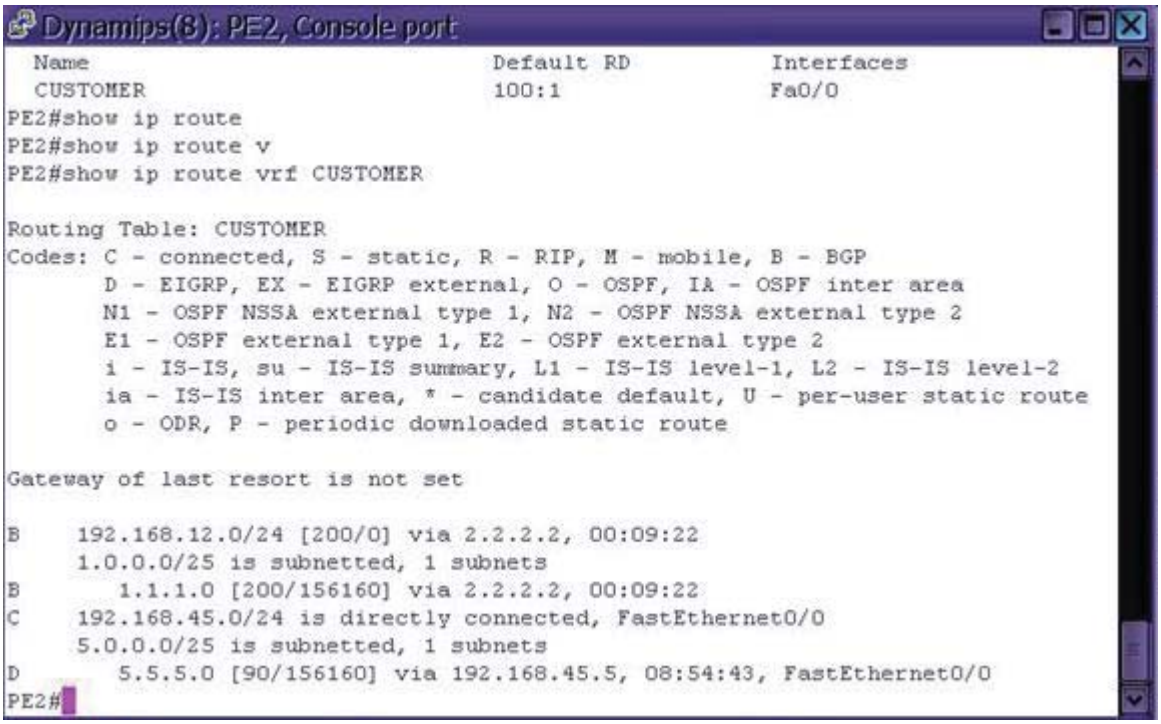


Fig. 6-7 Vista del comando *Show ip route vrf CUSTOMER* desde PE2

La figura 6-7 ilustra el uso del comando *show ip route vrf* en el router PE2, que permite ver las redes que se encuentran en este "enrutamiento virtual" para separar los tráficos de cada uno de los clientes que eventualmente podrían utilizar el mismo equipo PE2 como borde de entrada al proveedor.

6.2 Resumen del Capítulo

Se ha dado a conocer la configuración básica de la red MPLS con la cual posteriormente se realizó el montaje en laboratorio para probar en ella, algunas configuraciones de Calidad de Servicio, para esto se realizaron pruebas con distintos tipos de *software* tanto para simular el tráfico de datos como para tomar muestras y análisis de los mismos.

Se logró verificar el funcionamiento de la aplicación GNS3 y sus capacidades para cargar los distintos IOS de los equipos a utilizar y con esto se pudo montar la red básica con la que ya se cuenta para realizar las pruebas de tráfico de datos y luego montar en los equipos de laboratorio. Se consideró una red interna del cliente en protocolo EIGRP y por parte del proveedor una en red en base a OSPF, ambos comúnmente utilizados en las organizaciones.

CAPÍTULO 7

IMPLEMENTACIÓN DE LA RED EN LABORATORIO

7.1 Implementación para pruebas

7.1.1 Topología establecida

De acuerdo al estudio previo de los equipos disponibles en el laboratorio, se eligió la siguiente topología que se muestra en la figura 7-1. En la imagen se aprecia claramente el dominio MPLS representando un ISP y en los extremos, los *routers* de un cliente con dos sitios A y B, separados geográficamente.

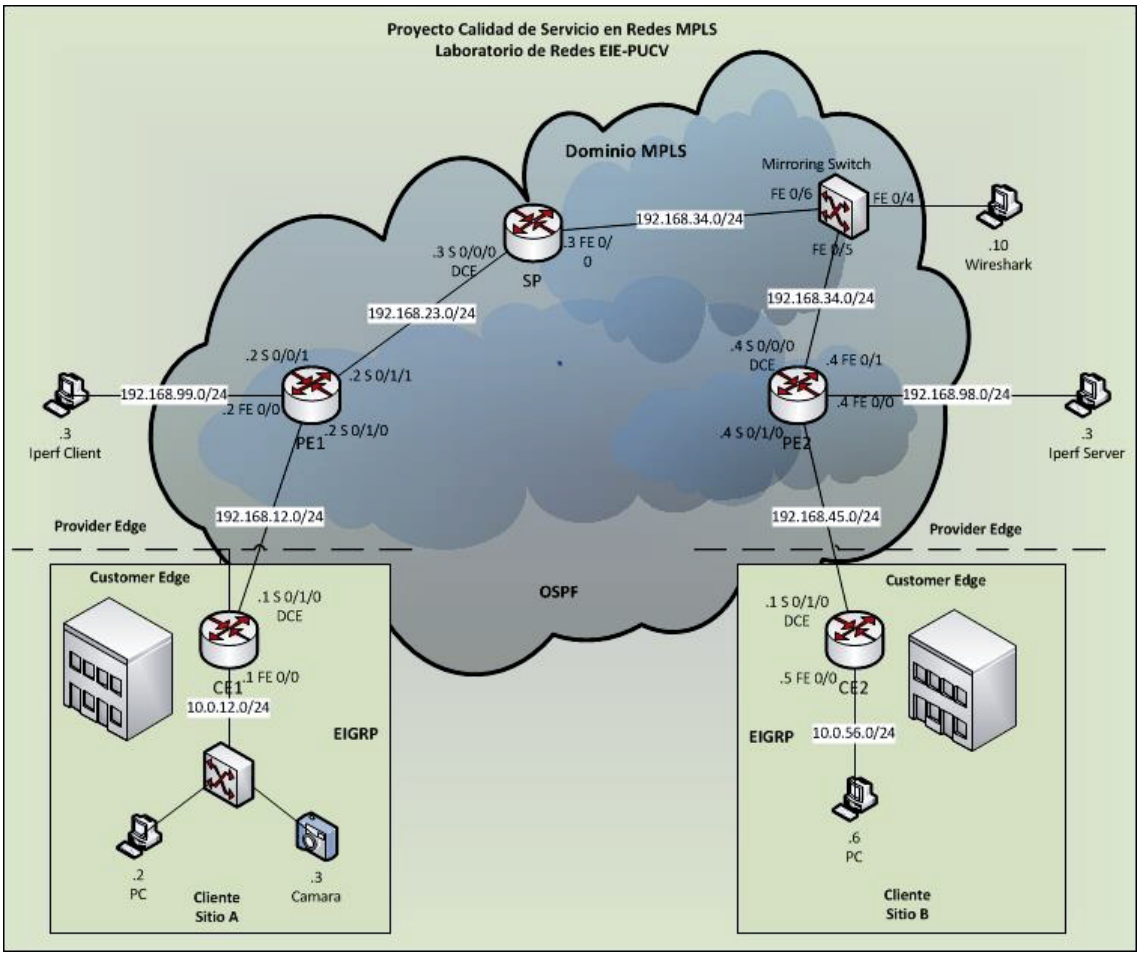


Fig. 7-1 Topología de la red MPLS en laboratorio

7.1.2 Composición de la Red

Todos los computadores del laboratorio cuentan con sistema operativo *Windows* XP profesional. En el caso de los computadores conectados directamente a los *routers* PE1 y PE2 cumplieron la función de generar distintos tipos de tráfico durante las pruebas y además se instaló en ellos la herramienta *iPerf* bajo su interfaz gráfica *jPerf* con motor Java, la cual permite facilitar las configuraciones para simular y realizar mediciones de tráfico tanto UDP como TCP, permitiendo elegir distintos parámetros.

Para poder realizar mediciones del tráfico que circulaba por la red se incorporó un *Switch* Cisco 3550 y conectando a éste un computador con el software *Wireshark* que permite revisar a nivel de capas del modelo OSI el tráfico que circula por la red MPLS.

Al igual que en la simulación con GNS3 los equipos del laboratorio se configuraron de la misma forma en el sentido más básico, vale decir que se configuraron los dominios EIGRP de cada sitio del cliente, el dominio OSPF en el proveedor y habilitando MPLS para el tráfico entre las *router* de borde en el proveedor. Se redistribuye el tráfico EIGRP a través de BGP para poder comunicar los clientes entre sí a través del *backbone* MPLS.

La Tabla 7-1 muestra un listado de las interfaces físicas que se utilizaron durante las pruebas.

Tabla 7-1 Interfaces utilizadas en la topología de red

Equipo	Modelo	Interfaz	IP
CE1	1841	Loopback 0	1.1.1.1
		Serial 0/0/0 DCE	192.168.12.1
		FastEthernet 0/0	10.0.12.1
Switch B	3550	FastEterhnet 0/2	-
		FastEterhnet 0/3	-
		FastEterhnet 0/4	-
PC11	Dell	FastEthernet	10.0.12.2
Cámara web	Cisco	FastEthernet	10.0.12.3
CE2	1841	Serial 0/1/0	192.168.45.5
		FastEthernet 0/0	10.0.56.5
		Loopback 0	5.5.5.5
PC10	Dell	FastEthernet	10.0.56.6

PE1	1841	Serial 0/1/1	192.168.23.2
		Serial 0/1/0	192.168.12.2
		Loopback 0	2.2.2.2
		FastEthernet 0/1	192.168.99.2
PC9	Dell	FastEthernet	192.168.99.3
SP	2811	Loopback 0	3.3.3.3
		Serial 0/0/0 DCE	192.168.23.3
		FastEthernet 0/0	192.168.34.3
Switch A	3550	FastEterhnet 0/4	-
		FastEterhnet 0/5	-
		FastEterhnet 0/6	-
PC5	Dell	FastEterhnet	192.168.34.10
PE2	1841	Loopback 0	4.4.4.4
		FastEterhnet 0/1	192.168.34.4
		Serial 0/1/0	192.168.45.4
PC4	Dell	FastEterhnet	192.168.98.3

7.1.3 Pruebas de conectividad

Se realizaron distintas pruebas de conectividad y verificación de la trazabilidad de los equipos. Estas pruebas se presentan en las figuras 7-2 y 7-3 a continuación:

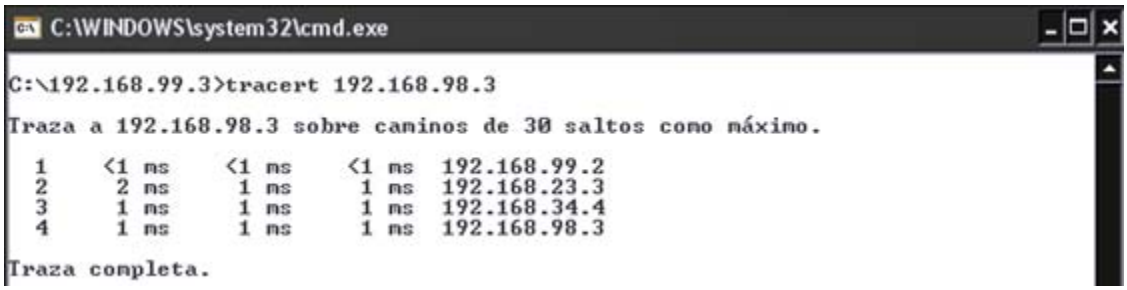


Fig. 7-2 Comando tracert desde PC 192.168.99.3 a 192.168.98.3

En la figura 7-2 se pueden ver los saltos de red utilizando el comando *tracert* desde el cliente al servidor, se logra apreciar las direcciones IP de los *routers* del proveedor de la red MPLS.

En la figura 7-3 se realiza una prueba de conectividad a través de una ventana DOS con el comando Ping desde el computador que cumple la función de cliente *jPerf* con IP 192.168.99.3 hacia el computador que es servidor *jPerf* con IP 192.168.98.3.

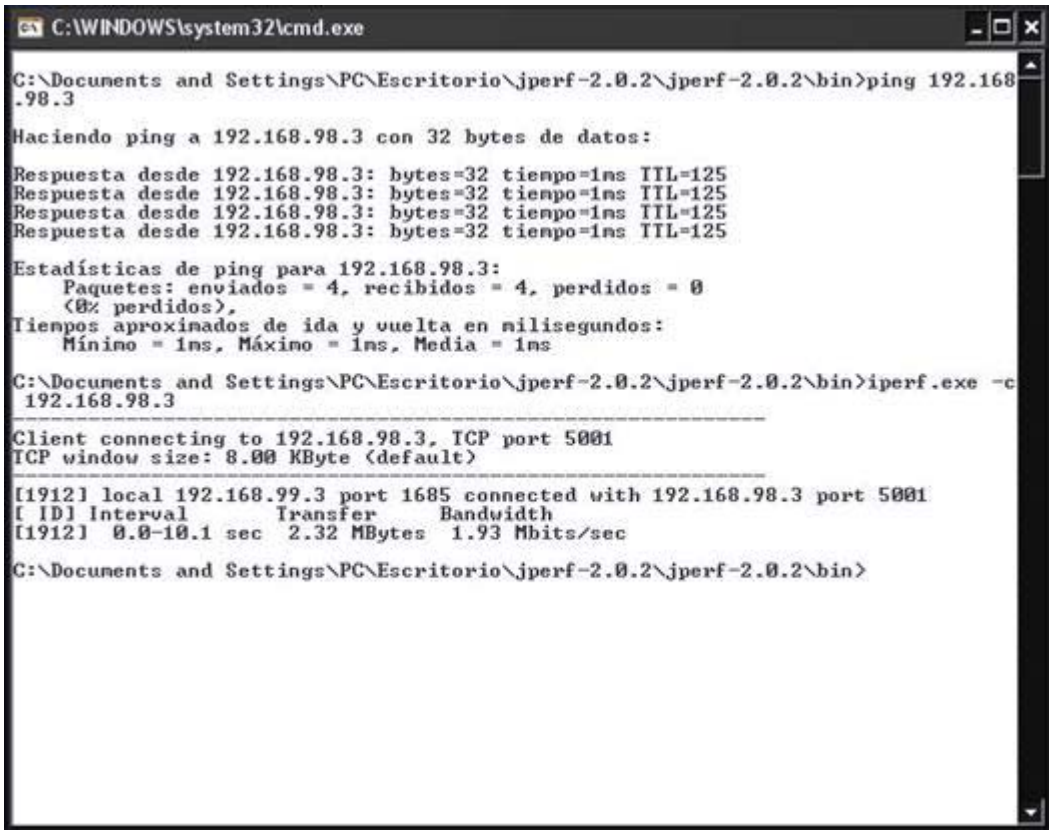


Fig. 7-3 Prueba de conectividad servidor-cliente

7.1.4 Pruebas con Tráfico Sobre la Red

Dentro de las pruebas que se realizaron con tráfico, se contempló una configuración de servidor FTP en el computador, conectado directamente al *router* CE1 del cliente en sitio A, una cámara IP de vigilancia con interfaz web, transmisión de video a través del reproductor VLC en modo emisor vía http y además generando tráfico entre cliente y servidor sobre la red MPLS del ISP con *jPerf*, para medir de qué manera iba disminuyendo el ancho de banda utilizado por esta aplicación al generar tráfico TCP.

Los resultados se presentan desde la figura 7-4 a 7-7, donde claramente se observa cómo es dividido el ancho de banda al iniciar la transmisión de datos con la aplicación *jPerf* y al ir iniciando las otras transmisiones una tras otra de manera independiente. Vale decir que para esta prueba con *jPerf*, primero se transmitió sólo tráfico a través del *software*, que se aprecia en color verde, luego se transmitió nuevamente en conjunto con el tráfico vía FTP (color azul). Los resultados se muestran a intervalos de 1 segundo.

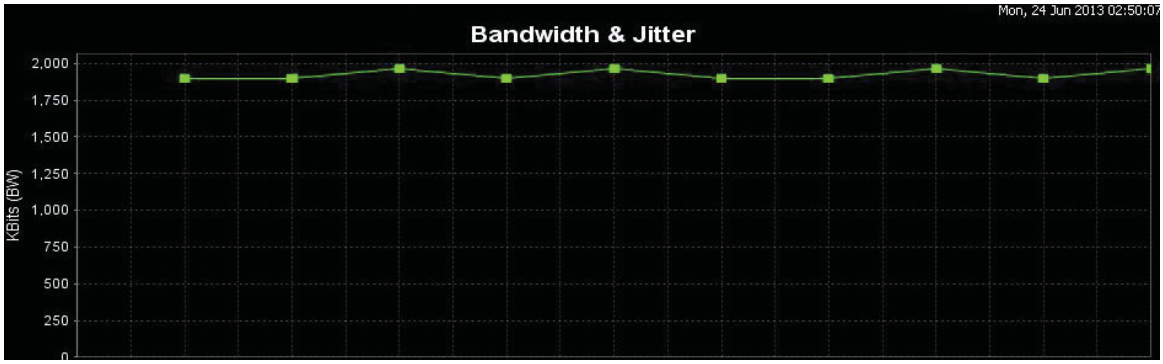


Fig. 7-4 Prueba sólo con tráfico de *jPerf*

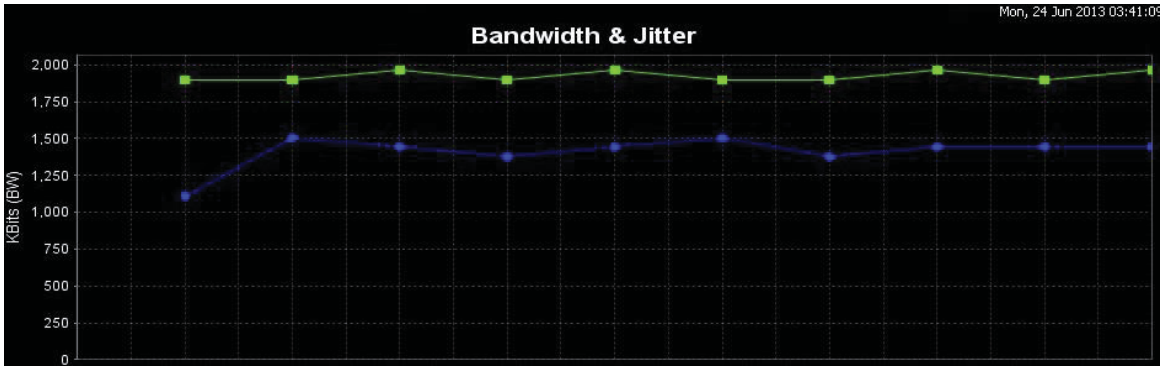


Fig. 7-5 Se suma tráfico FTP en azul

Después se habilitó la cámara web (color blanco) cuyo tráfico se observa en la figura 7-6, y finalmente en la figura 7-7, se aprecia el tráfico generado con el *software VLC Player* donde se transmitió video vía HTTP (color rojo).

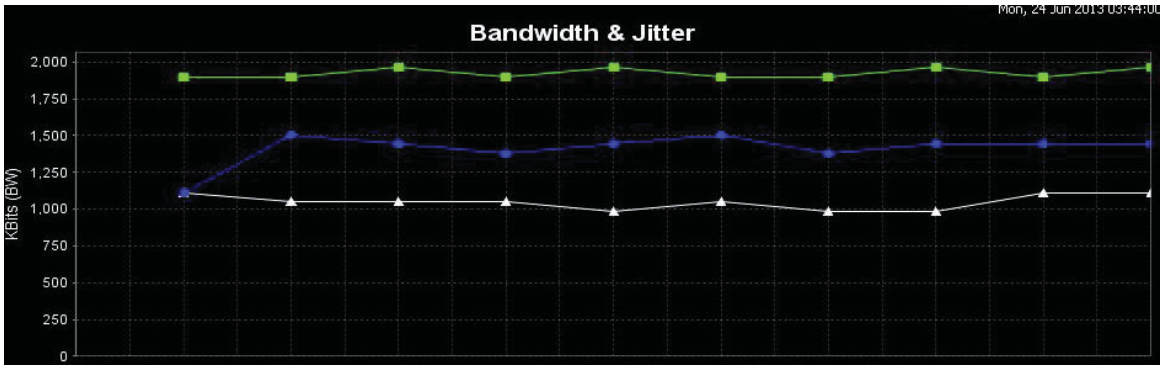


Fig. 7-6 Se suma tráfico de la cámara *web* blanco

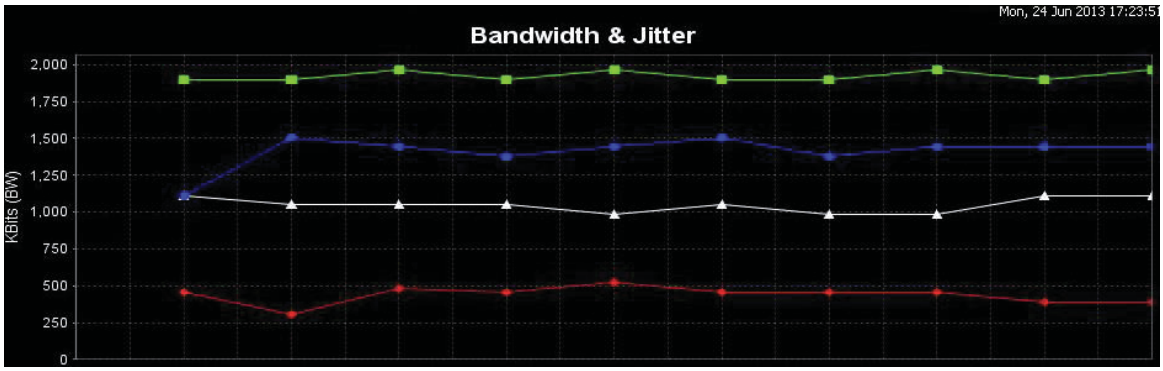


Fig. 7-7 Se suma tráfico de video

En la figura 7-8 y dado que en esta parte de la configuración el sistema aplica *best effort*, los anchos de banda por aplicación o tipo de tráfico son simplemente transmitidos por orden de llegada y queda claro que el ancho de banda es repartido de manera equitativa de acuerdo a esto.

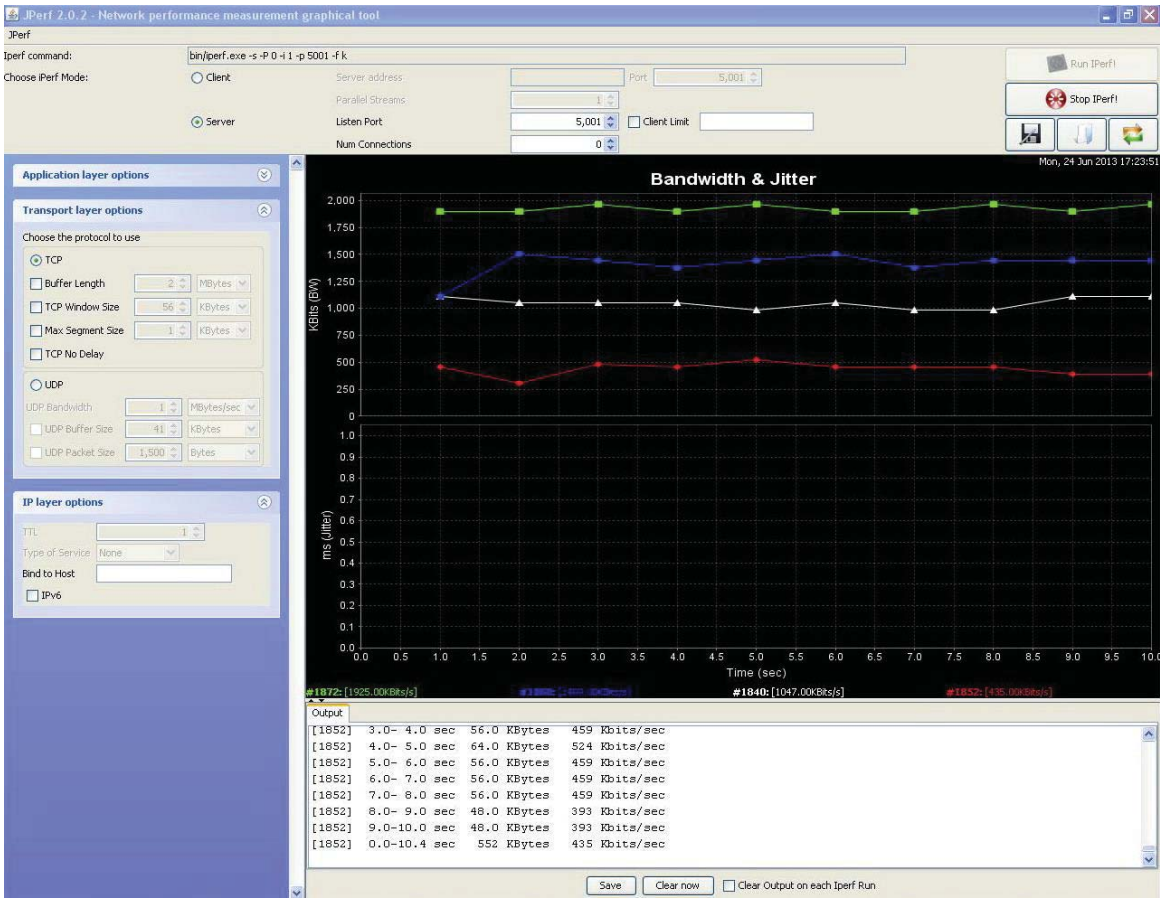


Fig. 7-8 Prueba con *jPerf* y distintos tráficos

La figura 7-9 muestra de qué manera varían los tiempos de respuesta de ping entre el servidor y cliente *jPerf* a medida que se van activando los distintos tipos de tráfico utilizados, que se mencionaron anteriormente.

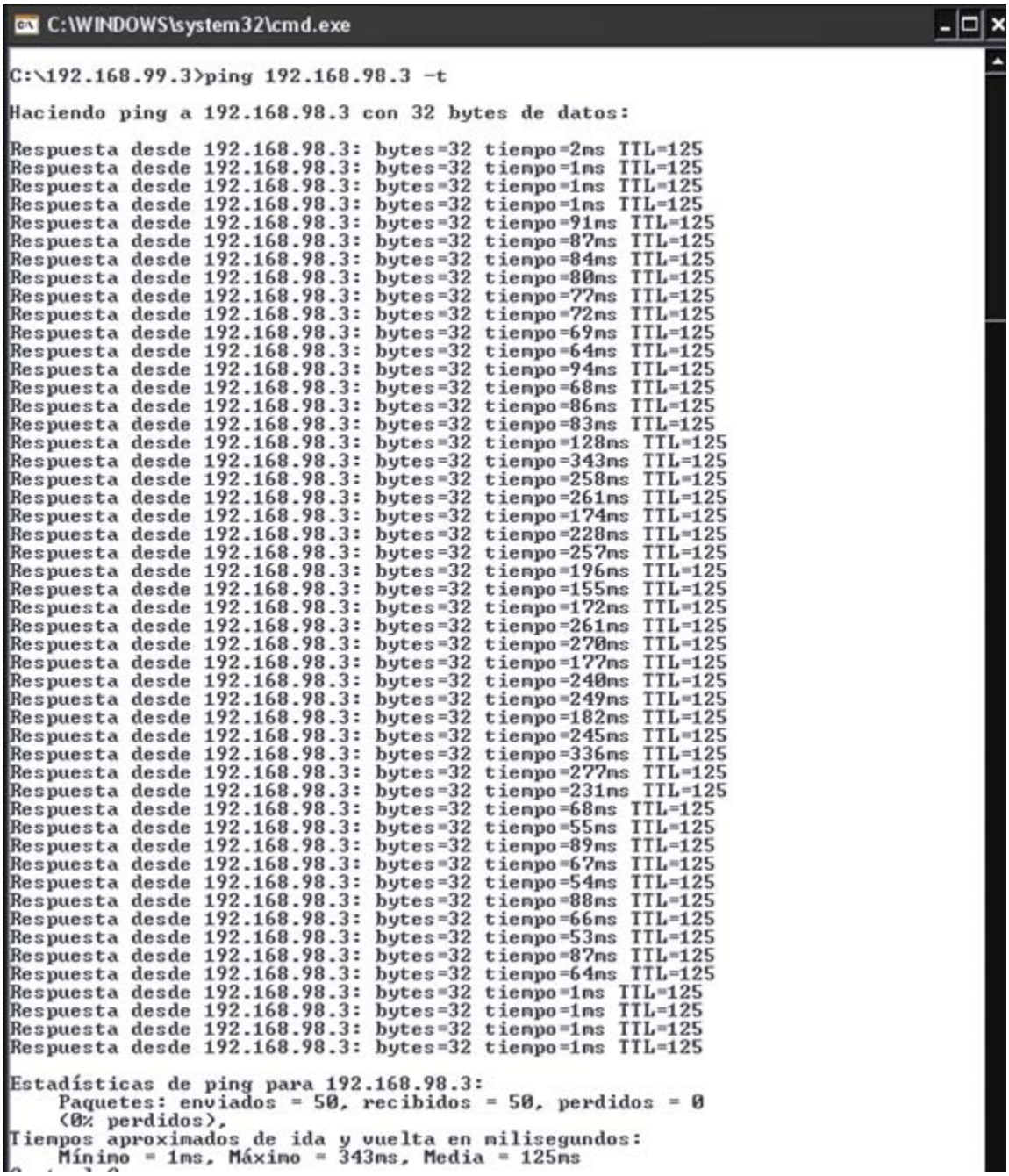


Fig. 7-9 Tiempos de respuesta de ping

7.2 Pruebas entre Sitios de cliente

7.2.1 Conectividad

La primera prueba fue la de conectividad entre la redes de los sitios A y B del cliente. Esta prueba tiene por finalidad mostrar que el tráfico entre los sitios del cliente representados por los equipos frontera CE1 y CE2, efectivamente pasa por la red establecida en el diagrama de la figura 7-1 presentada al principio de este capítulo. Para ello se utilizó el comando *tracert* desde cada uno de los equipos conectados a la red del cliente en ambos sitios A y B.

La figura 7-10 muestra el resultado de la "*ruta*" que recorre el tráfico desde el *router* frontera del cliente en el sitio A, CE1 hacia el sitio B, CE2.

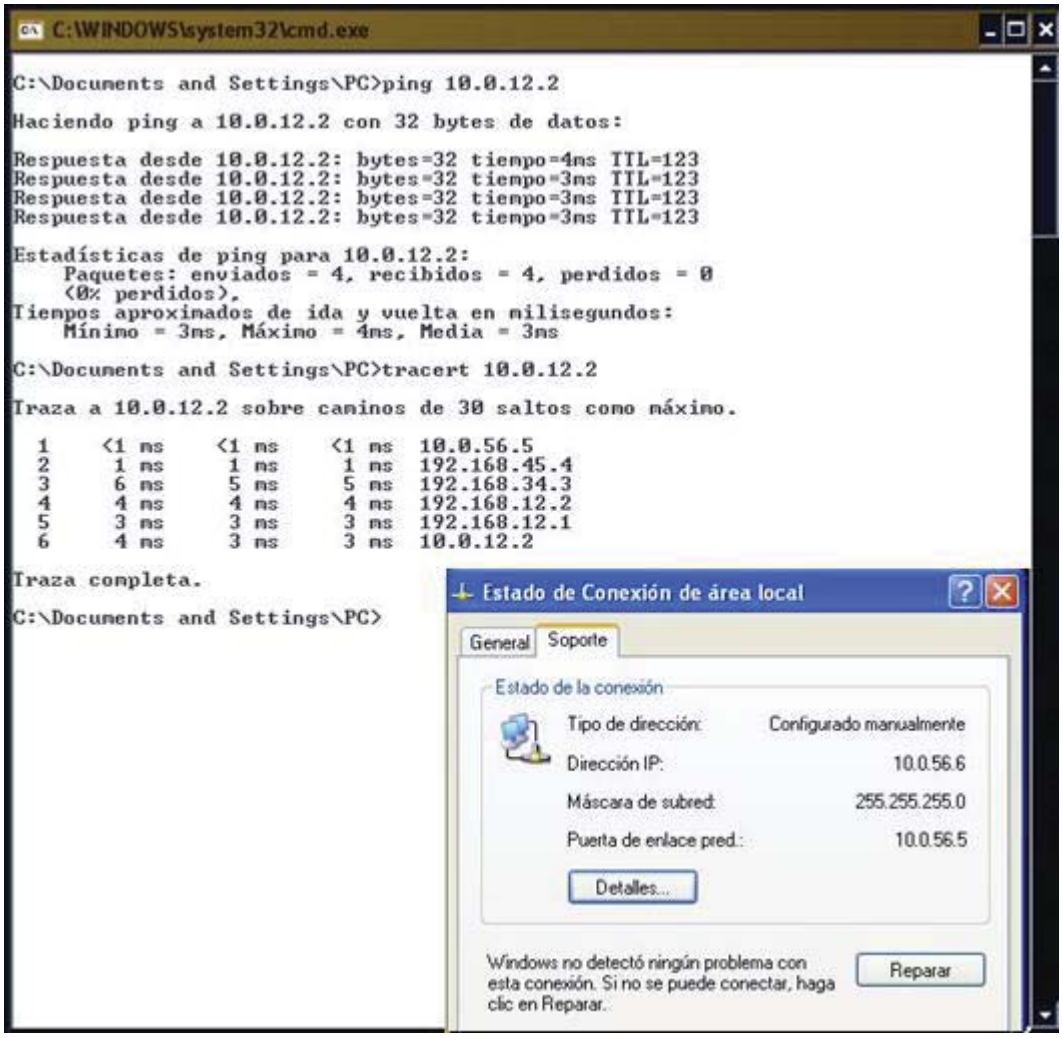


Fig. 7-10 Comando *tracert* entre Sitios Cliente CE1 a CE2

La figura 7-11 muestra el resultado de la "*ruta*" que recorre el tráfico desde el *router* frontera del cliente en el sitio B, CE2 hacia el sitio A, CE1.

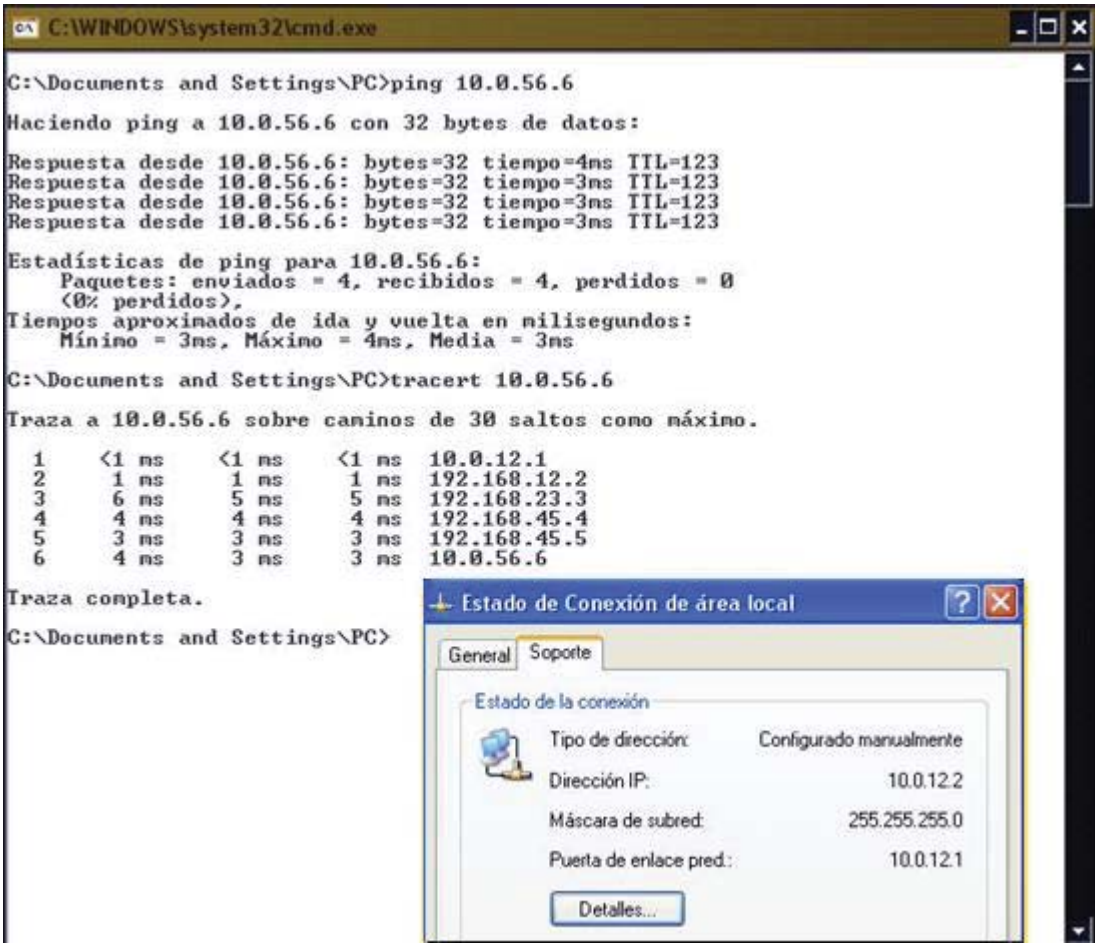


Fig. 7-11 Comando *tracert* entre Sitios Cliente CE2 a CE1

7.2.2 Consideraciones para pruebas de marcas de CoS en tráfico

Como se especificó anteriormente, los computadores utilizados en el laboratorio cuentan con sistema operativo *Windows* XP profesional, el cual no tiene soporte para QoS, es decir el tráfico no puede ser marcado por el sistema operativo. Por este motivo se realizó una búsqueda de algún *software* que pudiese manipular y modificar esta situación en *Windows* XP, sin resultados exitosos. Existía la posibilidad de utilizar máquinas virtuales con alguna distribución de Linux, lo cual se hizo para probar software que generaban tráfico UDP con marca de QoS, sin embargo existía el inconveniente de que las licencias no eran libres y se debían comprar a un alto costo, por otro lado los programas eran de compleja implementación. Por estas razones, para realizar las pruebas de aplicación de Calidad de Servicio, se optó por utilizar una herramienta de Cisco

incorporada en el mismo sistema operativo utilizado en los *Routers* de los sitios A y B como cliente. La herramienta en cuestión se denomina IP SLA.

La función de Cisco IP SLA sirve para poder reunir información detallada de algún tipo de tráfico específico de lado a lado dentro de una red. Básicamente, un equipo que tiene configurado IP SLA corre una prueba previamente configurada hacia un equipo en algún punto remoto de la red, de este modo, al recibir la respuesta a la prueba enviada al equipo remoto, IP SLA reúne información acerca del estado del camino por el cuál pasaron los paquetes.

Como se puede imaginar, esta es una herramienta útil cuando se requiere monitorear un enlace con un propósito específico, un buen ejemplo es el de monitorear enlaces para saber cuándo es conveniente hacer cambios de *router* activos.

Primero para configurar IP SLA se deben seguir los siguientes pasos:

1.- Habilitar IP SLA en el *Switch* de respuesta.

```
Switch(config)#ip sla responder
```

(Nota: Esto sólo se hace para mediciones de *jitter* o para mediciones más precisas.)

2.- Iniciar una operación de IP SLA en el *Switch* que se quiere monitorear.

```
Switch(config)#ip sla operation number
```

3.- Escoger el tipo de prueba de las opciones disponibles de acuerdo al SO.

```
Switch(config-ip-sla)#icmp-echo destination-ip-addr [source-ip-addr]
```

4.- Asignar la ejecución de la prueba.

```
Switch(config)#ip sla schedule operation-number
```

(Nota: con esta opción se puede iniciar la prueba en ese momento, hacerla recurrente o darle un tiempo de inicio y tiempo de vida)

7.2.3 Tráfico UDP con marca de *DiffServ*

Al utilizar IP SLA, se crearon simulaciones de tráfico VoIP desde el sitio A hacia el B del cliente y viceversa, además se seleccionaron distintos tipos de códec en cada uno de

los *routers*, tanto en CE1 como en CE2. De este modo es que se utilizó el códec g711alaw y g729a, ambos comúnmente usados en tráficos reales de voz.

La figura 7-12 corresponde a una captura de pantalla utilizando el programa *Wireshark*. La primera línea en gris muestra la selección de un paquete de datos que proviene del sitio B del cliente representado por el *router* CE2 a través de la IP 192.168.45.5 de la interfaz que comunica con el proveedor y que tiene como destino al *router* CE1 en el sitio A del cliente. El tipo de tráfico utiliza el protocolo UDP ya que está simulando datos de voz.

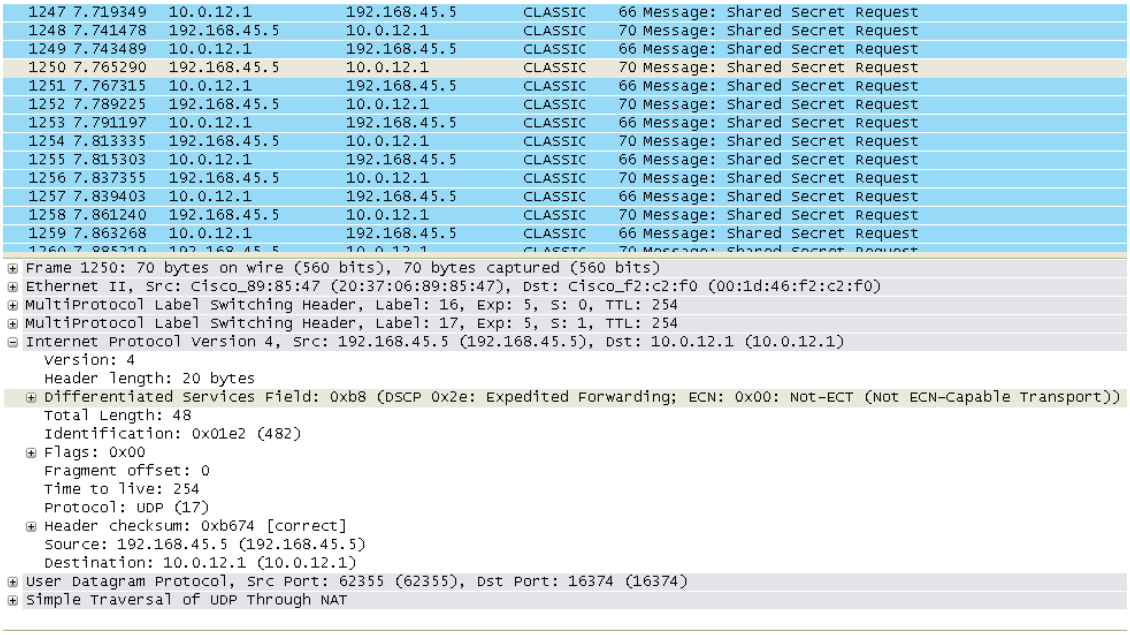


Fig. 7-12 Captura del tráfico IP SLA en *Wireshark*

Se aprecia además en la figura 7-12, el campo *DiffServ* que marca justamente el DSCP con el valor 0x2e correspondiente al valor 46 que a su vez equivale a *Expedited Forwarding* en el trato que debe recibir para QoS (ver Tabla 4-3). Esto confirma la marca de Calidad de Servicio que es requerida para poder realizar las pruebas de medición.

7.3 Pruebas Finales

7.3.1 Primera prueba de IP SLA con la red MPLS sin saturar ni configurar QoS

Para la realización de esta prueba, se dejó habilitada sólo la cámara web para envío de tráfico y de esa forma hacer las mediciones. Para comprender los valores analizados, el detalle del funcionamiento de IP SLA se adjunta en el **apéndice B**.

En la figura 7-13 se aprecia la configuración de IP SLA en ambos *routers* CE1 y CE2. La ejecución del comando, muestra qué tipo de operación se está realizando en la configuración. En este caso, corresponde a una medición de retardo o *jitter* a través de emisión de tráfico Voip en codificación **g711alaw**.

```
CE1#show ip sla con
IP SLAs Infrastructure Engine-II
Entry number: 1
Owner:
Tag:
Type of operation to perform: udp-jitter
Target address/Source address: 10.0.56.5/0.0.0.0
Target port/Source port: 16384/0
Type Of Service parameter: 0xB8
Target port/Source port: 16384/0
Operation timeout (milliseconds): 5000
Codec Type: g711alaw
Codec Number Of Packets: 1000
Codec Packet Size: 160
Codec Interval (milliseconds): 15
Advantage Factor: 0
Type Of Service parameters: 0xB8
Verify data: No
Vrf Name:
Control Packets: enabled
Schedule:
   Operation frequency (seconds): 20  (not considered if randomly scheduled)
   Next Scheduled Start Time: Start Time already passed
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): Forever
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000 (not considered if react RTT is configured)
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 20
Enhanced History:


CE1#
```

Fig. 7-13 Configuración de IP SLA en CE1 códec g711alaw

La figura muestra los detalles de la configuración realizada, entre los cuales se observa la IP de destino (en este caso el *router* CE2), puerto de destino, número de paquetes que se transmiten y su tamaño e intervalo de tiempo entre envío.

En la figura 7-14 pueden apreciarse los mismos detalles pero en la configuración realizada en el *router* CE2, cuyo destino para la medición es el *router* CE1.

```
CE2#show ip sla con
IP SLAs Infrastructure Engine-II
Entry number: 1
Owner:
Tag:
Type of operation to perform: udp-jitter
Target address/Source address: 10.0.12.1/0.0.0.0
Target port/Source port: 16374/0
Type Of Service parameter: 0xB8
Target port/Source port: 16374/0
Operation timeout (milliseconds): 5000
Codec Type: g729a
Codec Number Of Packets: 1000
Codec Packet Size: 20
Codec Interval (milliseconds): 20
Advantage Factor: 0
Type Of Service parameters: 0xB8
Verify data: No
Vrf Name:
Control Packets: enabled
Schedule:
   Operation frequency (seconds): 25  (not considered if randomly scheduled)
   Next Scheduled Start Time: Start Time already passed
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): Forever
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000 (not considered if react RTT is configured)
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 20
Enhanced History:

CE2#
```

Fig. 7-14 Configuración de IP SLA en CE2 códec g729a

Con diferencia de la configuración anterior, aquí se observa que el códec utilizado para voip es el **g729a** y los parámetros se detallan en la figura 7-14.

A continuación, la figura 7-15, corresponde a los valores que entrega el comando *show ip sla statistics* en CE1 gracias a la comunicación establecida con el *router* CE2 en el cual se configuró un *IP SLA RESPONDER* para que entre ambos equipos, se establezca la calidad de conexión de acuerdo a los parámetros fijados. Se observa un valor **MOS** (*Main Opinion Score*) igual a 4.34 (en una escala de 0.00 a 5.00) acompañado de un **ICPIF** igual a 1, esto significa que las pérdidas de paquetes es menor al 1% al igual que el retardo o *jitter*, por lo que de acuerdo a las tablas de ICPIF y MOS (ver apéndice B), la conexión establecida es de alta calidad.

```
CE1#show ip sla st
IPSLAs Latest Operation Statistics

IPSLA operation id: 1
Type of operation: udp-jitter
        Latest RTT: 6 milliseconds
Latest operation start time: *22:32:37.870 UTC Mon Jul 29 2013
Latest operation return code: OK
RTT Values:
        Number of RTT: 1000             RTT Min/Avg/Max: 6/6/8 milliseconds
Latency one-way time:
        Number of Latency one-way Samples: 0
        Source to Destination Latency one way Min/Avg/Max: 0/0/0 milliseconds
        Destination to Source Latency one way Min/Avg/Max: 0/0/0 milliseconds
Jitter Time:
        Number of SD Jitter Samples: 999
        Number of DS Jitter Samples: 999
        Source to Destination Jitter Min/Avg/Max: 0/1/1 milliseconds
        Destination to Source Jitter Min/Avg/Max: 0/1/1 milliseconds
Packet Loss Values:
        Loss Source to Destination: 0           Loss Destination to Source: 0
        Out Of Sequence: 0      Tail Drop: 0
        Packet Late Arrival: 0  Packet Skipped: 0
Voice Score Values:
        Calculated Planning Impairment Factor (ICPIF): 1
MOS score: 4.34
Number of successes: 64
Number of failures: 0
Operation time to live: Forever

CE1#
```

Fig. 7-15 Respuesta IP SLA en CE1 sin QoS en la red MPLS

```
CE2#show ip sla st
IPSLAs Latest Operation Statistics

IPSLA operation id: 1
        Latest RTT: 3 milliseconds
Latest operation start time: *22:20:55.290 UTC Mon Jul 29 2013
Latest operation return code: OK
RTT Values:
        Number of RTT: 1000             RTT Min/Avg/Max: 3/3/5 milliseconds
Latency one-way time:
        Number of Latency one-way Samples: 0
        Source to Destination Latency one way Min/Avg/Max: 0/0/0 milliseconds
        Destination to Source Latency one way Min/Avg/Max: 0/0/0 milliseconds
Jitter Time:
        Number of SD Jitter Samples: 999
        Number of DS Jitter Samples: 999
        Source to Destination Jitter Min/Avg/Max: 0/1/1 milliseconds
        Destination to Source Jitter Min/Avg/Max: 0/1/2 milliseconds
Packet Loss Values:
        Loss Source to Destination: 0           Loss Destination to Source: 0
        Out Of Sequence: 0      Tail Drop: 0
        Packet Late Arrival: 0  Packet Skipped: 0
Voice Score Values:
        Calculated Planning Impairment Factor (ICPIF): 11
MOS score: 4.06
Number of successes: 59
Number of failures: 0
Operation time to live: Forever

CE2#
```

Fig. 7-16 Respuesta IP SLA en CE2 sin QoS en la red MPLS

La figura 7-16 muestra las estadísticas recogidas de la comunicación establecida entre los *routers* CE1 y CE2 después de haber configurado un IP SLA en CE2 y un *IP SLA RESPONDER* en CE1. En el equipo CE2 se configuró el códec g729a, el cual difiere

en valores de ICPIF de acuerdo a las tablas de Cisco, sin embargo el valor de ICPIF es igual a 11, lo que corresponde a pérdidas de datos de un 1% y acompañado de un MOS de 4.06 sigue resultando una conexión de alta calidad.

Se debe tener presente que hasta este punto, en ninguno de los equipos tanto por el lado del ISP como del cliente, se ha configurado QoS.

7.3.2 Prueba con la red MPLS saturada y sin configuración de QoS

En esta prueba se saturó la red MPLS del proveedor, haciendo pasar tráfico entre el PC4 y PC9 realizando un ping con paquetes de 30 *kbytes*; y por otro lado, con la cámara *web*, FTP y *VLC Player* (como emisor vía http) desde el sitio A al B.

Los resultados fueron los siguientes:

- Entre el PC4 y PC9 se produjo una pérdida de paquetes de un 10% con el tráfico TCP realizado a través de ping con paquetes de 30 *kbytes* y TTL 123 milisegundos.
- En cuanto a la respuesta de la configuración de IP SLA, se pueden apreciar los resultados obtenidos en la figura 7-17.

En la figura 7-17 se observa con claridad una disminución considerable en el valor de ICPIF con 35 y un MOS de 2.85, los cuales representan una pérdida de paquetes entre un 5 a 6 por ciento y un retardo muy alto, en definitiva, este resultado indica una calidad baja.

```
CE2#show ip sla st
IPSLAs Latest Operation Statistics

IPSLA operation id: 1
        Latest RTT: 416 milliseconds
Latest operation start time: *16:43:52.998 UTC Mon Aug 5 2013
Latest operation return code: OK
RTT Values:
        Number of RTT: 182              RTT Min/Avg/Max: 123/416/674 milliseconds
Latency one-way time:
        Number of Latency one-way Samples: 0
        Source to Destination Latency one way Min/Avg/Max: 0/0/0 milliseconds
        Destination to Source Latency one way Min/Avg/Max: 0/0/0 milliseconds
Jitter Time:
        Number of SD Jitter Samples: 169
        Number of DS Jitter Samples: 181
        Source to Destination Jitter Min/Avg/Max: 9/14/25 milliseconds
        Destination to Source Jitter Min/Avg/Max: 0/9/160 milliseconds
Packet Loss Values:
        Loss Source to Destination: 818         Loss Destination to Source: 0
        Out Of Sequence: 0      Tail Drop: 0
        Packet Late Arrival: 0  Packet Skipped: 0
Voice Score Values:
        Calculated Planning Impairment Factor (ICPIF): 35
MOS score: 2.85
Number of successes: 45
Number of failures: 0
Operation time to live: Forever
```

Fig. 7-17 Respuesta de IP SLA en CE2 con la red saturada

7.3.3 Prueba con la red MPLS saturada y configuración de QoS

Para esta prueba se realizó configuración básica de QoS con el comando Auto-QoS que ofrece Cisco de manera de simplificar la prueba y asegurar que todas las interfaces relacionadas en la conexión de la topología tuviesen compatibilidad de configuración (ver apéndice A).

```
CE1#show ip sla st
IPSLAs Latest Operation Statistics

IPSLA operation id: 1
Type of operation: udp-jitter
        Latest RTT: 339 milliseconds
Latest operation start time: *22:43:58.182 UTC Mon Jul 29 2013
Latest operation return code: OK
RTT Values:
        Number Of RTT: 1000             RTT Min/Avg/Max: 25/339/403 milliseconds
Latency one-way time:
        Number of Latency one-way Samples: 0
        Source to Destination Latency one way Min/Avg/Max: 0/0/0 milliseconds
        Destination to Source Latency one way Min/Avg/Max: 0/0/0 milliseconds
Jitter Time:
        Number of SD Jitter Samples: 999
        Number of DS Jitter Samples: 999
        Source to Destination Jitter Min/Avg/Max: 0/9/274 milliseconds
        Destination to Source Jitter Min/Avg/Max: 0/3/47 milliseconds
Packet Loss Values:
        Loss Source to Destination: 0           Loss Destination to Source: 0
        Out Of Sequence: 0      Tail Drop: 0
        Packet Late Arrival: 0  Packet Skipped: 0
Voice Score Values:
        Calculated Planning Impairment Factor (ICPIF): 6
MOS score: 4.21
Number of successes: 98
Number of failures: 0
Operation time to live: Forever

CE1#
```

Fig. 7-18 Respuesta de IP SLA en CE1 con la red saturada y QoS

```
IPSLA operation id: 2
        Latest RTT: 72 milliseconds
Latest operation start time: *18:52:03.589 UTC Mon Aug 5 2013
Latest operation return code: OK
RTT Values:
        Number Of RTT: 2                RTT Min/Avg/Max: 70/72/75 milliseconds
Latency one-way time:
        Number of Latency one-way Samples: 0
        Source to Destination Latency one way Min/Avg/Max: 0/0/0 milliseconds
        Destination to Source Latency one way Min/Avg/Max: 0/0/0 milliseconds
Jitter Time:
        Number of SD Jitter Samples: 1
        Number of DS Jitter Samples: 1
        Source to Destination Jitter Min/Avg/Max: 5/5/5 milliseconds
        Destination to Source Jitter Min/Avg/Max: 0/0/0 milliseconds
Packet Loss Values:
        Loss Source to Destination: 0           Loss Destination to Source: 0
        Out Of Sequence: 0      Tail Drop: 0
        Packet Late Arrival: 0  Packet Skipped: 998
Voice Score Values:
        Calculated Planning Impairment Factor (ICPIF): 11
MOS score: 4.06
Number of successes: 79
Number of failures: 23
Operation time to live: Forever

CE2#
```

Fig. 7-19 Respuesta de IP SLA en CE2 con la red saturada y QoS

En las figuras 7-18 y 7-19 se aprecia cómo los valores de ICPIF y MOS vuelven a estar dentro de los rangos establecidos como de alta calidad, con pérdidas hasta de un 1 por ciento y poco retardo.

Al considerar estos últimos escenarios con la red en saturación y aplicando auto-QoS, se obtienen como resultado de los indicadores ICPIF y MOS los siguientes promedios, que se presentan en la Tabla 7-2, con sus respectivos niveles de calidad de acuerdo a lo que establece *Cisco* (ver Apéndice B).

Tabla 7-2 Indicadores de Calidad en último escenario

Indicador	Valor	Calidad de comunicación
ICPIF promedio	8,5	Buena a Muy Buena
MOS promedio	4,14	Buena a Excelente

Finalmente en el caso en que se ha saturado la red con distintos tráficos y dando prioridad a los paquetes de VoIP marcados con DSCP 46 como es el caso del tráfico generado con el comando *IP SLA* se ha logrado determinar y medir los niveles de Calidad de Servicio sobre estos paquetes que en la mayoría de los casos de redes son los que requieren los niveles más altos de priorización.

Bajo un mismo escenario, al aplicar QoS sobre la red MPLS se ha logrado llevar desde un nivel de calidad entre Pobre y Caso Límite (según valores de MOS e ICPIF respectivamente) a niveles de calidad entre Buena y Excelente.

CONCLUSIONES

El trabajo realizado en este informe ha demostrado la compatibilidad y eficiencia de los mecanismos que existen en la actualidad para ofrecer Calidad de Servicio en redes IP. El desarrollo del informe se centró en el estudio de las funcionalidades de cada herramienta utilizada y finalmente en la implementación de éstas como un todo para realizar las mediciones correspondientes que se mostraron en el Capítulo 7.

Del proceso de implementación de la red, se obtuvo una respuesta esperada al aplicar las configuraciones de *Auto-QoS* en cada una de las interfaces en juego. Vale decir, que al mantener la red saturada en un mismo nivel, se logró dar prioridad al tráfico marcado como DSCP *Expedited Forwarding*, pudiendo medir los porcentajes de pérdidas de algunos de los paquetes enviados por la red, como en el caso del Ping desde PC4 a PC9, con pérdidas de un 10% y obteniendo sólo un 1% de pérdidas en el caso del tráfico con prioridad, proveniente de los equipos CE1 y CE2 establecidos como los Sitios A y B del cliente.

Dentro de la parte técnica, se conoció una herramienta de gran utilidad como es IP SLA de Cisco, la cual permitió generar el tráfico requerido sin necesidad de algún *software* externo. Otra ventaja de las estadísticas disponibles en los mismos *routers*, fue deducir que en el caso del códec g711alaw, éste presentó una mayor sensibilidad al tráfico cuando la red se encontraba saturada, ya que al aplicar QoS en los equipos, el códec g729a se mostró con una mejor respuesta en cuanto a su disminución de retardo o *jitter*.

En este informe se han descrito MPLS, DS y QoS con detalles técnicos, tales como su arquitectura, diseño, modos de operación y capacidades. MPLS y DS proveen de una buena solución para QoS en las redes IP, es así como los mayores proveedores de servicios de telecomunicaciones han desarrollado e implementado MPLS en sus *Backbones* y DS ha tenido éxito en el desarrollo de las redes para implementación de los nuevos servicios que hoy se ofrecen en el mercado de las comunicaciones. En otras palabras, aplicar QoS en las redes IP ha sido en gran medida posible gracias a la implementación de MPLS y DS.

En términos de costos como un factor importante, MPLS ha traído consigo muchas ventajas probando así su eficiencia. Hoy en día, ambos, MPLS y DS han evolucionado, pero siguen requiriendo mejoras. Sin embargo, adoptar MPLS presenta una muy buena solución para obtener QoS aunque deben realizarse consideraciones importantes para la implementación de MPLS.

Actualmente es claro ver cómo MPLS ha desplazado las tecnologías de ATM y *Frame Relay* a un plano de las redes de borde y ha dominado las redes IP.

DS es una gran solución para garantizar QoS para redes de datos y MPLS, con su soporte para DS, ofrece la mejor alternativa para las redes de *Backbone*. Así es que adoptar MPLS como *Backbone* es un paso crítico y esencial para obtener y garantizar QoS tanto para proveedores como organizaciones.

REFERENCIAS

[1] A. Balchunas, «Introduction to QoS v1.21,» 2010. [En línea]. Available: http://www.routeralley.com/ra/docs/qos_intro.pdf.

[2] S. B. F. B. D. B. K. Nichols, «Request for Comments: 2474,» ietf.org, December 1998. [En línea]. Available: http://www.ietf.org/rfc/rfc2474.txt.

[3] S. William, Computer Networking with Protocols, Prentice Hall, 2004.

[4] D. R. Sánchez, Modelado y Simulacion de un Conmutador MPLS utilizando VHDL, Universidad de las Américas Puebla, Mayo 2004.

[5] D. Miras, «A Survey of Network QoS Needs of Advanced Internet Applications,» November 2002. [En línea]. Available: http://qos.internet2.edu/wg/apps/fellowship/Docs/Internet2AppsQoSNeeds.pdf.

[6] B. A. D. A. D. W. P. A. Stefano Avallone, «D-ITG V.2.6.1d Manual,» Universidad de Naples Federico II, Mayo 2008. [En línea]. Available: http://traffic.comics.unina.it/software/ITG/manual/D-ITG2.6.1d-manual.pdf.

[7] M. C. Karen, «Evaluación de algoritmos de control de retardo en voz sobre internet,» Universidad Autónoma Metropolitana Iztapalpa, Enero 2009. [En línea]. Available: http://pcyti.izt.uam.mx/~kmiranda/docs/MastersThesis.pdf.

[8] V. O. Cristina, «Estudio de VoIP aplicado en redes privadas,» Universidad Tecnica del Norte, Ecuador, Junio 2011. [En línea].

[9] G. V. A. Álvarez M. Sebastián, «Estudio y configuracion de Calidad de Servicio para Protocolos IPV4 e IPV6 en una red de fibra óptica WDM,» Facultad de Ingeniería - Universidad Tarapacá, Marzo 2005. [En línea]. Available: www.scielo.cl/pdf/rfacing/v13n3/art15.pdf.

[10] R. S. Delfino Adrián, «Evaluación de Performance en Redes de Telecomunicaciones,» Instituto de Ingeniería Eléctrica, Universidad de la Republica Uruguay, 2003. [En línea]. Available: http://iie.fing.edu.uy/ense/asign/perfredes/trabajos/trabajos_2003/diffserv/Trabajo%20Final.pdf.

[11] L. Díaz Cervantes, «Evaluación de la herramienta GNS3 con conectividad a enrutadores reales,» Escuela Técnica Superior de Ingeniería de Telecomunicaciones de Barcelona, Universitat Politecnica de Catalunya, 2010. [En línea]. Available: http://upcommons.upc.edu/pfc/bitstream/2099.1/9989/1/PFC_Lisset_D%C3%ADaz.pdf.

APÉNDICE A

CONFIGURACIONES DE EQUIPOS CISCO

CONFIGURACIONES DE EQUIPOS CISCO

A.1 Comandos para configuración de equipos *Customer Edge* CE1 y CE2:

Tabla A-1 Comandos de configuración para equipo CE1 y CE2

Comando	Propósito
class-map	Crea un mapa de clase que se utilizará para que los paquetes coincidan con una clase especificada.
match (*class-map configuration*)	Define los criterios de coincidencia para clasificar el tráfico.
match-any	(Opcional) Determina cómo se evalúan los paquetes cuando existen múltiples criterios de coincidencia. Encuentra declaraciones en relación con este mapa de clase basandose en la función lógica OR. Se acepta una declaración u otra. Si no se especifica el match-any o match-all (palabra clave), la palabra clave por defecto es match-all. [1]
auto qos	Instala los mapas de clase de QoS y mapas de políticas creados por las características de la función AutoQoS.
trust	(Opcional) Indica que las marcas el punto de código de servicios diferenciados (DSCP) de un paquete son de confianza (basado en) para la clasificación del tráfico de voz. Si no se especifica la palabra clave *trust* opcional, el tráfico de voz se clasifica mediante el reconocimiento de la aplicación basada en la red (NBAR), y los paquetes son marcados con el valor DSCP apropiado. [1]
policy-map	Crea o modifica un mapa de política que se puede conectar a una o más interfaces para especificar una política de servicio.
priority	Especifica la prioridad de una clase de tráfico que pertenece a un mapa de política.
percent	Especifica el porcentaje de ancho de banda garantizado en base a un porcentaje absoluto de ancho de banda disponible para ser reservado para la clase de prioridad. El porcentaje puede ser un número de 1 a 100.
bandwidth	Establece los valores de ancho de banda heredados y recibidos para una interfaz.
fair-queue (class-default)	Especifica el número de colas dinámicas a ser reservados para el uso de la Clase-*default* como parte de la política de clase predeterminada.

Comando	Propósito
service-policy	Se fija un mapa de política a una interfaz de entrada o circuito virtual (VC) o a una interfaz de salida o VC para ser utilizado como la política de servicio para dicha interfaz o VC.
ip sla responder	Configura el router como un IP SLA de respuesta. (detalle en apéndice B)
ip sla	Crear una operación de IP SLA, y entrar en el modo de configuración IP SLA.

A continuación se muestra la configuración completa del equipo CE1.

Tabla A-2 Configuración en *router* CE1

Comandos ingresados en *router* CE1

hostname CE1
 	
class-map match-any AutoQoS-VoIP-RTP-Trust
 match ip dscp ef
class-map match-any AutoQoS-VoIP-Control-Trust
 match ip dscp cs3
 match ip dscp af31
 	
policy-map AutoQoS-Policy-Trust
 class AutoQoS-VoIP-RTP-Trust
  priority percent 70
 class AutoQoS-VoIP-Control-Trust
  bandwidth percent 5
 class class-default
  fair-queue
 	
interface FastEthernet0/0
 ip address 10.0.12.1 255.255.255.0
 duplex auto
 speed auto
 auto qos voip trust
 service-policy output AutoQoS-Policy-Trust
 	
router eigrp 100
 network 1.0.0.0
 network 10.0.0.0
 network 192.168.12.0
 no auto-summary
 	
ip sla responder

Comandos ingresados en *router* CE1
ip sla 1 udp-jitter 10.0.56.5 16485 codec g711alaw codec-interval 8 codec-size 5000 tos 184 frequency 15 ip sla schedule 1 life forever start-time now ip sla 2 udp-jitter 10.0.56.5 16585 codec g729a codec-interval 8 codec-size 5000 tos 184 frequency 15 ip sla schedule 2 life forever start-time now snmp-server community COSTUMERA RO

Tabla A-3 Configuración en *router* CE2

Comandos ingresados en *router* CE2

hostname CE2
 	
class-map match-any AutoQoS-VoIP-RTP-Trust
 match ip dscp ef
class-map match-any AutoQoS-VoIP-Control-Trust
 match ip dscp cs3
 match ip dscp af31
 	
policy-map AutoQoS-Policy-Trust
 class AutoQoS-VoIP-RTP-Trust
  priority percent 70
 class AutoQoS-VoIP-Control-Trust
  bandwidth percent 5
 class class-default
  fair-queue
 	
interface FastEthernet0/0
 ip address 10.0.56.5 255.255.255.0
 duplex auto
 speed auto
 auto qos voip trust
 service-policy output AutoQoS-Policy-Trust
 	
interface Serial0/1/0
 ip address 192.168.45.5 255.255.255.0
 auto qos voip trust
 clock rate 2000000

Comandos ingresados en *router* CE2

```
 service-policy output AutoQoS-Policy-Trust

router eigrp 100
 network 5.0.0.0
 network 10.0.0.0
 network 192.168.45.0
 no auto-summary

    ip sla responder

    ip sla 1
     udp-jitter 10.0.12.1 16484 codec g711alaw codec-interval 10 codec-size 5000
     tos 184
     frequency 15
    ip sla schedule 1 life forever start-time now

    ip sla 2
     udp-jitter 10.0.12.1 16474 codec g729a codec-interval 8 codec-size 5000
     tos 184
     frequency 15
    ip sla schedule 2 life forever start-time now

    snmp-server community COSTUMERB RO
```

A.2 Comandos para configuración de equipos del ISP: PE1, SP y PE2.

Tabla A-4 Comandos de configuración para equipo PE1, SP y PE2

Comando	Propósito
Ip vrf *vrf name*	Nombre de la VRF y entra en el modo de configuración VRF.
rd *route-distinguisher*	Crea una tabla VRF especificando un distintintivo de ruta. Ingrese un número de AS(*Autonomous System*) y un número arbitrario (xxx:y) o una dirección IP y un número arbitrario (ABCD:y).
route-target *{export | import | both} route-target-ext-community*	Crea una lista de importación, de exportación o de importación y exportación; rutas objetivo de comunidades para el VRF especificado. Ingrese un número de AS del sistema y un número arbitrario (xxx:y) o una dirección IP y un número arbitrario (ABCD: y). El valor de route-target-ext-community debe ser el mismo que el valor de route-distinguisher ingresado anteriormente.
class-map *class-map-name*	Crea un mapa de clase que se utilizará para que los paquetes coincidan con una clase especificada.

Comando	Propósito
match (*class-map configuration*)	Define los criterios de coincidencia para clasificar el tráfico.
auto qos	Instala los mapas de clase de QoS y mapas de políticas creados por las características de la función AutoQoS.
trust	(Opcional) Indica que las marcas el punto de código de servicios diferenciados (DSCP) de un paquete son de confianza (basado en) para la clasificación del tráfico de voz. Si no se especifica la palabra clave trust opcional, el tráfico de voz se clasifica mediante el reconocimiento de la aplicación basada en la red (NBAR), y los paquetes son marcados con el valor DSCP apropiado.
policy-map	Crea o modifica un mapa de política que se puede conectar a una o más interfaces para especificar una política de servicio.
priority	Especifica la prioridad de una clase de tráfico que pertenece a un mapa de política.
percent	Especifica el porcentaje de ancho de banda garantizado en base a un porcentaje absoluto de ancho de banda disponible para ser reservado para la clase de prioridad. El porcentaje puede ser un número de 1 a 100.
bandwidth	Establece los valores de ancho de banda heredados y recibidos para una interfaz.
police	Crea un filtro de control por interfaz
fair-queue (class-default)	Especifica el número de colas dinámicas a ser reservados para el uso de la Clase-*default* como parte de la política de clase predeterminada.
mpls ip	Configura MPLS reenvío *hop-by-hop* (salto a salto) en la interfaz.
redistribute	Redistribuye rutas desde otros protocolos en BGP *{direct | {eigrp | isis | ospf | ospfv3 | rip} instance-tag | static} route-map map-name*
router bgp	Entra en el modo BGP y asigna el número de sistema autónomo al orador BGP local.
ip access-list	Configurar una lista de acceso
mpls ldp router-id	Especifica la dirección IP de una interfaz como el ID del *router* LDP

Tabla A-5 Configuración en *router* PE1

Comandos ingresados en *router* PE1

```
hostname PE1

ip vrf CUSTOMER
 rd 100:1
 route-target export 1:100
 route-target import 1:100

class-map match-all TCP_PC9
 match access-group 101
class-map match-any AutoQoS-VoIP-RTP-Trust
 match ip dscp ef
class-map match-any AutoQoS-VoIP-Control-Trust
 match ip dscp cs3
 match ip dscp af31

policy-map DROP
 class TCP_PC9
   priority percent 30 10000
   police 100000 10000 1000 conform-action transmit  exceed-action drop  violate-action drop
policy-map AutoQoS-Policy-Trust
 class AutoQoS-VoIP-RTP-Trust
   priority percent 70
 class AutoQoS-VoIP-Control-Trust
   bandwidth percent 5
class TCP_PC9
   police 100000 10000 1000 conform-action transmit  exceed-action drop  violate-action drop
 class class-default
   fair-queue

interface FastEthernet0/1
 ip address 192.168.99.2 255.255.255.0
 duplex auto
 speed auto

interface Serial0/1/0
 ip vrf forwarding CUSTOMER
 ip address 192.168.12.2 255.255.255.0
 auto qos voip trust
 service-policy output AutoQoS-Policy-Trust

interface Serial0/1/1
 ip address 192.168.23.2 255.255.255.0
 mpls ip
 auto qos voip trust
 service-policy output AutoQoS-Policy-Trust

    router eigrp 1
     auto-summary
```

Comandos ingresados en *router* PE1

```
address-family ipv4 vrf CUSTOMER
  redistribute bgp 1 metric 2000000 10 230 230 1500
  network 10.0.0.0
  network 192.168.12.0
  no auto-summary
  autonomous-system 100
 exit-address-family

router ospf 1
 log-adjacency-changes
 network 2.2.2.0 0.0.0.255 area 0
 network 192.168.23.0 0.0.0.255 area 0
 network 192.168.28.0 0.0.0.255 area 0
 network 192.168.34.0 0.0.0.255 area 0
 network 192.168.98.0 0.0.0.255 area 0
 network 192.168.99.0 0.0.0.255 area 0

router bgp 1
 no synchronization
 bgp log-neighbor-changes
 neighbor 4.4.4.4 remote-as 1
 neighbor 4.4.4.4 update-source Loopback0
 no auto-summary

address-family vpnv4
  neighbor 4.4.4.4 activate
  neighbor 4.4.4.4 send-community both
 exit-address-family

address-family ipv4 vrf CUSTOMER
  redistribute eigrp 100
  no synchronization
 exit-address-family

access-list 101 permit tcp 192.168.99.0 0.0.0.255 any

snmp-server community PE1 RO

mpls ldp router-id Loopback0
```

Tabla A-6 Configuración en *router* SP

Comandos ingresados en *router* SP

hostname SP
 	
class-map match-any AutoQoS-VoIP-RTP-Trust
 match ip dscp ef
class-map match-any AutoQoS-VoIP-Control-Trust
 match ip dscp cs3
 match ip dscp af31
 	
policy-map AutoQoS-Policy-Trust
  class AutoQoS-VoIP-RTP-Trust
  priority percent 70
  class AutoQoS-VoIP-Control-Trust
  bandwidth percent 5
  class class-default
  fair-queue
 	
interface FastEthernet0/0
 ip address 192.168.34.3 255.255.255.0
 duplex auto
 speed auto
 mpls ip
 auto qos voip trust
 service-policy output AutoQoS-Policy-Trust
 	
interface Serial0/0/0
 ip address 192.168.23.3 255.255.255.0
 mpls ip
 auto qos voip trust
 clock rate 2000000
 service-policy output AutoQoS-Policy-Trust
 	
router ospf 1
 log-adjacency-changes
 network 3.3.3.0 0.0.0.255 area 0
 network 192.168.23.0 0.0.0.255 area 0
 network 192.168.34.0 0.0.0.255 area 0
 network 192.168.98.0 0.0.0.255 area 0
 network 192.168.99.0 0.0.0.255 area 0
 	
mpls ldp router-id Loopback0

Tabla A-7 Configuración en *router* PE2

Comandos ingresados en *router* PE2

```
hostname PE2

ip vrf CUSTOMER
 rd 100:1
 route-target export 1:100
 route-target import 1:100

class-map match-all PC4
 match access-group 101
class-map match-any AutoQoS-VoIP-RTP-Trust
 match ip dscp ef

class-map match-any AutoQoS-VoIP-Control-Trust
 match ip dscp cs3
 match ip dscp af31

policy-map AutoQoS-Policy-Trust
 class AutoQoS-VoIP-RTP-Trust
   priority percent 70
 class AutoQoS-VoIP-Control-Trust
   bandwidth percent 5
 class PC4
   police 100000 10000 1000 conform-action transmit  exceed-action drop  violate-action drop
 class class-default
   fair-queue

interface FastEthernet0/0
 ip address 192.168.98.4 255.255.255.0
 duplex auto
 speed auto

interface FastEthernet0/1
 ip address 192.168.34.4 255.255.255.0
 duplex auto
 speed auto
 mpls ip
 auto qos voip trust
 service-policy output AutoQoS-Policy-Trust

interface Serial0/1/0
 ip vrf forwarding CUSTOMER
 ip address 192.168.45.4 255.255.255.0
 auto qos voip trust
 service-policy output AutoQoS-Policy-Trust

interface Serial0/1/1
 no ip address
 shutdown
```

Comandos ingresados en *router* PE2

```
 mpls ip
 clock rate 2000000

address-family ipv4 vrf CUSTOMER
  redistribute bgp 1 metric 2000000 10 230 230 1500
  network 10.0.0.0
  network 192.168.45.0
  no auto-summary
  autonomous-system 100
 exit-address-family

router ospf 1
 log-adjacency-changes
 network 4.4.4.0 0.0.0.255 area 0
 network 192.168.23.0 0.0.0.255 area 0
 network 192.168.34.0 0.0.0.255 area 0
 network 192.168.47.0 0.0.0.255 area 0
 network 192.168.98.0 0.0.0.255 area 0
 network 192.168.99.0 0.0.0.255 area 0

router bgp 1
 no synchronization
 bgp log-neighbor-changes
 neighbor 2.2.2.2 remote-as 1
 neighbor 2.2.2.2 update-source Loopback0
 no auto-summary

address-family vpnv4
  neighbor 2.2.2.2 activate
  neighbor 2.2.2.2 send-community both
 exit-address-family

address-family ipv4 vrf CUSTOMER
  redistribute eigrp 100
  no synchronization
 exit-address-family

access-list 101 permit tcp 192.168.98.0 0.0.0.255 any
access-list 2701 permit any 19 any any

snmp-server community PE2 RO

mpls ldp router-id Loopback0
```

Tabla A-8 Configuración en *Mirroring Switch* A

Comandos ingresados en *mirroring switch* A
monitor session 1 source interface Fa0/6 monitor session 1 destination interface Fa0/4

APÉNDICE B

IP SLA

IP SLA

B.1 Análisis de Niveles de Servicio usando la operación VoIP UDP *Jitter*

Este apéndice describe cómo utilizar los acuerdos de nivel de servicio IP de Cisco IOS (SLAs) operación UDP *jitter* para monitorear proactivamente Voz sobre IP (VoIP), los niveles de calidad en la red, lo que le permite garantizar niveles de calidad de VoIP a los usuarios. La operación *jitter* IP SLA VoIP UDP simula con precisión el tráfico de VoIP utilizando *codecs* comunes, y calcula las puntuaciones de calidad de voz consistentes (MOS y ICPIF) entre los dispositivos IOS de Cisco en la red.

Cisco IOS IP SLA es un conjunto de funciones incorporado en el software Cisco IOS que permite analizar los niveles de servicio para las aplicaciones IP y servicios IP, para aumentar la productividad, reducir los costos operacionales, y para reducir los casos de congestión o cortes de red. IP SLA utiliza el monitoreo de tráfico activo para medir el rendimiento de la red. La exactitud de los datos de medición es permitida gracias a la habilitación de IP SLA Responder, disponible en los *routers* de Cisco, en el dispositivo de destino. [1]

Nota: El término "Voz" en éste apéndice debe tomarse como significado de cualquier aplicación de telefonía en Internet. Y "Voz sobre IP" puede incluir la transmisión multimedia (ambos video y voz) sobre redes IP.

B.1.1 Pre-requisitos para IP SLA y operaciones VoIP *jitter*

Para utilizar esta característica, los dispositivos de red en ambos extremos de la conexión deben ser compatibles con Cisco IOS IP SLA. Cisco IOS IP SLA es una característica integrada en el IOS de software Cisco.

B.1.2 Restricciones para IP SLA y operaciones VoIP *jitter*

Esta función utiliza el tráfico UDP para generar voz sobre IP y generar puntajes. No proporciona soporte para el protocolo de transporte en tiempo real (RTP).

B.1.3 Información sobre la operación de IP SLA VoIP UDP *jitter*

Para utilizar la herramienta IP SLA VoIP UDP, deben comprenderse los siguientes conceptos:

- *The Calculated Planning Impairment Factor (ICPIF)*
- *Mean Opinion Scores (MOS)*
- *Voice Performance Monitoring Using IP SLAs*
- *Codec Simulation Within IP SLAs*
- *The IP SLAs ICPIF Value*

- *The IP SLAs MOS Value*

B.1.4 ICPIF - *The Calculated Planning Impairment Factor*

El Factor de deterioro calculado de planificación, ICPIF se originó en la versión de la ITU-T G.113 de 1996, "degradaciones de la transmisión", como parte de la fórmula Icpif = Itot - A. El ICPIF trata de cuantificar, a efectos de comparación y planificación, los impedimentos clave para la calidad de voz que se encuentran en la red. [1]

El ICPIF es la suma de los factores de degradación medidos (impedimentos totales, o itot) menos un factor definido por el usuario, Factor de acceso *Advantage* (A) que está destinado a representar las expectativas del usuario, en función de cómo se hizo la llamada (por ejemplo, una llamada de móvil en comparación con una llamada de línea terrestre). En su forma expandida, la fórmula completa se expresa como:

Icpif = Io + Iq + Idte + Idd + Ie – A

donde,

- *Io, representa impedimentos causados por no óptimo índice de sonoridad,*
- *Iq, representa impedimentos causados por la cuantificación de distorsión de PCM,*
- *IDTE, representa impedimentos causados por eco del hablante,*
- *Idd, representa deterioros causados por el tiempo de transmisión de una vía de un solo sentido (delay),*
- *Ie, representa impedimentos causados por los efectos de equipos, tales como el tipo de códec utilizado para la llamada y la pérdida de paquetes, y*
- *A, representa un factor de ventaja acceso (también llamado el factor de Expectativa de usuario) que compensa el hecho de que los usuarios pueden aceptar cierta degradación de la calidad a cambio de la facilidad de acceso.*

Los valores ICPIF se expresan en un rango típico de 5 (muy bajo deterioro) a 55 (muy alto deterioro). Los valores de ICPIF numéricamente menores de 20 son generalmente considerados "adecuados". Si bien la intención de ser una medida objetiva de la calidad de voz, el valor ICPIF también se utiliza para predecir el efecto subjetivo de combinaciones de impedimentos. La tabla B-1, tomada del ITU-T G.113 (02/96), muestra cómo se espera que los valores de muestra ICPIF correspondan a un juicio de calidad subjetivo.

Tabla B-1 Niveles de Calidad en función de ICPIF [1]

Límite superior ICPIF	Calidad de la comunicación
5	Muy Buena
10	Buena
20	Adecuada
30	Caso límite
45	Caso límite excepcional
55	Clientes que pueden reaccionar con severidad (reclamos, cambio de operador)

Versiones más recientes de la recomendación ITU-T G.113 (2001), ya no incluye el modelo ICPIF. En cambio se refiere a la recomendación G.107 "Método Factor de Deterioro", utilizado por el E-*Model* del ITU-T G.107.

La versión completa del E-*Model*, también llamado ITU-T "*Transmission Rating Model*", expresa como R=Ro-Is-Id-Ie+A, que ofrece la posibilidad de mediciones más precisas de la calidad de la llamada mediante el perfeccionamiento de las definiciones de los factores de degradación (ver la versión 2003 del G.107 para más detalles). A pesar de que ICPIF comparte términos de deficiencias con el E-*model*, los dos modelos no deben ser confundidos.

La herramienta de Cisco IP SLA VoIP UDP toma ventajas de correspondencias observadas entre el ICPIF, el factor de índices de transmisión R y los valores de MOS, pero todavía no admite el modelo E.

B.1.5 MOS – *Mean Opinion Score*

La calidad de la voz transmitida es una respuesta subjetiva del oyente. Cada códec utilizado para la transmisión de voz sobre IP proporciona un cierto nivel de calidad. Un punto de referencia común que se utiliza para determinar la calidad del sonido producido por los *codecs* específicos es MOS. Con MOS, una amplia gama de oyentes han juzgado a la calidad de las muestras de voz electrónica utilizando *codecs* particulares, en una escala de 1 (mala calidad) a 5 (excelente calidad). Las puntuaciones de opinión se promedian para proporcionar la media para cada muestra. La Tabla B-2 muestra las calificaciones MOS y la descripción correspondiente de la calidad para cada valor.

Tabla B-2 Valores de MOS [1]

Puntuación	Calidad	Descripción del deterioro de la calidad
5	Excelente	Imperceptible
4	Buena	Apenas perceptible, pero no molesta
3	Justa	Perceptible y un poco molesto
2	Pobre	Molesto, pero no objetable
1	Mala	Muy molesto y desagradable

A medida que se conocen las puntuaciones MOS para *codecs* y otros deterioros de transmisión, un estimado de MOS puede ser calculado y desplegados basándose en las de mediciones de deficiencias. Este valor estimado es designado como MOS-CQE (Mean Opinion Score; calidad conversacional, Estimada) por la ITU con el fin de distinguirlo de los valores MOS objetivos o subjetivos (ver ITU-T P.800.1 para más detalles).

B.1.6 Monitoreo del rendimiento de voz usando IP SLA

Uno de los indicadores clave para medir la calidad de voz y vídeo a través de una red IP es *jitter*. *Jitter* es el nombre que se utiliza para indicar la variación de retardo entre paquetes que llegan. *Jitter* afecta a la calidad de voz, causando brechas irregulares en el patrón de voz de la persona que habla. Otros parámetros de rendimiento clave para la transmisión de voz y video a través de redes IP incluyen latencia (retardo) y la pérdida de paquetes. IP SLA es una función de vigilancia activa incrustada en el IOS del software Cisco que proporciona un medio para la simulación y la medición de estos parámetros con el fin de garantizar que la red está cumpliendo o excediendo los acuerdos de nivel de servicio con los usuarios. [1]

IP SLA proporciona una operación de UDP *jitter*, que consiste en probar paquetes UDP enviados a través de la red desde un dispositivo de origen a un destino específico (llamado el objetivo operativo). Este tráfico sintético se utiliza para registrar la cantidad de jitter para la conexión, así como el tiempo de ida y vuelta, la pérdida de paquetes por la dirección, y el tiempo de retardo en un sentido (latencia en un solo sentido). (El término "tráfico sintético" indica que el tráfico de la red es simulado, es decir, el tráfico se genera por IP SLAs.) Datos, en forma de estadísticas recogidas, se pueden mostrar para múltiples pruebas durante un periodo de tiempo definido por el usuario, que le permite ver, por ejemplo, cómo se comporta la red en diferentes momentos del día, o en el transcurso de una semana. La prueba de jitter tiene la ventaja de utilizar el IP SLA Responder para proporcionar un mínimo de latencia en el extremo receptor. [1]

La operación *jitter* del IP SLA VoIP UDP modifica la operación UDP *jitter* estándar mediante la adición de la capacidad para devolver las puntuaciones de MOS e ICPIF en los datos recogidos por la herramienta, además de las métricas ya recogidos por la operación UDP *jitter*. Esta aplicación VoIP específica proporciona información aún más útil para determinar el rendimiento de una red VoIP, mejorando así la capacidad para realizar la evaluación de la red, solución de problemas y el monitoreo de su integridad.

B.1.7 Simulación de Códecs con IP SLA

La operación de *jitter* IP SLA VoIP UDP calcula estadísticas mediante el envío de n paquetes UDP, cada uno de tamaño s, enviado cada t milisegundos, a partir de un *router* de origen dado a *router* de destino dado, a una frecuencia f. El *router* de destino debe ejecutar el IP SLA Responder (de respuesta) con el fin de procesar las operaciones de prueba.

Para generar valores de MOS y puntajes ICPIF, se especifica el tipo de códec a utilizar en la conexión al configurar la operación *jitter* VoIP UDP. Basado en el tipo de códec que se configura para la operación, el número de paquetes (n), el tamaño de cada carga útil (s), el intervalo de tiempo entre paquetes (T), y la frecuencia de funcionamiento (f) se auto-configuran con los valores predeterminados. Sin embargo, se le da la opción, si es necesario, para configurar manualmente estos parámetros en la sintaxis del comando tipo *jitter dest-ipaddr*.

La Tabla B-3 muestra los parámetros predeterminados que están configurados para la operación con el *codec*.

Tabla B-3 Parámetros por defecto según Códec [1]

Codec	Tamaño soicitado por defecto (Packet Payload) (s)	Intervalo entre paquetes (t)	Nro. de paquetes por defecto (n)	Frecuencia de operación (f)
G.711 mu-Law (g711ulaw)	160 + 12 RTP bytes	20ms	1000	Cada 1 minuto
G.711 A-Law (g711alaw)	160 + 12 RTP bytes	20ms	1000	Cada 1 minuto
G.729A (g729a)	20 + 12 RTP bytes	20ms	1000	Cada 1 minuto

Por ejemplo, si configura el funcionamiento jitter VoIP UDP para usar las características del códec g711ulaw, de forma predeterminada una operación de la sonda será enviada una vez por minuto (f). Cada operación sonda consistiría de 1000 paquetes (n), con cada paquete que contiene 180 bytes de datos sintéticos (s), enviado cada 20 milisegundos (t).

B.1.8 El valor ICPIF

Valor ICPIF calculado con el IOS del software Cisco se basa principalmente en los dos más importantes factores que pueden afectar la calidad de la voz: retraso de paquetes y la pérdida de paquetes. Debido a retardo de paquetes y pérdida de paquetes se pueden medir por IP SLA, la fórmula completa ICPIF, Icpif = Io + Iq + Idte + Idd + Ie - A, se simplifica suponiendo que los valores de Io, Iq, y IDTE son iguales a cero, lo que resulta en la siguiente fórmula:

Factor Deterioro Total (Icpif) = Factor de discapacidad de retardo (IDD) + Factor de degradación del Equipo (Ie) - Expectativa o *Advantage Factor* (A)

Esto significa que el valor ICPIF se calcula mediante la adición de un factor de deterioro de retardo, que se basa en una medición del retraso de los paquetes, y un factor de degradación del equipo, que se basa en una medición de la pérdida de paquetes. A

partir de esta suma de los impedimentos totales medidos en la red, una variable de deterioro (el Factor de Expectativa) se resta para obtener el ICPIF.

Esta es la misma fórmula usada por puertas de enlace Cisco para calcular el ICPIF para los flujos de datos de VoIP recibidos.

- ***El Factor de Deterioro de retardo***

El Factor de Deterioro de retardo (Idd) es un número basado en dos valores. Un valor es fijo y se obtiene utilizando los valores estáticos (como se define en los estándares de la ITU) para el retardo de Códec, "*Look Ahead Delay*" y retardo de Procesamiento de Señal Digital (DSP). El segundo valor es variable y se basa en el retardo en un sentido de medición (medición del tiempo de ida y vuelta dividido por 2). El valor de retardo de una vía se asigna a un número con una tabla de asignación que se basa en una expresión analítica de la G.107 (versión 2002). Tabla B-4 muestra las correspondencias de muestreo entre la demora de ida medido por IP SLA y los valores del factor de retardo por deterioro.

Tabla B-4 Correspondencia entre *One-way Delay* y retardo de deterioro ICPIF [1]

One-way delay	Factor de retardo de deterioro
50	1
100	2
150	4
200	7

- ***El factor de degradación del equipo***

El factor de degradación del equipo (Ie) es un número basado en la cantidad medida de pérdida de paquetes, expresado como un porcentaje del número total de paquetes enviados, se define de acuerdo al códec. Tabla B-5 muestra las correspondencias entre la muestra medida de pérdida de paquetes IP SLA y los valores del Factor de degradación del equipo.

Tabla B-5 Correspondencia: *Packet-Loss* y Factor de degradación por Equipo

Pérdida de paquetes (por centaje del total enviado)	Factor de deterioro por Equipo para códecs PCM (G.711)	Factor de deterioro por equipo para códecs CS-ACELP (G.729A)
2%	12	20
4%	22	30
6%	28	38
8%	32	42

- ***El Factor Expectativa***

El Factor de Expectativa, también llamado el factor de ventaja (A), está destinado a representar el hecho de que los usuarios pueden aceptar cierta degradación de la calidad a cambio de la facilidad de acceso. Por ejemplo, un usuario de teléfono móvil en una ubicación de difícil alcance, puede tener la expectativa de que la calidad de la conexión no será tan buena como la de una conexión de línea fija tradicional. Esta variable es también llamado el *Advantage Factor* (de Factor de ventaja de acceso) debido a que intenta equilibrar un aumento de la ventaja de acceso frente a una reducción en la calidad de voz. [1]

La Tabla B-6, adaptada de la ITU-T Rec. G.113, define un conjunto de valores máximos provisionales de A en términos del proveedor de servicios.

Tabla B-6 Valores máximos recomendados de Factor de Ventaja

Servicio de Comunicación	Factor de Ventaja o Expectativa Máximo valor de A
Línea terrestre convencional (línea fija)	0
Movilidad dentro de un edificio (celular)	5
Movilidad en un área geográfica o en un vehículo	10
Acceso a lugares de dificil acceso (ej. A través de conexión satelital multi-hop)	20

Estos valores son sólo sugerencias. Para ser significativos, el uso del factor A y su valor seleccionado en una aplicación específica debe ser utilizado consistentemente en cualquier modelo de planificación que adopte. Sin embargo, los valores en la Tabla B-6 deben ser considerados como los límites superiores absolutos para A.

El factor de Expectativa predeterminado para las operaciones de IP SLA VoIP UDP *jitter* es siempre cero.

B.1.9 Valor MOS de IP SLA

IP SLA utiliza una correspondencia observada entre ICPIF y los valores MOS para estimar un valor de MOS. El uso de la abreviatura MOS en el contexto de esta característica debe entenderse como representación del MOS-CQE (*Mean Opinion Score*; calidad conversacional, estimada).

El modelo E, tal como se define en la G.107 (03/2003), predice la calidad subjetiva que es experimentada por un oyente promedio combinando el deterioro causado por los parámetros de transmisión (tales como la pérdida y retraso) en una sola clasificación, el factor de transmisión R. Esta clasificación, expresada en una escala de 0 (el peor) a 100 (el mejor) se puede utilizar para predecir las reacciones de usuario subjetivas, tales como MOS. Específicamente, MOS puede obtenerse a partir del Factor R con una fórmula de

conversión. A la inversa, una forma de modificación contraria puede ser utilizada para calcular los factores R de los valores de MOS.

También hay una relación entre el valor ICPIF y el factor R. IP SLA se aprovecha de esta correspondencia mediante la derivación de la puntuación MOS aproximada de un factor R estimado, el cual, a su vez, se deriva de la puntuación ICPIF. La Tabla B-7 muestra los valores de MOS resultantes que serán generados por los valores ICPIF correspondientes.

Tabla B-7 Correspondencia de valores ICPIF y valores MOS

Rango ICPIF	MOS	Categoría de Calidad
0 – 3	5	Mejor
4 – 13	4	Alta
14 – 23	3	Media
24 – 33	2	Baja
34 – 43	1	Pobre

IP SLA siempre expresará el valor MOS estimado como un número en el rango de 1 a 5, siendo 5 la mejor calidad. Un valor MOS de 0 (cero) indica que los datos de MOS no pudieron generarse para la operación.

B.1.10 Cómo configurar la Operación IP SLA VoIP UDP jitter

La Operación IP SLA VoIP UDP *jitter* contiene diferentes opciones de configuración en comparación con la operación *jitter* UDP estándar. Tan pronto como se especifica la palabra clave codec en la sintaxis de comandos dentro de *type jitter dest-ipaddr*, va a configurar la aplicación VoIP específico de la operación de *jitter*.

Restricciones

Actualmente, IP SLA sólo admite los siguientes códecs de voz (métodos de compresión):

- *G.711 A Law (g711alaw: 64 kbps PCM compression method)*
- *G.711 mu Law (g711ulaw: 64 kbps PCM compression method)*
- *G.729A (g729a: 8 kbps CS-ACELP compression method)*

Los siguientes comandos disponibles en el modo de configuración de *jitter* UDP, no son válidos para las operaciones UDP *jitter* (códec):

- *distributions-of-statistics-kept*

- *statistics-distribution-interval*
- *request-data-size*

Resumen de pasos:

1. **enable**
2. **configure terminal**
3. **ip sla monitor** operation-number
4. **type jitter dest-ipaddr** {hostname | ip-address} **dest-port** port-number **codec** codec-type [**codec-numpackets** number-of-packets] [**codec-size** number-of-bytes] [**codec-interval** milliseconds] [**advantage-factor** value] [**source-ipaddr** {hostname | ip-address}] [**source-port** port-number] [**control** {**enable | disable**}]
5. **dest-ipaddr** ip-address
6. **dest-port** port-number
7. **enhanced-history** [**interval** seconds] [**buckets** number-of-buckets]
8. **frequency** seconds
9. **hours-of-statistics**-kept hours
10. **owner** owner-id
11. **tag** text
12. **threshold** milliseconds
13. **timeout** milliseconds
14. **tos** number
15. **verify-data**
16. **vrf** vrf-name
17. **exit**
18. **ip sla monitor schedule** operation-number [**life {forever** | seconds}] [**start-time** {hh:mm[:ss] [month day | day month] | **pending | now | after** hh:mm:ss] [**ageout** seconds] [**recurring**]
19. **exit**
20. **show ip sla monitor configuration** [operation-number]

APÉNDICE C

ESTUDIO DE EQUIPAMIENTO

ESTUDIO DE EQUIPAMIENTO

C.1 Equipamiento

Los equipos que se ocuparán son *routers* de la serie 2800, un Cisco 2801, cuatro *routers* Cisco 1841, 2 Cisco 2621, todos ellos con soporte MPLS.

A modo de comparación en cuanto al protocolo MPLS, en la figura C-1 se muestra una tabla con las características MPLS de los *routers* 1841 y 2801.

New Features	Cisco 1841	Cisco 2801
Multiprotocol Label Switching (MPLS) Support—Basic MPLS Capabilities		
Basic MPLS forwarding and signaling	Yes	Yes
Label Distribution Protocol (LDP)	Yes	Yes
Resource Reservation Protocol (RSVP)	Yes	Yes
MPSL Class of Service (CoS)		
Congestion management	Yes	Yes
Packet marking, policing, and shaping	Yes	Yes
MPLS Traffic Engineering		
MPLS Traffic Engineering	Yes	Yes
TE-RSVP	Yes	Yes
Guaranteed Bandwidth Traffic Engineering (GB-TE)	Yes	Yes
MPLS DiffServ-Aware Traffic Engineering (DS-TE)	Yes	Yes
Intermediate System-to-Intermediate System Traffic Engineering (ISIS-TE)	Yes	Yes
MPLS VPN Features		
MPLS VPN	Yes	Yes
Guaranteed Bandwidth VPN	Yes	Yes
Interprovider VPN	Yes	Yes
Carrier Supporting Carrier (CSC) VPN	Yes	Yes
Border Gateway Protocol (BGP) attributes	Yes	Yes

Fig. C-1 Características MPLS de CISCO 1841 y 2801

Fig. C-2 *Router* Cisco 2801

Fig. C-3 *Router* Cisco 1841

Fig. C-4 *Router* Cisco 2621

C.1.1 Características de los equipos utilizados

Tabla C-1 Características *router* Cisco 2621

Cisco 2621	
Tipo de dispositivo	*Router*
Factor de forma	Externo - modular - 1U
Ancho	43.8 cm, 43.82 cm
Profundidad	41.7 cm, 41.66 cm
Altura	4.5 cm, 4.45 cm
Peso	6.4 kg
Memoria RAM	256 MB (instalados) / 768 MB (máx.) - DDR SDRAM, 256 MB (instalados) / 760 MB (máx.) - DDR SDRAM Memoria Flash 64 MB (instalados) / 256 MB (máx.)
Protocolo de interconexión de datos	Ethernet, Fast Ethernet
Red / Protocolo de transporte	IPSec
Protocolo de gestión remota	SNMP 3
Indicadores de estado	Actividad de enlace, alimentación
Cumplimiento de normas	IEEE 802.3af
Protección firewall, cifrado del hardware, soporte de MPLS, Diseño modular, protección firewall, criptografía 128 bits, cifrado del hardware, asistencia técnica VPN, soporte de MPLS, filtrado de URL, cifrado de 256 bits	

Tabla C-2 Características *router* Cisco 1841

Cisco 1841	
Tipo de dispositivo	*Router*
Factor de forma	Externo - modular - 1U
Cantidad de módulos instalados (máx.)	0 (instalados) / 3 (máx.)

Cisco 1841	
Anchura	34.3 cm
Profundidad	27.4 cm
Altura	4.8 cm
Peso	2.7 kg
Memoria RAM	128 MB (instalados) / 384 MB (máx.) – SDRAM
Memoria Flash	32 MB (instalados) / 128 MB (máx.)
Tecnología de conectividad	Cableado
Protocolo de interconexión de datos	Ethernet, Fast Ethernet
Red / Protocolo de transporte	IPSec
Protocolo de gestión remota	SNMP, http
Total ranuras de expansión	(libres) 1 (0) x Tarjeta CompactFlash 1 (1) x AIM 2 (2) x HWIC
Interfaces	2 x red - Ethernet 10Base-T/100Base-TX - RJ-45 1 x USB - 4 PIN USB tipo A 1 x gestión - consola 1 x gestión – auxiliary
Algoritmo de cifrado	DES, Triple DES, SSL, AES de 128 bits, AES de 192 bits, AES de 256 bits
Método de autentificación	Secure Shell v.2 (SSH2)
Características Protección firewall, compresión del hardware, cifrado del hardware, asistencia técnica VPN, soporte VLAN, Sistema de prevención de intrusiones (IPS), montable en pared, Dynamic Multipoint VPN (DMVPN), Network Admissions Control (NAC)	

Tabla C-3 Características *router* 2801

Cisco 2801	
Tipo de dispositivo	*Router*
Factor de forma	Externo - modular - 1U
Cantidad de módulos instalados (máx.)	0 (instalados) / 3 (máx.)
Anchura	43.8 cm
Profundidad	41.7 cm
Altura	4.5 cm
Peso	6.4 kg
Memoria RAM	128 MB (instalados) / 384 MB (máx.) – SDRAM
Memoria Flash	64 MB (instalados) / 128 MB (máx.)
Tecnología de conectividad	Cableado
Protocolo de interconexión de datos	*Ethernet, Fast Ethernet*
Red / Protocolo de transporte	IPSec
Protocolo de gestión remota	SNMP3
Total ranuras de expansión	(libres) 2 (2) x HWIC 2 (2) x AIM 2 (2) x PVDM 1 (1) x WIC 1 (1) x VIC 1 Tarjeta CompactFlash
Interfaces	2 x red - Ethernet 10Base-T/100Base-TX - RJ-45 1 x USB - 4 PIN USB tipo A 1 x gestión - consola 1 x gestión – auxiliary
Algoritmo de cifrado	DES, Triple DES, AES
Método de autentificación	Secure Shell v.2 (SSH2)

Características Protección firewall, cifrado del hardware, alimentación mediante *Ethernet* (PoE), asistencia técnica VPN, soporte de MPLS, filtrado de URL. Cumplimiento de normas IEEE 802.3af, CISPR 22 Class A, CISPR 24, EN 61000-3-2, VCCI Class A ITE, IEC 60950, EN

Cisco 2801
61000-3-3, EN55024, EN55022 Class A, UL 60950, EN50082-1, CSA 22.2 No. 60950, AS/NZ 3548 Class A, JATE, FCC Part 15, ICES-003 Class A, CS-03, EN 61000-6-2.

Contenido

ÍNDICE DE FIGURAS

Referencias

[1] I. Cisco System, «Cisco IOS Quality of Service Solutions Command Reference,» Abril 2011. [En línea]. Available: http://www.cisco.com/en/US/docs/ios/qos/command/reference/qos_cr.pdf.

[2] I. Cisco System, «Cisco IOS IP SLAs Configuration Guide, Release 12.4,» Agosto 2008. [En línea]. Available: http://www.cisco.com/en/US/docs/ios/12_4/ip_sla/configuration/guide/sla_12_4_book.pdf.