ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE INGENIERÍA

IMPLEMENTACIÓN Y SIMULACIÓN DE CASOS DE ESTUDIO PARA ANALIZAR LA ARQUITECTURA MPLS "MULTIPROTOCOL LABEL SWITCHING" UTILIZANDO RUTEADORES CISCO DE LAS SERIES 3600 Y 2600

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

LEONARDO JAVIER BRAVO BRAVO

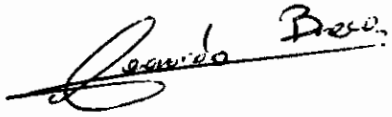
DIRECTOR: MSC. CARLOS EGAS

Quito, Marzo 2006

DECLARACIÓN

Yo Leonardo Javier Bravo Bravo, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normatividad institucional vigente.



Leonardo Javier Bravo Bravo

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Leonardo Javier Bravo Bravo, bajo mi supervisión.

MSc. Carlos Egas

DIRECTOR DE PROYECTO

AGRADECIMIENTOS

Parte fundamental en el desarrollo del presente trabajo ha sido el MSc. Carlos Egas, ya que su incondicional apoyo y motivación fue de gran ayuda para la culminación del mismo.

Al personal docente y administrativo del Centro de Transferencia y Desarrollo de Tecnología en Electrónica Telecomunicaciones y Redes de Información – CTTETRI, quienes me brindaron las facilidades y me ayudaron de varias formas para la realización del presente trabajo.

A mis familiares y amigos, quienes me brindaron su tiempo y apoyo incondicional a lo largo de mi carrera universitaria.

DEDICATORIA

El presente trabajo lo dedico en su totalidad a mis queridos padres: **Amable y Enith**; y a mis queridas hermanas **Patricia y Christina**, ya que sin el apoyo brindado por ellos durante toda mi vida no hubiese podido culminar con éxito esta etapa universitaria. Muchas gracias familia.

CONTENIDO

RESUMEN

El crecimiento imparable de la Internet, así como la demanda sostenida de nuevos y más sofisticados servicios, supone cambios tecnológicos fundamentales respecto a las prácticas habituales desarrolladas a mitad de los años 90. Nuevas tecnologías de transmisión tales como *Multi-Prototocol Label Switching* (MPLS) se considera fundamental en la construcción de los nuevos cimientos para la Internet del siglo XXI. Debido a la importancia que esta arquitectura ha tomado en la actualidad se ve la necesidad de elaborar un procedimiento conformado por varios casos de estudio para realizar la configuración y pruebas del funcionamiento de una red MPLS para facilitar las implementaciones futuras de redes basadas en dicha arquitectura.

La arquitectura MPLS se ha convertido en la solución más prometedora a las nuevas necesidades de las redes de backbone que tienen el enorme reto de gestionar redes cada vez más complejas y extensas, con una mayor gama de servicios y con creciente demanda de ancho de banda, calidad y garantías. En este proyecto configuraremos paso a paso una red MPLS que soporta VPNs y brinda características de QoS. Comenzaremos con una configuración básica de MPLS pasando por la configuración de VPNs sobre MPLS y finalmente configuraremos la característica QoS.

En el capítulo I se realizará el estudio de la arquitectura MPLS y sus principales aplicaciones (VPNs y Qos), en el capítulo II se realizará la configuración de los equipos con las características antes mencionadas, en el capítulo III se diseñarán los casos de estudio que facilitarán la comprensión de la configuración MPLS, estos casos de estudio están formados por una parte de configuración y otra de pruebas. Finalmente como complemento al proyecto utilizaremos una herramienta de simulación que ayudará a visualizar el funcionamiento de una red MPLS en situaciones como congestión de LSRs y caída de enlaces.

PRESENTACIÓN

El rápido crecimiento del Internet, la aparición de nuevas aplicaciones de datos, video, voz y multimedia, en las áreas educativa, de investigación y comercial, traen consigo la necesidad de crear mecanismos que hagan posible el funcionamiento eficaz de tales aplicaciones y motivan la creación de otras más innovadoras, así como los mecanismos que permitan la transición a un esquema de red de convergencia (donde no se deba tener redes diferentes para aplicaciones diferentes), ha sido la motivación principal para el desarrollo de la arquitectura MPLS (Multiprotocol Label Switching) como una solución versátil para hacer frente a las necesidades actuales, como son: velocidad, escalabilidad, manejo de Calidad del Servicio (QoS) e ingeniería de tráfico entre otras. MPLS representa el siguiente nivel de evolución en estándares, donde se combinan las tecnologías de conmutación de capa dos (enlace de datos) y tecnologías de enrutamiento de capa tres (red). MPLS aparece como una solución elegante para alcanzar los requerimientos de ancho de banda y servicios para la nueva generación de redes de backbone basadas en el protocolo IP.

En este trabajo se presenta la configuración de una red MPLS que soporta VPNs y brinda características QoS, además se presenta el estudio de la arquitectura MPLS y sus principales aplicaciones.

El diseño de los casos de estudio está basado en las configuraciones realizadas en el capítulo II y se complementa con la utilización de la herramienta de simulación "OpenSimMPLS".

CAPÍTULO I

1 MULTIPROTOCOL LABEL SWITCHING (MPLS)

MPLS fue presentado originalmente como una solución para mejorar la velocidad en los ruteadores, pero ahora está emergiendo como una tecnología de estándares crucial, la cual ofrece nuevas capacidades para redes IP a gran escala. Ejemplos de aplicaciones de MPLS son: Calidad de Servicio (QoS) y soporte para Redes Privadas Virtuales (VPN). Estos son dos ejemplos de aplicaciones claves donde MPLS es superior a cualquier tecnología IP disponible en la actualidad. Aunque MPLS fue concebido para ser independiente de la capa 2, gran parte del interés generado por MPLS gira alrededor de la promesa de implementar de una forma más efectiva redes IP a través de redes de backbone WAN-ATM.

1.1 INTRODUCCIÓN A MPLS

Comúnmente, para enviar un paquete IP se debe analizar la dirección IP destino contenida en la cabecera del paquete de capa 3 a medida que éste se desplaza por la red desde su origen hasta su destino. El encargado de realizar este análisis en cada salto es el ruteador. El enrutamiento estático y los protocolos de enrutamiento dinámico construyen bases de datos necesarias para enrutar los paquetes a través de la red. El proceso de llevar a cabo el enrutamiento IP tradicional también se denomina *enrutamiento unidifusión basado en el destino salto a salto.*

El método mencionado es ampliamente usado a pesar de sus debilidades que disminuyen su flexibilidad. Por ello, se impone contar con nuevas técnicas para expandir la funcionalidad de una infraestructura de red basada en el protocolo IP.

El envío tradicional de paquetes por la capa de red se apoya en la información proporcionada por los protocolos de enrutamiento de la capa red (por

ejemplo: OSPF o BGP), o el enrutamiento estático, para tomar una decisión de envío independiente en cada salto dentro de la red. La decisión de envío se basa estrictamente en la dirección IP unidifusión de destino. Todos los paquetes para el mismo destino siguen la misma ruta a través de la red, si no existen otras rutas de acceso de igual costo. Siempre que un ruteador tenga dos rutas de igual costo hacia su destino, los paquetes dirigidos hacia ese destino podrán tomar una o ambas rutas, con lo que en cierta medida se compartiría la carga. •

MPLS está basado en el concepto de conmutación de etiquetas: una "etiqueta" independiente y única es agregada a cada paquete de datos y ésta es utilizada para enrutar y conmutar el paquete de datos a través de la red. La etiqueta es simple (es una versión corta de la cabecera de un paquete de información IP) lo que favorece a la optimización de los equipos de red en lo que al procesamiento de etiquetas y envío de tráfico se refiere. Este concepto ha sido difundido en la industria de las comunicaciones durante años. X.25, Frame Relay y ATM son ejemplos de tecnologías de conmutación de etiquetas.

Los ruteadores IP convencionales contienen "tablas de enrutamiento", las cuales son utilizadas en base al encabezado IP del paquete para tomar la decisión de cómo el paquete será enviado. Estas tablas son construidas basándose en protocolos de enrutamiento IP (ej. OSPF, RIP, etc.), los cuales transportan información de topología y alcance en la red en forma de direcciones IP. En la práctica encontramos que el plano de envío (búsqueda de dirección IP en la tabla) y el plano de control (generación de las tablas de enrutamiento) están íntimamente ligados. En cambio en MPLS el envío está basado en etiquetas, es posible separar de forma clara el plano de envío basado en etiquetas del plano de control. Separando estos dos planos cada uno puede ser modificado de forma independiente. Con esta separación no es necesario cambiar los dispositivos de envío por ejemplo para migrar a una nueva estrategia de enrutamiento en la red.

En MPLS, el asignamiento de un paquete particular a una FEC particular (Forwarding Equivalence Class), es realizado una sola vez al momento que el

• Referencia bibliográfica: "Arquitecturas MPLS y VPN" [1]

paquete ingresa a la red. La FEC al cual el paquete es asignado es codificada como un valor de longitud fija y corta llamada "etiqueta". Cuando el paquete es enviado a su próximo salto la etiqueta es enviada junto con éste. Lo que significa que los paquetes son etiquetados antes de ser enviados.

En el siguiente salto, no hay un análisis adicional a la cabecera del paquete de capa red. En lugar de esto, la etiqueta es utilizada como un índice dentro de la tabla que especifica el próximo salto. La vieja etiqueta es reemplazada con una nueva y el paquete es enviado a su próximo salto.

Una vez que el paquete es asignado a una FEC, no es necesario analizar la cabecera en los siguientes ruteadores, todo el proceso de envío es manejado por la etiquetas. Esto tiene gran ventaja sobre los mecanismos convencionales de envío de paquetes de capa red.

La conmutación de etiquetas multiprotocolo (MPLS) es una tecnología emergente destinada a encauzar muchos de los retos actuales que plantea el envío de paquetes en las redes modernas. Ya que múltiples soluciones independientes para el desarrollo de tecnologías basadas en conmutación de etiquetas es claramente una solución no aceptable, se reconoció la necesidad de desarrollar estándares y se creó el grupo de trabajo de la IETF para este propósito, el cual fue creado en abril del 1997; desde entonces MPLS se encuentra en proceso de estandarización.

El grupo de trabajo MPLS es responsable de la estandarización de una tecnología base usando conmutación de etiquetas y la implementación de rutas de etiquetas conmutadas sobre varios paquetes basados en tecnologías de nivel de enlace, tales como Packet-Over-Sonet, ATM, Frame Relay y tecnologías LAN (ejemplo: todas las formas de Ethernet, Token Ring, etc.). Esto incluye procedimientos y protocolos para la distribución de etiquetas entre ruteadores y encapsulación.

1.1.1 MPLS e IP [•]

Es importante comprender las diferencias entre la manera en que MPLS y el enrutamiento IP envían datos a través de una red. El envío de paquetes IP tradicional usa la dirección IP destino ubicada en la cabecera del paquete para realizar una decisión de envío independiente en cada ruteador de la red. Estas decisiones salto a salto son basadas en protocolos de enrutamiento de capa red, tales como OSPF o BGP, los mismo que buscan la ruta más corta a través de la red para llegar al destino.

MPLS crea un modelo overlay basado en conexión dentro de las tradicionales redes enrutadas IP no orientadas a conexión. Esta arquitectura orientada a conexión da apertura a nuevas posibilidades de administración de tráfico en una red IP.

1.1.2 ARQUITECTURA MPLS [•]

La arquitectura MPLS describe los mecanismos para realizar la conmutación de etiquetas, que combina los beneficios del envío de paquetes basados en la conmutación de Capa 2 con los beneficios del enrutamiento de Capa 3. De forma similar a las redes de Capa 2 (ejemplo: Frame Relay o ATM), MPLS asigna etiquetas a los paquetes para su transporte a través de redes basadas en paquetes o celdas. El mecanismo de envío a través de la red es el intercambio de etiquetas, en cuyas unidades de datos se añade una etiqueta, de longitud fija, que indica a los nodos de conmutación que hay a lo largo de la ruta de los paquetes la forma de procesar y enviar los datos.

La diferencia más significativa entre las tecnologías MPLS y la tradicional WAN es la forma en que se asignan las etiquetas y la capacidad de transportar una pila de etiquetas adjuntas a un paquete. El concepto de una pila de etiquetas

---

[•] Referencia bibliográfica: "MPLS Conformance and Performance Testing" [2]
[•] Referencia bibliográfica: "Arquitecturas MPLS y VPN" [1]

habilita nuevas aplicaciones, como la ingeniería de tráfico, las redes privadas virtuales, el reenrutamiento rápido alrededor de un enlace, etc.

Las redes sin conexión contrastan con el envío de paquetes en MPLS, en las redes sin conexión cada paquete se analiza sobre la base de salto a salto, se comprueba su cabecera de capa Red, y se toma una decisión de envío independiente en base a la información extraída de un algoritmo de enrutamiento de capa Red.

La arquitectura MPLS se divide en dos componentes separados:

Componente de envío.- También denominado **plano de datos**, emplea una base de datos de envío de etiquetas mantenida por una conmutación de etiquetas, para ejecutar el envío de paquetes de datos basándose en las etiquetas transportadas por los paquetes.

Componente de Control.- Es el responsable de la creación y mantenimiento de la información de envío de etiquetas (denominadas también **enlaces**) entre un grupo de *switches* de etiquetas interconectados.

En cada nodo MPLS se deben ejecutar uno o varios protocolos de enrutamiento IP para intercambiar la información de enrutamiento IP con otros nodos MPLS de la red. En este sentido, cada nodo MPLS es un router IP en el plano de control, como muestra la figura 1.1.

En un nodo MPLS, la tabla de enrutamiento IP se utiliza para determinar el intercambio de enlace de etiquetas, donde los nodos MPLS adyacentes intercambian etiquetas para las subredes individuales almacenadas en la tabla de enrutamiento IP. El intercambio de enlaces de etiquetas para el enrutamiento IP basado en destinos unidifusión se realiza utilizando el Protocolo de distribución de etiquetas identificativas (TDP) patentado por Cisco o el Protocolo de distribución de etiquetas (LDP) especificado por el IETF.

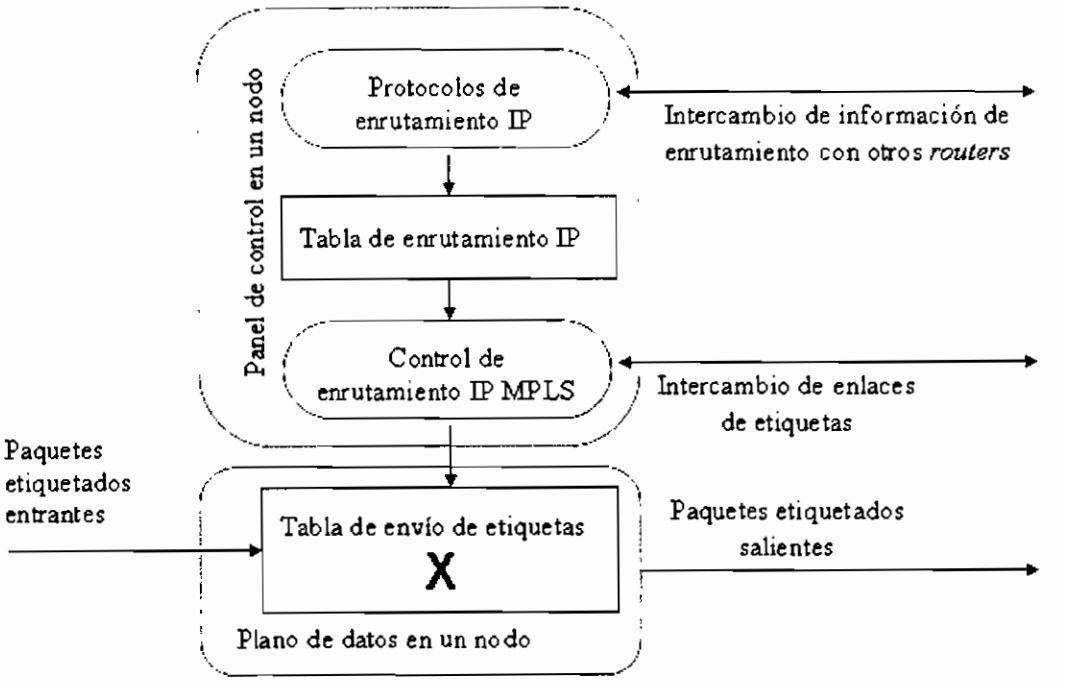


Figura 1.1: Arquitectura básica de un nodo MPLS realizando el enrutamiento IP [•]

El proceso de control del enrutamiento IP MPLS utiliza las etiquetas intercambiadas entre nodos MPLS continuos para crear la tabla de envío de etiquetas denominada Label Forwarding Information Base (LFIB), que es la base de datos del plano de envío que se emplea para enviar los paquetes etiquetados a través de la red MPLS.

1.1.2.1 Dispositivos MPLS (Construcción de bloques)

En toda nueva tecnología, la aparición de nuevos términos para describir los dispositivos que conforman la arquitectura es inevitable. Estos nuevos términos describen la funcionalidad de cada dispositivo y sus funciones dentro de la red MPLS.

Router de conmutación de etiquetas (LSR).- Es un router de alta velocidad en el corazón de la red MPLS, el cual debe soportar los protocolos de enrutamiento IP y participa en el establecimiento de los LSP (Label Switched

[•] Referencia bibliográfica: "Arquitecturas MPLS y VPN" [1]

Path) utilizando el protocolo de señalización de etiquetas adecuado. Permite conmutación de tráfico de datos a alta velocidad basado en los caminos establecidos y es capaz de enviar paquetes basándose en etiquetas, típicamente es un conmutador ATM modificado. La función básica de los procedimientos de distribución de etiquetas es permitir que un LSR distribuya sus enlaces de etiquetas a otros LSR del mismo dominio MPLS.

Existen diferentes tipos de LSR cuya diferencia es la función que desempeñan en la infraestructura de red y es puramente arquitectónica. A estos diferentes tipos de LSR se los denomina **LSR de contorno, LSR ATM y LSR ATM de contorno.**

LSR de contorno.- Es un router que básicamente realiza dos funciones: imposición de etiquetas y la determinación de etiquetas en el contorno de la red.

Imposición de etiquetas.- Llamada también acción ***push*** es la acción de añadir una etiqueta o una pila de etiquetas, a un paquete en el punto de entrada (con respecto al flujo de tráfico desde el origen al destino) de la red MPLS.

Determinación de etiquetas.- Llamada también acción ***pop*** consiste en eliminar la última etiqueta de un paquete en el punto de salida antes de que se envíe a un vecino que está fuera de la red MPLS, es decir, lo contrario a la imposición de etiquetas.

Un LSR de contorno es reconocido como un LSR que tenga vecinos que no sean MPLS y si éste tuviera una interfaz conectada a un LSR ATM a través de MPLS será también un **LSR ATM de contorno**. Los LSR de contorno emplean la tradicional tabla de envíos IP que además contiene información de etiquetado para realizar las acciones de pop o push según sea el caso.

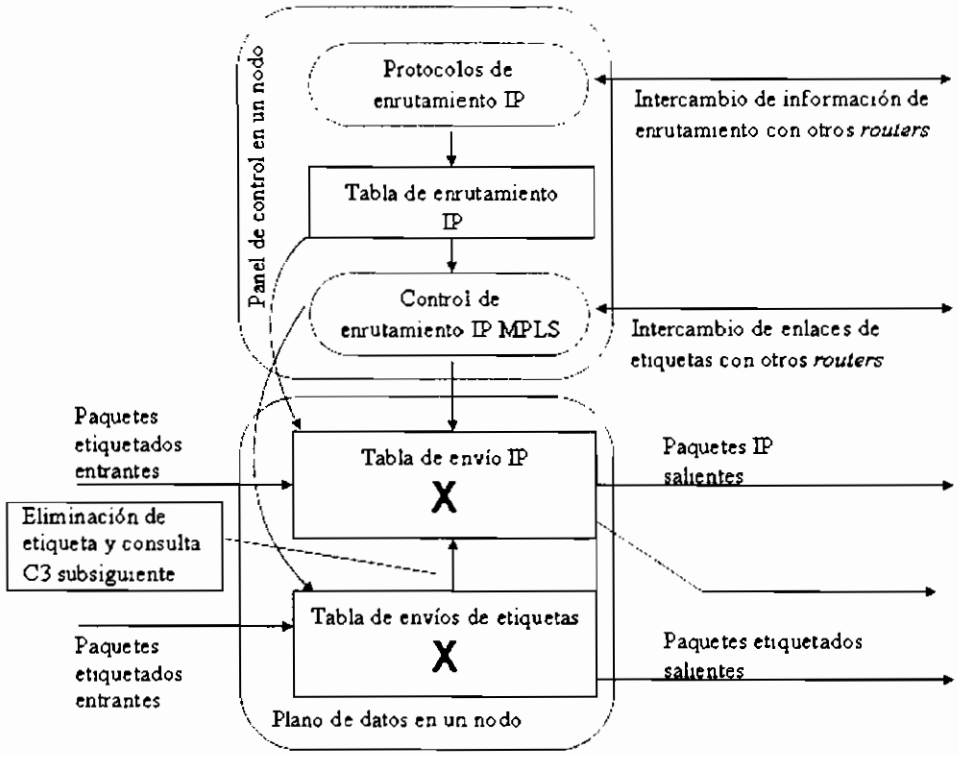


Figura 1.2: Arquitectura de un LSR de contorno [•]

En un LSR de contorno existen componentes adicionales en el plano de datos. La tabla de envío IP se construye sobre la base de la tabla de enrutamiento IP y se amplia con información de etiquetado. Los paquetes IP que ingresan se pueden enviar como paquetes IP puros hacia nodos que no son MPLS, o pueden ser etiquetados y enviados a un nodo MPLS. Los paquetes etiquetados que ingresan se envían a otro nodo MPLS como paquetes etiquetados, por otro lado, cuando estos paquetes son destinados a nodos que no son MPLS, se elimina la etiqueta y se realiza el enrutamiento IP hacia el destino que no es MPLS.

Un **LSR ATM** es un switch ATM que actúa como LSR, realiza el enrutamiento IP y la asignación de etiquetas en el plano de control y envía paquetes utilizando mecanismos de conmutación por celdas ATM adicionales en el plano de datos. Las funciones desempeñadas por los diferentes tipos de LSR se listan a continuación, cabe recalcar que cualquier dispositivo en la red puede

[•] Referencia bibliográfica: "Arquitecturas MPLS y VPN" [1]

desempeñar más de una función, por ejemplo, un LSR de contorno puede ser al mismo tiempo un LSR ATM de contorno.

Tipo de LSR	Función
LSR	**Envía paquetes etiquetados**
LSR ATM	**Ejecuta protocolos MPLS en el plano de control para establecer circuitos ATM virtuales.** **Envía paquetes etiquetados como celdas ATM.**
LSR de contorno	**Recibe paquetes IP, efectúa consultas de Capa 3, e impone una pila de etiquetas antes de enviar el paquete dentro del dominio MPLS.** **Recibe paquetes etiquetados, elimina etiquetas, realiza consultas de Capa 3 y envía el paquete IP hacia su siguiente destino.**
LSR ATM de contorno	**Puede recibir tanto paquetes etiquetados como no etiquetados, segmentarlo en celdas ATM y enviar las celdas hacia su próximo salto LSR ATM.** **Puede recibir celdas ATM de un LSR ATM vecino, reensamblar estas celdas en el paquete original y después enviar el paquete como paquete etiquetado o no etiquetado.**

Tabla 1.1: Tipos de LSR y sus funciones [•]

1.1.3 CARACTERÍSTICAS DE LA CONMUTACIÓN DE ETIQUETAS MULTIPROTOCOLO [•]

MPLS, en conjunto con otras tecnologías estandarizadas, ofrecen entre otras las siguientes características críticas a quienes deciden implementarla, que por lo general son los proveedores de servicios:

- MPLS, en combinación con los protocolos estandarizados de enrutamiento IP tales como OSPF o IS-IS, provee un soporte completo y altamente escalable de enrutamiento IP dentro de una infraestructura ATM.

[•] Referencia bibliográfica: "Arquitecturas MPLS y VPN" [1]
[•] Referencia bibliográfica: "Cisco MPLS Controller Software Configuration Guide" [3]

- MPLS, en combinación con el protocolo BGP (Border Gateway Protocol), brinda soporte a servicios de Redes Privadas Virtuales IP altamente escalables. Los servicios VPN IP son un invaluable desarrollo en las redes del proveedor, dando a la empresa cliente un servicio que satisface sus necesidades de privacidad y entrega de servicios IP no orientados a conexión.

- Acuerdos de Niveles de Servicio (SLA) pueden ser provistos en forma adecuada para tráfico no orientado a conexión, la combinación de MPLS con el estándar DiffServ (Diferenciación de Servicios) ayuda a este proceso. Junto con el soporte de VPN y la habilidad de brindar Niveles de Servicio adecuado para tráfico IP hacen que MPLS sea una solución a las nuevas demandas de servicios IP.

- MPLS y el enrutamiento IP pueden fácilmente ser introducidos en redes tradicionales ATM usando técnicas de tunelamiento como PVP o PVC, ya que los conmutadores con capacidad MPLS son introducidos continuamente.

1.1.4 BENEFICIOS DE LA CONMUTACIÓN DE ETIQUETAS MULTIPROTOCOLO [*]

La tecnología MPLS ofrece muchas ventajas sobre la tradicional IP-over-ATM. Cuando se integra con conmutadores ATM, la conmutación de etiquetas toma ventaja del hardware optimizado del conmutador para aprovechar la longitud fija de las celdas ATM y la conmutación de celdas a altas velocidades. Para redes multiservicio, la conmutación de etiquetas habilita al conmutador para proveer ATM, Frame Relay, servicio IP Internet y servicios IP-VPN todos en una sola plataforma de forma sumamente escalable. El soporte de estos servicios en una plataforma común provee ahorro en los costos operacionales y simplifica la labor de los proveedores multiservicio.

---

[*] Referencia bibliográfica: "Cisco MPLS Controller Software Configuration Guide" [3]

Estos beneficios de MPLS son listados a continuación:

- Integración
- Alta confiabilidad
- Mejor Eficiencia
- Implementación directa de Clases de Servicios
- Escalabilidad y administrabilidad de VPNs
- Capacidad de Ingeniería de tráfico

1.1.5 ESTRUCTURA DE UNA RED MPLS

Una típica estructura de una red MPLS usada por un proveedor de servicio es la mostrada en la figura 1.3. Como se describió anteriormente los elementos básicos de una red MPLS son:

LSR de contorno.- está ubicado en los límites de la red dándole valor agregado a los servicios de capa red y aplicando etiquetas a los paquetes. Estos dispositivos pueden ser ruteadores o conmutadores multicapa LAN.

LSR ATM.- Estos dispositivos conmutan paquetes etiquetados o celdas basadas en etiquetas. Estos dispositivos pueden soportar enrutamiento de Capa 3 o conmutación de Capa 2 además de la conmutación de etiquetas. Un Switch Router Multiservicio es un LSR ATM.

LSR ATM de contorno.- Puede recibir tanto paquetes etiquetados como no etiquetados, segmentarlo en celdas ATM y enviar las celdas hacia su próximo salto LSR ATM

Protocolo de Distribución de Etiquetas (LDP).- El protocolo de distribución de etiquetas es usado en conjunto con los protocolos de enrutamiento estandarizados de capa Red para distribuir información de etiquetado entre dispositivos en una red de conmutación de etiquetas.

Una red MPLS consiste de LSRs de contorno alrededor de un núcleo de LSRs. Los clientes están conectados a la red MPLS por medio de los LSR de contorno.

Típicamente existen cientos de clientes por LSR de contorno. EL Equipo de Propiedad del Cliente (Customer Premises Equipment CPE) está ejecutando el envío IP ordinario pero usualmente no ejecuta MPLS. Si el CPE no ejecuta MPLS, lo usa independientemente del proveedor.

Es importante notar que los LSRs de contorno son parte de la red MPLS del proveedor y son controlados por éste. Los LSRs de contorno son críticos para la operación de la red y no se pretenden que sean CPE bajo ninguna circunstancia. El proveedor puede localizar y administrar los ruteadores de los clientes, a pesar de que éstos estén ejecutando IP ordinario y se encuentren fuera de la red MPLS.

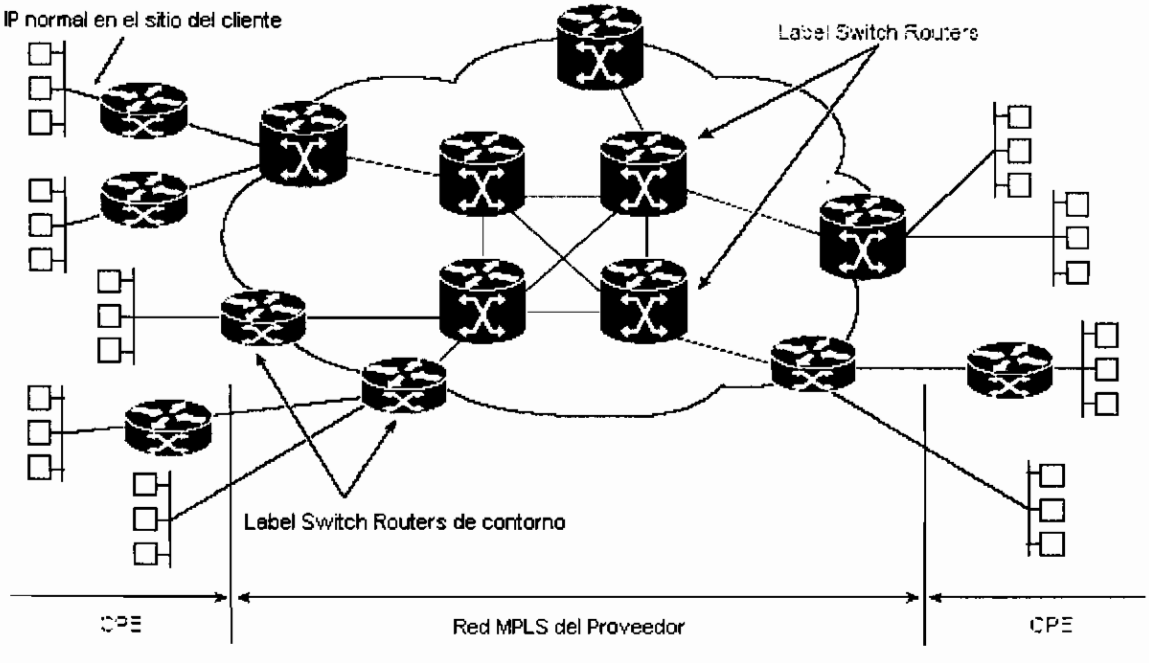


Figura 1.3: Estructura básica de una red MPLS

1.1.6 FUNCIONAMIENTO DE MPLS

MPLS es la tecnología usada para optimizar el envío de tráfico a través de una red. Aunque MPLS puede ser aplicado en diferentes ambientes de red, en este proyecto el enfoque principal será hacia MPLS en redes de paquetes IP, por ser una de las aplicaciones más comunes de MPLS en la actualidad. MPLS asigna etiquetas a los paquetes para transportarlos a través de la red, las etiquetas están contenidas en una cabecera MPLS insertada dentro del paquete de datos como muestra la figura 1.4:

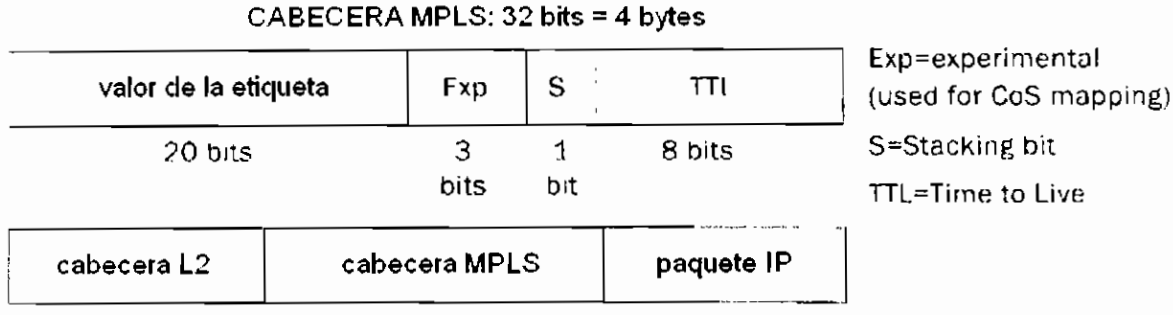


Figura 1.4: Formato de la cabecera MPLS en un paquete MPLS [•]

Estas etiquetas cortas y de longitud fija llevan la información que le dice a cada nodo de conmutación como procesar y enviar el paquete, desde la fuente al destino. Esto tiene significado sólo en una conexión nodo-a-nodo local. Como cada nodo envía el paquete, éste intercambia la etiqueta actual por la apropiada para enrutar el paquete al próximo nodo. Este mecanismo permite muy altas velocidades de conmutación de paquetes a través del núcleo de la red MPLS.

MPLS combina lo mejor del enrutamiento de Capa 3 y la conmutación de Capa 2, de hecho, algunas veces es llamado protocolo de "Capa 2½". Mientras los ruteadores requieren procesamiento a nivel de Capa red para determinar a donde enviar el tráfico, los conmutadores solo envían datos al próximo salto, entonces son intrínsicamente más simples, rápidos y menos costosos. MPLS se apoya en los tradicionales protocolos de enrutamiento IP para anunciar y establecer la topología de red, MPLS es entonces tratado en lo más alto de esta

[•] Referencia bibliográfica: "MPLS Conformance and Performance Testing" [2]

topología. MPLS predetermina la ruta que los datos toman a través de la red y codifica esa información dentro de una etiqueta que los ruteadores de la red interpretan sin problemas

Dado que el planeamiento de la ruta ocurre con anticipación y en el contorno de la red (donde el cliente y la red del proveedor de servicio se encuentran), el etiquetado MPLS de datos requiere menos capacidad en los ruteadores para atravesar el núcleo de la red del proveedor de servicio.

1.1.6.1 Imposición de etiquetas en el contorno de la red MPLS.*

La imposición de etiquetas es el acto de adición de una etiqueta a un paquete, cuando éste entra en el dominio MPLS. Se trata de una función de frontera o contorno, lo que quiere decir que los paquetes se etiquetan antes de enviarse al dominio MPLS.

Para desempeñar esta función, un LSR de contorno debe comprender dónde se ha encabezado el paquete y qué etiqueta, o pila de etiquetas, se debería asignar al paquete. En un envío de etiqueta de Capa 3 convencional, cada salto en la red realiza una consulta en la tabla de envíos IP para la dirección de destino IP almacenada en al cabecera de Capa 3 del paquete, selecciona una dirección IP de siguiente salto para el paquete en cada iteración de la consulta y, eventualmente, envía el paquete fuera de una interfaz hacia su destino final.

Algunos mecanismos de envío, como CEF (Cisco Express Forwarding) permiten al router asociar cada prefijo de destino conocido de la tabla de enrutamiento al siguiente salto adyacente del prefijo de destino; así se resuelve el problema de las consultas repetitivas. Toda la recursividad se resuelve mientras el router puebla la caché o la tabla de envíos y no cuando tiene que enviar paquetes.

---

* Referencia bibliográfica: "Arquitecturas MPLS y VPN" [1]

La elección para el siguiente salto del paquete IP es una combinación de dos funciones, La primera función separa el conjunto de paquetes posibles en un conjunto de prefijos de destino IP. La segunda función efectúa la asignación de cada prefijo de destino IP a una dirección IP del siguiente salto. Esto significa que cada destino en la red se alcanza mediante una ruta respecto al flujo de tráfico que va desde un dispositivo de entrada hasta el dispositivo de salida de destino (podrían habilitarse múltiples rutas si el equilibrado de la carga se realiza utilizando rutas de costo equivalente o rutas de costo desigual, como ocurre con algunos protocolos IGP, como, por ejemplo, IGRP mejorado).

Clases equivalentes de envío (FEC).- En la arquitectura MPLS, los resultados de la primera función se conoce como *clases equivalentes de envío*, éstas vendrían a ser como un conjunto de paquetes IP que se envían de la misma manera, por la misma ruta y con idéntico tratamiento.

> Una clase equivalente de envío podría corresponder a una subred de destino, pero también podría corresponder a cualquier clase de tráfico que un LSR de contorno considere significativa. Por ejemplo, todo el tráfico interactivo hacia un cierto destino o todo tráfico con un cierto valor de precedencia IP podría constituir una FEC. Otro ejemplo; una FEC puede ser un subconjunto de la tabla BGP, incluyendo todos los prefijos de destino alcanzables a través del mismo punto de salida.

Con el envío IP convencional, el procesamiento de paquetes se efectúa en cada salto en la red. No obstante, cuando se introduce MPLS, se asigna un paquete particular a una FEC particular una sola vez, y esto tiene lugar en el dispositivo de contorno a medida que el paquete entra en la red. La FEC a la que se asigna el paquete se codifica entonces como un identificador corto de longitud fija, conocido como etiqueta.

Cuando se envía un paquete al siguiente salto, se añade la etiqueta al final del paquete IP, de modo que el siguiente dispositivo en la ruta del paquete pueda

enviarlo basándose en la etiqueta codificada en lugar de a través del análisis de la información de la cabecera de Capa 3. La figura 1.5 muestra todo el proceso de imposición y envío de etiquetas.

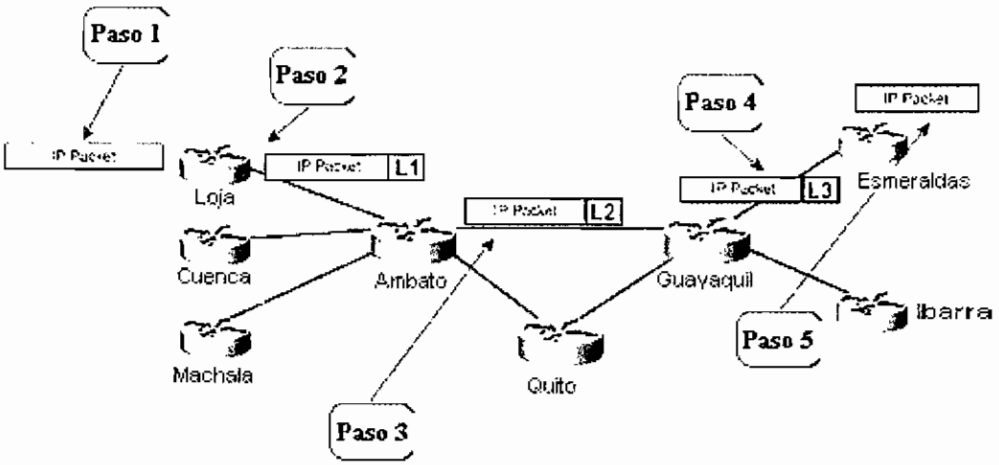


Figura 1.5: Imposición y envío de etiquetas MPLS [•]

Paso 1: Los paquetes IP llegan al router de Loja.

Paso 2: El router de Loja efectúa la consulta de Capa 3, añade la etiqueta y envía el paquete hacia el router de Ambato.

Paso 3: El router de Ambato consulta la etiqueta, intercambia etiquetas y envía el paquete hacia el router de Guayaquil

Paso 4: El router de Guayaquil consulta la etiqueta, intercambia etiquetas y envía el paquete hacia el router de Esmeraldas.

Paso 5: El router de Esmeraldas consulta la etiqueta, omite la etiqueta, hace la consulta de Capa 3 y envía el paquete hacia el router del siguiente salto externo.

1.1.6.2 Enrutamiento MPLS

Todos los paquetes que entran a una red MPLS lo hacen por medio de un LSR de entrada y salen de la misma forma por medio de un LSR de salida. Este

[•] Referencia bibliográfica: "Arquitecturas MPLS y VPN" [1]

mecanismo crea una ***ruta conmutada por etiqueta (LSP)***, que es esencialmente el conjunto de LSR que atraviesa el paquete desde el LSR de entrada hasta llegar al LSR de salida para una FEC particular. Esta LSP es unidireccional, lo que quiere decir que el tráfico de retorno desde una FEC determinada utilizará otra LSP.

La creación de la LSP es un esquema orientado a conexión porque la ruta se establece antes que cualquier flujo de tráfico. Sin embargo, el establecimiento de esta conexión se basa en información topológica más que en la necesidad de un flujo de tráfico. Esto significa que la ruta se crea independientemente de si en ese momento hay tráfico esperando a pasar por la ruta hacia un conjunto particular de FEC.

A medida que el paquete atraviesa la red MPLS, cada LSR intercambia la etiqueta entrante por otra de salida, parecido al mecanismo utilizado actualmente en ATM, donde el VPI/VCI se intercambia por un par VPI/VCI diferente cuando sale del switch ATM. El proceso es repetido hasta llegar al último LSR denominado LSR de salida.

El equivalente MPLS de la matriz de conmutación de un Switch ATM es la base de información de envío de etiquetas. La información relacionada con el componente de envío se almacena en dos tablas, cada LSR mantiene estas dos tablas.

La primera, conocida en el IOS de Cisco como ***Base de información de etiquetas (TIB) y LIB en términos MPLS estándar,*** mantiene todas las etiquetas asignadas por este LSR y las asignaciones de estas etiquetas a las etiquetas recibidas de cualquiera de los vecinos. Estas asignaciones de las etiquetas se distribuyen mediante el uso de protocolos de distribución de etiquetas.

Así como varios vecinos pueden enviar etiquetas para un mismo prefijo IP aunque pudiera no ser el siguiente salto IP actualmente en uso en la tabla de enrutamiento para el destino, no todas las etiquetas de la TIB/LIB deben utilizarse para el envío de paquetes. La segunda tabla, conocida en el IOS de Cisco como ***Base de información de envío de etiquetas (TFIB, y LFIB en términos MPLS estándar)***, se utiliza durante el envío de paquetes y almacena sólo las etiquetas que en ese momento está usando el componente de envío MPLS.

Las figuras 1.6a y 1.6b muestran un LSR de contorno en términos MPLS estándar y en términos del IOS de cisco y de la terminología del Envío expreso de Cisco (CEF) (se eligió un LSR de contorno ya que su función es un superconjunto de un LSR que no es de contorno).

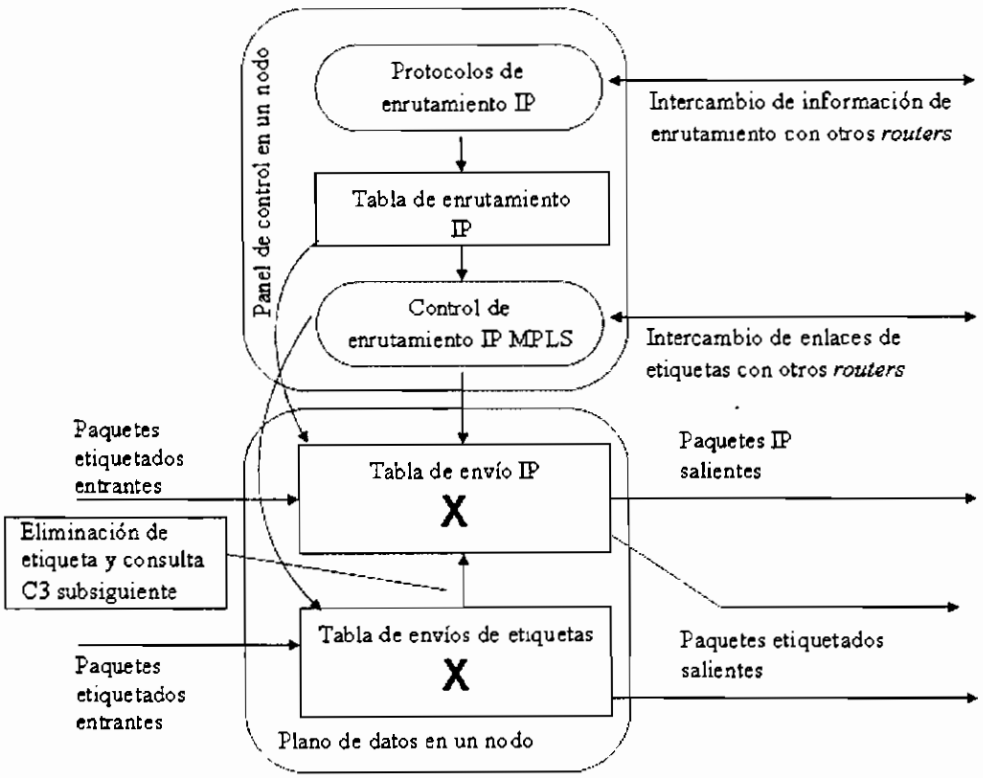


Figura 1.6a: Arquitectura LSR de contorno [•]

[•] Referencia bibliográfica: "Arquitecturas MPLS y VPN" [1]

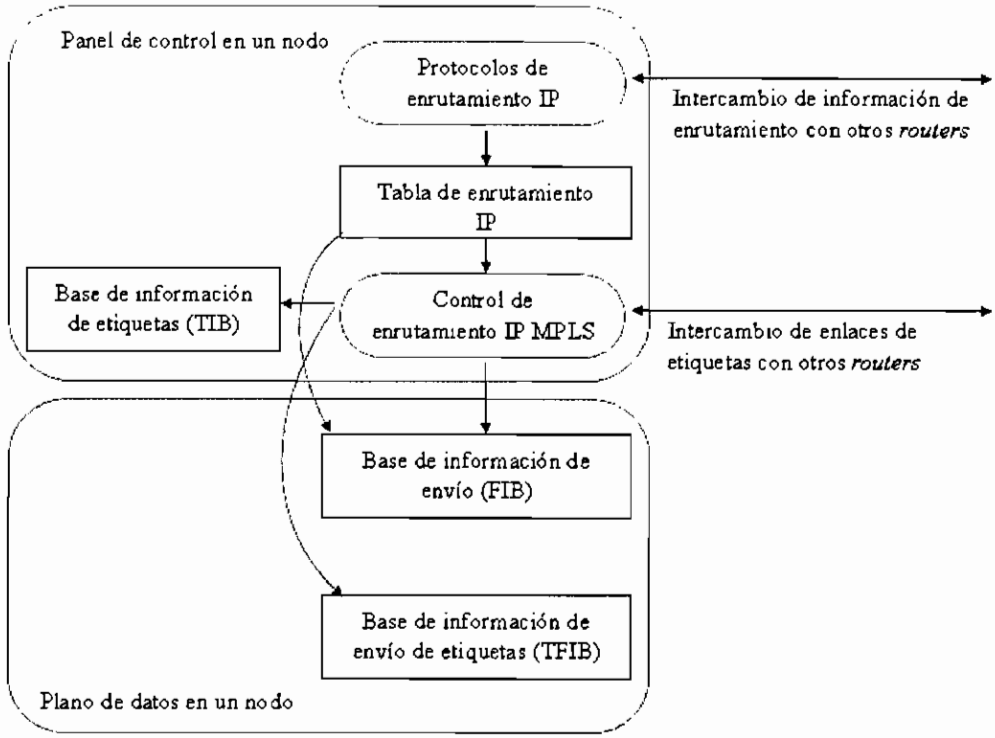


Figura 1.6b: Arquitectura LSR de contorno en términos del IOS de Cisco [•]

1.1.6.3 Funcionamiento de MPLS en modo Trama

Para comprender el funcionamiento de MPLS en modo trama, abordaremos en primer lugar el plano de datos MPLS, ya que mediante éste se establece un acuerdo entre routers. Luego explicaremos los mecanismos exactos empleados para distribuir las etiquetas entre los routers, y finalmente veremos la interacción entre los protocolos de distribución de etiquetas, el Protocolo de gateway interior (IGP) y el Protocolo de gateway fronterizo (BGP) en una red de un proveedor de servicios.

A lo largo de este capítulo se hace referencia a la arquitectura genérica de un router de conmutación de etiquetas MPLS (LSR) que se muestra en la figura 1.7a y empleamos de ejemplo una red de un proveedor de servicios llamada ***LOJANet*** ilustrada en la figura 1.7b. La red de LOJANet utiliza enlaces serie sin numerar basados en interfaces *loopback* que tienen las direcciones IP de la tabla 1.2.

[•] Referencia bibliográfica: "Arquitecturas MPLS y VPN" [1]

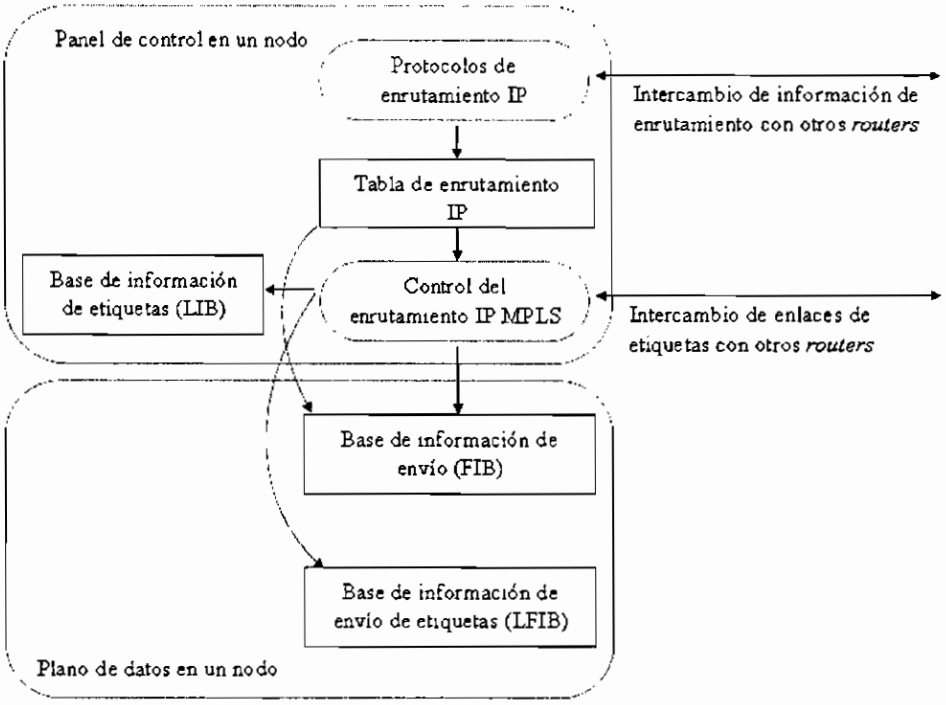


Figura 1.7a.: Arquitectura LSR de contorno [•]

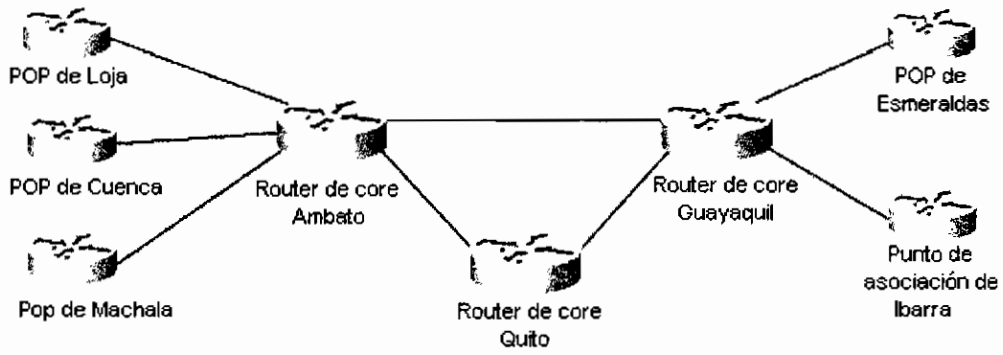


Figura 1.7b: Red del proveedor de servicios LOJANet

Router	loopback
Loja	172.16.1.1/32
Cuenca	172.16.1.2/32
Machala	172.16.1.3/32
Ambato	172.16.1.4/32
Quito	172.16.2.1/32
Guayaquil	172.16.3.1/32
Esmeraldas	172.16.3.2/32
Ibarra	172.16.4.1/32

Tabla 1.2: Direcciones loopback en la red LOJANet

[•] Referencia bibliográfica: "Arquitecturas MPLS y VPN" [1]

1.1.6.3.1 Funcionamiento del plano de datos en MPLS en modo trama

Cuando un paquete se propaga a través de un backbone MPLS realiza el siguiente proceso:

- El LSR de contorno de entrada recibe un paquete IP, Clasifica el paquete en una clase equivalente de envío (FEC), y etiqueta el paquete con la pila de etiquetas de salida correspondiente a la FEC. Para el enrutamiento IP unidifusión basado en el destino, la FEC corresponde a la subred de destino y la clasificación del paquete es una consulta tradicional de Capa 3 en la tabla de envío.

- Los LSR de core reciben sus paquetes etiquetados y emplean las tablas de envío para intercambiar etiquetas entrantes en el paquete entrante con la etiqueta saliente correspondiente a la misma FEC. (subred IP).

- Cuando el LSR de contorno de salida para esta FEC particular recibe el paquete etiquetado, le quita la etiqueta y realiza una consulta tradicional de Capa 3 en el paquete IP resultante.

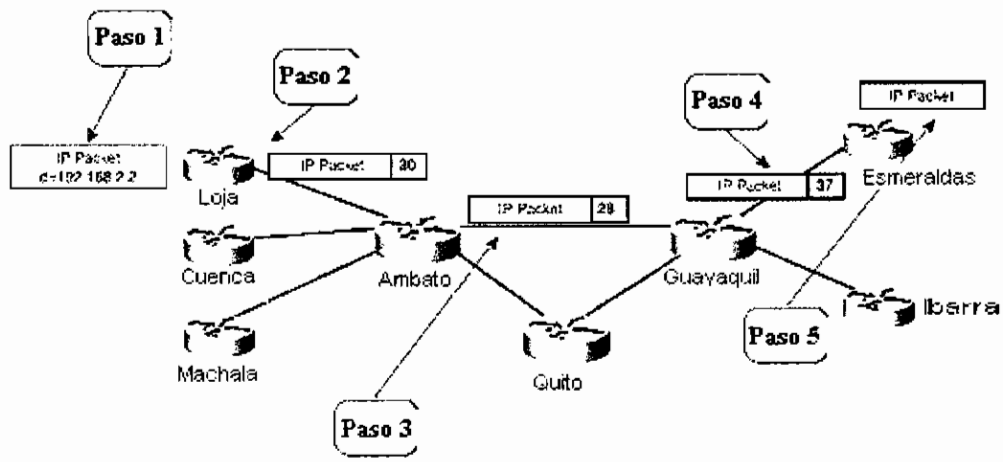


Figura 1.8: Envío de un paquete entre el POP de Loja y el cliente en Esmeraldas

En la figura 1.8 se muestra este proceso para la red LOJANet para un paquete que atraviesa la red desde el router Loja hacia un cliente conectado en el router Esmeraldas.

Paso 1: El paquete IP con la dirección de destino 192.168.2.2 llega al router de Loja.

Paso 2: El router de Loja hace una consulta de Capa 3 a través de la tabla de envío IP (denominada también Base de información de envío FIB), añade la etiqueta y envía el paquete hacia el router de Ambato.

Paso 3: EL router de Ambato consulta la etiqueta, intercambia las etiquetas y envía el paquete hacia el router de Guayaquil

Paso 4: El router de Guayaquil consulta la etiqueta, intercambia las etiquetas y envía el paquete hacia el router de Esmeraldas.

Paso 5: El router de Esmeraldas consulta la etiqueta, omite la etiqueta, hace la búsqueda de Capa 3 y envía el paquete hacia el router externo del siguiente salto.

1.1.6.3.2 Cabecera de la pila de etiquetas MPLS

Por motivo de brindar mejor rendimiento a la conmutación, la etiqueta MPLS se debe insertar por delante de los datos etiquetados en una implementación en modo trama de la arquitectura MPLS. La etiqueta MPLS se inserta así entre la cabecera de Capa 2 y los contenidos de la Capa 3 de la trama de Capa 2, como se muestra en la figura 1.9a.

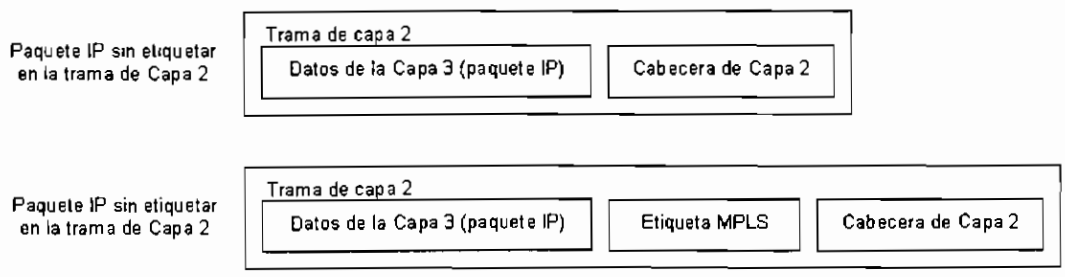


Figura 1.9a: Posición de la etiqueta MPLS en una trama de Capa 2. •

• Referencia bibliográfica: "Arquitecturas MPLS y VPN" [1]

La cabecera de la etiqueta MPLS se la denomina cabecera ***shim*** debido a la forma que es insertada entre el paquete de Capa 3 y la cabecera de Capa 2. La cabecera de la etiqueta MPLS está formada por la etiqueta (20 bits), la información de clase de servicio (3 bits, llamados también bits experimentales) y el campo de tiempo de existencia (TTL 8 bits, tiene funciones idénticas al campo IP TTL en cuanto a la detección de bucles se refiere) y un bit llamado parte inferior de la pila. La figura 1.9b muestra una cabecera de etiqueta MPLS.

MPLS permite que múltiples etiquetas (llamada una pila de etiquetas -label stack-) sean llevadas en un paquete. La pila de etiquetas habilita a los nodos MPLS para diferenciar entre diferentes tipos de flujos, para establecer y distribuir LSPs como consecuencia. Un uso común de la pila de etiquetas es el establecimiento de túneles a través de redes MPLS para aplicaciones VPN.

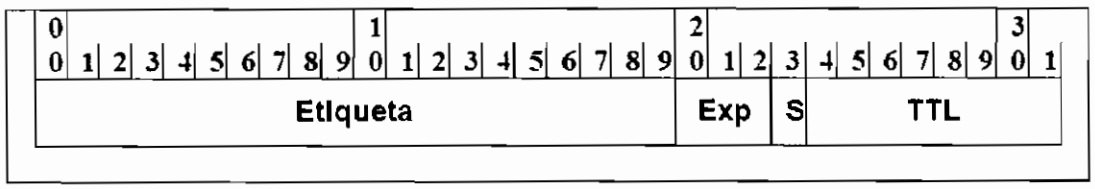


Figura 1.9b: Cabecera de pila de etiquetas MPLS. [•]

1.1.6.3.3 Conmutación de etiquetas en MPLS en modo trama

Un router Cisco que disponga del IOS adecuado y que está operando como un LSR MPLS en un dominio MPLS y en modo trama, puede realizar varias acciones sobre el paquete etiqueta. A continuación listamos estas acciones:

- **Acción *pop* de etiqueta (omisión de la etiqueta)**. Elimina la etiqueta superior de la pila de etiquetas MPLS y propaga la sobrecarga restante, ya sea como un paquete etiquetado (si el bit de la parte inferior de la pila es cero) o como un paquete IP sin etiquetar (el campo Tag Stack de la FIB está vacío.

[•] Referencia bibliográfica: "Arquitecturas MPLS y VPN" [1]

- **Intercambiar la etiqueta.** Sustituye la etiqueta superior de la pila de etiquetas MPLS por otro valor (el campo Tag Stack de la LIB es una etiqueta larga).

- **Acción *push* de etiqueta.** Sustituye la etiqueta superior de la pila de etiquetas MPLS por un conjunto de etiquetas (el campo Tag Stack de la LFIB contiene varias etiquetas).

- **Agregar.** Elimina la etiqueta superior de la pila de etiquetas MPLS y hace una búsqueda de Capa 3 en el paquete IP subyacente. La etiqueta eliminada es la etiqueta de la parte inferior de la pila de etiquetas MPLS; de lo contrario, se descarta el datagrama.

- **Desetiquetar.** Elimina la etiqueta superior de la pila de etiquetas MPLS y envía el paquete IP subyacente al siguiente salto IP especificado. La etiqueta eliminada es la etiqueta de la parte inferior de la pila de etiquetas MPLS; de lo contrario, se descarta el datagrama.

1.1.6.3.4 Señalización y distribución de etiquetas MPLS en modo trama

Hasta el momento hemos visto el mecanismo básico de envío de paquetes a través de los LSPs mediante el procedimiento de intercambio de etiquetas según las tablas de los LSRs. Pero queda revisar dos aspectos fundamentales:

- Cómo se generan las tablas de envío que establecen los LSPs.
- Cómo se distribuye la información sobre las etiquetas a los LSRs.

El primero de ellos está relacionado con la información que se tiene sobre la red: topología, patrón de tráfico, características de los enlaces, etc. Es la información de control típica de los algoritmos de encaminamiento. MPLS necesita esta información de *enrutamiento* para establecer los caminos virtuales LSPs. Lo más lógico es utilizar la propia información de enrutamiento que

manejan los protocolos internos IGP (OSPF, IS-IS, RIP...) para construir las tablas de enrutamiento (recuerde que los LSR son *routers* con funcionalidad añadida). Esto es lo que hace MPLS precisamente: para cada "ruta IP" en la red se crea un "camino de etiquetas" a base de concatenar las de entrada/salida en cada tabla de los LSRs; el protocolo interno correspondiente se encarga de pasar la información necesaria.

El segundo aspecto se refiere a la información de "señalización", pero siempre que se quiera establecer un circuito virtual se necesita algún tipo de señalización para marcar el camino, es decir, para la distribución de etiquetas entre los nodos. Sin embargo, la arquitectura MPLS no asume un único protocolo de distribución de etiquetas; de hecho se han estandarizando algunos existentes con las correspondientes extensiones; unos de ellos es el protocolo *Resource reSerVation Protocol with Tunneling Extensions (RSVP-TE)* del Modelo de Servicios Integrados del IETF. Pero, además, en el IETF se están definiendo otros nuevos, específicos para la distribución de etiquetas, cual es el caso del *Label Distribution Protocol* (LDP). Como alternativa, la asignación de etiquetas puede ser transportada sobre los protocolos de enrutamiento existentes tales como BGP.

El protocolo de señalización MPLS más usado es LDP. LDP define un conjunto de procedimientos usados por los ruteadores MPLS para intercambiar información de etiquetado y mapeo de flujo. Es también usado para señalar al router de contorno de una red MPLS (el punto donde el tráfico no MPLS ingresa.

RSVP-TE es también utilizado para la distribución de etiquetas, comúnmente en el núcleo de las redes que requieren ingeniería de tráfico y QoS. Un conjunto de extensiones fueron agregadas al protocolo RSVP original, RSVP-TE provee funcionalidad adicional más allá de la distribución de etiquetas, tales como enrutamiento LSP explícito, reenrutamiento dinámico alrededor de las fallas de la red, preferencia de LSPs, y detección de lazos. RSVP-TE puede distribuir los parámetros de ingeniería de tráfico tales como ancho de banda, reservaciones y requerimientos de QoS.

Las extensiones multi-protocolo han sido definidas por BGP, habilitando el protocolo para que sea también usado para distribuir etiquetas MPLS. Las etiquetas MPLS son llevadas sobre los mismos mensajes BGP usadas para distribuir las rutas asociadas.

La figura 1.10 expone una red MPLS en operación con un flujo de datos sobre la misma y el proceso de etiquetado realizado en los routers involucrados en la trayectoria que los paquetes deben atravesar para ir desde el origen hasta su destino.

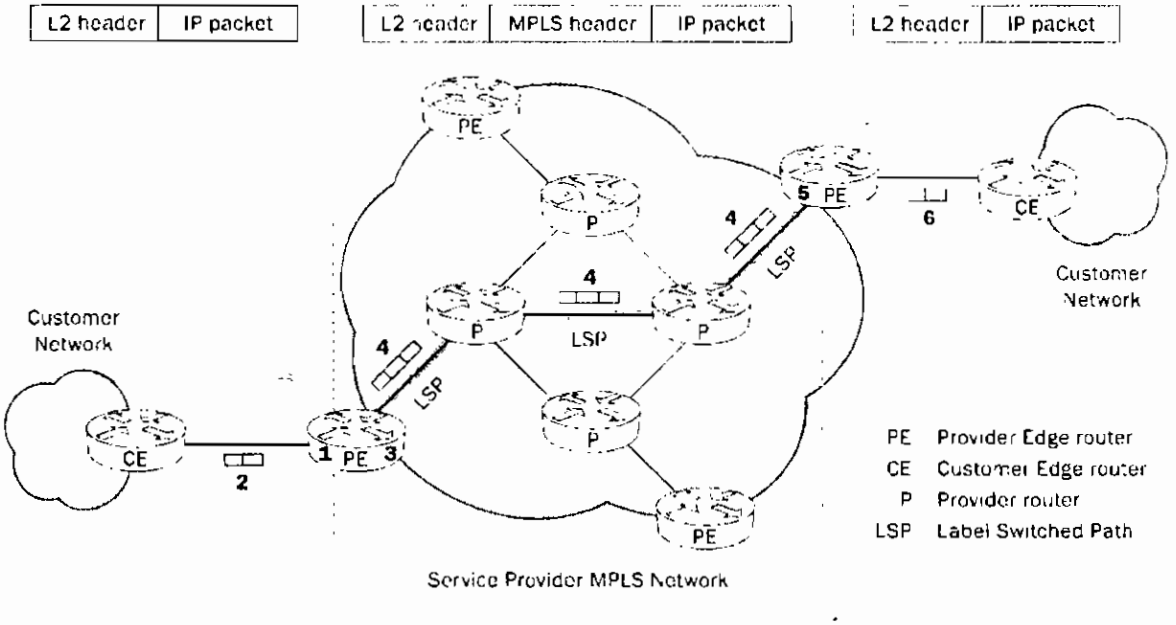


Figura 1.10: Red MPLS en operación *

1.1.6.4 Convergencia en una red MPLS

Un aspecto importante de las redes MPLS es el tiempo de convergencia de la red. Algunas aplicaciones MPLS (ejemplo: diseño MPLS/VPN o BGP basado en MPLS) no operan correctamente a menos que se pueda enviar un paquete etiquetado a lo largo de todo el trayecto, desde el LSR de contorno de entrada hasta el LSR de contorno de salida. En estas aplicaciones, el tiempo de

* Referencia bibliográfica: "MPLS Conformance and Performance Testing" [2]

convergencia necesario para el Protocolo de Gateway Interior (IGP) para converger en torno a un fallo en la red principal puede aumentarse retrasando la propagación de la etiqueta.

En una red MPLS en modo trama, si se utiliza el modo de retención liberal, en combinación con el control de etiqueta independiente y la distribución de etiquetas de flujo descendente no solicitada, se minimiza el retraso de convergencia TDP/LDP. Cada router que emplea el modo de retención liberal, normalmente tiene asignaciones de etiqueta para un prefijo dado desde todos sus vecinos TDP/LDP, de manera que siempre puedan encontrar una etiqueta de salida apropiada siguiendo la convergencia de la tabla de enrutamiento sin preguntar a su nuevo router de siguiente salto acerca de la asignación de etiqueta.

1.1.6.5 Interacción de MPLS con el Protocolo de Gateway Fronterizo

En la tabla de enrutamiento IP de un router que actúa como LSR se asigna una etiqueta a cada prefijo IP, siendo la única excepción las rutas aprendidas a través del Protocolo de gateway fronterizo (BGP). A estas rutas no se les asignan etiquetas y el LSR de contorno de entrada utiliza la etiqueta asignada al siguiente salto BGP para etiquetar los paquetes enviados hacia los destinos BGP.

La interacción entre MPLS, IGP y BGP brinda al diseñador de redes una perspectiva completamente nueva del diseño de redes. De acuerdo con el funcionamiento tradicional, BGP se tiene que ejecutar en cada router del núcleo de la red del proveedor de servicios para permitir el envío correcto de paquetes. Por ejemplo, la información BGP del router de Ibarra tiene que propagarse hacia todos los routers de core de la red de LOJANet (Quito, Guayaquil y Ambato). En caso de no ser esto posible, los routers de core no podrían enrutar los paquetes hacia el destino BGP.

Si a pesar de todo esto, la red de LOJANet ejecuta MPLS, el router de Loja propaga el paquete hacia un destino BGP como un paquete etiquetado con la etiqueta asociada con el siguiente salto BGP. Debido a que el siguiente salto BGP debe anunciarse en el IGP de la red que lo está ejecutando, todos los routers intermedios ya deben tener una asignación de etiqueta entrante a saliente para ese destino en su LFIB y deben propagar el paquete etiquetado hacia el LSR de salida (Ibarra), pero no necesitan ejecutar BGP.

Las ventajas de eliminar el proceso BGP de los routers principales (o de core) de la red de un proveedor de servicios son las siguientes:

- Debido a que los routers principales no procesan las fluctuaciones de las rutas en Internet, las tablas de enrutamiento de los routers principales se hacen mucho más estables.

- Al no tener que almacenar las rutas de Internet el ahorro de memoria en los routers principales es notoria ya que almacenar de 70.000 a 80.000 rutas requieren de aproximadamente entre 20 a 40 MB de memoria.

- Debido a que los routers principales no tienen que procesar actualizaciones BGP se reduce enormemente la utilización de la CPU de los mismos.

Es sumamente recomendable el despliegue de MPLS, aún cuando el backbone IP del proveedor de servicios esté formado estrictamente por routers.

1.1.7 APLICACIONES DE MPLS

Como se ha visto hasta ahora, la arquitectura MPLS permite una perfecta integración de los switches ATM y routers tradicionales en un backbone IP. No obstante, lo que hace que MPLS se convierta en una arquitectura fuerte son sus aplicaciones, desde ingeniería de tráfico hasta redes privadas virtuales igual a

igual. Todas las aplicaciones de MPLS usan la funcionalidad del plano de control, similar al plano de control de enrutamiento IP para establecer la base de datos de conmutación de etiquetas. La figura 1.11 expone la interacción existente entre estas aplicaciones y la matriz de conmutación de etiquetas.

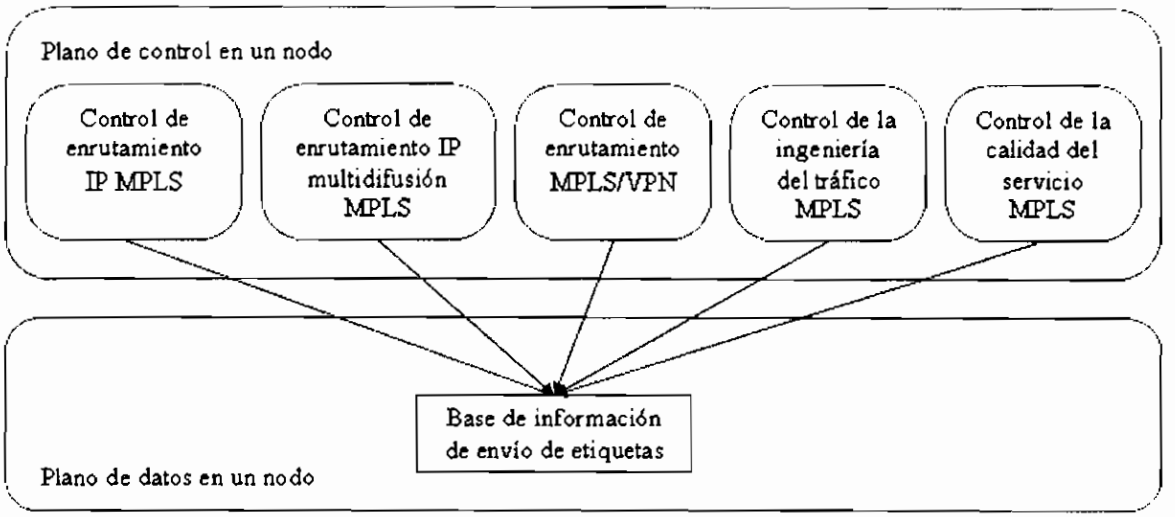


Figura 1.11: Aplicaciones MPLS y sus interacciones [•]

Cada aplicación MPLS tiene el mismo conjunto de componentes que la aplicación de enrutamiento IP:

- Una base de datos que define la tabla de clases equivalentes de envío (FEC) para la aplicación (la tabla de enrutamiento IP en una aplicación de enrutamiento IP).

- Los protocolos de control que intercambian los contenidos de la tabla FEC entre los LSR (protocolos de enrutamiento IP o enrutamiento estático en una aplicación de enrutamiento IP).

- El proceso de control que realiza el enlace de etiquetas a las FEC y un protocolo para intercambiar los enlaces de etiquetas entre los LSR (TDP o LDP en una aplicación de enrutamiento IP).

[•] Referencia bibliográfica: "Arquitecturas MPLS y VPN" [1]

- Opcionalmente, una base de datos interna de asignación de etiquetas a las FEC (base de información de etiquetas en una aplicación de enrutamiento IP).

Para realizar el proceso de intercambio de la tabla de FEC o la asignación FEC-a-etiqueta entre nodos, cada aplicación tiene su propio conjunto de protocolos. La tabla 1.3 resume los protocolos y las estructuras de datos.

Aplicación	Tabla FEC	Protocolo de control utilizado para construir la tabla FEC	Protocolo de control utilizado para intercambiar asignación FEC-a-etiqueta
Enrutamiento IP	Tabla de enrutamiento IP	Cualquier protocolo de enrutamiento IP	**Protocolo de distribución de etiquetas (TDP o LDP)**
Enrutamiento multidifusión IP	Tabla de enrutamiento multidifusión	PIM	**Extensiones PIM versión 2**
Enrutamiento VPN	Tabla de enrutamiento por cada VPN	La mayoría de los protocolos de enrutamiento IP entre el proveedor de servicios y el cliente, BGP multiprotocolo en la red del proveedor de servicios	**BGP multiprotocolo**
Ingeniería de tráfico	Definición de túneles MPLS	Definiciones de interfaz manual, extensiones a IS-IS u OSPF	**RSVP o CR-LDP**
Calidad del Servicio MPLS	Tabla de enrutamiento IP	Protocolos de enrutamiento IP	**Extensiones a TDP LDP**

Tabla 1.3: Aplicaciones MPLS y los protocolos utilizados [•]

[•] Referencia bibliográfica: "Arquitecturas MPLS y VPN" [1]

1.2 REDES PRIVADAS VIRTUALES (VPN) *

La implementación inicial de una red de computadoras brindaba una buena seguridad a sus clientes, pero no era buena económicamente hablando debido a:

- El tráfico de una red varía dependiendo de la hora, del día del mes, tipo de empresa entre otros.

- Los usuarios finales necesitan respuestas rápidas, lo que implica mayor ancho de banda que en ciertas ocasiones será desperdiciado, por ejemplo cuando los usuarios estén sin actividad.

Estas dos razones forzaron a la industria de comunicación de datos y a los ISPs a desarrollar e implementar un esquema de multiplexación estadístico que proporcionara un servicio parecido al de las líneas dedicadas. Sin embargo, este servicio era más económico debido a los beneficios que el proveedor podía alcanzar al brindar un servicio estadístico a una mayor cantidad de clientes. Las primeras VPNs se basaban en tecnologías como X.25 y Frame Relay, y, posteriormente, en ATM.

Una Red Privada Virtual (VPN) es un servicio de red privada sobre una red pública compartida. Las VPNs benefician a los clientes permitiéndoles conectarse a locaciones remotas de manera segura sobre una red pública, sin necesidad de comprar o arrendar un enlace dedicado. MPLS habilita las VPNs proveyendo un circuito orientado a conexión y permitiendo a los proveedores de servicio desarrollar VPNs sobre la infraestructura de redes IP sin conexión tradicionales.

Las VPNs basadas en IP pueden extender las intranets sobre enlaces WAN a oficinas remotas o usuarios móviles. Estas pueden soportar extranets enlazando socios de negocios, clientes y proveedores para brindar una mejor satisfacción al cliente y reducir los costos de producción. Las VPNs pueden

* Referencia bibliográfica: "MPLS Conformance and Performance Testing" [2]

también conectar comunidades con los mismos intereses, proveyendo un forum seguro para la discusión de tópicos comunes.

Los principales componentes de una red VPN global son:

- El propietario de la infraestructura (equipos y medios de transmisión) es el proveedor de servicios que ofrece líneas dedicadas emuladas a sus clientes. El servicio ofrecido al cliente es el de Red Privada Virtual.

- El cliente se conecta al proveedor por medio de un dispositivo denominado CPE (*Equipo Terminal del Cliente*). Por lo general el CPE es un dispositivo ensamblador/desensamblador de paquetes (PAD) que proporciona total conectividad de Terminal, un bridge o un router. El CPE se denomina también dispositivo *límite del cliente o contorno del cliente (CE Customer Edge).*

- El dispositivo CPE es conectado a través de los medios de transmisión (dial-up o comúnmente una línea dedicada) al equipo del proveedor de servicios, que pueden ser ATM, X.25, Frame Relay o incluso un Router IP. Este dispositivo es llamado *límite del proveedor o contorno del proveedor (PE Provider Edge)*

- El proveedor de servicio usualmente tiene equipamiento en el núcleo de su red llamado red P. Estos dispositivos son llamados dispositivos P por ejemplo switches P o routers P.

- La parte contigua a la red del cliente es llamada un sitio, un sitio puede conectarse a la red P por medio de una o varias líneas de transmisión, usando uno o varios dispositivos CPE y PE, basándose en los requerimientos de redundancia.

- La línea dedicada emulada proporcionada al cliente por el proveedor de servicio en el modelo *VPN overlay*, a menudo es llamado *Circuito Virtual*

VC. El VC puede estar constantemente disponible (*Circuito Virtual Permanente PVC)* o se puede establecer bajo demanda (*Circuito Virtual Conmutado SVC)*. Algunas tecnologías utilizan términos especiales para los VCs como *Identificador de Conexión de Enlace de Datos (DLCI)* en Frame Relay.

- El proveedor de servicios puede cargar o una tarifa plana para el servicio VPN, el cual normalmente depende del ancho de banda disponible para el cliente, o una tarifa basada en el uso, lo cual puede depender del volumen de datos intercambiados o la duración del proceso de intercambio de datos.

La figura 1.12 expone una solución VPN global con sus principales componentes.

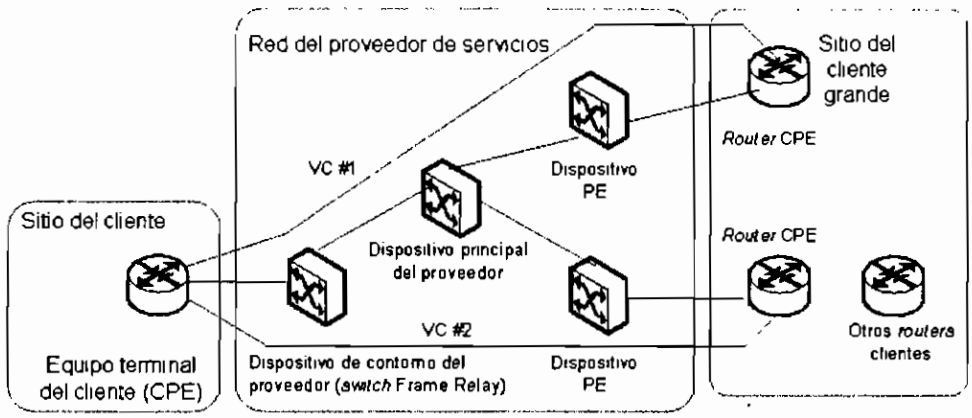


Figura 1.12: Red Frame Relay común [•]

Las VPNs tradicionales se han venido construyendo sobre infraestructuras de transmisión compartidas con características implícitas de seguridad y respuesta predeterminada. Tal es el caso de las redes de datos Frame Relay, que permiten establecer PVCs entre los diversos nodos que conforman la VPN. La seguridad y las garantías las proporcionan la separación de tráficos por PVC y el caudal asegurado (CIR). Algo similar se puede hacer con ATM, con diversas clases de garantías. Los inconvenientes de este tipo de solución es que la configuración de las rutas se basa en procedimientos más bien artesanales, al

[•] Referencia bibliográfica: "Arquitecturas MPLS y VPN" [1]

tener que establecer cada PVC entre nodos, con la complejidad que esto supone al proveedor en la gestión (y los mayores costes asociados). Si se quiere tener conectados a todos con todos, en una topología lógica totalmente mallada, añadir un nuevo emplazamiento supone retocar todos los CPEs (equipo terminal del cliente) del cliente y restablecer todos los PVCs.

Además, la popularización de las aplicaciones TCP/IP, así como la expansión de las redes de los proveedores de servicios de Red (NSPs), ha llevado a tratar de utilizar estas infraestructuras IP para el soporte de VPNs, tratando de conseguir una mayor flexibilidad en el diseño e implantación y unos menores costes de gestión y provisión de servicio. La forma de utilizar las infraestructuras IP para servicio VPN (IP VPN) ha sido la de construir túneles IP de diversos modos.

1.2.1 REDES PRIVADAS VIRTUALES MODERNAS [•]

Debido a la gran variedad de tecnología, proveedores de servicios, fabricantes y nuevos requisitos de los clientes, el concepto VPN se hace cada vez más complicado. Los fabricantes han introducido nuevos términos, y con frecuencia confusos, que aumenta aún más la complejidad. Para afrontar esta variedad de tecnologías y topologías de VPNs las clasificaremos tomando en cuenta los siguientes 6 criterios:

- ***El problema comercial que la VPN va a solucionar.-*** Las principales clases de problemas comerciales son la comunicación interna de la empresa llamada ***intranet***, la comunicación entre empresas llamada ***extranet*** y el acceso para usuarios móviles llamada ***red de marcación privada virtual*** o ***VPDN.***

[•] Referencia bibliográfica: "Arquitecturas MPLS y VPN" [1]

- ***La capa OSI*** en la que el proveedor de servicios intercambia la información de topología con el cliente. Aquí, las principales categorías son el ***modelo overlay***, donde el proveedor de servicios ofrece al cliente sólo un grupo de enlaces punto a punto o multipunto entre los sitios del cliente, y el ***modelo igual a igual,*** donde el proveedor de servicios y el cliente intercambian información de enrutamiento de Capa 3.

- ***La tecnología de Capa 2 o Capa 3*** empleada para implementar el servicio VPN dentro de la red del proveedor de servicios, que puede ser ***X.25, Frame Relay, ATM*** o ***IP.***

- ***La topología de la red,*** que puede ir desde una simple topología ***radial*** hasta topologías de redes en ***malla completa*** y ***jerárquicas*** de varios niveles en redes mayores.

- ***Basadas en el Cliente:*** la VPN es configurada exclusivamente en el equipo localizado en el lado del cliente y utiliza protocolos de tunelamiento a lo largo de la red pública, el protocolo comúnmente usado es IPSec.

- ***Basadas en la Red:*** La red es configurada en el equipo del proveedor de servicios y administrada por el proveedor. Las VPNs-MPLS son un ejemplo de VPNs basada en Red.

1.2.2 REDES PRIVADAS VIRTUALES BASADAS EN MPLS [•]

Las nuevas tecnologías y productos permiten una implementación más confiable, escalable y de un costo más económico del mismo producto. Todo esto asociado a las nuevas tecnologías VPN, no es sorprendente que los servicios

[•] Referencia bibliográfica: "Cisco MPLS Controller Software Configuration Guide" [3]

VPN estén entre los principales motivos por los cuales la tecnología MPLS se ha esparcido en redes de proveedores de servicios y empresas.

VPNs basadas en MPLS vs VPNs basadas en IPSec [•]

IPSec agrega encriptación segura para IP. Esto es típicamente administrado por el cliente final, fuera de la red del proveedor de servicios, donde hay un alto grado de vulnerabilidad a la privacidad de los datos. En las VPNs basadas en túneles IPSec, la seguridad requerida se garantiza mediante el cifrado de la información de los datos y de la cabecera de los paquetes IP, que se encapsulan con una nueva cabecera IP para su transporte por la red del proveedor. Es relativamente sencillo de implementar, bien sea en dispositivos especializados, tales como cortafuegos, como en los propios *routers* de acceso del NSP. Además, como es un estándar, IPSec permite crear VPNs a través de redes de distintos NSPs que sigan el estándar IPSec. Pero como el cifrado IPSec oculta las cabeceras de los paquetes originales, las opciones QoS son bastante limitadas, ya que la red no puede distinguir flujos por aplicaciones para asignarles diferentes niveles de servicio. Además, sólo vale para paquetes IP nativos, IPSec no admite otros protocolos.

Las VPNs-MPLS son albergadas en el equipo del NSP, lo cual puede proveer un significante ahorro económico y un incremento en la escalabilidad comparada con otras tecnologías VPN. Las VPNs-MPLS mantienen separado el tráfico de diferentes clientes únicamente identificando cada flujo VPN y levantando conexiones como circuitos. Este mecanismo provee separación de tráfico y es transparente para el usuario final dentro del grupo VPN. Las VPNs-MPLS proveen seguridad intrínsicamente, esencialmente haciendo a IP tan seguro como Frame Relay o ATM, y reduce la necesidad de encripción.

> "Miercom, una consultora y laboratorio de pruebas de redes independiente, probó la seguridad de una red VPN-MPLS en una red de varios ruteadores, y

[•] Referencia bibliográfica: "MPLS Conformance and Performance Testing" [2]

concluyó (2001): "Nuestros resultados de las pruebas han demostrado que la redes VPN basadas en MPLS ofrecen el mismo nivel de seguridad que una red ATM o Frame Relay."

Tomado de: IXIA "Conformance and Performance Testing"

VPNs de Capa 3

Las VPNs-MPLS están dentro del grupo de VPNs basadas en la Red, aquellas que operan en la Capa 2 y Capa 3. Las VPNs de Capa 3 fueron las primeras en ser investigadas y estandarizadas en RFCs. Las VPNs de Capa 3 basadas en el RFC 2547bis han sido las más ampliamente desarrolladas hasta la fecha y utilizan extensiones a BGP, específicamente el Multi-Protocolo interno BGP (MP-iBGP), para distribuir la información de enrutamiento VPN a través del backbone del proveedor. Los mecanismos MPLS estandarizados son usados para enviar el tráfico VPN a través del backbone. En una VPN L3, los routers CE y PE están enrutando IP peer-to-peer. EL router CE provee al router PE con información de enrutamiento para la red privada del cliente detrás de ésta. El router PE almacena su información de enrutamiento privada en una Tabla de Enrutamiento y Envío Virtual (VRF); cada VRF es en esencia una red privada IP. El router PE mantiene una tabla VRF separada por cada VPN, de tal modo provee seguridad y aislamiento apropiado. Los usuarios de la VPN tienen acceso sólo a los sitios o hosts dentro de la misma VPN. Además para las tablas VRF, el router PE también almacena la información de enrutamiento normal que es necesaria para enviar tráfico sobre la Internet pública.

Las VPNs L3 utilizan una pila de etiquetas de nivel dos (ver figura 1.13). La etiqueta entrante lleva información específica de la VPN de PE a PE. La etiqueta saliente lleva la información de envío MPLS de salto a salto. Los routers P en la red MPLS solo leen e intercambian la etiqueta saliente en cuanto el paquete pasa a través de red. Ellos no leen o actúan sobre la etiqueta VPN entrante, esa información es tunelada a través de la red.

EL acercamiento a VPN L3 tiene varias ventajas. El espacio de direccionamiento IP del cliente es administrado por el proveedor, simplificando significativamente el papel del IT del cliente, por ejemplo, nuevos sitios VPN del cliente son fácilmente conectados y administrados por el proveedor.

El acercamiento a Capa 3 también tiene desventajas. Las VPNs de Capa 3 soportan sólo tráfico del cliente IP o encapsulado IP. El escalamiento puede ser un problema significante con los routers PE requeridos para soportar tablas de enrutamiento que son más grandes que las normales con la agregación de rutas VPN.

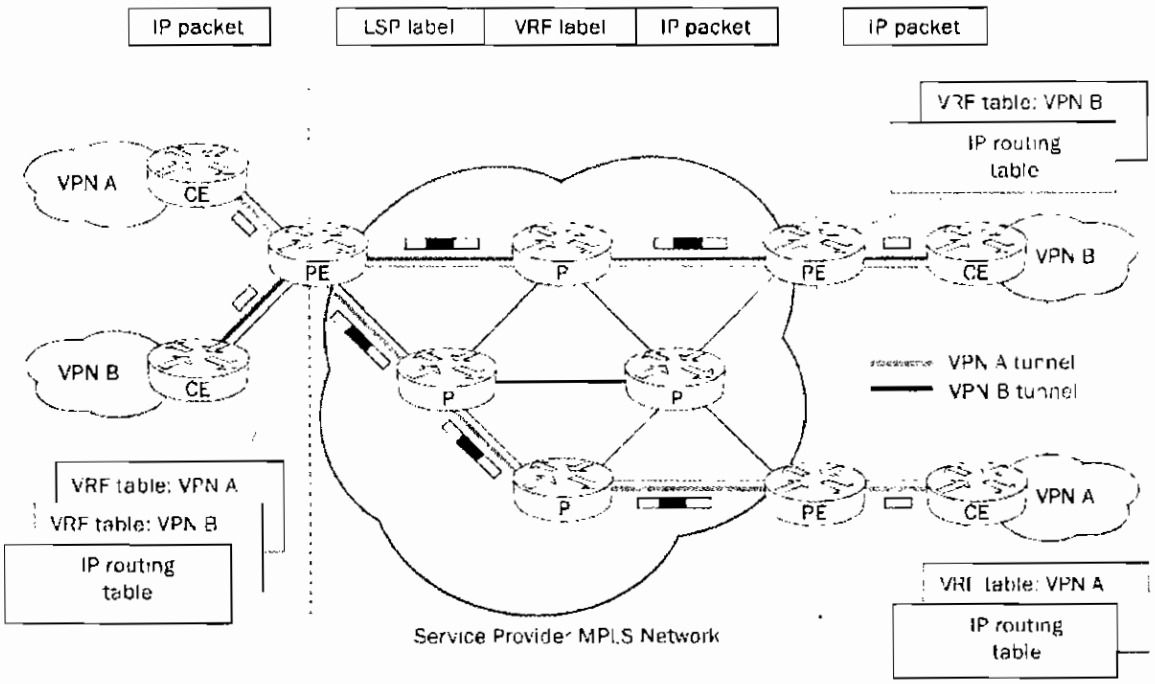


Figura 1.13: Red VPN-MPLS de Capa 3 [•]

VPNs de Capa 2

Las VPNs-MPLS de Capa 2 generaron un interés de los proveedores y vendedores al punto que comenzaron a desarrollarlas (2003). La industria se ha centralizado en los borradores Martini de la IETF, llamados así por su autor Luca Martini. Estos borradores definen un método para levantar túneles VPN L2 a

[•] Referencia bibliográfica: "MPLS Conformance and Performance Testing" [2]

través de una red MPLS que pueda manejar todo tipo de tráfico de Capa 2, incluyendo Ethernet, Frame Relay, ATM, TDM y PPP/HDLC.

Existen dos clases de VPNs L2 que usan la metodología Martini:

- Point-to-Point: similar a ATM y Frame Relay usando conexiones point-to-point (LSPs) fijas a través de la red.

- Multi-point: soportando topologías malla y jerárquica.

VPLS (Virtual Private LAN Services) es un modelo VPN L2 multi-point que ha generado gran interés. VPLS utiliza Ethernet como tecnología de acceso entre el cliente y el proveedor de red y habilita una red Ethernet corporativa privada para ser extendida sobre la infraestructura MPLS administrada por el proveedor

1.2.3 CARACTERÍSTICAS DE REDES VPN-MPLS [•]

A continuación se listan las principales características de una red VPN-MPLS:

Desempeño.- Cuando las VPNs-MPLS son configuradas usando LSRs ATM, las capacidades del servicio IP no orientado a conexión escalables son combinadas con las capacidades de desempeño y administración de tráfico de ATM.

Servicio no orientado a conexión.- Una ventaja técnica significante de las VPNs-MPLS es el servicio no orientado a conexión. El Internet debe su éxito a su tecnología básica, TCP/IP. Esto significa que no es necesaria una acción previa para establecer la comunicación entre hosts, haciendo fácil la comunicación entre hosts.

[•] Referencia bibliográfica: "Cisco MPLS Controller Software Configuration Guide" [3]

Para establecer privacidad en un ambiente IP no orientado a conexión, la actual solución VPN es un modelo overlay point-to-point orientado a conexión. Incluso si está corriendo sobre una red no orientada a conexión, las VPNs actuales no pueden tomar ventaja de la facilidad de la conectividad y múltiples servicios disponibles en las redes no orientadas a conexión.

Debido a la creación de VPN-MPLS no orientadas a conexión, no se requieren tunelamiento y encriptación para brindar una red con privacidad, de esta manera se elimina significantemente la complejidad de configuración de la red.

Servicio Centralizado.- La construcción de VPNs de Capa 3 tiene la ventaja adicional de permitir la entrega de servicios dirigidos a un grupo de usuarios representados por una VPN.

Una VPN debe dar al proveedor de servicios más de un mecanismo, para privadamente, permitir a los usuarios conectarse a los servicio de intranet. Esta debe también proveer una manera flexible de entregar servicios de valor agregado a cierto grupo de clientes. La escalabilidad es crítica, porque los clientes quieren utilizar los servicios privados no solo en sus intranets sino también en sus extranets.

Debido a que las VPNs-MPLS son vistas como intranets privadas, facilita el levantamiento de nuevos servicios IP como:

- Multicast
- Calidad de Servicio
- Soporte de Telefonía dentro de una VPN
- Servicios Centralizados tales como contenido y hospedaje Web a una VPN.

Escalabilidad.- La escalabilidad es la mayor deficiencia de las VPNs creadas usando enlaces orientados a conexión, point-to-point overlay, Frame Relay, o VCs ATM. Específicamente, las VPNs orientadas a conexión requieren una malla full N^2 (full N^2 mesh) de conexiones entre las localidades del cliente para dar soporte a una comunicación any-to-any (todos contra todos).

En lugar de eso las VPNs-MPLS usan el modelo peer to peer (igual a igual) y la arquitectura de Capa 3 no orientada a conexión para impulsar una solución VPN altamente escalable. El modelo igual a igual requiere un punto del cliente para hacer una conexión peer con un solo router PE contrariamente a todos los otros router CPE o CE que son miembros de la VPN. La arquitectura no orientada a conexión permite la creación de VPNs en la Capa 3, eliminando la necesidad de túneles o VCs.

Seguridad.- Las redes VPNs-MPLS proveen el mismo nivel de seguridad que una VPN orientada a conexión. Los paquetes desde una VPN no irán inadvertidamente hacia otra VPN. La seguridad es suministrada en el contorno y núcleo de la red del proveedor:

- En el contorno, la seguridad garantiza que los paquetes recibidos de un cliente son puestos en la VPN correcta.
- En el backbone, el tráfico de la VPN se mantiene separado.

El spoofing malicioso de un router PE es casi imposible debido a que los paquetes recibidos son paquetes IP. Estos paquetes IP deben ser recibidos en una interfaz particular o subinterfaz para ser únicamente identificados con la etiqueta de la VPN.

Fácil de crear.- Para tomar completa ventaja de la VPNs, debe ser fácil crear nuevas VPNs y comunidades de usuarios. Debido a que las VPNs-MPLS son no orientadas a conexión, no se requieren mapas o topologías de conexiones punto a punto. Esto facilita para agregar sitios a intranets y

extranets y formar grupos de usuarios cercanos. Administrando las VPNs de esta manera habilita un miembro de cualquier sitio dado en múltiples VPNs, maximizando la flexibilidad en intranets y extranets ya construidas.

Direccionamiento flexible.- Para hacer un servicio VPN más flexible, los usuarios deben ser capaces de diseñar su propio plan de direccionamiento, independientemente del plan de direccionamiento de otra VPN del proveedor de servicios.

Muchas organizaciones utilizan direcciones privadas, como es definido actualmente en el RFC 1918, y no desean sobrellevar el tiempo y dinero que implica la implementación de direcciones IP registradas para habilitar la conectividad de la intranet. Las VPNs-MPLS permiten a los clientes continuar utilizando su actual esquema de direccionamiento sin NAT (Network Address Translation) proveyendo una visión pública y privada de las direcciones. Si dos VPNs quieren comunicar y ambas tienen direcciones sobrelapadas, esta comunicación requiere NAT en un punto final. Esto permite a los clientes usar su propio esquema de direccionamiento privado no registrado y comunicarse libremente a través de la red IP pública.

Soporte de Clase de Servicio (CoS) Integrado.- CoS es un ingrediente esencial de una VPN IP porque esto provee la capacidad para proveer dos requerimientos fundamentales de las VPNs.

- Desempeño confiable y política de implementación
- Soporte para múltiples Clases de Servicios en una VPN-MPLS

El tráfico de la red es clasificado y etiquetado en el contorno de la red antes de que sea enrutado de acuerdo a las políticas definidas por los suscriptores e implementado por el proveedor y transportado por el núcleo del proveedor. El tráfico en el contorno y núcleo de la red puede entonces ser diferenciado en diferentes clases por probabilidad de ser desechados o por retardo.

Migración Directa.- Para un despliegue rápido de los servicios VPN los proveedores de servicio requieren un camino de migración directo. Las VPNs-MPLS son las únicas que ofrecen ese camino, porque pueden ser construidas sobre múltiples arquitecturas de red, incluyendo, IP, ATM, Frame Relay y redes híbridas.

La migración para el cliente final es también simplificada porque no se requiere que el router CE (router en el contorno del cliente) soporte MPLS ni se necesitan realizar modificaciones en la intranet del cliente.

1.2.4 FUNCIONAMIENTO DE LA ARQUITECTURA VPN/MPLS

Es importante destacar que en el borde de la nube MPLS tenemos una red convencional de routers IP. El núcleo MPLS proporciona una arquitectura de transporte que hace aparecer a cada par de routers a una distancia de un sólo salto. Funcionalmente es como si estuvieran unidos mediante una topología mallada, esa unión a un solo salto se realiza por medio de los LSPs de la arquitectura MPLS (puede haber más de un LSP por cada par de routers). La diferencia con topologías conectivas reales es que en MPLS la construcción de caminos virtuales es mucho más flexible y que no se pierde la visibilidad sobre paquetes IP.

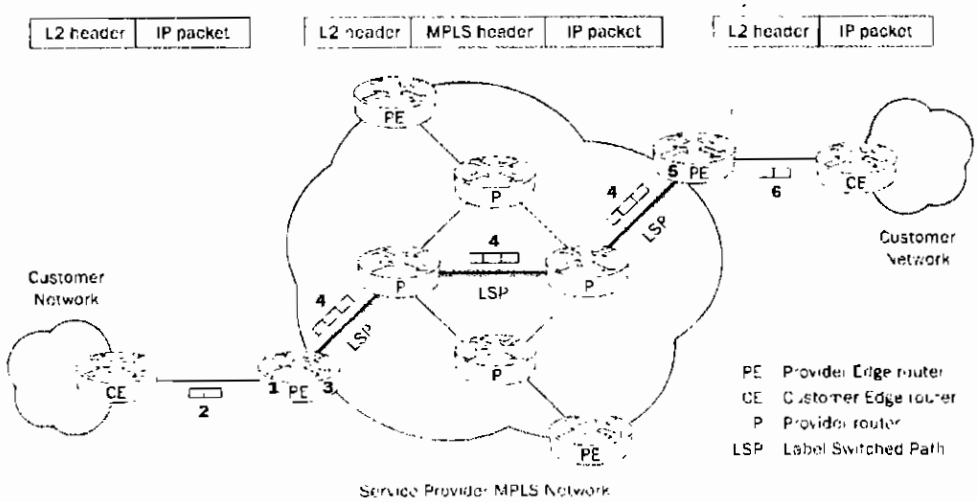


Figura 1.14.: Funcionamiento de una Red MPLS *

* Referencia bibliográfica: "MPLS Conformance and Performance Testing" [2]

En la arquitectura VPN-MPLS, un único router es dividido en muchos routers virtuales para proveer funcionabilidad VPN-MPLS. Virtual Routing and Forwarding (VRFs) son usadas para crear routers virtuales. Cada router virtual contiene sus propias tablas de enrutamiento, CEF caches e interfaces.

Un RD (Router Distinguisher) es usado para diferenciar entre redes IPv4 pertenecientes a diferentes VPNs. El único propósito del RD es proveer direcciones únicas VPNv4.

El intercambio de información topológica entre routers CE requiere el envío de actualizaciones utilizando uno de los protocolos de enrutamiento soportados entre los routers PE y CE (por ejemplo: enrutamiento estático, eBGP, RIPv2, OSPF). Un router PE recibiendo una actualización desde el router CE tiene que redistribuir la información dentro de BGP. La información es entonces trasladada dentro del formato MP-BGP. Sobre este formato es agregado un RT (Router Target). La información VPNv4 se enviará a otros routers PE donde será llevada en VRFs que están usando los mismos RTs. Entonces los routers PE remotos redistribuirán la información actualizada en formato IGP y la envía a los routers CE.

1.3 CALIDAD DE SERVICIO (QOS) EN REDES MPLS

Uno de los defectos de las redes basadas en IP, comparada con redes Frame Relay y ATM, ha sido su inhabilidad para proveer servicios garantizados para tráfico que transportan, por ejemplo, tráfico de tiempo real como voz o video necesitan alta calidad de servicio (baja latencia, bajo jitter, etc.) para atravesar exitosamente una red. Similarmente, datos con una importancia crítica, tales como transacciones comerciales electrónicas, deben tener prioridad sobre el tráfico web normal.

La arquitectura MPLS provee el entramado necesario para brindar garantías de calidad al tráfico IP. A pesar de que QoS y Clase de Servicio (CoS) no son características fundamentales de MPLS, estas pueden ser aplicadas en redes MPLS.

MPLS está diseñado para poder cursar servicios diferenciados, según el Modelo DiffServ del IETF. Este modelo define una variedad de mecanismos para poder clasificar el tráfico en un reducido número de clases de servicio, con diferentes prioridades. Según los requisitos de los usuarios, DiffServ permite diferenciar servicios tradicionales tales como el WWW, el correo electrónico o la transferencia de ficheros (para los que el retardo no es crítico), de otras aplicaciones mucho más dependientes del retardo y de la variación del mismo (voz, video). Para ello se emplea el campo ToS (*Type of Service*), rebautizado en DiffServ como el octeto DS. Esta es la técnica QoS de marcar los paquetes que se envían a la red.

MPLS se adapta perfectamente a ese modelo, ya que las etiquetas MPLS tienen el campo EXP para poder propagar la clase de servicio en el correspondiente LSP.

De este modo, una red MPLS puede transportar distintas clases de tráfico, ya que:

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

- Entre cada par de LSR exteriores se pueden proveer múltiples LSPs, cada uno de ellos con distintas prestaciones y con diferentes garantías de ancho de banda.

  Por ejemplo un LSP puede ser para tráfico de máxima prioridad, otro para una prioridad media y un tercero para tráfico *best-effort*, tres niveles de servicio, primera, preferente y turista, que, lógicamente, tendrán distintos precios.

La característica CoS de MPLS puede ser usada opcionalmente con redes VPN-MPLS.

1.3.1 FUNCIONAMIENTO [•]

Cuando un cliente transmite paquetes de un sitio a otro, el campo de precedencia IP (los tres primeros bits del campo DSCP de la cabecera IP) especifica la clase de servicio (CoS). El paquete recibe un tratamiento dependiendo de la CoS contenida en su precedencia IP, esto puede afectar de diferentes maneras como latencia o porcentaje de ancho de banda. Si la red del proveedor de servicios es una red MPLS entonces los bits de precedencia IP son copiados dentro del campo EXP de la etiqueta MPLS en el borde de la red. Sin embargo, el proveedor de servicios podría querer configurar una CoS de un paquete MPLS a un diferente valor determinado por el servicio ofrecido.

Esta característica permite a los proveedores de servicio configurar el campo EXP MPLS en lugar de sobrescribir el valor en el campo DSCP de la precedencia IP del paquete del cliente. La cabecera IP permanece disponible para

[•] Referencia bibliográfica: "Cisco MPLS Class of Service Enhancements" [4]

ser usado por el cliente; la CoS del paquete IP no es cambiada mientras el paquete viaja a través de una red MPLS.

1.3.2 BENEFICIOS DE MPLS CoS EN UN BACKBONE IP *

Básicamente, los beneficios de usar un backbone formado por routers IP y que estén configurados con MPLS son:

- Asignación eficiente de los recursos.- WFQ es utilizado para asignar ancho de banda de tal modo que garantiza un porcentaje de ancho de banda para el tráfico de la red.
- Diferenciación de paquetes.- Cuando los paquetes IP atraviesan una red MPLS, los paquetes son diferenciados mapeando los bits de precedencia IP de los paquetes IP con los bits CoS del campo EXP de los paquetes MPLS. Este mapeo de bits permite al proveedor de servicios mantener la garantías end-to-end de la red y proveer al cliente acuerdos de niveles de servicio (SLAs).
- Servicios de mejoramiento futuros.- MPLS CoS provee construcción de bloques para futuros servicios de mejoramiento tales como líneas arrendadas virtuales satisfaciendo requerimientos de ancho de banda.

Beneficios para el proveedor de servicios:

La característica CoS de MPLS permite a los proveedores de servicio clasificar paquetes de acuerdo a su tipo, interfaz de entrada y otros factores marcando a cada paquete dentro del campo EXP MPLS sin cambiar la precedencia IP en el campo DSCP. Por ejemplo, los proveedores de servicios pueden clasificar paquetes con o sin considerar la tasa que el PE recibe, si la tasa se debe considerar entonces el proveedor marca los paquetes diferenciando los in-rate de los out-rate.

* Referencia bibliográfica: "Cisco MPLS Class of Service Enhancements" [4]

Beneficios para el Cliente:

La característica CoS de MPLS conserva el campo DSCP de la precedencia IP dentro de la red del proveedor, lo que le permite al cliente diferenciar el tráfico dentro de su red sin necesidad de comprar múltiples niveles de servicio al proveedor.

El funcionamiento y configuración de la característica CoS de MPLS se la detalla en el Capítulo II.

CAPITULO II

2 CONFIGURACIÓN DE EQUIPOS

En este capítulo se describe la configuración de los equipos que conforman la red MPLS, pasando desde una configuración básica hasta la configuración de una red MPLS que proporcione servicio de redes privadas virtuales (VPN) y calidad de servicio (QoS).

La red que se configurará e implementará en este proyecto básicamente consta de tres routers que conforman la nube MPLS y están distribuidos de la siguiente manera:

- Router Cisco 3640: Router de Core (LSR)
- Routers Cisco 2610: Routers de Acceso (LSRs de contorno)

La figura 2.1 muestra la red MPLS a configurarse.

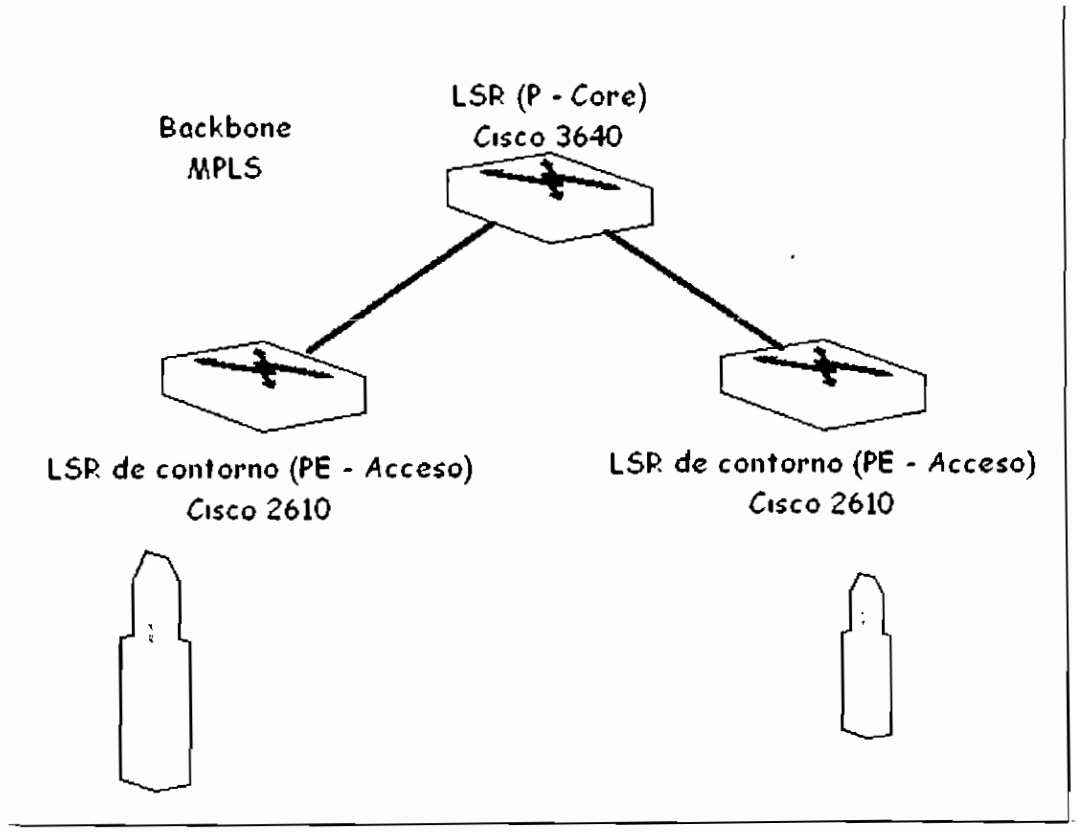


Figura 2.1: Red MPLS a configurarse e implementarse

Los equipos utilizados en este proyecto son de las siguientes características:

Routers de la serie 3600:

- Plataforma 3640.
- Versión del IOS: c3640-jk9o3s-mz.123-10.
- 4 interfaces seriales.
- 2 interfaces ethernet.
- 128 MB de memoria NVRAM
- 32 MB de memoria Flash

Routers de la serie 2600:

- Plataforma 2610.
- Versión del IOS: c2600-telco-mz.12.3-12.
- 2 interfaces seriales.
- 1 interfaz ethernet.
- 64 MB de memoria NVRAM
- 16 MB de memoria Flash

PCs (características más importantes):

- Pentium IV.
- Windows XP.
- 1 interfaz Fast ethernet.
- 256 MB de memoria RAM

Un parámetro que se debe tomar en cuenta es la versión del IOS de los routers ya que la misma debe soportar comandos MPLS, caso contrario no se podrá llevar a cabo la configuración deseada. Las características de los IOS utilizados se detallan en los anexos 2.a y 2.b.

Para lograr comprender con facilidad la configuración de cada router se la realizará paso a paso, explicando qué hace cada comando introducido y qué sintaxis debemos utilizar. De esta manera se indica qué se está haciendo y por qué.

2.1 CONFIGURACIÓN BÁSICA DE LA NUBE MPLS.

En esta sección se configurará la parte básica de MPLS, esto no significa que no se puedan tomar otros enfoques o revisar otros temas relevantes como el costo de la red o la administración que no se los verá debido a que son muy extensos y no permitiría ofrecer una explicación en este proyecto. Sin embargo hay que estar consciente que se debe investigar estos temas antes de llevar a cabo una configuración o migración a MPLS.

Una Red MPLS es comúnmente una red de backbone comprendida por routers habilitados para transportar tráfico MPLS denominados LSR. Generalmente, la red consiste de un LSR de Core con LSRs de contorno responsables de aplicar las etiquetas a los paquetes.

Operación:

La operación de una red MPLS es la siguiente:

1. Las tablas de enrutamiento de los diferentes LSRs son computadas usando un Protocolo de Gateway Interior (IGP). Se puede usar un protocolo de estado de enlace como RIPv2, OSPF o IS-IS.
2. Un protocolo de distribución de etiquetas (LDP) anuncia los enlaces entre rutas y etiquetas, estas correspondencias son chequeadas en la tabla de enrutamiento. Si la ruta aprendida por medio de LDP corresponde a la ruta aprendida por IGP, se crea una entrada en la LFIB del LSR.

Los LSRs utilizan el siguiente mecanismo de envío:

1. Un vez que el LSR de contorno recibe un paquete no etiquetado, se revisa la tabla CEF y se coloca una etiqueta al paquete si es necesario. Este LSR es llamado LSR de ingreso.

2. Al arribo de un paquete etiquetado por una interfaz de un LSR de core, la LFIB provee la interfaz de salida y la nueva etiqueta que será asociada al paquete saliente.
3. El router anterior al último LSR (penúltimo salto) realiza la acción pop al paquete y lo transmite sin etiqueta. El último salto es llamado LSR de salida.

La figura 2.2 muestra la red MPLS a configurarse en esta sección. El router 3600 de Core realiza la acción de conmutación de etiquetas MPLS (swap) y los 2600 las acciones push y pop según sea el caso.

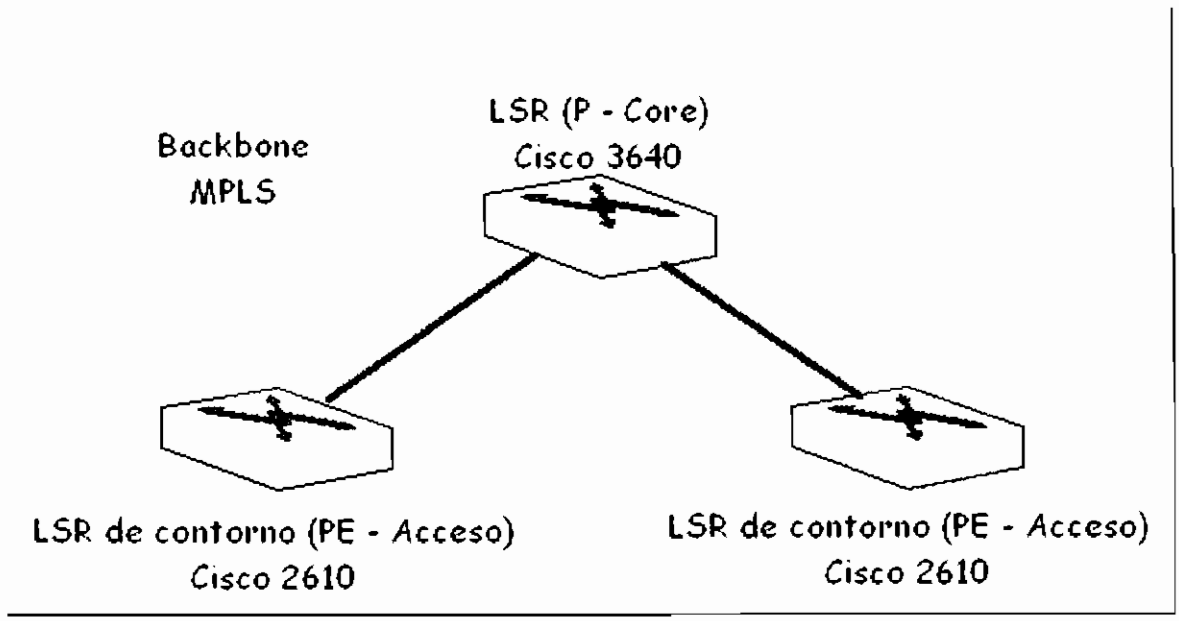


Diagrama 2.2: Configuración Básica MPLS

Requisitos:

Para llevar a cabo la correcta configuración de los equipos, es necesario cumplir con los siguientes requisitos:

- El usuario debe estar familiarizado con la operación básica y terminología MPLS.
- Cisco IOS® Software Releases 12.3(12) o superior.

- Mínimo plataforma 3600 para LSR
- Mínimo plataforma 2610 para LSR de contorno

2.1.1 CONFIGURACIÓN DEL ROUTER LSR (P – CORE).

A continuación se detalla paso a paso la configuración del LSR.

PASO 1: Borrar el archivo de configuración.

El archivo de configuración del router debe ser borrado para evitar problemas, ya que es posible que se encuentren configurados parámetros no deseados que afecten al correcto funcionamiento del equipo.

Para esto se utiliza el comando ***erase startup-config*** en el modo de configuración global. Luego debemos reiniciar el equipo con el comando ***reload*** en el modo de configuración global. Al reiniciarse el router

Comando	***erase startup-config***
Modo	Modo privilegiado
Sintaxis	***erase [startup-config]***
Descripción	***startup-config***: borra el contenido del archivo de configuración que se encuentra almacenado en la nvram.

Ejemplo	`Router# erase startup-config`
	`Erasing the nvram filesystem will remove all files! Continue? [confirm]`
	`[OK]`
	`Erase of nvram: complete`
	`Router# reload`
	.

Tabla 2.1: Características de comando *erase startup-config*

PASO 2: Configuración inicial del router.

Previo a la configuración MPLS de los routers, éstos deben estar configurados con parámetros básicos como:

- Nombre del router
- Passwords
- Interfaces

Para esto se debe seguir el procedimiento que se indica en el primer Caso de Estudio del Capítulo III.

PASO 3: Protocolo de enrutamiento

A continuación se detallan los pasos a seguir para la configuración del protocolo de enrutamiento.

PASO 3.1: Elección del protocolo de enrutamiento

RIPv2, OSPF o IS-IS son protocolos que pueden ser utilizados en este proyecto debido a sus características, las cuales satisfacen las necesidades que debe brindar el backbone MPLS que vamos a configurar. El protocolo que elegimos es OSPF ya que nos permite utilizar MPLS, VLSM y nos brinda la posibilidad de migrar a un backbone más grande en el futuro con mayor facilidad. A continuación se muestra una tabla comparativa de los protocolos de enrutamientos que utilizaremos.

Protocolo de Enrutamiento	
RIPv2	OSPF
Copia la tabla de enrutamiento de los vecinos	Usa la ruta más corta
Se actualiza frecuentemente	Las actualizaciones son desencadenadas por eventos
Usa el número de saltos como métrica	Envía paquetes de estado de enlace a todos los routers de la red
Visualiza la red desde la perspectiva de los vecinos	Tiene una vista común de la red
Converge lentamente	Converge rápidamente

Es susceptible a los bucles de enrutamiento	No es susceptible a los bucles de enrutamiento
Fácil de configurar y administrar	Es más difícil de configurar
Consume una gran cantidad de ancho de banda	Consume una menor cantidad de ancho de banda
Usa topología plana	Permite el diseño jerárquico para grandes internetworks

Tabla 2.2: Protocolos de enrutamiento

Cabe indicar que la elección del protocolo de enrutamiento es muy importante en el desarrollo de cualquier red de backbone, lo que implica que se deben tomar en cuenta algunos parámetros como: número de routers que pertenecen a un sistema autónomo, si vamos o no a utilizar VLSM y otros mencionados en la tabla 2.2.

PASO 3.2: Configuración del protocolo de enrutamiento.

Asegurarse que el protocolo de enrutamiento esté funcionando correctamente. La configuración del protocolo de enrutamiento la realizaremos paso a paso pero sin muchos detalles, debido a que no entra en los objetivos de este proyecto, la configuración se muestra en los Casos de Estudio 2 y 3.

PASO 4: Habilitar IP CEF

CEF es una tecnología de conmutación avanzada de Capa 3. CEF optimiza el rendimiento y escalabilidad de redes cuyo tráfico es de carácter dinámico y poseen una topología dispersa tales como aplicaciones basadas en web y sesiones interactivas.

El Envío Expreso de Cisco (CEF) es el único mecanismo de envío de Capa 3 que emplea la tabla FIB, debido a esto, ***CEF debe estar habilitado*** en todos los routers que ejecutan MPLS y en todas las interfaces de entrada que reciben paquetes IP no etiquetados que se propagan como paquetes etiquetados a través de un backbone MPLS que debe soportar conmutación CEF.

Para habilitar CEF en los routers se usa el comando *ip cef* en el modo de configuración global, para deshabilitar cef se usa la forma *no* del comando.

NOTA: Los routers principales no realizan conmutación CEF (solamente conmutan paquetes etiquetados), pero deben tener habilitado CEF para poder asignar las etiquetas.

Comando	*ip cef*
Modo	Modo de configuración global
Sintaxis	*ip cef [distributed]* *no ip cef [distributed]*
Descripción	*distributed*: es opcional y habilita la operación de CEF distribuido, que distribuye la información CEF a las tarjetas en línea

Ejemplos	`Router(config)#ip cef`
	`Router(config)#ip cef [distributed]`
	`Router(config)#no ip cef`
	`Router(config)#no ip cef [distributed]`

Tabla 2.3: Características del comando *ip cef*

PASO 5: Habilitar MPLS

El siguiente paso es habilitar la conmutación de etiquetas multiprotocolo. Habilitando MPLS en el modo de configuración global no se habilita mpls en las interfaces ni se inicia la distribución de etiquetas, luego veremos como habilitar MPLS en cada interfaz e iniciar la distribución de etiquetas.

Para esto introducimos el comando ***mpls ip*** en el modo de configuración global, para detener la conmutación de etiquetas usamos la forma ***no*** del comando.

NOTA: Para detener la conmutación de etiquetas en todas las interfaces, no es necesario ingresar el comando en cada una de ellas, basta con hacerlo desde el modo de configuración global y lo mismo para volver a habilitar.

Comando	*mpls ip*
Modo	Modo de configuración global
Sintaxis	*mpls ip* *no mpls ip*
Descripción	Este comando no tiene argumentos o palabra clave
Ejemplos	`Router(config)#mpls ip` `Router(config)#no mpls ip`

Tabla 2.4: Características del comando *mpls ip*

PASO 6: Habilitar MPLS en cada interfaz

Es necesario habilitar la conmutación de etiquetas en cada interfaz que estará involucrada en la red MPLS.

Para esto se utiliza el comando ***mpls ip*** en el modo de configuración de interfaz. La forma ***no*** del comando hace que los paquetes salientes de la interfaz se envíen sin etiquetar, esta forma del comando también termina la distribución de etiquetas en esa interfaz.

NOTA: La forma no del comando no afecta al envío de paquetes etiquetados a través de túneles LSP que pueden estar usando esa interfaz. El comando ***mpls ip*** inicia el etiquetado de paquetes MPLS y utiliza de forma predeterminada el protocolo TDP como protocolo de distribución de etiquetas.

Comando	*mpls ip*
Modo	Modo de configuración de interfaz
Sintaxis	*mpls ip* *no mpls ip*
Descripción	Este comando no tiene argumentos o palabra clave
Ejemplos	`Router(config-if)#mpls ip` `Router(config-if)#no mpls ip`

Tabla 2.5: Características del comando *mpls ip*

PASO 7: Configurar el Protocolo de Distribución de Etiquetas

Si queremos cambiar el protocolo de distribución de etiquetas de TDP a LDP es necesario configurar una interfaz *loopback* para que el protocolo LDP, pueda funcionar correctamente.

Para configurar una interfaz de loopback se lo hace con el comando ***interface loopback*** *number* en el modo de configuración global y luego se configura su dirección ip con el comando ***ip address*** *ip-address netmask*

Comando	***interface loopback***
Modo	Modo de configuración global
Sintaxis	***interface loopback*** *number* ***no interface loopback*** *number*
Descripción	*number:* indica el número de la interfaz loopback
Ejemplos	`Router(config)#interface loopback 0` `Router(config)#no interface loopback 0`

Tabla 2.6a: Características del comando *interface loopback*

Comando	***ip address ip-address netmask***
Modo	Modo de configuración de interfaz
Sintaxis	***ip address*** *ip-address netmask* ***no ip address*** *ip-address netmask*
Descripción	*ip-address*: dirección ip asignada a esa interfaz *netmask:* mascara de red a la que pertenece esa interfaz

Ejemplos	`Router(config-if)#ip address 192.168.10.10 255.255.255.255`
	
	`Router(config-if)#no ip address 192.168.20.1 255.255.255.0`

Tabla 2.6b: Características del comando *ip address ip-address netmask*

PASO 7.1: Elección del Protocolo de Distribución de Etiquetas

El IOS de Cisco implementa dos protocolos de distribución de etiquetas (LDP y TDP) que pueden usarse para asociar dos redes IP con etiquetas MPLS con la finalidad de un enrutamiento unidifusión basado en destinos.

Tag Distribution Protocol TDP (antiguo protocolo de distribución de etiquetas). Es el protocolo patentado por Cisco y disponible en las versiones del IOS 11.1CT, 12.0 y todas la siguientes.

Label Distribution Protocol LDP (Protocolo de distribución de etiquetas). Es el protocolo de enlace de etiquetas estándar de la IETF disponible desde la versión 12.2T del IOS de Cisco.

TDP y LDP son equivalentes funcionalmente y pueden ser utilizados al mismo tiempo dentro de una red. Debido a esto y a que LDP es el protocolo estándar lo hemos seleccionado para utilizarlo en el proyecto. Cabe indicar que no hay diferencia al utilizar LDP o TDP.

Si se desea cambiar el protocolo de distribución de etiquetas TDP / LDP se lo hace con el comando ***mpls label protocol.*** Este comando puede ser utilizado tanto en el modo de configuración global como en el modo de configuración de interfaz.

NOTA: Si no se especifica el protocolo de distribución de etiquetas mediante el comando ***mpls label protocol*** el protocolo TDP es usado por defecto. Si se desea que todas las interfaces usen LDP, basta con ingresar el comando en el modo de configuración global.

Comando	***mpls label protocol*** *protocol*
Modo	Modo de configuración global Modo de configuración de interfaz
Sintaxis	***mpls label protocol*** *protocol* ***no mpls label protocol*** *protocol*
Descripción	*protocol:* indica el protocolo de distribución de etiquetas a usarse, puede ser: *tdp o ldp.*
Ejemplos	`Router(config)#mpls label protocol ldp` `Router(config-if)#no mpls label protocol tdp`

Tabla 2.7: Características del comando *mpls label protocol protocol*

PASO 7.2: Elección del Router ID LDP

Se debe elegir el router ID LDP, que es necesario para que el protocolo LDP inicie su operación.

Para especificar a una interfaz como el router ID LDP, use el comando ***mpls ldp router-id*** en el modo de configuración global. Para remover la interfaz preferida como router ID LDP, use la forma ***no*** del comando

NOTA: Si el comando ***mpls ldp router-id*** no es ejecutado, el router ID LDP es determinado de la siguiente manera:

1.- Se examina la dirección IP de todas las interfaces
2.- Si esas direcciones IP incluyen direcciones IP de interfaces loopback, la dirección IP más grande de una loopback es elegida como router ID LDP.
3.- De no haber direcciones IP loopback, La dirección IP más grande perteneciente a una interfaz activa es elegida como router ID LDP

Es recomendable configurar una interfaz de loopback como router ID LDP, debido a que si se elige una interfaz normal esta puede dejar de funcionar o no

estar siendo anunciada por el protocolo de enrutamiento, provocando un mal funcionamiento del protocolo LDP.

Comando	***mpls ldp router-id*** *interface*
Modo	Modo de configuración global
Sintaxis	***mpls ldp router-id*** *interface* ***no mpls ldp router-id*** *interface*
Descripción	*interface:* provoca que la dirección IP de la interfaz especificada sea seleccionada como router ID LDP, siempre y cuando la interfaz se encuentre operacional
Ejemplos	`Router(config)# mpls ldp router-id loopback 0` `Router(config-if)#no mpls ldp router-id ethernet 0`

Tabla 2.8: Características del comando *mpls ldp router-id interface*

2.1.2 CONFIGURACIÓN DE LOS ROUTERS LSR DE CONTORNO (PE - ACCESO).

MPLS necesita una conexión IP estándar para establecer bases de envío, para lo cual se debe configurar una red IP usual.

Es importante asegurarse que el protocolo de enrutamiento utilizado esté trabajando correctamente. La configuración MPLS de los LSR de contorno es igual a la del LSR así que se repiten los pasos del 1 al 7 de la sección anterior.

2.1.3 RESULTADOS DE LA CONFIGURACIÓN.

A continuación se muestra la red obtenida en la figura 2.3, luego los archivos de configuración de cada router; los comandos introducidos se detallan en los anexos debidamente numerados.

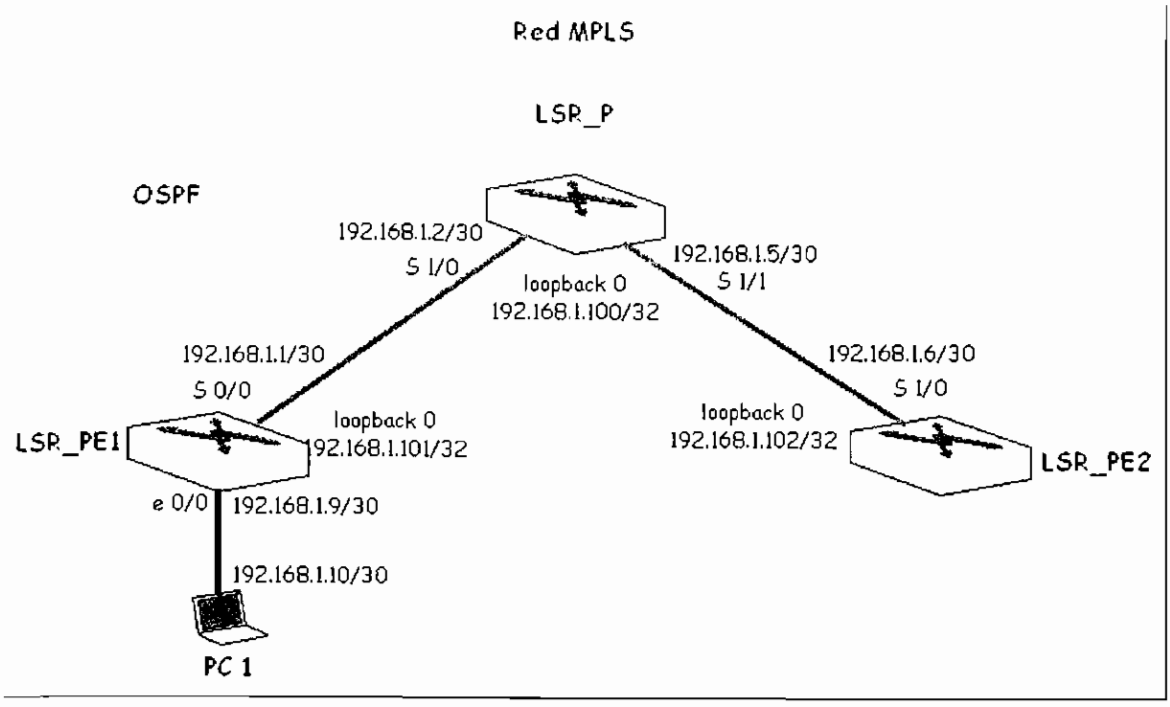


Diagrama 2.3: Resultados de la configuración Básica MPLS

Archivo de configuración

ROUTER LSR_P (Cisco 3640)

```
LSR_P#
LSR_P#show running
Building configuration...

Current configuration : 1286 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname LSR_P
!
boot-start-marker
boot-end-marker
!
enable password lsrp
!
no aaa new-model
ip subnet-zero
!
!
!
ip cef
ip audit po max-events 100
mpls label protocol ldp
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
!
```

```
!
interface Loopback0
 ip address 192.168.1.100 255.255.255.255
 no clns route-cache
!
interface Ethernet0/0
 ip address 192.168.57.1 255.255.255.0
 shutdown
 half-duplex
 no clns route-cache
!
interface TokenRing0/0
 no ip address
 shutdown
 ring-speed 16
 no clns route-cache
!
interface Serial1/0
 ip address 192.168.1.2 255.255.255.252
 tag-switching ip
 no fair-queue
 no clns route-cache
!
interface Serial1/1
 ip address 192.168.1.5 255.255.255.252
 tag-switching ip
 clockrate 56000
 no clns route-cache
!
interface Serial1/2
 no ip address
 shutdown
 no clns route-cache
!
interface Serial1/3
 no ip address
 shutdown
 no clns route-cache
!
router ospf 1
 log-adjacency-changes
 network 192.168.1.0 0.0.0.255 area 0
!
ip http server
no ip http secure-server
ip classless
!
!
!
line con 0
 password lsrp
 login
line aux 0
line vty 0 4
 password lsrp
 login
!
!
end
LSR_P#
```

Archivo de configuración:

ROUTER LSR_PE_1 (Cisco 2610)

```
LSR_PE_1#sh run
Building configuration...

Current configuration : 1082 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname LSR_PE_1
!
boot-start-marker
boot-end-marker
!
enable password lsr1
!
memory-size iomem 10
no aaa new-model
ip subnet-zero
!
!
ip cef
mpls label protocol ldp
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
!
!
!
interface Loopback0
 ip address 192.168.1.101 255.255.255.255
!
interface Ethernet0/0
 ip address 192.168.1.9 255.255.255.252
 half-duplex
 tag-switching ip
!
interface Serial0/0
 ip address 192.168.1.1 255.255.255.252
 tag-switching ip
 clockrate 56000
 no fair-queue
!
interface Serial0/1
 no ip address
 shutdown
!
router ospf 1
 log-adjacency-changes
 network 192.168.1.0 0.0.0.255 area 0
!
ip http server
ip classless
!
!
!
!
line con 0
 password lsr1
 login
```

```
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
 password lsr1
 login
 transport preferred all
 transport input all
 transport output all
!
end
```

Archivo de configuración

ROUTER LSR_PE_2 (Cisco 2610)

```
LSR_PE_2#show running
Building configuration...

Current configuration : 957 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname LSR_PE_2
!
boot-start-marker
boot-end-marker
!
enable password lsr2
!
no aaa new-model
ip subnet-zero
!
!
ip cef
mpls label protocol ldp
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
!
!
!
interface Loopback0
 ip address 192.168.1.102 255.255.255.255
!
interface Ethernet0/0
 no ip address
 shutdown
 half-duplex
!
interface Serial0/0
 no ip address
 shutdown
!
interface Serial0/1
 ip address 192.168.1.6 255.255.255.252
 tag-switching ip
!
router ospf 1
```

```
 log-adjacency-changes
 network 192.168.1.0 0.0.0.255 area 0
!
ip http server
ip classless
!
!
!
!
line con 0
 password lsr2
 login
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
 password lsr2
 login
 transport preferred all
 transport input all
 transport output all
!
end
```

Configuración: LSR_P (Ver anexo2.1.3.a)

Configuración: LSR_PE_1 (Ver anexo2.1.3.b)

Configuración: LSR_PE_2 (Ver anexo2.1.3.c)

2.2 CONFIGURACIÓN DE VPNs SOBRE MPLS.

Las Redes Privadas Virtuales (VPNs) proveen funciones y usos parecidos a los de las redes privadas dedicadas. Las VPNs son soportadas por las redes del proveedor de servicio sobre la cual los paquetes etiquetados son enviados desde LSRs de contorno hacia otros LSRs de contorno. El servicio de VPN crea múltiples redes privadas ambientadas en una infraestructura pública. Los proveedores de servicio pueden usar las VPNs para dar a sus clientes un servicio de red privada individualizado, dándole al cliente un ambiente IP seguro utilizando una infraestructura pública

Operación:

Cada VPN está asociada con uno o más VRFs, los cuales definen una VPN a un sitio del cliente ligado a un router PE. Una tabla VRF contiene la siguiente información:

- Tabla de enrutamiento IP.
- Tabla CEF derivada.
- Un conjunto de interfaces que usan la tabla de envío.
- Un conjunto de reglas y variables del protocolo de enrutamiento que determina que va en la tabla de envío.

Un sitio del cliente puede ser miembro de múltiples VPNs. Sin embargo, un sitio puede ser asociado con un solo VRF. El VRF de un sitio del cliente contiene todas las rutas disponibles para el sitio desde las VPNs asociadas.

Las tablas de enrutamiento IP y CEF almacenan la información de envío del paquete por cada VRF (estas tablas juntas son análogas a la tabla FIB usada en MPLS). El router mantiene por separado las tablas de enrutamiento y CEF para prevenir que la información sea enviada fuera de la VPN y permite que la misma subred sea usada en varias VPNs sin causar problemas de duplicación de direcciones IP.

La distribución de la información de enrutamiento de las VPNs es controlada a través del uso de comunidades VPN, implementada por comunidades extendidas BGP, la distribución trabaja de la siguiente manera:

- Cuando una ruta VPN es inyectada en BGP, esta es asociada con una lista de la comunidad VPN correspondiente. Esta lista es creada a través de la lista de exportación asociada con el VRF desde el cual la ruta fue aprendida.
- Asociada con cada VRF es una lista de importación de la comunidad correspondiente, la cual define los valores a ser verificados por la tabla VRF antes de que la ruta sea estimada elegible por la importación dentro de la instancia VPN de enrutamiento. Por ejemplo, si una lista de importación de un VRF dado incluye los distintivos de ruta A, B y C, entonces cualquier ruta VPN que lleve A, B o C es importada dentro del VRF.

Requisitos:

Para llevar a cabo la correcta configuración de los equipos, es necesario cumplir con los siguientes prerrequisitos:

- El usuario debe estar familiarizado con la operación básica y terminología MPLS.
- Cisco IOS® Software Releases 12.3(12) o superior.
- Mínimo plataforma 3600 para LSR
- Mínimo plataforma 2610 para LSR de contorno
- CEF switching habilitado en cada router
- Configurar MPLS
- Conectividad MPLS entre todos los LSR de contorno con servicio VPN o MPLS en todo el proveedor
- Habilitar BGP entre los routers del proveedor para la distribución de la información de enrutamiento de las VPN.

La funcionalidad MPLS/VPN reside en los routers PE. Se puede usar como routers CE cualquier router capaz de intercambiar información de enrutamiento con su router PE.

2.2.1 CONFIGURACIÓN DE LOS ROUTERS LSR DE CONTORNO (PE).

A continuación se detalla paso a paso la configuración de los routers LSR de contorno.

PASO 1: Configurar MPLS

Para realizar la configuración MPLS se debe seguir los pasos del 1 al 7 de la sección 2.1.1.

PASO 2: Crear y configurar las VRFs

Como primer paso para la configuración del servicio de VPN basado en MPLS, se debe crear una instancia VRF por cada VPN conectada usando el comando ***ip vrf*** *vrf name*[*] en el modo de configuración global. Cuando se introduce el comando ***ip vrf*** *vrf name*, el router entra en el submodo vrf configuration. Ahora puede configurar las variables asociadas con esta VRF, como el distintivo de ruta y las normas de importación y exportación. El comando ***ip vrf*** *vrf name* crea las VRF correspondientes y las tablas de envío y enrutamiento CEF únicas. Sin embargo, las VRF no están totalmente configuradas, se debe seguir configurando para proporcionar rutas para las tablas y para crear etiquetas MPLS asociadas.

Comando	***ip vrf*** *vrf name*
Modo	Modo de configuración global
Sintaxis	***ip vrf*** *vrf name* ***no ip vrf*** *vrf name*

[*] Se debe tomar en cuenta que el comando ***ip vrf*** *vrf name* hace distinción entre mayúsculas y minúsculas.

Descripción	*vrf name:* Nombre asignado al VRF creado.
Ejemplos	`Router(config)#ip vrf VPN1` `Router(config)#no ip vrf VPN1`

Tabla 2.9: Características del comando *ip vrf vrf name*

PASO 3: Definir y Configurar los distintivos de ruta.

Las rutas VPN del cliente deben ser publicadas a través del backbone MPLS/VPN entre los routers PE. Debido a la capacidad de administrar un gran número de rutas y su flexibilidad para transportar parámetros opcionales (conocidos como atributos) sin cambiar mucho el protocolo, BGP (protocolo de Gateway fronterizo) es el protocolo elegido para este propósito. Estos factores hacen que el protocolo sea muy adaptable y conveniente para usarlo con la arquitectura MPLS/VPN.

BGP en su formato estándar, sólo puede administrar rutas IPv4. En la arquitectura MPLS/VPN, como cada VPN puede usar los mismos prefijos IP que otras VPN, es necesario agregar un ***distintivo de ruta*** a la dirección IPv4. Esto requiere que el protocolo BGP pueda manejar extensiones para que la información de la VPN sea única en el backbone MPLS/VPN y para que los portavoces BGP puedan identificar las actualizaciones de enrutamiento y no transporten información de prefijo IPv4 estándar. El multiprotocolo (MP-BGP) y la información de enrutamiento VPN-IPv4 proporcionan estas extensiones.

El ***distintivo de ruta*** está formado por una secuencia de 64 bits al inicio de la dirección IPv4 que contiene la actualización MP-iBGP* y es distinto para cada VPN para que las direcciones contenidas en todas las VPN sean únicas en el backbone MPLS/VPN. BGP considera una dirección IPv4 diferente a otra dirección IPv4 que tenga la misma red y máscara si los distintivos de ruta son diferentes.

* Aunque se puede usar MP-BGP con enlaces igual a igual internos y externos al dominio MPLS/VPN, nos vamos a referir a este protocolo como MP-iBGP debido a que las sesiones entre routers PE existen via iBGP (BGP interior) en el mismo dominio MPLS/VPN.

Para configurar un distintivo de ruta se lo debe hacer utilizando el comando *rd route-distinguisher* en el submodo de *vrf configuration*.

Se debe establecer la asignación de un valor particular al distintivo de ruta por cada VRF del router PE. La estructura de este valor puede ser ASN:nn o direciónIP:nn. Se recomienda el uso de ASN:nn* con un ASN (número de sistema autónomo que es asignado por la IANA para que sea único entre los proveedores de servicios. Utilice el formato direcciónIP:nn sólo cuando la red MPLS/VPN use un número de AS privado pero las direcciones VPN-IPv4 sean propagadas más allá del AS privado como cuando se intercambian rutas VPN entre distintos proveedores de servicio.

Comando	*rd route-distinguisher*
Modo	Submodo vrf configuration
Sintaxis	*rd route-distinguisher*
Descripción	*route-distinguisher:* Agrega 8 bytes de prefijo a un prefijo IPv4 para crear un prefijo VPN-IPv4.

Ejemplos	`Router(config)# ip vrf vrf_loja`
	`Router(config-vrf)# rd 100:3`
	`Router(config-vrf)# exit`
	`Router(config)# ip vrf vrf_red`
	`Router(config-vrf)# rd 173.13.0.12:2`

Tabla 2.10: Características del comando *rd route-distinguisher*

PASO 4: Configurar las propiedades *import y export*.

El paso final para configurar una VFR es añadir las normas de importación y exportación que debe usar la VRF. Estas normas se las añade para poner rutas en la VRF y para publicar rutas fuera de la VRF.

* Incluso cuando se utilice el formato ASN:nn para el distintivo de ruta, es importante hacer notar que el distintivo de ruta no tiene semántica y solamente es interpretado por BGP como una secuencia de bits que forman parte de la dirección VPN-IPv4.

La comunidad extendida BGP de objetivo de ruta especifica las normas usadas por la VRF. El objetivo de ruta debe configurarse para especificar, las rutas, que contienen este valor de objetivo de ruta, que se importan a la VRF, así como el objetivo de ruta que se añade a las rutas que son exportadas desde la VRF. El comando ***route-target {import | export | both}*** *route-target-ext-community* cuya sintaxis se muestra en la tabla 2.11, controla esto en el submodo de vrf configuration.

Utilice las palabras clave *export* o *import* con el comando ***route-target*** para especificar por separado las normas de importación y exportación para cada VRF. Normalmente, las normas de importación y exportación son predeterminadas y son las mismas, de forma que pueda especificar ambas usando el comando ***route-target*** *ASN:nn* o ***route-target both*** *ASN:nn*. Este es el caso por ejemplo de una topología, donde las VRF de dos clientes necesiten exportar sus rutas con un objetivo de ruta que es importado por otros miembros de la VPN de su organización.

Comando	***route-target {import | export | both}*** *route-target-ext-community*
Modo	Submodo vrf configuration
Sintaxis	***route-target {import | export | both}*** *route-target-ext-community*
Descripción	*import:* Importa la información de enrutamiento desde la comunidad extendida de objetivo VPN. *export:* Exporta la información de enrutamiento a la comunidad extendida de objetivo VPN. *both:* Importa y exporta la información de enrutamiento a la comunidad extendida de objetivo VPN. *route-target-ext-community:* Agrega los atributos de la comunidad extendida de objetivo de ruta a la lista de importación o exportación de la VRF.

Ejemplos	`Router(config)# ip vrf vrf_loja`
	`Router(config-vrf)# route-target both 1000:1`
	`Router(config-vrf)# route-target export 1000:2`

```
Router(config-vrf)# route-target import
173.27.0.130:200
Router(config)# ip vrf vrf_loja
Router(config-vrf)# route-target both 1000:1
Router(config-vrf)# route-target export 1000:2
Router(config-vrf)# route-target import
173.27.0.130:200
```

Tabla 2.11: Características de comando *route-target*

PASO 5: Configurar los detalles de envío de las respectivas interfaces.

Se debe asociar la o las diferentes VRFs a una o varias interfaces según nuestras necesidades, para esto utilizamos el comando ***ip vrf forwarding*** *vrf-name* en el modo de configuración de interfaz.

Este comando asocia una interfaz con una VRF. Se debe tomar en cuenta que ejecutando este comando la dirección IP es removida de la interfaz, por lo tanto se debe reconfigurar la dirección IP de dicha interfaz. Para desasociar una VRF utilice la forma no del comando.

NOTA: Sólo las interfaces que ejecutan la conmutación CEF pueden asociarse con las VRF, debido a que el mecanismo de conmutación CEF es un requisito previo necesario para un envío satisfactorio de los datos MPLS/VPN al conseguir la imposición de etiquetas a través de la ruta de conmutación CEF.

Comando	***ip vrf forwarding*** *vrf-name*
Modo	Configuración de interfaz
Sintaxis	***ip vrf forwarding*** *vrf-name* ***no ip vrf forwarding*** *vrf-name*
Descripción	*vrf-name:* Asocia la interfaz con la VRF especifica

Ejemplos	`Router(config)# interface atm0/0` `Router(config-if)# ip vrf forwarding vpn1`

Tabla 2.12: Características de comando *ip vrf forwarding*

PASO 6: Configurar el enlace PE-CE

Para proporcionar el servicio de VPN, los routers PE deben ser configurados de tal manera que cualquier información de enrutamiento que aprenden de una interfaz de cliente VPN pueda asociarse con una VRF particular. Puede hacerse esto durante el proceso del protocolo de enrutamiento, lo que se conoce como **contexto de enrutamiento**, cada VRF usa un contexto de enrutamiento diferente. Cualquier ruta aprendida a través de una interfaz asociada con el contexto del protocolo de enrutamiento particular es instalada en la VRF asociada, caso contrario es colocada en la tabla de enrutamiento global. Esto permite la separación de la información de enrutamiento en contextos diferentes, aunque la información sea aprendida por el mismo proceso del protocolo de enrutamiento lo que permite que los protocolos de enrutamiento sean consistentes.

Dependiendo del protocolo de enrutamiento que se esté usando entre los routers PE y CE, éste puede ser enrutamiento estático o se puede usar protocolos de enrutamiento como RIP, OSPF o BGP. En el desarrollo del Capítulo III se hará uso de estos protocolos para realizar la configuración y pruebas de los Casos de Estudios planteados..

PASO 6.1: Configurar el protocolo de enrutamiento RIPv2

RIPv2 es un protocolo de enrutamiento dinámico que se configura dando al protocolo de enrutamiento el nombre de RIP versión 2 y luego asignando números de red IP sin especificar los valores de máscara de subred. Para ello se utiliza el comando ***router rip*** en el modo de configuración global y para especificar la versión de RIP introducimos el comando ***version 2*** en el modo *router configuration*, de este modo configuramos a RIPv2 como protocolo de enrutamiento.

Comando	*router rip*
Modo	Configuración global
Sintaxis	*router rip*
Descripción	*rip:* Especifica a RIP como protocolo de enrutamiento.
Ejemplos	`Router(config)# router rip` `Router(config-router)# version 2`

Tabla 2.13: Características de comando *router rip*

Cuando se elige RIPv2 como protocolo de enrutamiento, el proceso RIP necesita que le digan qué rutas RIP publicar y desde qué interfaces, para ello, se utiliza el comando ***network*** *ip-address* en el modo *router configuration*. Este comando indica al proceso RIP que interfaces tienen habilitado RIP y desde qué interfaces enviar las actualizaciones RIP. Estas actualizaciones contiene la tabla de enrutamiento más cualquier interfaz conectada directamente que tenga habilitado RIP.

El comando que le indica al proceso RIP qué rutas publicar, es el comando ***network*** *network-address,* ingrese este comando en el modo *router configuration*. Para remover una entrada, utilice la forma ***no*** del comando.

Comando	***network*** *network-address*
Modo	Router configuration
Sintaxis	***network*** *network-address* ***no network*** *network-address*
Descripción	*network-address:* Especifica una lista de redes para el proceso de enrutamiento.

Ejemplos	`Router(config)# router rip`
	`Router(config-router)# version 2`
	`Router(config-router)# network 192.168.77.0`
	`Router(config-router)# network 10.30.0.0`

Tabla 2.14: Características de comando *network network-address*

PASO 6.2: Configurar *address-family*

Es indeseable para el proceso RIP, publicar todas las rutas de cualquier interfaz que pertenezcan al intervalo de direcciones especificado por el comando ***network*** *ip-address*. Para evitar esto, se utiliza el submodo *address-family* durante la configuración del proceso RIP principal, cualquier comando introducido en este submodo es interpretado como perteneciente a la VRF especificada. Así, cualquier comando ***network*** introducido en el submodo es asociado con la VRF configurada para esa familia de direcciones, con lo cual se evita que cualquier ruta RIP que pertenezca a la tabla global de enrutamiento, o cualquier otra VRF, sea publicada aunque el proceso RIP tenga en cuenta estas rutas y las rutas que serán publicadas son la pertenecientes a las interfaces asociadas con la familia de direcciones.

Para configurar lo antes mencionado, se debe ingresar el comando ***address-family ipv4 vrf*** *vrf-name* en el submodo *address-family*, para deshabilitar el submodo *address-family*, cuando se esté configurando el protocolo de enrutamiento, utilice la forma ***no*** del comando

Comando	***address-family ipv4 vrf*** *vrf-name*
Modo	Submodo address-family
Sintaxis	***address-family ipv4 vrf*** *vrf-name* ***no address-family ipv4 vrf*** *vrf-name*
Descripción	*vrf-name:* Especifica el nombre de la VRF a asociarse con los comandos del submodo.

Ejemplos	`Router(config)# router rip`
	`Router(config-router)# version 2`
	`Router(config-router)# address-family ipv4 vrf vrf_Loja`

Tabla 2.15: Características de comando *address-family ipv4 vrf vrf-name*

NOTA: Esto también es aplicado cuando se utiliza OSPF o BGP.

PASO 6.3: Redistribuir el proceso BGP dentro del proceso RIP

Para que cualquier ruta VPN (aprendida a través de las sesiones MP-iBGP) sea redistribuida dentro del proceso RIP, es necesario ingresar el comando ***redistribute*** *protocol process-id* ***metric*** *metric-value*. Estas rutas iBGP normalmente no son redistribuidas, pero se logra hacerlo cuando la interfaz está asociada a una VRF.

Cuando se utiliza RIP como configuración PE-CE, es necesario especificar la métrica BGP como métrica predeterminada, que es *unreachable,* sin embargo, es posible transportar métricas RIP de forma transparente a través del backbone MPLS/VPN con el uso del comando ***redistribute*** *bgp* ***metric*** *transparent*. Este comando hace que RIP utilice la métrica de la tabla de enrutamiento para rutas redistribuidas como si fuera la métrica RIP, con la métrica original siendo transportada a través del backbone MPLS/VPN.

Comando	***redistribute*** *protocol process-id* ***metric*** *metric-value*
Modo	Router configuration
Sintaxis	***redistribute*** *protocol process-id* ***metric*** *metric-value* ***no redistribute*** *protocol process-id* ***metric*** *metric-value*
Descripción	*protocol:* El protocolo de enrutamiento que está siendo utilizado. *process-id:* Es el número de sistema autónomo utilizado por el proceso de enrutamiento. *metric-value:* Métrica usada por la ruta redistribuida. Puede usarse la opción *transparent* para redistribuir la ruta con una métrica de forma transparente.

Ejemplos	`Router(config)# router rip`
	`Router(config-router)# version 2`
	`Router(config-router)# redistribute bgp 1 metric 1`

Tabla 2.16: Características de comando *redistribute protocol process-id metric metric-value*

PASO 7: Configurar el multiprotocolo BGP (MP-BGP)

Ya se ha visto que se usa MP-BGP, que es la extensión del protocolo BGP-4 existente, para publicar rutas VPN de cliente entre routers PE que fueron aprendidas de los routers CE conectados. Estas rutas pueden ser aprendidas a través del estándar BGP-4, de la versión 2 de RIP, de rutas estáticas o de OSPF. Futuras versiones del IOS de Cisco puede que soporten protocolos de enrutamiento CE-PE adicionales.

MP-BGP sólo es necesario en el backbone del proveedor de servicios. Por tanto, todas las sesiones MP-BGP son sesiones BGP internas, porque la sesión es entre dos routers que pertenecen al mismo sistema autónomo.

MP-iBGP es necesario en la arquitectura MPLS-VPN porque la actualización BGP necesita transportar más información que solo una dirección IPv4, esto es una dirección VPN-IPv4 que está formada por información de etiquetado, comunidades BGP extendidas y posiblemente comunidades BGP estándar.

La configuración BGP requiere varios pasos y varios comandos de configuración. Se debe tomar en cuenta que la configuración debe permitir cualquier sesión MP-iBGP PE a PE a través del backbone MPLS/VPN.

Los pasos que debemos seguir para lograr la configuración antes mencionada se detallan a continuación:

PASO 7.1: Configurar una sesión BGP

Para configurar una sesión BGP lo hacemos introduciendo el comando ***router bgp*** *as-number* en el modo de configuración global.

Comando	*router bgp as-number*
Modo	Configuración Global
Sintaxis	*router bgp as-number*
Descripción	*as-number:* Número de Sistema autónomo.
Ejemplos	`Router(config)# router bgp 1`

Tabla 2.17: Características de comando *router bgp as-number*

PASO 7.2: Evitar activaciones de sesiones BGP, IPv4 o VPN-IPv4 por defecto.

El comportamiento predeterminado cuando se configura una sesión BGP en un router Cisco es que esta sesión permite transportar prefijos unidifusión IPv4, esto puede resultar un problema en un entorno puramente MPLS/VPN, donde BGP es utilizado para transportar únicamente prefijos VPN-IPv4. Para evitar la activación de cualquier sesión BGP, IPv4 o VPN-IPv4 por defecto utilizamos el comando ***no bgp default ipv4-unicast*** .

Comando	***bgp default ipv4-unicast***
Modo	Configuración de protocolo
Sintaxis	***bgp default ipv4-unicast*** ***no bgp default ipv4-unicast***
Descripción	*vrf-name:* Asocia la interfaz con la VRF especifica
Ejemplos	`Router(config-router)#bgp default ipv4-unicast` `Router(config-router)#no bgp default ipv4-unicast`

Tabla 2.18: Características de comando *bgp default ipv4-unicast*

PASO 7.3: Activación de las sesiones BGP IPv4 estándar.

El siguiente paso en la configuración de MP-iBGP es definir y activar las sesiones BGP entre routers PE. Algunas de estas sesiones transportan rutas VPN-IPv4 e IPv4. La configuración de las sesiones BGP que transportan rutas IPv4 desde la tabla de enrutamiento global es exactamente igual que la

configuración BGP estándar, con la excepción de que se debe activar la sesión. EL comando ***neighbor*** controla la activación de la sesión, como se muestra en el ejemplo 2.2.1.7.3 este ejemplo muestra todos los comandos usados para establecer una sesión BGP IPv4 entre routers PE.

Ejemplo 2.2.1.7.3:

```
Router (config)#router bgp 1
Router (config-router)#neighbor 160.89.2.3 remote-as 1
Router (config-router)#neighbor 160.89.2.3 update-source loopback0
Router (config-router)#neighbor 160.89.2.3 active
```

Para agregar una entrada a la tabla de vecinos BGP o MP-iBGP, se usa el comando ***neighbor*** *ip-address* ***remote-as*** *as-number* en el modo *router configuration*. Para remover una entrada utilice la forma ***no*** del comando.

Especificando un vecino con un número de sistema autónomo que corresponde al número del sistema autónomo ingresado con el comando ***router bgp*** *as-number* identifica al vecino como interno al sistema autónomo local, de otra manera, el vecino es considerado externo.

Comando	***neighbor*** *ip-address* ***remote-as*** *as-number*
Modo	Configuración de protocolo
Sintaxis	***neighbor*** *ip-address* ***remote-as*** *as-number* ***no neighbor*** *ip-address* ***remote-as*** *as-number*
Descripción	*ip-address:* Dirección IP del vecino *as-number:* Número del sistema autónomo
Ejemplos	`Router(config)#router bgp 1` `Router(config-router)#neighbor 131.108.1.2 remote-as 1`

Tabla 2.19: ***neighbor*** *ip-address* ***remote-as*** *as-number*

EL IOS de Cisco permite a las sesiones BGP usar una interfaz operacional específica para conexiones TCP mediante el comando ***neighbor*** *ip-address* ***update-***

source *interface-type* en el modo *router configuration*. Esta característica trabaja con cualquier interfaz especificada en el router. La interfaz loopback es la interfaz que se usa comúnmente.

Comando	***neighbor*** *ip-address* ***update-source*** *interface-type*
Modo	Configuración de protocolo
Sintaxis	***neighbor*** *ip-address* ***update-source*** *interface-type* ***no neighbor*** *ip-address* ***update-source*** *interface-type*
Descripción	*ip-address:* Dirección IP del vecino BGP-speaking *interface-type:* Interface a ser usada como la fuente

Ejemplos	`Router (config)#router bgp 1`
	`Router (config-router)#neighbor 160.89.2.3 remote-as 1`
	`Router (config-router)#neighbor 160.89.2.3 update-source loopback0`

Tabla 2.20: ***no neighbor*** *ip-address* ***update-source*** *interface-type*

Para habilitar el intercambio de información con un vecino BGP, use el comando ***neighbor*** *ip-address* ***activate*** en el modo *router configuration*. Para deshabilitar el intercambio de una dirección con un vecino BGP utilice la forma ***no*** del comando.

Comando	***neighbor*** *ip-address* ***activate***
Modo	Router configuration
Sintaxis	***neighbor*** *ip-address* ***activate*** ***no neighbor*** *ip-address* ***activate***
Descripción	*ip-address:* Dirección IP del router vecino

Ejemplos	`Router (config)#router bgp 1`
	`Router (config-router)#neighbor 160.89.2.3 remote-as 1`
	`Router (config-router)#neighbor 160.89.2.3 update-source loopback0`
	`Router (config-router)#neighbor 160.89.2.3 active`

Tabla 2.21: Características del commando ***no neighbor*** *ip-address* ***update-source*** *interface-type*[*]

[*] Utilice este comando para anunciar o intercambiar información en forma de un prefijo IP, IPv6 o VPN-IPv4. La información del prefijo es conocida en BGP como *Network Layer Reachability Information (NLRI).*

PASO 7.4: Configuración de la familia de direcciones para el intercambio de rutas VPN-IPv4.

El proceso BGP activa la sesión MP-iBGP que transporta prefijos VPN-IPv4 mediante una familia de direcciones propia de BGP. Esta configuración crea un contexto de enrutamiento para el intercambio de prefijos VPN-IPv4.

Utilice el comando ***address-family*** *vpnv4* en el modo *router configuration* para que las sesiones de enrutamiento BGP estén habilitadas a transportar prefijos VPN-IPv4. Utilice la forma **no** del comando para deshabilitar esta función. Para salir de este modo de configuración utilice el comando ***exit-address-family.***

Comando	***address-family*** *vpnv4*
Modo	Router configuration
Sintaxis	***address-family*** *vpnv4* ***no address-family*** *vpnv4*
Descripción	*vpnv4:* Configura a la sesión para que transporte prefijos VPN-IPv4, cada uno de los cuales ha sido configurados únicos globalmente agregándoles el distintivo de ruta.

Ejemplos	`Router (config)#router bgp 1`
	`Router (config-router)#address-family vpnv4`
	`Router (config-router)#exit-address-family`

Tabla 2.22: Características del comando *address-family vpnv4*

PASO 7.5: Publicación del atributo de comunidad extendida.

La configuración de la familia de direcciones vpnv4 también añade un comando adicional a la configuración de BGP. Este comando es ***neighbor*** *ip-address* ***send-community*** *extended*, se añade por defecto y es necesario porque instruye a BGP a publicar el atributo de comunidad extendida.

El comportamiento predeterminado es mandar solamente el atributo de comunidad extendida, así que no es necesario introducir el comando si es lo que es lo que requerimos. Si el diseño de la red requiere que se adjunte el atributo de comunidad estándar a las rutas VPN, se debe cambiar la configuración predeterminada utilizando el comando ***neighbor*** *ip-address* ***send-community*** *both.*

Comando	***neighbor*** *ip-address* ***send-community*** *extended*
Modo	Router configuration
Sintaxis	***neighbor*** *ip-address* ***send-community*** *extended|standar|both*
Descripción	*ip-address:* Dirección IP del vecino BGP. *both:* Envía los atributos de comunidad extendida y estándar.
Ejemplos	`Router (config)#router bgp 1` `Router (config-router)#neighbor 160.89.2.3 send-community extended`

Tabla 2.23: Características del comando *neighbor ip-address send-community extended 4*

2.2.2 CONFIGURACIÓN DEL ROUTER LSR (P).

La configuración del router LSR (P) es muy similar a la de los routers de contorno excepto por la configuración de las VPNs y el protocolo de enrutamiento BGP. Esto se nota claramente en el ejemplo de la sección 2.2.3

2.2.3 RESULTADOS DE LA CONFIGURACIÓN.

A continuación se muestra la red obtenida en la figura 2.4, luego los archivos de configuración de cada router; los comandos introducidos se detallan en los anexos debidamente numerados. Los sitios A y B de los clientes a y b serán simulados con interfaces de loopback.

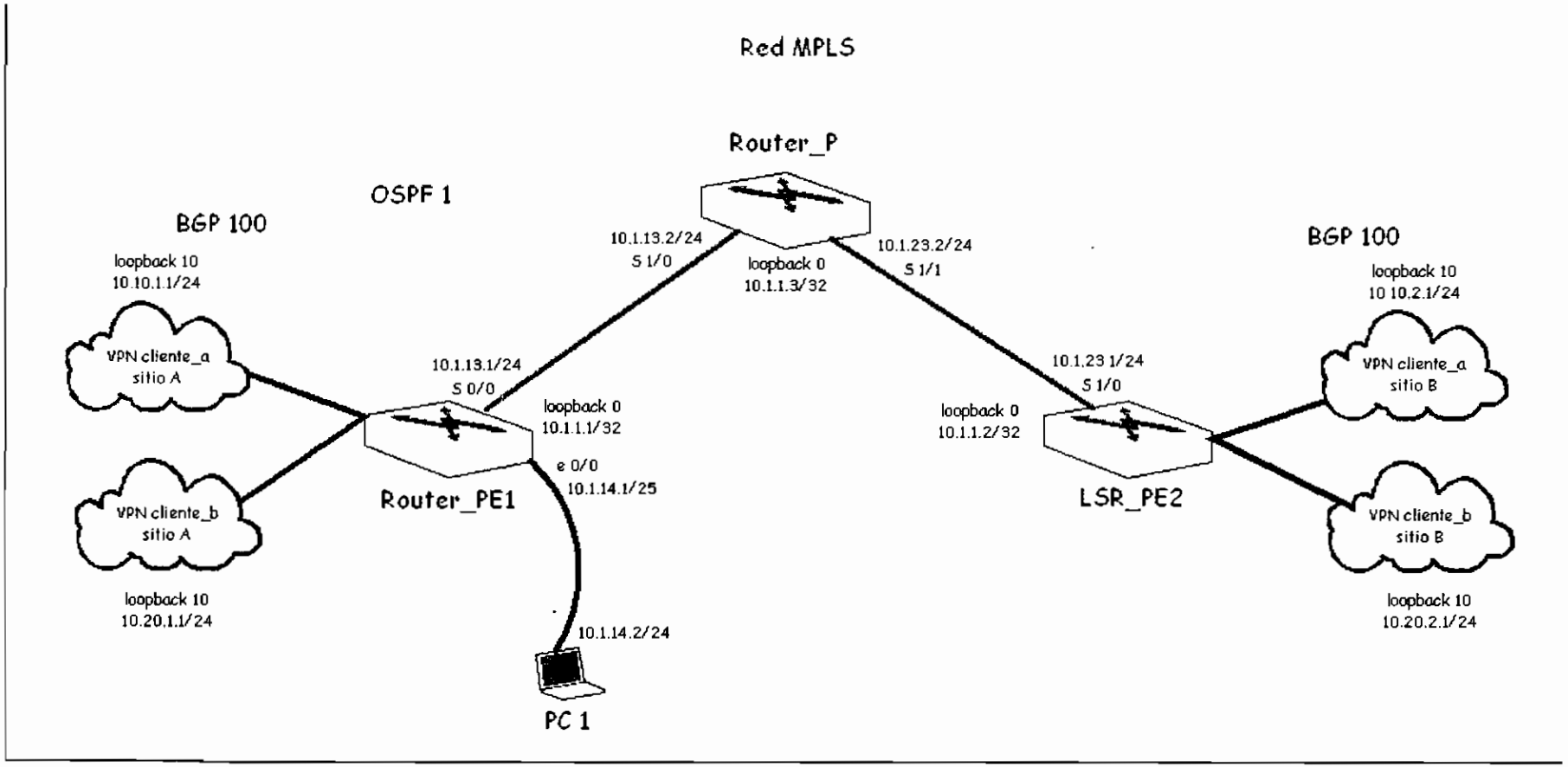


Figura 2.4: Resultados de la configuración VPN/MPLS

Archivo de configuración

ROUTER Router_P (Cisco 3640)

```
Router_P#show running-config
Building configuration...

Current configuration : 1227 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router_P
!
boot-start-marker
boot-end-marker
!
enable password p
!
no aaa new-model
ip subnet-zero
!
!
!
ip cef
ip audit po max-events 100
mpls label protocol ldp
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
!
!
!
interface Loopback0
 ip address 10.1.1.3 255.255.255.255
 no clns route-cache
!
interface Ethernet0/0
 no ip address
 shutdown
 half-duplex
 no clns route-cache
!
interface TokenRing0/0
 no ip address
 shutdown
 ring-speed 16
 no clns route-cache
!
interface Serial1/0
 ip address 10.1.13.2 255.255.255.0
 tag-switching ip
 no clns route-cache
!
interface Serial1/1
 ip address 10.1.23.2 255.255.255.0
 tag-switching ip
 clockrate 56000
 no clns route-cache
!
interface Serial1/2
 no ip address
 shutdown
```

```
 no clns route-cache
!
interface Serial1/3
 no ip address
 shutdown
 no clns route-cache
!
router ospf 1
 log-adjacency-changes
 network 10.1.0.0 0.0.255.255 area 0
!
ip http server
no ip http secure-server
ip classless
!
!
!
!
line con 0
 password p
 login
line aux 0
line vty 0 4
 password p
 login
!
!
end

Router_P#
Router_P#
```

Archivo de configuración

ROUTER Router_PE1 (Cisco 2610)

```
Router_PE1#show running-config
Building configuration...

Current configuration : 2075 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router_PE1
!
boot-start-marker
boot-end-marker
!
enable password pe1
!
memory-size iomem 10
no aaa new-model
ip subnet-zero
!
!
ip cef
ip vrf cliente_a
 rd 100:110
 route-target export 100:1000
```

```
 route-target import 100:1000
!
ip vrf cliente_b
 rd 100:120
 route-target export 100:2000
 route-target import 100:2000
!
mpls label protocol ldp
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
!
!
!
interface Loopback0
 ip address 10.1.1.1 255.255.255.255
!
interface Loopback10
 description VPN cliente_a
 ip vrf forwarding cliente_a
 ip address 10.10.1.1 255.255.255.0
!
interface Loopback20
 description VPN cliente_b
 ip vrf forwarding cliente_b
 ip address 10.20.1.1 255.255.255.0
!
interface Ethernet0/0
 ip address 10.1.14.1 255.255.255.0
 half-duplex
!
interface Serial0/0
 ip address 10.1.13.1 255.255.255.0
 tag-switching ip
 clockrate 56000
 no fair-queue
!
interface Serial0/1
 no ip address
 shutdown
!
router ospf 1
 log-adjacency-changes
 network 10.1.0.0 0.0.255.255 area 0
!
router bgp 100
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 10.1.1.2 remote-as 100
 neighbor 10.1.1.2 update-source Loopback0
 !
 address-family ipv4
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.1.1.2 activate
 neighbor 10.1.1.2 route-reflector-client
 neighbor 10.1.1.2 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf cliente_b
 redistribute connected
 no auto-summary
```

```
 no synchronization
 exit-address-family
 !
 address-family ipv4 vrf cliente_a
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
!
ip http server
ip classless
!
!
!
!
line con 0
 password pe1
 login
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
 password pe1
 login
 transport preferred all
 transport input all
 transport output all
!
end

Router_PE1#
Router_PE1#
Router_PE1#
```

Archivo de configuración

ROUTER Router_PE2 (Cisco 2610)

```
Router_PE2#show running-config
Building configuration...

Current configuration : 1972 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router_PE2
!
boot-start-marker
boot-end-marker
!
enable password pe2
!
no aaa new-model
ip subnet-zero
!
!
ip cef
ip vrf cliente_a
```

```
 rd 100:110
 route-target export 100:1000
 route-target import 100:1000
!
ip vrf cliente_b
 rd 100:120
 route-target export 100:2000
 route-target import 100:2000
!
mpls label protocol ldp
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
!
!
!
interface Loopback0
 ip address 10.1.1.2 255.255.255.255
!
interface Loopback10
 ip vrf forwarding cliente_a
 ip address 10.10.2.1 255.255.255.0
!
interface Loopback20
 ip vrf forwarding cliente_b
 ip address 10.20.2.1 255.255.255.0
!
interface Ethernet0/0
 no ip address
 shutdown
 half-duplex
!
interface Serial0/0
 no ip address
 shutdown
 no fair-queue
!
interface Serial0/1
 ip address 10.1.23.1 255.255.255.0
 tag-switching ip
!
router ospf 1
 log-adjacency-changes
 network 10.1.0.0 0.0.255.255 area 0
!
router bgp 100
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 10.1.1.1 remote-as 100
 neighbor 10.1.1.1 update-source Loopback0
 !
 address-family ipv4
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.1.1.1 activate
 neighbor 10.1.1.1 route-reflector-client
 neighbor 10.1.1.1 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf cliente_b
 redistribute connected
 no auto-summary
```

```
 no synchronization
 exit-address-family
 !
 address-family ipv4 vrf cliente_a
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
!
ip http server
ip classless
!
!
!
!
line con 0
 password pe2
 login
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
 password pe2
 login
 transport preferred all
 transport input all
 transport output all
!
end

Router_PE2#
```

Configuración: Router_P (Anexo2.2.3.a)

Configuración: Router_PE1 (Anexo2.2.3.b)

Configuración: Router_PE2 (Anexo2.2.3.c)

2.3 CONFIGURACIÓN DE CoS PARA BRINDAR QoS A LA RED MPLS.

La característica de Clase de Servicio usada en conjunto con MPLS permite al administrador de red proveer servicios diferenciados a través de una red MPLS. Los servicios diferenciados satisfacen un rango de requerimientos proveyendo para cada paquete transmitido la clase de servicio especificada para el paquete por su CoS. El servicio puede ser especificado de diferentes maneras, y uno de los más usados es usando el Bit de precedencia IP colocado en los paquetes IP.

MPLS/CoS soporta los siguientes servicios: packet classification (CAR), congestion avoidance (WRED) y congestion management (WFQ). En la tabla 2.3 se muestra los diferentes tipos de servicio con sus respectivas funciones CoS asociadas.

Servicio	Función CoS	Descripción
Packet Classification	Commited Access Rate (CAR). Los paquetes son clasificados en el borde de la red antes de que los paquetes sean etiquetados	**CAR utiliza los bits TOS (type of service) en la cabecera IP para clasificar los paquetes de acuerdo a las tasas de transmisión de entrada y salida. CAR es a menudo configurado en interfaces en el borde de una red con el propósito de controlar el tráfico entrante o saliente de la red. Usted puede utilizar los comandos de clasificación CAR para clasificar o reclasificar un paquete.**
Congestion avoidance	Weighted Random Early Detection (WRED). Las clases de paquetes son diferenciadas en base a su probabilidad de ser desechados (drop probability)	**WRED monitorea el tráfico de red, tratando de anticipar o prevenir congestión y cuellos de botella en la misma. WRED puede selectivamente descartar tráfico de prioridad más baja cuando una interfaz comienza a congestionarse. Esto puede también proveer características de rendimiento diferenciado para diferentes clases de servicio.**
Congestion management	**Weighted Fair Queueing (WFQ). Las clases de paquetes son diferenciadas en base al ancho de banda y retardo definido.**	**WFQ es un sistema de programación automatizado que provee el ancho de banda asignado a todo el tráfico de la red. WFQ utiliza pesos (prioridades) para determinar cuanto ancho de banda es asignado a cada clase de tráfico.**

Tabla 2.24: Servicios CoS y características

Como se mencionó anteriormente, existen algunos métodos diferentes para brindar CoS a través de un backbone MPLS, la elección depende de si el Core está formado por LSRs o ATM LSRs. Para este proyecto disponemos de LSRs como Core del backbone la operación de este backbone se detalla a continuación:

Operación:

Los LSRs de un backbone MPLS por lo general son routers Cisco de las series 7200 y 7500 en los cuales debe estar corriendo software MPLS (debido a que la red que se está configurando no se trata de un backbone de las dimensiones tales que amerite la utilización de un router de las series 7200 o 7500, se puede utilizar el router Cisco 3640 como LSR, con la condición de que en éste se esté corriendo software MPLS), otras series que soportan CoS son 4500, 3600 y 2600.

En una red MPLS que soporta CoS los paquetes son procesados de la siguiente manera:

1. Los paquetes IP ingresan al borde de la red MPLS.

2. Los LSRs de contorno invocan CAR para clasificar los paquetes IP y posiblemente fijar la precedencia IP. Como alternativa, los paquetes IP pueden ser recibidos con su precedencia IP previamente fijada.

3. Por cada paquete, el router ejecuta un lookup en la dirección IP para determinar el LSR de próximo salto.

4. La etiqueta apropiada es colocada en el paquete con los bits de precedencia IP copiados dentro de cada etiqueta ingresada en la cabecera MPLS.

5. EL paquete etiquetado es entonces enviado a la interfaz de salida apropiada para su procesamiento.

6. Los paquetes son diferenciado por clase, Esto es realizado de acuerdo a la probabilidad de baja (WRED) o al ancho de banda y retardo (WFQ). En ambos casos, los LSRs implementan la

diferenciación definida continuando con el uso WRED o WFQ en cada salto.

Requisitos:

Para utilizar la característica CoS/MPLS, la red debe estar corriendo las siguientes características del IOS de Cisco:

- CEF en cada router que se habilitará para MPLS
- MPLS
- El usuario debe estar familiarizado con la operación básica y terminología MPLS.
- Cisco IOS® Software Releases 12.3(12) o superior.
- Mínimo plataforma 3600 para LSR
- Mínimo plataforma 2610 para LSR de contorno

2.3.1 CONFIGURACIÓN DE LOS ROUTERS LSR DE CONTORNO CISCO 2600 (PE).

PASO 1: Habilitar MPLS

Para realizar la configuración MPLS se debe seguir los pasos del 1 al 9 de la sección 2.1.1.

PASO 2: Configurar el campo EXP MPLS (experimental MPLS)

Configurando el valor del campo EXP se satisface el requerimiento de los proveedores de servicio quienes no quieren que el valor de la precedencia IP sea alterado dentro de los paquetes IP que son transportados a través de sus redes.

Escogiendo diferentes valores para el campo EXP, se puede marcar los paquetes basándose en sus características, tales como tasa o tipo y de esa forma los paquetes tienen la prioridad que requieren durante los periodos de congestión.

Para clasificar los paquetes IP, se debe configurar el router LSR de ingreso. Los paquetes IP son recibidos por el router de ingreso y transmitidos como paquetes MPLS. Esta configuración se la puede realizar de dos formas:

- *Modular QoS CLI.*- se utiliza este método cuando no se quiere tomar en cuenta la tasa de los paquetes que se recibe por el router de ingreso.
- *CAR.*- se utiliza CAR cuando se desea considerar la tasa de los paquetes entrantes.

Utilizaremos CAR como método para configurar el campo EXP debido a que los equipos que disponemos solo soportan este método.

PASO 2.1: Configurar EXP usando CAR.

Para configurar el LSR de ingreso usando CAR haga lo siguiente:

1.- Configure una lista de acceso *IP rate-limit* para clasificar los paquetes de acuerdo a su precedencia IP
2.- Configure *rate-limit* en la interfaz de entrada para setear los paquetes MPLS (escribe la clasificación de los paquetes dentro del campo EXP MPLS).

PASO 2.1.1: Configurar una lista de acceso *IP rate-limit.*

Para configurar una lista de acceso que será utilizada con políticas CAR (committed access rate) utilice el comando ***access-list rate-limit*** *acl-index precedence* en el modo de configuración global. Para remover la lista de acceso use la forma ***no*** del comando.

Comando	***access-list rate-limit***
Modo	Configuración global

Sintaxis	*access-list* ***rate-limit*** *acl-index precedence*
Descripción	*acl-index:* Especifica el número de la lista de acceso para clasificar los paquetes. *precedence:* Especifica la precedencia IP. Valores entre 0 y 7.
Ejemplos	`Router (config)# access-list rate-limit 24 4` `Router (config)# end`

Tabla 2.25: Características del comando *access-list rate-limit acl-index precedence*

PASO 2.1.2: Configurar Rate-Limit en la interfaz de entrada.

Para configurar las políticas CAR, utilice el comando ***rate-limit*** en el modo de configuración de interfaz. Para remover el rate-limit desde la configuración, utilice la forma ***no*** del comando. En el ejemplo de la tabla 2.3.1.2.1.2, el campo exp de los paquetes MPLS salientes es seteado como 4 si los paquetes IP entrantes coinciden con la lista de acceso y están dentro del tasa especificada. El campo exp es seteado como 0 si los paquetes coinciden con la lista de acceso pero exceden la tasa especificada.

Comando	***rate-limit***
Modo	Configuración de interfaz
Sintaxis	***rate-limit input [access-group [rate-limit]****acl-index] bps burst-normal burst-max* ***conform-action set-mpls-exp-transmit*** *exp* ***exceed-action set-mpls-exp-transmit*** *exp* **rate-limit** {**input** | **output**} [**access-group** [**rate-limit**] *acl-index*] *bps burst-normal burst-max* **conform-action** *conform-action* **exceed-action** *exceed-action*
Descripción	*input* | *output:* Aplica esta política de tráfico CAR a los paquetes recibido/enviados en esta interfaz de entrada/salida. *acl-index:* Especifica el número de la lista de acceso para clasificar los paquetes. *bps:* Tasa promedio, en bps. El valor debe ser en incrementos de 8Kbps. *burst-normal:* Tamaño del burst normal en bytes. El valor mínimo

	es bps dividido para 2000. *burst-max:* Exceso del tamaño del burst en bytes. *conform-action:* Acción que se tomará con los paquetes que conforman la tasa limite especificada. *exp:* Especifica el campo experimental MPLS. Los valores válidos están enre 0 y 7. *exceed-action:* Acción que se tomará con los paquetes que exceden la tasa limite especificada.

Ejemplos	`Router(config)# interface et 1/0/0`
	
	`Router(config-int)# rate-limit input access-group rate-limit 24 8000 8000 8000 conform-action set-mpls-exp-transmit 4 exceed-action set-mpls-exp-transmit 0`
	
	`Router(config-int)# end`

Tabla 2.26: Características del comando *rate-limit*

NOTA: Los valores de la precedencia IP y el campo EXP tienen la misma prioridad, es decir, que un paquete con valor de precedencia IP = 3 tiene la misma prioridad que un paquete etiquetado con campo EXP = 3.

2.3.2 CONFIGURACIÓN DEL ROUTER LSR CISCO 3640 (P).

La configuración del router LSR (P) es muy similar a la de los router P de la sección 2.2. Su configuración se detalla más adelante.

2.3.3 RESULTADOS DE LA CONFIGURACIÓN.

A continuación se muestra la red obtenida en la figura 2.5, luego los archivos de configuración de cada router; los comandos introducidos se detallan en los anexos debidamente numerados. La característica Qos configurada indica que los paquetes que ingresen a la red con precedencia IP = 4 mantendrán su prioridad dentro de la nube MPLS con el campo EXP = 4

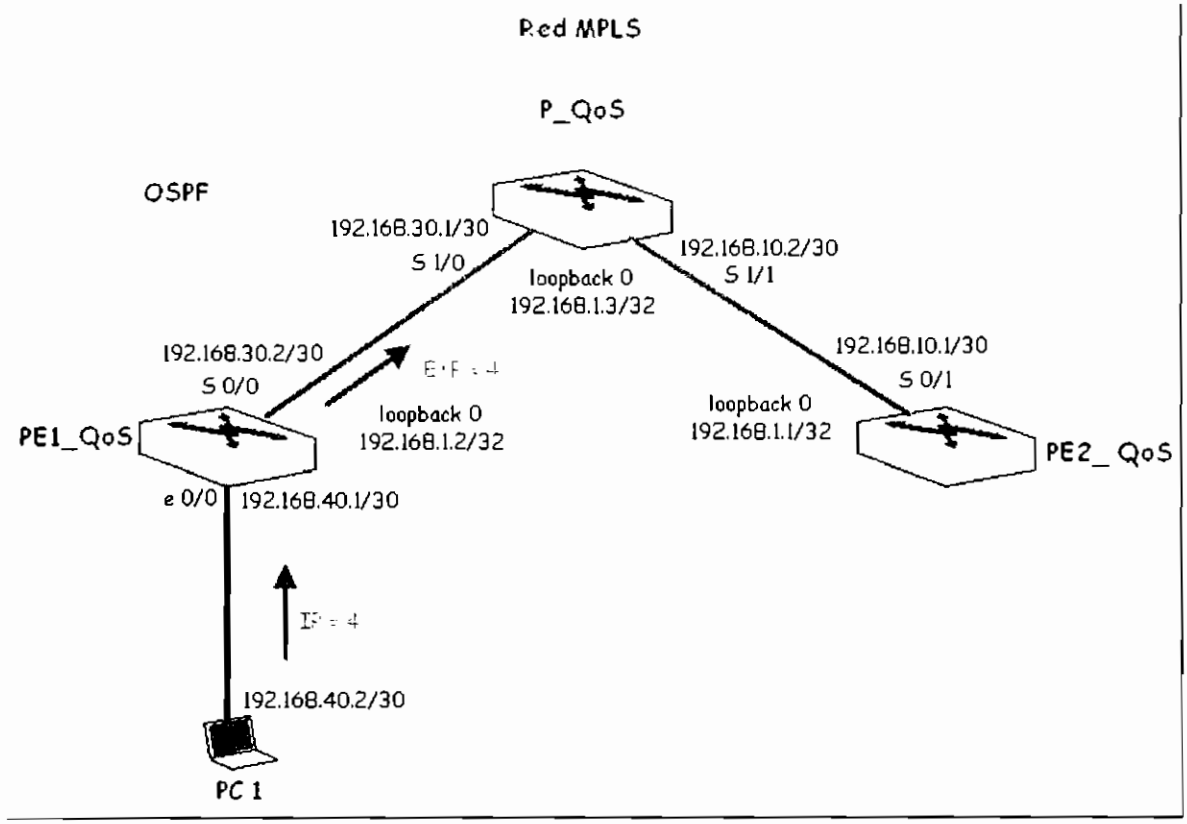


Figura 2.5: Red MPLS con QoS

Archivo de configuración

ROUTER P_QoS (Cisco 3640)

```
P_QoS#show running-config
Building configuration...

Current configuration : 1236 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P_QoS
!
boot-start-marker
boot-end-marker
!
enable password p
!
no aaa new-model
ip subnet-zero
!
!
ip cef
```

```
ip audit po max-events 100
mpls label pr
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
!
!
interface Loopback0
 ip address 192.168.1.3 255.255.255.255
 no clns route-cache
!
interface Ethernet0/0
 no ip address
 shutdown
 half-duplex
 no clns route-cache
!
interface TokenRing0/0
 no ip address
 shutdown
 ring-speed 16
 no clns route-cache
!
interface Serial1/0
 ip address 192.168.30.1 255.255.255.0
 tag-switching ip
 no clns route-cache
!
interface Serial1/1
 ip address 192.168.10.2 255.255.255.0
 tag-switching ip
 clockrate 56000
 no clns route-cache
!
interface Serial1/2
 no ip address
 shutdown
 no clns route-cache
!
interface Serial1/3
 no ip address
 shutdown
 no clns route-cache
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
ip http server
no ip http secure-server
ip classless
!
!
!
line con 0
 password p
 login
line aux 0
line vty 0 4
 password p
 login
!
!
end

P_QoS#
```

Archivo de configuración

ROUTER PE1_QoS (Cisco 2610)

```
PE1_QoS#show running-config
Building configuration...

Current configuration : 1019 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE1_QoS
!
boot-start-marker
boot-end-marker
!
enable password pe1
!
memory-size iomem 10
no aaa new-model
ip subnet-zero
!
!
ip cef
mpls label protocol ldp
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
!
!
!
interface Loopback0
 ip address 192.168.1.2 255.255.255.255
!
interface Ethernet0/0
 ip address 192.168.40.1 255.255.255.0
 rate-limit input access-group rate-limit 24 8000 8000 8000
 conform-action set-mpls-exp-imposition-transmit 4 exceed-
 action set-mpls-exp-imposition-transmit 0
 half-duplex
!
interface Serial0/0
 ip address 192.168.30.2 255.255.255.0
 tag-switching ip
 clockrate 56000
 no fair-queue
!
interface Serial0/1
 no ip address
 shutdown
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
ip http server
ip classless
!
!
!
```

```
access-list rate-limit 24 4
!
!
!
line con 0
 password pe1
 login
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
 password pe1
 login
 transport preferred all
 transport input all
 transport output all
!
end

PE1_QoS#
```

Archivo de configuración

ROUTER PE2_QoS (Cisco 2610)

```
PE2_QoS#show running-config
Building configuration...

Current configuration : 1060 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE2_QoS
!
boot-start-marker
boot-end-marker
!
enable password pe2
!
no aaa new-model
ip subnet-zero
!
!
ip cef
mpls label protocol ldp
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
!
!
!
interface Loopback0
 ip address 192.168.1.1 255.255.255.255
!
```

```
interface Ethernet0/0
 no ip address
 shutdown
 half-duplex
!
interface Serial0/0
 no ip address
 shutdown
!
interface Serial0/1
 ip address 192.168.10.1 255.255.255.0
  tag-switching ip
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
ip http server
ip classless
!
!
line con 0
 password pe2
 login
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
 password pe2
 login
 transport preferred all
 transport input all
 transport output all
!
end

PE2_QoS#
```

Configuración: P_QoS (Anexo 2.3.3.a)

Configuración: PE1_QoS (Anexo 2.3.3.b)

Configuración: PE2_QoS (Anexo 2.3.3.c)

2.4 SIMULACIÓN. •

La simulación es una técnica que permite inferir las características de un sistema real mediante el estudio de un modelo que imita su comportamiento cuando la experimentación con el sistema real es más costosa o imposible. La simulación no es un ejercicio de programación que da una respuesta exacta.

La primera aplicación de simulación informática a gran escala se llevó a cabo en el proyecto Manhattan que dirigió EEUU en la década de los cuarenta, cuando modeló el proceso de la detonación nuclear.

Desde entonces, las técnicas de simulación se han desarrollado mano a mano con la rápida evolución de la informática hasta tal punto que hoy en día no se utilizan solamente como herramientas fundamentales en ámbitos de vanguardia de la ciencia y el descubrimiento, sino que contribuyen igualmente al éxito de taquilla de las últimas superproducciones cinematográficas.

Existen muchas razones por las que la simulación es importante, tanto en el ámbito de la ciencia como en el de la empresa, la simulación ayuda a explorar y comprender sistemas complejos. No es cuestión de predecir el futuro, sino de intentar comprender la complejidad o las normas subyacentes.

La mayoría de los sistemas complejos no son lineales, lo que hace muy difícil predecir el comportamiento futuro, por el contrario, tienden a ser puntos masivos críticos, por ejemplo en la divulgación de una idea o la propagación de un virus, cuando el sistema comienza a mostrar nuevos o extraños comportamientos. A través de la simulación se puede ver cuándo ocurren estos puntos masivos críticos.

Simulación es también la experimentación con un modelo de una hipótesis de trabajo. La experimentación puede ser un trabajo de campo o de laboratorio. El modelo de método usado para la simulación seria teórico, conceptual o sistémico.

• Referencia bibliográfica: http://icadc.cordis.europa.eu.int/fep-cgi/srchidadb?CALLER=ES_NEWS&ACTION=D&SESSION=&RCN=25109 [18]

Después de confirmar la hipótesis se puede ya diseñar un teorema. Finalmente si este es admitido puede convertirse en una teoría o en una ley.

El modelo teórico debe contener los elementos que se precisen para la simulación. Un ejemplo con trabajo de laboratorio es un programa de estadística con ordenador que genere números aleatorios y que contenga los estadísticos de la media y sus diferentes versiones: cuadrática- aritmética-geométrica-armónica. Además debe ser capaz de determinar la normalidad en términos de probabilidad de las series generadas. La hipótesis de trabajo es que la media y sus versiones también determinan la normalidad de las series. Es un trabajo experimental de laboratorio. Si es cierta la hipótesis se puede establecer la secuencia teorema, teoría, ley.

El modelo conceptual desea establecer por un cuestionario y con trabajo de campo, la importancia de la discriminación o rechazo en una colectividad y hacerlo por medio de un cuestionario en forma de una simulación con una escala de actitud. Después de ver si la población es representativa o adecuada, ahora la simulación es la aplicación del cuestionario y el modelo es el cuestionario para confirmar o rechazar la hipótesis de si existe discriminación en la población y hacia que grupo de personas y en que cuestiones. Gran parte de las simulaciones son de este tipo con modelos conceptuales.

El modelo sistémico es más pretencioso y es un trabajo de laboratorio. Se simula el sistema social en una de sus representaciones totales. El análisis de sistemas es una representación total. Un plan de desarrollo en el segmento de transportes con un modelo de ecología humana, por ejemplo. El énfasis en la teoría general de sistemas es lo adecuado en este tipo de simulaciones. Este método, que es para un Sistema complejo, es sumamente abstracto, no se limita a la descripción del sistema, sino que debe incluir en la simulación las entradas y salidas de energía y retroalimentación.

Tanto el programa de estadística, la escala de actitud, como el sistema total, son perfectas simulaciones de la realidad y modelizan todos los elementos en sus respectivas hipótesis de trabajo. Son también un microclima y el ambiente

o el escenario en los procesos de simulación/experimentación. Otras propiedades que deben contener las simulaciones es que sean repetibles indefinidamente. Que eviten el efecto de aprendizaje que incita al encuestador a rellenar el mismo los cuestionarios y que se podrá evitar con algún control, que sean flexibles o mejorables y que no sea invasivo o cambiar la población de las muestras sucesivas.

En el mercado se encuentran varias herramientas de simulación las mismas que se diferencian entre otras cosas por sus limitantes y costo. Entre las más importantes tenemos: Network Simulator (NS) y OMNeT ++ que se describen brevemente a continuación:

Network Simulator (NS):

Network Simulator es un simulador de eventos discretos que está basado en dos lenguajes: un simulador escrito en C++, y una extensión de TCL (orientada a objetos) que sirve para ejecutar los "scripts". Existe un "linkage" entre los scripts y el simulador, que crea una correspondencia entre los objetos de OTcl y los de C++. Los usuarios crean nuevos objetos de simulación a través del intérprete, que son instanciados dentro de éste, y posteriormente "mirrored" a la jerarquía compilada. NS dispone de una gran libraría de redes y protocolos, y es un "estándar" en la simulación de redes de comunicaciones.

NS viene compilado con una librería "básica", que permite la realización casi inmediata de simulaciones TCP/IP bastante completas (ejemplo: FTP, TCP, fuentes de tráfico, RIP, OSPF...), sin necesidad de compilar código C++. Incluso es posible implementar algoritmos en lenguaje interpretado, aunque ello conlleva un tiempo de ejecución mayor. La instalación de nuevos protocolos o elementos precisa de recompilar todo el entorno, para el enlace de los nuevos objetos en el código e implementar el comportamiento. Si bien la ampliación del entorno con modelos probados es más o menos directa, la creación de nuevas funcionalidades implica desarrollo en dos lenguajes diferentes interpretado y compilado), por lo que el proceso de depuración puede resultar muy tedioso.

Plataformas: Linux, Windows.

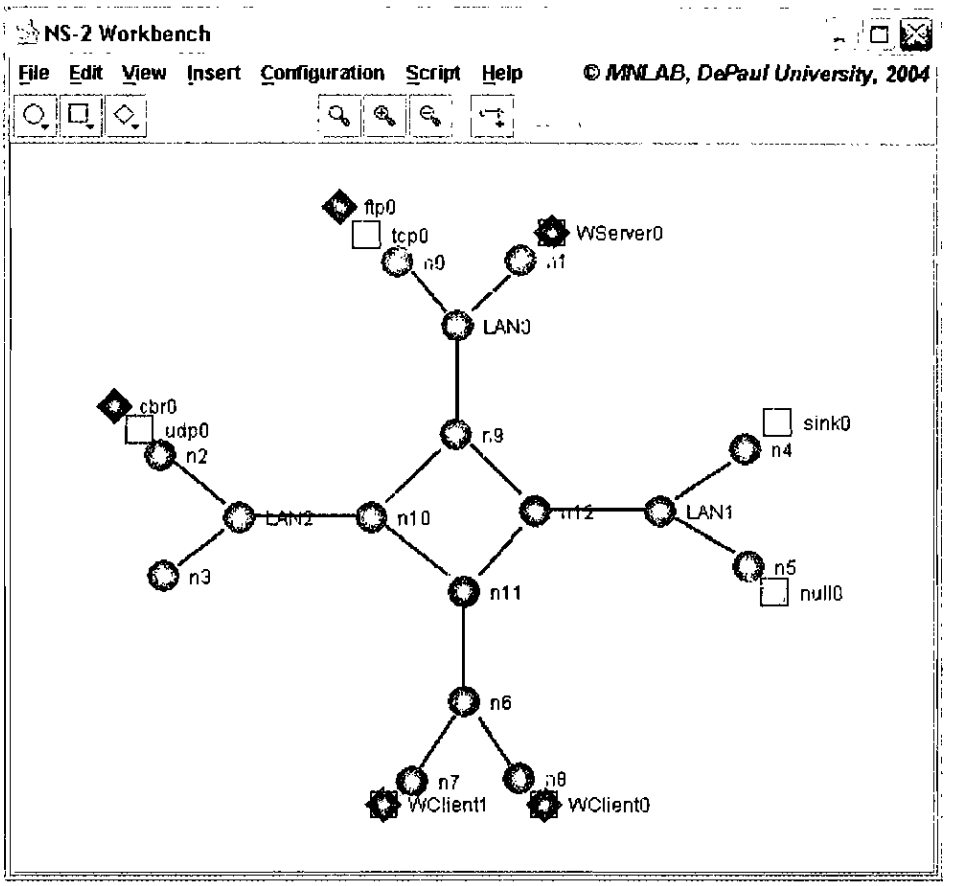


Figura 2.6: Simulador "Network Simulator"

Esta herramienta no será utilizada en este proyecto debido a que no se disponen de librerías que soporten MPLS.

OMNeT++:

OMNeT++ (Ojective Modular Network Testbed in C++) es un simulador de eventos discretos modular, orientado a objetos. Un modelo en Omnet consiste en módulos jerárquicamente anidados, que se comunican mediante paso de mensajes. Presenta dos interfaces de ejecución: gráfico y de comandos, lo que permite una depuración más sencilla. El modo gráfico permite una simulación paso a paso (de mensajes), o de forma contínua (con diferente granularidad). La interfaz visual, además, es una herramienta didáctica y de debugging fundamental (frente a la animación offline, que no aporta nada). Existe un gran esfuerzo desarrollador, tanto del entorno de simulación como de las librerías (IPv6, TCP, Mobility, etc). Con el entorno de Omnet se pueden ejecutar simulaciones

compiladas que requieran la herramienta gráfica (los ejecutables en modo línea de comandos son "stand-alone", lo que facilita su ejecución), mientras que para desarrollar nuevas es preciso implementar los módulos y compilarlos. Implementa la orientación a procesos y la orientación a eventos y forza a una jerarquización de los modelos (lo que facilita una programación ordenada, reutilización de código, etc.

Salvo que se disponga del ejecutable, para ejecutar cualquier simulación es preciso compilar primero el código fuente. Por ello, la "única" dificultad consiste en obtener dicho código fuente (bien por librerías disponibles, bien por desarrollo) y ejecutar un compilador de C++. Frente a Network Simulator, ello supone una gran ventaja (sólo un lenguaje). Además, las herramientas visuales son bastante útiles para depurar el código. Las librerías disponibles se encuentran, por lo general, muy bien documentadas. Debido al desarrollo modular de las mismas, es posible emplear únicamente aquellos bloques de interés.

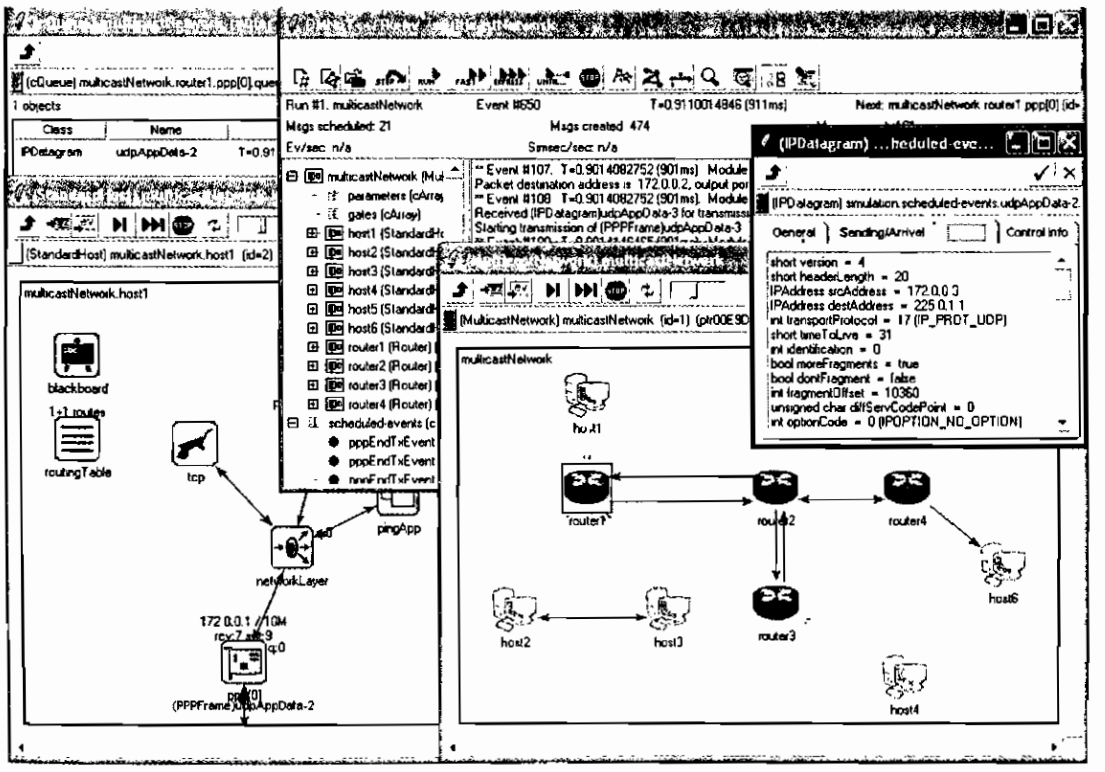


Figura 2.7: Simulador "OMNeT++"

Debido a que el costo de las librerías que soportan MPLS en el simulador *OMNet++* no pudo ser cubierto dentro del presupuesto del proyecto, no se utilizó esta herramienta para la simulación complementaria del mismo.

OpenSimMPLS 1.0:

Como complemento de este proyecto se vio la necesidad de simular escenarios que no se pueden configurar debido a que no se cuenta con las herramientas necesarias, éstos pueden ser fácilmente simulados. La herramienta que se utilizará para realizar la simulación es la aplicación "Open SimMPLS 1.0" que es un software libre cofinanciado en gran parte por el Excmo. Ayuntamiento de Zafra (Badajoz, ESPAÑA) y por la Universidad de Extremadura con fondos provenientes de la "Convocatoria 2003 de Ayudas para la realización de Memoria de Licenciatura o Proyecto Fin de Carrera y de mejores expedientes". Esta aplicación pretende conseguir dos objetivos principales:*

- La implementación de un simulador MPLS multilingüe y portable como plataforma para la prueba empírica de las conclusiones que se deriven de las investigaciones.
- Investigar los posibles caminos para poder dar soporte de garantía de servicio a flujos privilegiados sobre Multiprotocol Label Switching.

Este simulador no implementa todos los protocolos existentes ni el conjunto completo de características de MPLS. Estas son las características y funciones soportadas por Open SimMPLS 1.0

Tecnología

- TPC como payload de paquetes IP.
- IPv4 sobre MPLS.

---

* Referencia bibliográfica: http://patanegra.unex.es/opensimmpls [17]

- Flujos IPv4.
- Flujos MPLS.
- Flujos IPv4 marcado con Garantía de Servicio (GoS).
- Flujos MPLS marcados con Garantía de servicio (GoS).
- Tráfico constante.
- Tráfico variable con distribución de tamaños de paquetes real de Internet.
- Distribución de etiquetas bajo demanda.
- Soporte de TLDP.
- Soporte de GPSRP.
- Soporte de RLPRP.
- Soporte de RABAN.
- Implementación de DMGP.
- Soporte del algoritmo de Floyd tradicional.
- Algoritmo de gestión de búferes Round Robin Priorizado.
- Soporte de EPCD en los búfferes.
- Recuperación local de paquetes.
- Recuperación local de LSP.

Simulación

- Simulación completa del escenario.
- Simulación de LER, LER activos, LSR, LSR activos, emisores y receptores.
- Simulación de caídas de enlaces.
- Simulación de casos de congestión.
- Estadísticas completas de los nodos.
- Simulación del establecimiento de LSP y LSP de respaldo.
- Simulación de paquetes descartados.
- Impresión de gráficas estadísticas.
- Exportación de gráficas estadísticas a imágenes PNG .
- Simulación del retardo de los enlaces.
- Simulación de la recuperación de paquetes.
- Simulación de cada tipo de tráfico.

En este proyecto se utilizarán todas las características del simulador que ayuden a complementar los objetivos que se plantearon. Se simulará un backbone MPLS en situaciones como pérdidas de enlace, reenrutamiento y congestión, para lo cual se diseñarán tres casos de estudios los cuales nos ayudarán a familiarizarse con la aplicación y a simular escenarios con las características antes mencionadas.

Figura 2.8: Simulador "OpenSimMPLS 1.0"

CAPITULO III

3 PRUEBAS Y DISEÑO DE CASOS DE ESTUDIO

En el capítulo anterior se describió la configuración de los equipos y durante el desarrollo de éste se vio la necesidad de elaborar tres casos de estudio, dedicados uno a la configuración inicial de un router Cisco y dos a la configuración de los protocolos de enrutamiento utilizados en este proyecto (RIPv2, OSPF, BGP).

Los Casos de Estudio desarrollados en este capítulo constan de dos partes: una de configuración y otra de pruebas las mismas que contribuyen a esclarecer y comprobar el correcto funcionamiento de la red implementada. Así mismo, se detallan los requisitos necesarios para llevar a cabo el desarrollo de dichos casos.

Los comandos utilizados en cada una de las configuraciones de detallan en los respectivos anexos y los archivos de configuración están listados luego del esquema de configuración de cada router en la sección de configuración de cada Caso de Estudio. La aplicación utilizada para elaborar los diagramas de red es el SmartDraw7 en la misma que hemos utilizado las convenciones mostradas en la figura 3:

Figura 3.1: Elemento utilizados en el SmartDraw7

CASO DE ESTUDIO 1

CONFIGURACIÓN INICIAL DE UN ROUTER CISCO

DESCRIPCIÓN GENERAL Y OBJETIVOS

Este primer caso de estudio servirá para familiarizar al usuario con la terminología y la configuración inicial de un equipo Cisco. Configuraremos desde el nombre del router hasta las interfaces, cabe indicar, que la secuencia de configuración inicial que utilizaremos es la que ha dado mejores resultados en el desarrollo del proyecto pero no la única.

La red de la figura 3.2 es la red a configurarse, consta de dos routers Cisco 2610.

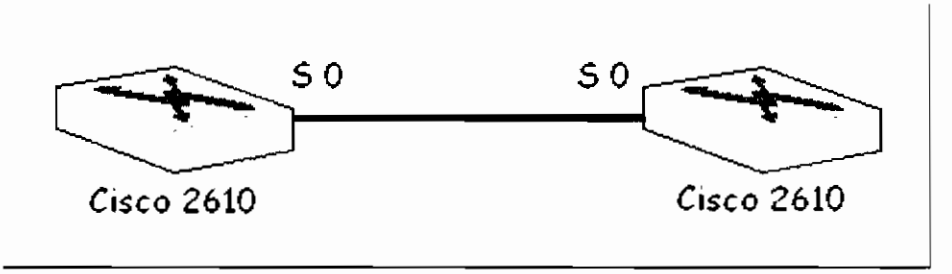


Figura 3.2: Red utilizada en el Caso de Estudio 1

TRABAJO PREPARATORIO

Para desarrollar este Caso de Estudio se deben tener conocimientos básicos sobre el funcionamiento de una red de comunicaciones, también se debe revisar los comandos utilizados para realizar la configuración básica de un router estos son: *enable, configure terminal, hostname, enable password, line console, line vty, interface* y *copy.*

REQUISITOS

La empresa IntelNet desea comunicar sus dos oficinas que se encuentran en edificios distintos y que pertenecen a redes diferentes, para lo cual dispone de dos routers Cisco 2610 y desea utilizar una conexión serial entre estos, una vez finalizada la configuración los routers deben comunicarse entre sí y se debe poder acceder mediante *Telnet* desde y hacia cualquiera de ellos.

En este caso de estudio no configuraremos el protocolo de enrutamiento, ésto se lo hará en el Caso de Estudio 2.

CONFIGURACIÓN

Para lograr una configuración exitosa, se debe planificarla en base a los objetivos y requerimientos antes mencionados, para lo cual se apoya en esquemas de configuración que se utilizan para registrar los datos que se deben configurar. Estos esquemas son de gran ayuda en la administración de la red y para solucionar problemas que podrían presentarse.

ROUTER 1 (CISCO 2610)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router 1:

ROUTER 1 (Cisco 2610)			
Nombre	Kimera		
Enable password	kimera123		
Console/VTY password	kimera		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 0	10.0.1.1/16	DTE

Archivo de configuración

KIMERA

```
Kimera#show running-config
Building configuration...

!
Version 12.1
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Kimera
enable password kimera123
!
ip subnet-zero
!
!
!
!
interface Serial0
 ip address 10.0.1.1 255.255.0.0
 no ip directed-broadcast
 bandwidth 1544
!
interface Serial1
 no ip address
 no ip directed-broadcast
 bandwidth 1544
 shutdown
!
interface Ethernet0
 no ip address
 no ip directed-broadcast
 bandwidth 10000
 shutdown
!
!
ip classless
no ip http server
!
!
!
!
line con 0
 login
 transport input none
 password kimera
line aux 0
line vty 0 4
 login
 password kimera
!
no scheduler allocate
end

Kimera#
```

Configuración KIMERA: Anexo 3.1.a

ROUTER 2 (CISCO 2610)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router 2:

ROUTER 2 (Cisco 2610)			
Nombre	Tekra		
Enable password	tekra123		
Console/VTY password	Tekra		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 0	10.0.1.2/16	DCE

Archivo de configuración

TEKRA

```
Tekra#show running-config
Building configuration...
!
Version 12.1
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Tekra
enable password tekra123
!
!
!
ip subnet-zero
!
interface Serial0
 ip address 10.0.1.2 255.255.0.0
 no ip directed-broadcast
 clock rate 56000
 bandwidth 1544
!
interface Serial1
 no ip address
 no ip directed-broadcast
 bandwidth 1544
 shutdown
!
interface Ethernet0
 no ip address
 no ip directed-broadcast
 bandwidth 10000
 shutdown
!
!
ip classless
no ip http server
```

```
!
line con 0
 login
 transport input none
 password tekra
line aux 0
line vty 0 4
 login
 password tekra
!
no scheduler allocate
end
```

Configuración TEKRA: Anexo 3.1.b

PRUEBAS

Una vez terminada la configuración se obtendrá la red de la figura 3.3, a la cual se le realizará un proceso de verificación para comprobar que la configuración trabaje correctamente, para esto, utilizaremos los comandos *ping* y *telnet* del IOS de Cisco de la siguiente manera:

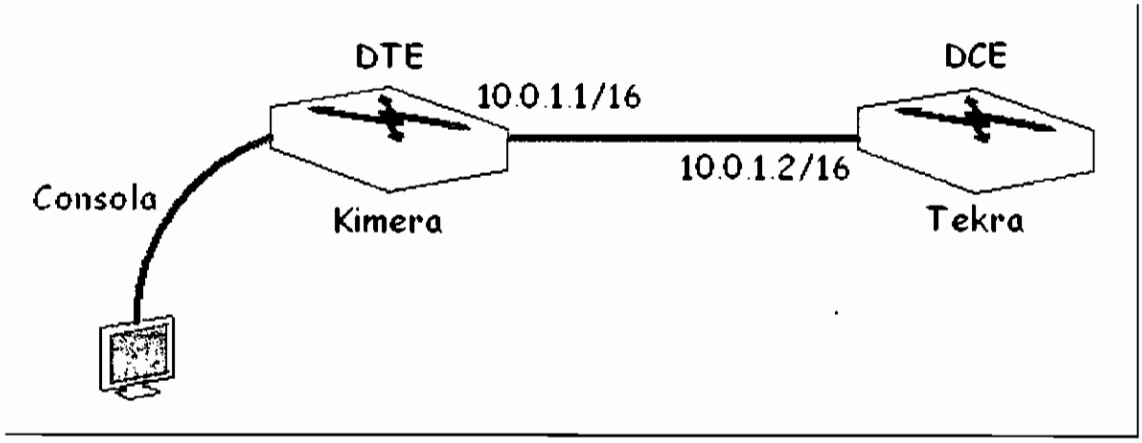


Figura 3.3: Red configurada en el Caso de Estudio 1

Proceso de pruebas:

Los resultados de la verificación se muestran seguidos del comando utilizado. El comando ***ping*** *(Packet InterNet Groper)* es un método muy común para verificar la accesibilidad hacia los dispositivos y utiliza mensajes ICMP para determinar si un host remoto está activo o inactivo, los retardos en la comunicación con el host remoto y si hay o no pérdida de paquetes. El comando

telnet (TCP, puerto 23) permite utilizar una máquina como terminal virtual de otra a través de la red, de forma que se crea un canal virtual de comunicaciones*.

- ***Desde el router KIMERA:***

```
Kimera# ping 10.0.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

```
Kimera#telnet 10.0.1.2

Enter password:tekra

Tekra>enable
Enter password:tekra123
Tekra#
```

- ***Desde el router TEKRA:***

```
Tekra# ping 10.0.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

```
Tekra#telnet 10.0.1.1

Enter password:kimera

Kimera>enable
Enter password:kimera123
Kimera#
```

* Tomado de la página: http://es.tldp.org/Manuales-LuCAS/doc-unixsec/unixsec-html/node221.html

CASO DE ESTUDIO 2

CONFIGURACIÓN DEL PROTOCOLO DE ENRUTAMIENTO RIPv2

DESCRIPCIÓN GENERAL Y OBJETIVOS

El objetivo de este caso es realizar la configuración del protocolo de enrutamiento de la red del Caso de Estudio 1. La red debe conectar las dos localidades de la empresa IntelNet para lo cual utilizará RIPv2 como protocolo de enrutamiento. La red a configurarse se muestra en la figura 3.4.

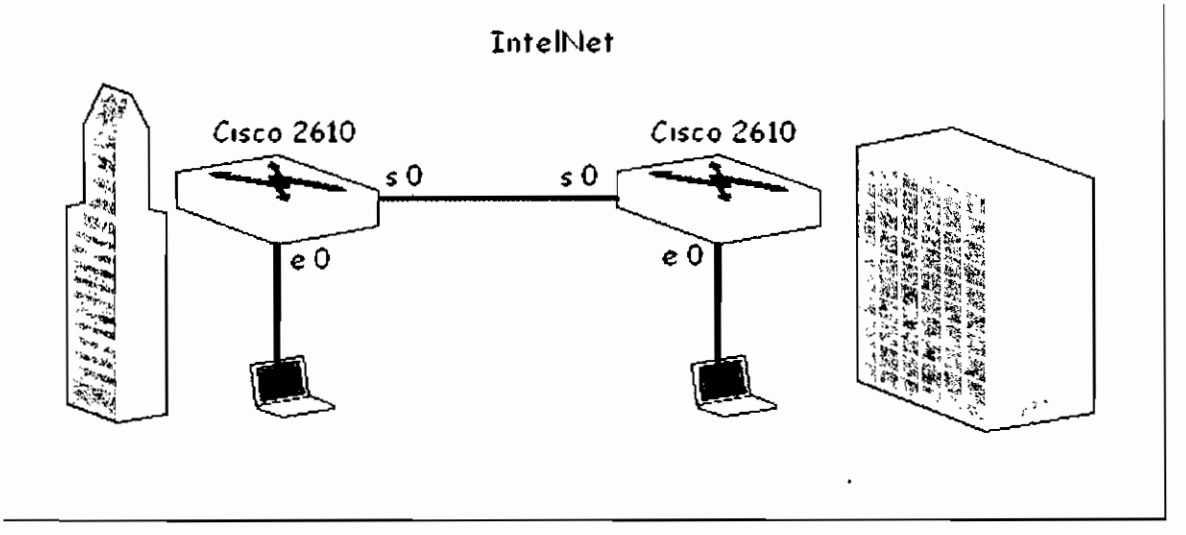


Figura 3.4: Red utilizada en el Caso de Estudio 2

TRABAJO PREPARATORIO

Para desarrollar este Caso de Estudio se deben tener conocimientos básicos sobre el funcionamiento de una red de comunicaciones y el protocolo de enrutamiento RIPv2, también se debe revisar los comandos utilizados para realizar la configuración del protocolo de enrutamiento RIPv2 estos son: *enable, configure terminal, hostname, enable password, line console, line vty, interface, copy, router rip* y *network.*

REQUISITOS

La empresa IntelNet desea comunicar sus dos oficinas que se encuentran en edificios distintos y que pertenecen a redes diferentes, para lo cual dispone de dos routers Cisco 2610 y desea utilizar una conexión serial entre estos, el protocolo de enrutamiento que se utilizará es RIPv2 debido a que es lo que solicita IntelNet. Al final de la configuración las dos redes deberán tener conectividad total y se podrá acceder a los dos routers mediante telnet.

CONFIGURACIÓN

Para lograr una configuración exitosa, se debe planificarla en base a los objetivos y requerimientos antes mencionados, para lo cual se apoya en esquemas de configuración que se utilizan para registrar los datos que se deben configurar.

ROUTER 1 (CISCO 2600)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router 1:

ROUTER 1 (Cisco 2610)			
Nombre	Rocafuerte		
Enable password	rocafuerte123		
Console/VTY password	rocafuerte		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 0	192.168.10.1/24	DCE
	ethernet 0	10.0.0.1/8	LAN
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	RIPv2	192.168.10.0/24	
		10.0.0.0/8	

Archivo de configuración

ROCAFUERTE

```
Rocafuerte#show running-config
Building configuration...

!
Version 12.1
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Rocafuerte
enable password rocafuerte123
!
ip subnet-zero
!
!
!
interface Serial0
 ip address 192.168.10.1 255.255.255.0
 no ip directed-broadcast
 clock rate 56000
 bandwidth 1544
!
interface Serial1
 no ip address
 no ip directed-broadcast
 bandwidth 1544
 shutdown
!
interface Ethernet0
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 bandwidth 10000
!
!
router rip

 version 2
 network 192.168.10.0
 network 10.0.0.0
!
ip classless
no ip http server
!
!
!
line con 0
 login
 transport input none
 password rocafuerte
line aux 0
line vty 0 4
 login
 password rocafuerte
!
no scheduler allocate
end
```

Configuración ROCAFUERTE: Anexo 3.2.a

ROUTER 2 (CISCO 2610)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router 2

ROUTER 2 (Cisco 2610)			
Nombre	Roldos		
Enable password	roldos123		
Console/VTY password	roldos		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 0	192.168.10.2/24	DTE
	ethernet	172.20.0.1/16	LAN
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	RIPv2	192.168.10.0/24	
		172.20.0.0/16	

Archivo de configuración

ROLDOS

```
Roldos#show running-config
Building configuration...

!
Version 12.1
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Roldos
enable password roldos123
!
!
ip subnet-zero
!
!
interface Serial0
 ip address 192.168.10.2 255.255.255.0
 no ip directed-broadcast
 bandwidth 1544
!
interface Serial1
 no ip address
 no ip directed-broadcast
 bandwidth 1544
 shutdown
!
interface Ethernet0
 ip address 172.20.0.1 255.255.0.0
```

```
 no ip directed-broadcast
 bandwidth 10000
!
router rip

 version 2
 network 192.168.10.0
 network 172.20.0.0
!
ip classless
no ip http server
!
line con 0
 login
 transport input none
 password roldos
line aux 0
line vty 0 4
 login
 password roldos
!
no scheduler allocate
end
```

Configuración ROLDOS: Anexo 3.2.b

PRUEBAS

Una vez terminada la configuración se obtendrá la red de la figura 3.5, a la cual se le realizará un proceso de verificación para comprobar que la configuración trabaje correctamente, para esto, se utilizará los comandos *show, ping* y *telnet* del IOS de Cisco de la siguiente manera:

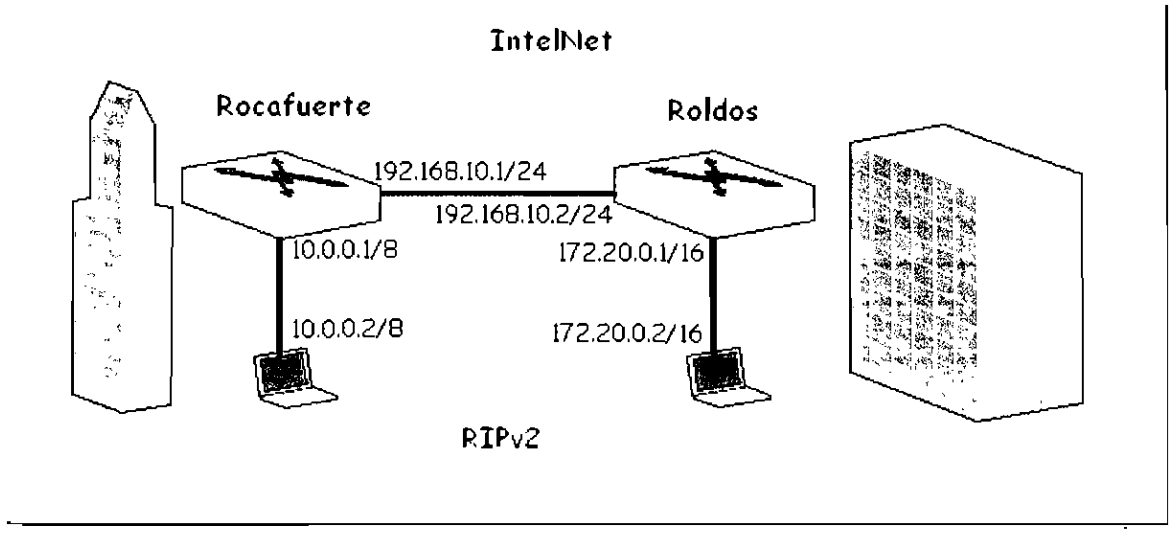


Figura 3.5: Red configurada en el Caso de Estudio 2

Proceso de pruebas:

Los resultados de la verificación se muestran seguidos del comando utilizado.

- ***Desde el router ROCAFUERTE:***

Rocafuerte#show ip route

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default
       U - per-user static route

Gateway of last resort is not set

     192.168.10.0/24 is subnetted, 1 subnets
C       192.168.10.0 is directly connected, Serial0
C    10.0.0.0 is directly connected, Ethernet0
R    172.20.0.0 [120/1] via 192.168.10.2, 00:05:25, Serial0
```

Rocafuerte#ping 192.168.10.2

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Rocafuerte#ping 10.0.0.2

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Rocafuerte#ping 172.20.0.2

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.20.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Rocafuerte#telnet 192.168.10.2

```
Enter password:roldos

Roldos>enable
Enter password:roldos123
Roldos#
```

- ***Desde el router ROLDOS:***

Roldos#show ip route

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default
       U - per-user static route

Gateway of last resort is not set

     192.168.10.0/24 is subnetted, 1 subnets
C       192.168.10.0 is directly connected, Serial0
C    172.20.0.0 is directly connected, Ethernet0
R    10.0.0.0 [120/1] via 192.168.10.1, 00:05:45, Serial0
```

Roldos#ping 192.168.10.1

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Roldos#ping 172.20.0.2

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.20.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Roldos#ping 10.0.0.2

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Roldos#telnet 192.168.10.1

```
Enter password:rocafuerte

Rocafuerte>enable
Enter password:rocafuerte123
Rocafuerte#
```

CASO DE ESTUDIO 3

CONFIGURACIÓN DEL PROTOCOLO DE ENRUTAMIENTO BGP Y OSPF

DESCRIPCIÓN GENERAL Y OBJETIVOS

En este Caso de Estudio se configurará el protocolo OSPF. OSPF es un protocolo de enrutamiento de estado de enlace basado en estándares abiertos. Se describe en diversos estándares de la Fuerza de Tareas de Ingeniería del Internet (IETF). El término "libre" en "OSPF" significa que está abierto al público y no es propiedad de ninguna empresa.

En comparación con RIPv1 y RIPv2, OSPF es el IGP preferido porque es escalable. RIP se limita a 15 saltos, converge lentamente y a veces elige rutas lentas porque pasa por alto ciertos factores críticos como por ejemplo el ancho de banda a la hora de determinar la ruta. OSPF ha superado estas limitaciones y se ha convertido en un protocolo de enrutamiento sólido y escalable adecuado para redes modernas. OSPF se puede usar y configurar en una sola área en las redes pequeñas. También se puede utilizar en redes grandes, las redes grandes OSPF utilizan un diseño jerárquico. Varias áreas se conectan a un área de distribución o a un área 0 que también se denomina backbone.

Debido a la importancia de este protocolo se ha decidido utilizarlo en los Casos de Estudios y elaborar uno que sirva de guía en las configuraciones posteriores. La red que se configurará esta vez es la de la empresa Lutor Corp, la cual se muestra en la figura 3.6

TRABAJO PREPARATORIO

Para desarrollar este Caso de Estudio se deben tener conocimientos básicos sobre el funcionamiento de una red de comunicaciones y los protocolos de enrutamiento OSPF y BGP, también se debe revisar los comandos utilizados

para realizar la configuración de dichos protocolos, estos son: *enable, configure terminal, hostname, enable password, line console, line vty, interface, copy, router ospf, router bgp, network y neighbor.*

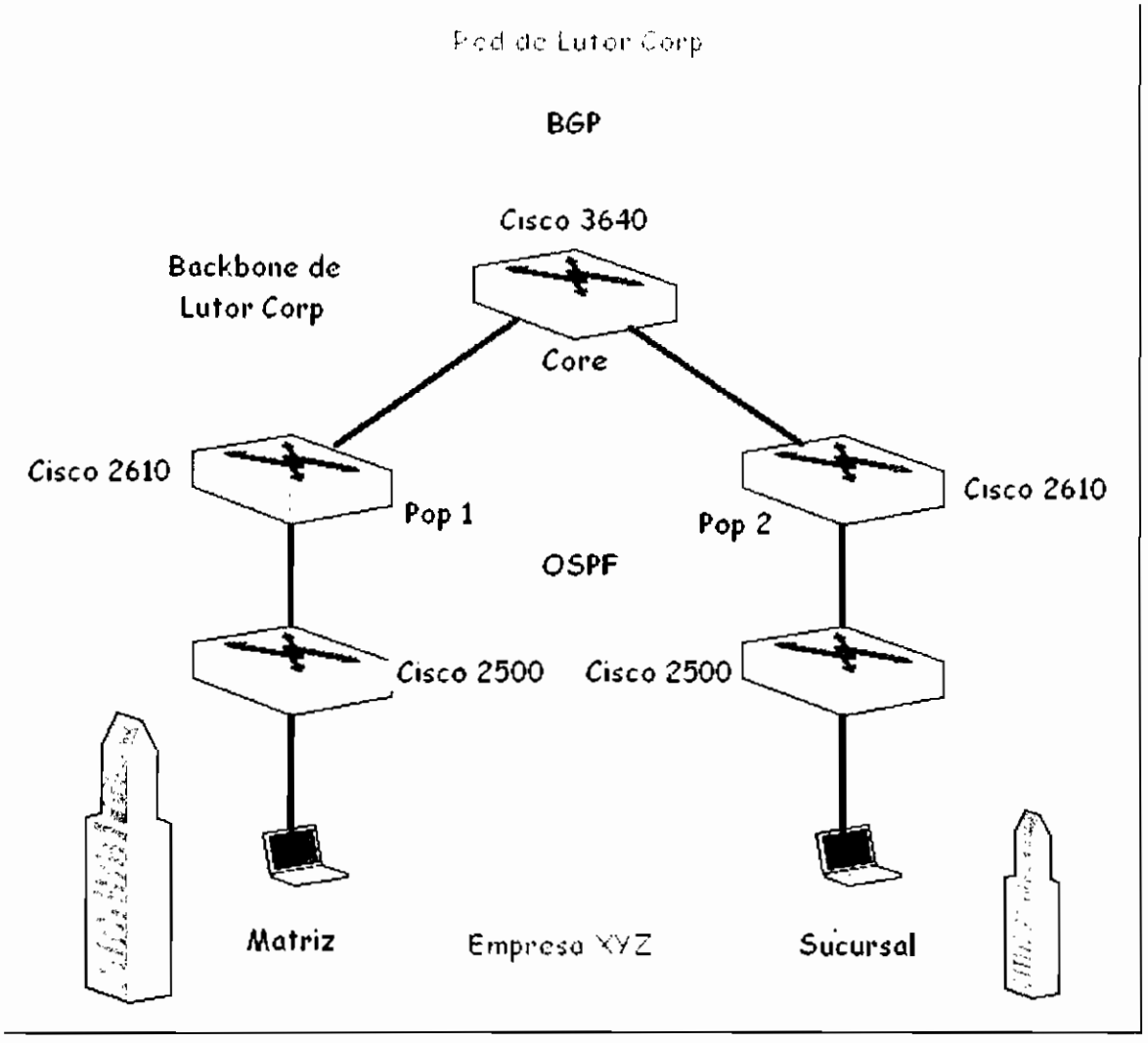


Figura 3.6: Red utilizada en el Caso de Estudio 3

REQUISITOS

La matriz de la empresa XYZ debe conectarse con la Sucursal mediante el backbone de Lutor Corp el mismo que tiene configurado BGP como protocolo de enrutamiento. Para lo cual XYZ utilizará un ruteador en cada sitio, los mismos que estarán corriendo OSPF como protocolo de enrutamiento.

CONFIGURACIÓN

ROUTER MATRIZ (CISCO 2500)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router MATRIZ:

ROUTER Matriz (Cisco 2500)			
Nombre	Matriz		
Enable password	matriz123		
Console/VTY password	matriz		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 0	192.168.10.1/30	DTE
	ethernet 0	10.0.0.1/8	LAN
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF	192.168.10.0/24 10.0.0.0/8	

Archivo de configuración

MATRIZ

```
Matriz#show running
Building configuration...

Current configuration : 957 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Matriz
!
boot-start-marker
boot-end-marker
!
enable password matriz123
!
no aaa new-model
ip subnet-zero
!
interface Ethernet0/0
 ip address 10.0.0.1 255.0.0.0
 half-duplex
!
interface Serial0/0
 ip address 192.168.10.1 255.255.255.252
```

```
!
interface Ethernet0/1
 no ip address
 shutdown
 half-duplex
!
interface Serial0/1
 no ip address
 shutdown
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 0
 network 192.168.10.0 0.0.0.255 area 0
!
no ip http server
ip classless
!
!
gatekeeper
 shutdown
!
line con 0
 password matriz
 login
line aux 0
line vty 0 4
 password matriz
 login
!
end
```

Configuración MATRIZ: Anexo 3.3.a

ROUTER POP 1 (CISCO 2610)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router POP 1:

ROUTER Pop 1 (Cisco 2610)			
Nombre	Pop_1		
Enable password	pop1123		
Console/VTY password	pop1		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 0	192.168.10.2/30	DCE
	Serial 1	192.168.10.5/30	DCE
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF	192.168.10.0/24	
	BGP	192.168.10.0	
		192.168.10.4	

Archivo de configuración

POP 1

```
Pop_1#show running
Building configuration...

Current configuration : 957 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Pop_1
!
boot-start-marker
boot-end-marker
!
enable password pop1123
!
memory-size iomem 10
no aaa new-model
ip subnet-zero
!
!
ip cef
no ftp-server write-enable
!
!
!
!
interface Ethernet0/0
 no ip address
 shutdown
 half-duplex
!
interface Serial0/0
 ip address 192.168.10.2 255.255.255.252
 clockrate 56000
!
interface Serial0/1
 ip address 192.168.10.5 255.255.255.252
 clockrate 56000
!
router ospf 1
 log-adjacency-changes
 network 192.168.10.0 0.0.0.255 area 0
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 network 192.168.10.0
 network 192.168.10.4
 redistribute ospf 1
 neighbor 192.168.10.10 remote-as 1
 no auto-summary
!
ip http server
ip classless
!
!
!
!
line con 0
```

```
 password pop1
 login
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 5
 password pop1
 login
 transport preferred all
 transport input all
 transport output all
!
end
```

Configuración POP 1: Anexo 3.3.b

ROUTER CORE (CISCO 3640)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router CORE:

ROUTER Core (Cisco 3640)			
Nombre	Core		
Enable password	core123		
Console/VTY password	core		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 0	192.168.10.6/30	DTE
	Serial 1	192.168.10.9/30	DCE
Protocolo de enrutamiento	*Protocolo*	*Redes*	
		192.168.10.0/24	

Archivo de configuración

CORE

```
Core#show running
Building configuration...
Current configuration : 957 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Core
```

```
!
boot-start-marker
boot-end-marker
!
enable password core123
!
no aaa new-model
ip subnet-zero
!
ip cef
ip audit po max-events 100
no ftp-server write-enable
!
interface Ethernet0/0
 no ip address
 shutdown
 half-duplex
 no clns route-cache
!
interface TokenRing0/0
 no ip address
 shutdown
 ring-speed 16
 no clns route-cache
!
interface Serial1/0
 ip address 192.168.10.6 255.255.255.252
 no fair-queue
 no clns route-cache
!
interface Serial1/1
 ip address 192.168.10.9 255.255.255.252
 clockrate 56000
 no clns route-cache
!
interface Serial1/2
 no ip address
 shutdown
 no clns route-cache
!
interface Serial1/3
 no ip address
 shutdown
 no clns route-cache
!
router ospf 1
 log-adjacency-changes
 network 192.168.10.0 0.0.0.255 area 0
!
ip http server
no ip http secure-server
ip classless
!
line con 0
 password core
 login
line aux 0
line vty 0 4
 password core
 login
!
end
```

Configuración CORE: Anexo 3.3.c

ROUTER POP 2 (CISCO 2610)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router POP 2:

ROUTER Pop 2 (Cisco 2610)			
Nombre	Pop_2		
Enable password	pop2123		
Console/VTY password	pop2		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 0	192.168.10.13/30	DTE
	Serial 1	192.168.10.10/30	DTE
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF	192.168.10.0/24	
	BGP	192.168.10.8 192.168.10.12	

Archivo de configuración

POP 2

```
Pop_2#show running
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Pop_2
!
boot-start-marker
boot-end-marker
!
enable password pop2123
!
no aaa new-model
ip subnet-zero
!
ip cef
no ftp-server write-enable
!
interface Ethernet0/0
 no ip address
 shutdown
 half-duplex
!
interface Serial0/0
 ip address 192.168.10.13 255.255.255.252
 no fair-queue
!
interface Serial0/1
```

```
 ip address 192.168.10.10 255.255.255.252
!
router ospf 1
 log-adjacency-changes
 network 192.168.10.0 0.0.0.255 area 0
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 network 192.168.10.8
 network 192.168.10.12
 neighbor 192.169.10.5 remote-as 1
 no auto-summary
!
ip http server
!
line con 0
 password pop2
 login
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
 password pop2
 login
 transport preferred all
 transport input all
 transport output all
!
end
```

Configuración POP 2: Anexo 3.3.d

ROUTER SUCURSAL (CISCO 2500)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router SUCURSAL:

ROUTER Sucursal (Cisco 2500)			
Nombre	Sucursal		
Enable password	sucursal123		
Console/VTY password	sucursal		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 0	192.168.10.14/30	DCE
	ethernet 0	172.20.0.1/16	LAN
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF	192.168.10.0/24 172.20.0.0/16	

Archivo de configuración

SUCURSAL

```
Sucursal#show running
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Sucursal
!
boot-start-marker
boot-end-marker
!
enable password sucursal123
!
no aaa new-model
ip subnet-zero
!
ip cef
ip audit po max-events 100
no ftp-server write-enable
!
interface Ethernet0/0
 ip address 172.20.0.1 255.255.0.0
 half-duplex
 no clns route-cache
!
interface Serial0/0
 ip address 192.168.10.14 255.255.255.252
 clockrate 56000
 no fair-queue
 no clns route-cache
!
interface Serial0/1
 no ip address
 shutdown
 no clns route-cache
!
router ospf 1
 log-adjacency-changes
 network 172.20.0.0 0.0.255.255 area 0
 network 192.168.10.0 0.0.0.255 area 0
!
ip http server
no ip http secure-server
ip classless
!
line con 0
 password sucursal
 login
line aux 0
line vty 0 4
 password sucursal
 login
!
end
```

Configuración SUCURSAL: Anexo 3.3.e

PRUEBAS

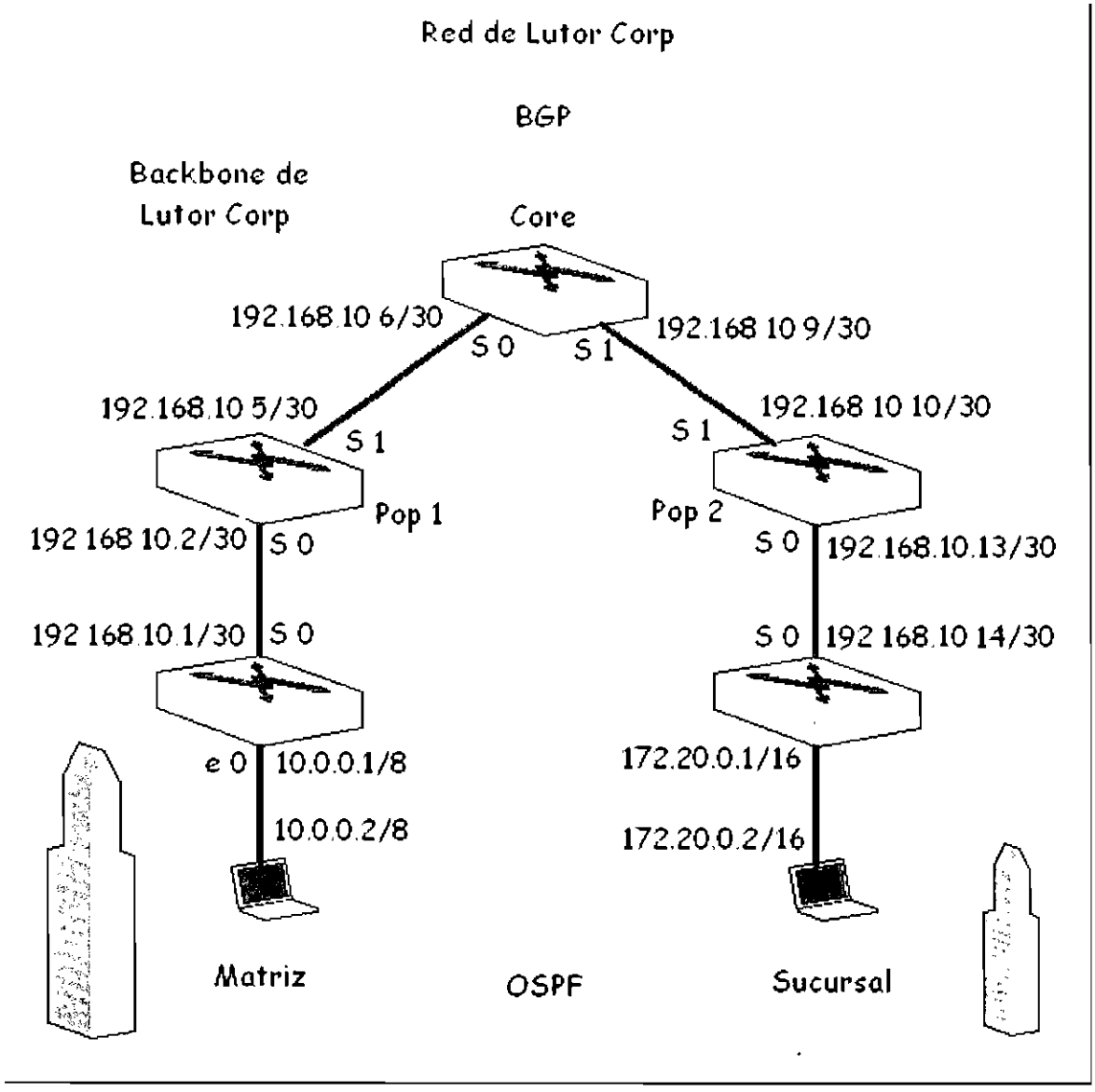


Figura 3.7: Red configurada en el Caso de Estudio 3

Una vez terminada la configuración se obtendrá la red de la figura 3.7, a la cual se le realizará un proceso de verificación para comprobar que la configuración trabaje correctamente, para esto, se utilizará los comandos *show, ping* y *tracert* y otros comandos del IOS de Cisco y del MS-DOS de la siguiente manera:

Proceso de pruebas:

Los resultados de la verificación se muestran seguidos del comando utilizado. Se debe verificar que las rutas mostradas en las tablas de enrutamiento

de los diferentes routers estén correctamente ingresadas y aprendidas y que el proceso OSPF hay convergido en su totalidad, esto se puede verificar constatando que en cada uno de los routers se encuentre una ruta hacia cada destino

Matriz#sh 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

     192.168.10.0/30 is subnetted, 4 subnets
C       192.168.10.0 is directly connected, Serial0/0
O       192.168.10.4 [110/845] via 192.168.10.2, 00:11:33, Serial0/0
O       192.168.10.8 [110/1626] via 192.168.10.2, 00:11:33, Serial0/0
O       192.168.10.12 [110/2407] via 192.168.10.2, 00:11:33, Serial0/0
     172.20.0.0/32 is subnetted, 1 subnets
O       172.20.0.1 [110/2408] via 192.168.10.2, 00:11:33, Serial0/0
C    10.0.0.0/8 is directly connected, Ethernet0/0
```

Pop_1#sh 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

     192.168.10.0/30 is subnetted, 4 subnets
C       192.168.10.0 is directly connected, Serial0/0
C       192.168.10.4 is directly connected, Serial0/1
O       192.168.10.8 [110/1562] via 192.168.10.6, 00:12:42, Serial0/1
O       192.168.10.12 [110/2343] via 192.168.10.6, 00:12:42, Serial0/1
     172.20.0.0/32 is subnetted, 1 subnets
O       172.20.0.1 [110/2344] via 192.168.10.6, 00:12:42, Serial0/1
O    10.0.0.0/8 [110/791] via 192.168.10.1, 00:12:42, Serial0/0
```

Core#sh 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

     192.168.10.0/30 is subnetted, 4 subnets
O       192.168.10.0 [110/1562] via 192.168.10.5, 00:13:11, Serial1/0
C       192.168.10.4 is directly connected, Serial1/0
C       192.168.10.8 is directly connected, Serial1/1
O       192.168.10.12 [110/1562] via 192.168.10.10, 00:13:11, Serial1/1
     172.20.0.0/32 is subnetted, 1 subnets
O       172.20.0.1 [110/1563] via 192.168.10.10, 00:13:11, Serial1/1
O    10.0.0.0/8 [110/1572] via 192.168.10.5, 00:13:11, Serial1/0
```

Pop_2#sh 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

     192.168.10.0/30 is subnetted, 4 subnets
O       192.168.10.0 [110/2343] via 192.168.10.9, 00:16:24, Serial0/1
O       192.168.10.4 [110/1562] via 192.168.10.9, 00:16:24, Serial0/1
C       192.168.10.8 is directly connected, Serial0/1
C       192.168.10.12 is directly connected, Serial0/0
     172.20.0.0/32 is subnetted, 1 subnets
O       172.20.0.1 [110/782] via 192.168.10.14, 00:16:24, Serial0/0
O    10.0.0.0/8 [110/2353] via 192.168.10.9, 00:16:24, Serial0/1
```

Sucursal#sh 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

     192.168.10.0/30 is subnetted, 4 subnets
```

```
O       192.168.10.0 [110/2407] via 192.168.10.13, 00:17:11, Serial0/0
O       192.168.10.4 [110/1626] via 192.168.10.13, 00:17:11, Serial0/0
O       192.168.10.8 [110/845] via 192.168.10.13, 00:17:11, Serial0/0
C       192.168.10.12 is directly connected, Serial0/0
C    172.20.0.0/16 is directly connected, Loopback0
O    10.0.0.0/8 [110/2417] via 192.168.10.13, 00:17:11, Serial0/0
```

Se puede revisar la tabla de rutas BGP ingresando el comando ***sh ip bgp*** en los routers en los cuales esté configurado el protocolo BGP de la siguiente manera.

Pop_1#sh ip bgp

```
BGP table version is 7, local router ID is 192.168.10.5
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.0.0.0         192.168.10.1           791         32768 ?
*> 172.20.0.1/32    192.168.10.6          2344         32768 ?
*> 192.168.10.0/30  0.0.0.0                  0         32768 ?
*> 192.168.10.4/30  0.0.0.0                  0         32768 ?
*> 192.168.10.8/30  192.168.10.6          1562         32768 ?
*> 192.168.10.12/30 192.168.10.6          2343         32768 ?
```

Pop_2#sh ip bgp

```
BGP table version is 7, local router ID is 192.168.10.13
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.0.0.0         192.168.10.9          2353         32768 ?
*> 172.20.0.1/32    192.168.10.14          782         32768 ?
*> 192.168.10.0/30  192.168.10.9          2343         32768 ?
*> 192.168.10.4/30  192.168.10.9          1562         32768 ?
*> 192.168.10.8/30  0.0.0.0                  0         32768 ?
*> 192.168.10.12/30 0.0.0.0                  0         32768 ?
```

Finalmente para verificar la convergencia de la red se debe enviar un ping de extremo a extreme de la red y también comprobar la ruta que está tomando un paquete para viajar de un extreme a otro, para esto se utiliza los comandos ping y tracert desde el MS-DOS de la máquina conectada al router Matriz de la siguiente forma:

C:\WINDOWS\system32>ipconfig

```
Configuración IP de Windows

Adaptador Ethernet Conexiones de red inalámbricas          :

        Estado de los medios. . . .: medios desconectados

Adaptador Ethernet Conexión de área local          :

        Sufijo de conexión específica DNS :
        Dirección IP. . . . . . . . . . . : 10.0.0.2
        Máscara de subred . . . . . . . . : 255.0.0.0
        Puerta de enlace predeterminada   : 10.0.0.1
```

C:\WINDOWS\system32>ping 172.20.0.1

```
Haciendo ping a 172.20.0.1 con 32 bytes de datos:

Respuesta desde 172.20.0.1: bytes=32 tiempo=82ms TTL=251
Respuesta desde 172.20.0.1: bytes=32 tiempo=82ms TTL=251
Respuesta desde 172.20.0.1: bytes=32 tiempo=82ms TTL=251
Respuesta desde 172.20.0.1: bytes=32 tiempo=82ms TTL=251

Estadísticas de ping para 172.20.0.1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 82ms, Máximo = 82ms, Media = 82ms
```

C:\WINDOWS\system32>tracert 172.20.0.1

```
Traza a 172.20.0.1 sobre caminos de 30 saltos como máximo.

  1     1 ms    <1 ms     1 ms  10.0.0.1
  2    25 ms    25 ms    25 ms  192.168.10.2
  3    49 ms    49 ms    49 ms  192.168.10.6
  4    74 ms    74 ms    74 ms  192.168.10.10
  5   131 ms   118 ms   118 ms  172.20.0.1

Traza completa.
```

CASO DE ESTUDIO 4

CONCEPTOS BÁSICOS DE MPLS

DESCRIPCIÓN GENERAL Y OBJETIVOS

El objetivo principal de este Caso de Estudio es familiarizar al usuario con la terminología y con los tipos de nodos que comprenden una red MPLS. Está basada en conceptos y en esencia es un resumen de la parte teórica de la arquitectura pero también incluye ciertos tips que hay que tomar en cuenta al momento de realizar la configuración o migración a la Arquitectura MPLS.

DESARROLLO

El desarrollo de este Caso se lo llevará a cabo utilizando la modalidad Pregunta/Respuesta en el cual se realizan preguntas cuyo objetivo es hacer sobresalir los aspectos más importantes en este caso de la arquitectura MPLS.

¿Qué es la arquitectura MPLS?

Es una tecnología emergente destinada a encauzar muchos de los actuales retos que plantea el envío de paquetes en las redes modernas y que utiliza la conmutación de etiquetas para enrutar los paquetes por la red.

¿Qué es la conmutación de etiquetas MPLS?

Una independiente y única "etiqueta" es agregada a cada paquete de datos y ésta es utilizada para enrutar y conmutar el paquete de datos a través de la red. La etiqueta es simple (es una versión corta de la cabecera de un paquete de información) lo que favorece a la optimización de los equipos de red en lo que al procesamiento de etiquetas y envío de tráfico se refiere

¿Qué es la etiqueta MPLS?

Es una versión corta de la cabecera de un paquete de información y está conformada por 32 bits los mismos que se detallan en la figura 3.8.

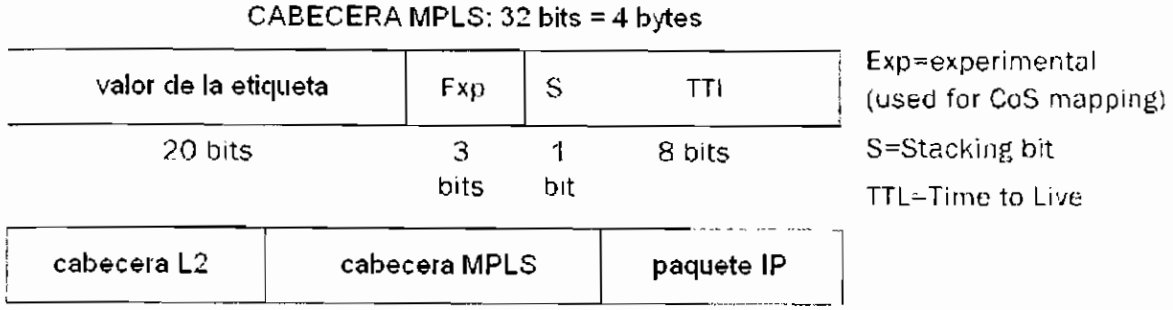


Figura 3.8: Formato de la cabecera MPLS en un paquete MPLS

¿Por qué se dice que la arquitectura MPLS es una arquitectura de Capa 2½?

MPLS combina lo mejor del enrutamiento de Capa 3 y la conmutación de Capa 2, de hecho, por esto es llamado protocolo de "Capa 2½". Mientras los ruteadores requieren procesamiento a nivel de Capa red para determinar a donde enviar el tráfico, los conmutadores solo envían datos al próximo salto, entonces son intrínsicamente más simples, rápidos y menos costosos

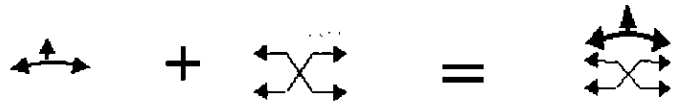
¿Porqué al utilizar MPLS se reduce el retardo de a red?

El envío tradicional basado en software es demasiado lento para manejar las grandes cantidades de tráfico de Internet o de las redes interconectadas. El tiempo de la búsqueda en las tablas de enrutamiento es demasiado elevado aún teniendo en cuenta métodos de aceleración de la búsqueda. Esto se traduce en pérdida de paquetes y conexiones, y por tanto en un bajo rendimiento.

La conmutación de etiquetas es más rápida. El motivo es simple: el valor de la etiqueta que se pone en la cabecera de los paquetes es el que se usa para acceder a la tabla de enrutamiento del enrutador (es el índice para acceder a la tabla). Esto requiere un solo acceso a la tabla (cómo tenemos el índice, accedemos directamente). Por tanto, el tiempo para transmitir un paquete es menor que con el enrutamiento tradicional IP. Se reduce el retardo y el tiempo de respuesta.

¿Cómo se separan las funciones de control y de envío?

MPLS se ha creado para combinar las ventajas del enrutamiento y el envío sin conexión de Capa 3 con la conmutación de Capa 2, que si emplea conexiones. La arquitectura se divide en el componente de envió (plano de datos) y el componente de control (plano de control)



Por la conmutación de etiquetas, se tiene equipos con características de precio/rendimiento de un conmutador, pero con funcionalidades de un ruteador

¿Qué tipo de nodos conforman la red MPLS?

Router de conmutación por etiquetas (LSR).- Es un router de alta velocidad en el corazón de la red MPLS, el cual debe soportar los protocolos de enrutamiento IP y participa en el establecimiento de los LSP (Label Switched Paths) utilizando el protocolo de señalización de etiquetas adecuado.

LSR de contorno.- Es un router que básicamente realiza dos funciones: imposición de etiquetas y la determinación de etiquetas en el contorno de la red.

LSR ATM.- Es un switch ATM que actúa como LSR, realiza el enrutamiento IP y la asignación de etiquetas en el plano de control y envía paquetes utilizando mecanismos de conmutación por celdas ATM adicionales en el plano de datos.

¿Cómo opera una red MPLS básica?

La operación de una red MPLS es la siguiente:

1. Las tablas de enrutamiento de los diferentes LSRs son computadas usando un Protocolo de Gateway Interior (IGP). Se puede usar un protocolo de estado de enlace como RIPv2, OSPF o IS-IS.
2. Un protocolo de distribución de etiquetas (LDP) anuncia los enlaces entre rutas y etiquetas, estas correspondencias son chequeadas en la tabla de enrutamiento. Si la ruta aprendida por medio de LDP

corresponde a la ruta aprendida por IGP, se crea una entrada en la LFIB del LSR.

Los LSRs utilizan el siguiente mecanismo de envío:

1. Un vez que el LSR de contorno recibe un paquete no etiquetado, se revisa la tabla CEF y se coloca una etiqueta al paquete si es necesario. Este LSR es llamado LSR de ingreso.
2. Al arribo de un paquete etiquetado por una interfaz de un LSR de core, la LFIB provee la interfaz de salida y la nueva etiqueta que será asociada al paquete saliente.
3. El router anterior al último LSR (penúltimo salto) realiza la acción pop al paquete y lo transmite sin etiqueta. El último salto es llamado LSR de salida.

¿Qué es recomendable realizar en un router antes de empezarlo a configurar?

Se debe borrar el archivo de configuración para evitar realizar configuraciones superpuestas ya que esto podría causar algún tipo de problema al momento de la implementación.

¿Por qué la versión del IOS utilizada es importante?

La versión del IOS debe satisfacer las necesidades de configuración, es decir, soportar los comandos que se necesitan ejecutar para llevar a cabo la configuración requerida. Las versiones utilizadas en este proyecto y que se requiere para ejecutar comandos MPLS son: c3640-jk9o3s-mz.123-10, c2600-telco-mz.12.3-12. Cabe señalar que no son las únicas pero si las mínimas necesarias.

¿Cuáles son los pasos a seguir en caso de que la red implementada no funcione correctamente?

1. Revisión de las conexiones físicas:
   1.1. Revisar que los cables estén correctamente conectados y sujetados.

1.2. Revisar que los cables DTE y DCE estén conectados de acuerdo a la configuración.

1.3. Descartar cable en mal estado.

2. Descartar problemas de hardware en el router:

2.1. En caso de encontrar problemas de hardware cambiar la parte afectada.

3. Revisión de la configuración:

3.1. Revisar que las ip configuradas en las interfaces correspondan a las planificadas con el esquema de configuración.

3.2. Revisar que las características de las interfaces correspondan a las planificadas.

3.3. Comprobar que la versión del IOS corresponda a las recomendadas en este proyecto.

4. Si los problemas persisten:

4.1. Reiniciar el router siguiendo el procedimiento de aislamiento del problema, es decir, localizar el router problema y reiniciar ese router.

4.2. Si no se soluciona, cargar el archivo de configuración y en caso de no disponer de un respaldo reconfigurar el router desde el inicio.

4.3. Si no se soluciona, cargar nuevamente el IOS del router

RESULTADOS

Al completar este Caso de Estudio el usuario tendrá claros los conceptos básicos de la arquitectura MPLS y comprenderá el funcionamiento de una red que utilice conmutación de etiquetas como la que se configurará en los Casos de Estudio posteriores.

Es importante comprender en su totalidad los conceptos arriba señalados ya que son la base fundamental de una correcta configuración como se señala en los requisitos de los casos de estudio.

CASO DE ESTUDIO 5

CONFIGURACIÓN BÁSICA MPLS DE UN LSR DE CONTORNO

DESCRIPCIÓN GENERAL Y OBJETIVOS

El siguiente caso de estudio se utilizará para ilustrar el proceso de configuración de un LSR de contorno en una red básica MPLS. Se presenta un escenario en el cual la empresa LojaNet, ubicada al norte de la ciudad de Loja, desea comunicarse a través de un backbone MPLS con la sucursal ubicada en el sur de la ciudad

Para este caso de estudio utilizaremos la red que se muestra en la figura 3.9, donde la empresa LojaNet será representada con el router Cisco 2500 (CE) y la sucursal con la interfaz *loopback* configurada en el router Cisco 3600(P).

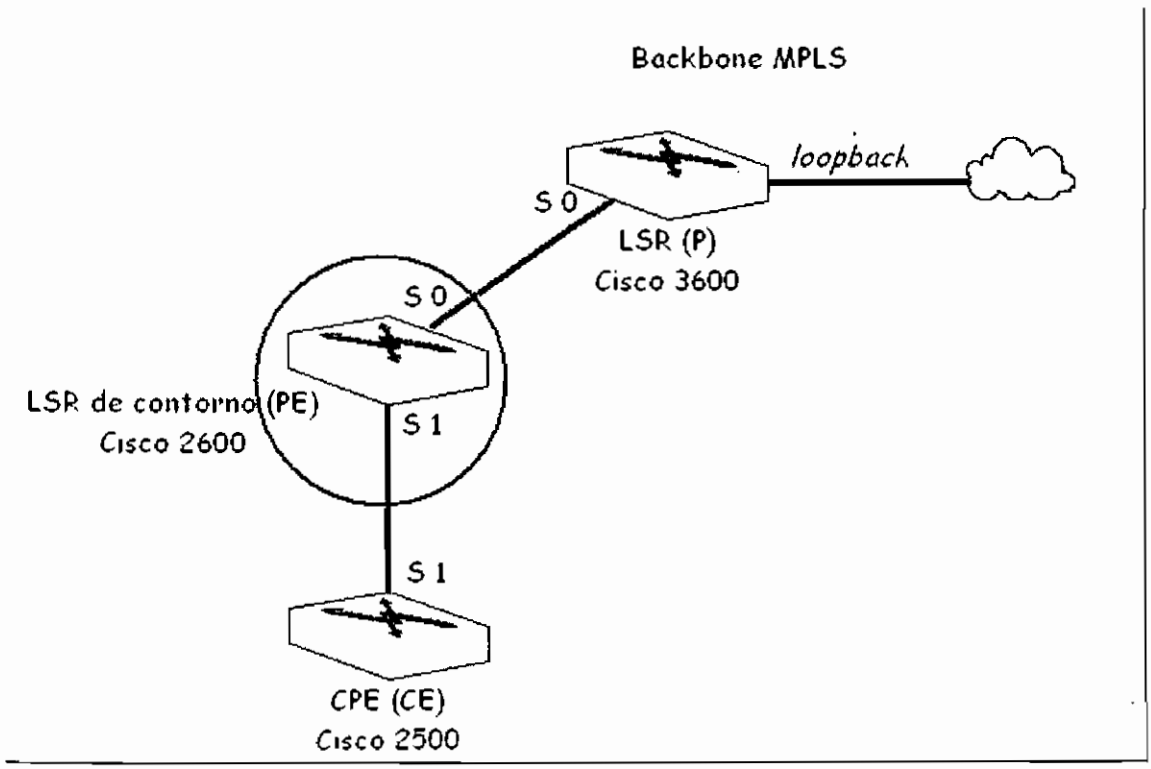


Figura 3.9: Red utilizada en el Caso de Estudio 5

TRABAJO PREPARATORIO

Para desarrollar este Caso de Estudio se deben haber desarrollado los casos de estudio del 1 al 4, se debe tener conocimientos básicos sobre el funcionamiento de una red de comunicaciones y los protocolos OSPF y MPLS, también se debe revisar los comandos utilizados para realizar la configuración de dichos protocolos, estos son: *enable, configure terminal, hostname, enable password, line console, line vty, interface, copy, router ospf, router bgp, network, ip cef y mpls ip.* Para poder realizar las pruebas de funcionamiento, el router P (Cisco 3640) y el router CE (Cisco 2500) estarán pre-configurados ya que la misma no es un objetivo de este caso de estudio*.

Al final de la configuración se realizarán las pruebas de funcionamiento, las cuales comprenden: pruebas de conectividad, verificar que los procesos de enrutamiento se lleven a cabo correctamente y finalmente comprobar que los paquetes sean etiquetados dentro de la nube MPLS.

REQUISITOS

La empresa LojaNet ha informado que su red utilizará OSPF como protocolo de enrutamiento al igual que en su sucursal y que el router que utilizará como CE es un Cisco 2500 el cual, entre otras, tiene una interfaz serial que será destinado para conectarse al LSR de contorno (PE).

El protocolo de enrutamiento que se utilizará en el backbone MPLS será OSPF. La sucursal de la empresa y el resto de la nube MPLS será simulada con un interfaz de loopback configurado en el router P (Cisco 3640). La configuración completa la veremos en el Caso de Estudio número 7. La empresa requiere comunicación total entre todos sus equipos y no alerta de alguna restricción para alguno de ellos.

* La configuración del router de *Core* utilizada en este caso se la realizará en el Caso de Estudio de estudio 6 y la del router CE se la realizó en los casos de estudio 1, 2 y 3.

CONFIGURACIÓN

Para lograr una configuración exitosa, se debe planificarla en base a los objetivos y requerimientos antes mencionados, para lo cual se apoya en esquemas de configuración que se utilizan para registrar los datos que se deben configurar.

ROUTER CE (CISCO 2500)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router CE:

ROUTER CE (Cisco 2500)			
Nombre	LojaNet		
Enable password	loja123		
Console/VTY password	loja		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 1	192.168.21.1/30	DCE
	Ethernet 0	192.168.20.1/24	LAN
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF	192.168.0.0/16	

ROUTER PE (CISCO 2610)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router PE:

ROUTER PE (Cisco 2610)			
Nombre	Pop_de_LojaNet		
Enable password	pop123		
Console/VTY password	poploja		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 0	192.168.22.1/30	DCE
	Serial 1	192.168.21.2/30	DTE
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF	192.168.0.0/16	

Archivo de configuración:*

Pop_de_LojaNet

```
Pop_de_LojaNet#show running-config
Building configuration...

Current configuration : 917 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Pop_de_LojaNet
!
boot-start-marker
boot-end-marker
!
enable password pop123
!
memory-size iomem 10
no aaa new-model
ip subnet-zero
!
!
ip cef
no ftp-server write-enable
!
!
interface Ethernet0/0
 no ip address
 shutdown
 half-duplex
!
interface Serial0/0
 ip address 192.168.22.1 255.255.255.252
 tag-switching ip
 clockrate 56000
 no fair-queue
!
interface Serial0/1
 ip address 192.168.21.2 255.255.255.252
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
ip http server
ip classless
!
!
line con 0
 password poploja
 login
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
```

* Si tiene alguna duda sobre algún comando introducido, puede consultarlo en el Caso de estudio 1, 2, 3 o en el Capítulo II

```
 password poploja
 login
 transport preferred all
 transport input all
 transport output all
!
end

Pop_de_LojaNet#
```

Configuración: Pop_de_LojaNet (Ver anexo 3.5.a)

ROUTER P (CISCO 3640)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router P:

ROUTER P (Cisco 3640)			
Nombre	Core		
Enable password	core123		
Console/VTY password	core		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 0	192.168.22.2/30	DTE
	Loopback 0	192.168.23.1/32	Simula nube MPLS.
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF	192.168.0.0/16	

PRUEBAS

Una vez terminada la configuración se obtiene la red de la figura 3.10, a la cual se realizará un proceso de verificación para comprobar que la configuración trabaje correctamente, para esto, utilizaremos los comandos *show* y *ping* del IOS de Cisco de la siguiente manera:

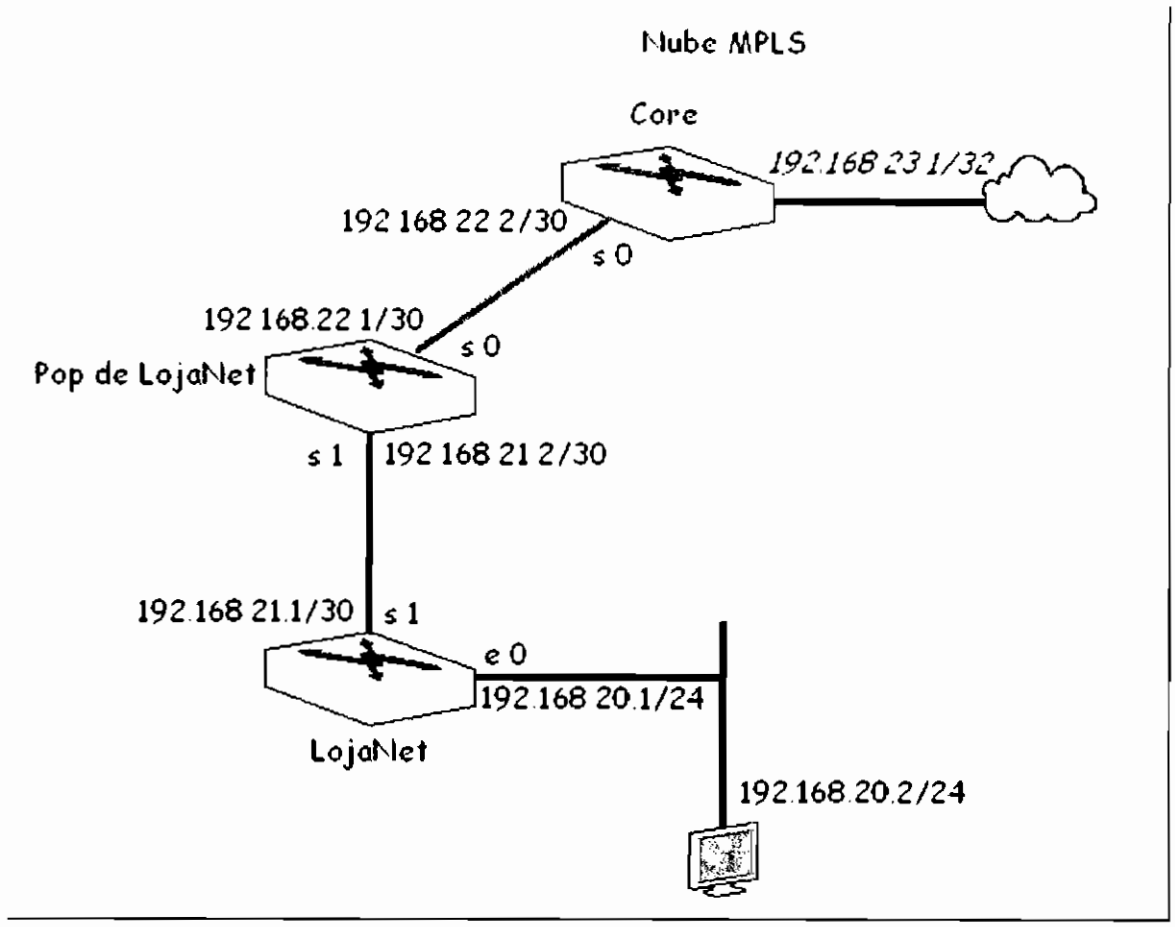


Figura 3.10: Red configurada en el Caso de Estudio 5

Proceso de pruebas:

Primero se comprobará que el protocolo de enrutamiento esté trabajando correctamente para lo cual se utiliza el siguiente procedimiento (los resultados de la verificación se muestran seguidos del comando utilizado):

Comprobación de las rutas que se encuentran configuradas y aprendidas mediante el proceso OSPF en el router Pop_de_LojaNet:

- ***Desde el router Pop_de_LojaNet:***

```
Pop_de_LojaNet#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
rou
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     192.168.21.0/30 is subnetted, 1 subnets
C       192.168.21.0 is directly connected, Serial0/1
O    192.168.20.0/24 [110/791] via 192.168.21.1, 00:25:17, Serial0/1
     192.168.23.0/32 is subnetted, 1 subnets
O       192.168.23.1 [110/782] via 192.168.22.2, 00:25:17, Serial0/0
     192.168.22.0/30 is subnetted, 1 subnets
C       192.168.22.0 is directly connected, Serial0/0
```

En la tabla de enrutamiento se nota las redes 192.168.20.0/24 y 192.168.23.1 que fueron aprendidas por el proceso OSPF.

Pop_de_LojaNet#ping 192.168.20.2

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.20.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/44/60 ms
```

Pop_de_LojaNet#ping 192.168.23.1

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/36 ms
```

Los pings exitosos enviados desde el router PE hacia la máquina conectada a la LAN de LojaNet y a la interfaz de loopback que simula la nube MPLS indican que el enrutamiento se está llevando a cabo satisfactoriamente.

Verificación de la configuración MPLS en el router Pop_de_LojaNet (se debe tomar en cuenta que los resultados se ven afectados al utilizar una interfaz de loopback, ya que no se puede configurar el protocolo MPLS en este tipo de interfaces. En el Caso de Estudio 7 se realizará la configuración completa, lo que nos permitirá apreciar mejor las pruebas realizadas):

- ***Desde el router Pop_de_LojaNet:***

Pop_de_LojaNet#show tag forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	Untagged	192.168.20.0/24	2312	Se0/1	point2point
17	Pop tag	192.168.23.1/32	0	Se0/0	point2point

Este comando muestra la LFIB. En esta tabla se puede verificar la etiqueta asignada al paquete cuyo destino se especifica en su cabecera o prefijo IP.

Pop_de_LojaNet#sh tag forwarding-table tags 16 detail

```
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
16     Untagged    192.168.20.0/24  13704      Se0/1       point2point
        MAC/Encaps=0/0, MRU=1504, Tag Stack{}
        No output feature configured
    Per-packet load-sharing
```

Pop_de_LojaNet#sh tag forwarding-table tags 17 detail

```
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
17     Pop tag     192.168.23.1/32  0          Se0/0       point2point
        MAC/Encaps=4/4, MRU=1504, Tag Stack{}
        0F008847
        No output feature configured
    Per-packet load-sharing
```

Al introducir este comando se muestra la LFIB con más detalle de la etiqueta o etiquetas que se especifiquen.

Pop_de_LojaNet#sh mpls interfaces

```
Interface              IP            Tunnel   Operational
Serial0/0              Yes (tdp)     No       Yes
```

Muestra las interfaces en las que se encuentre activado el protocolo MPLS.

Pop_de_LojaNet#sh mpls ldp neighbor

```
Peer TDP Ident: 192.168.23.1:0; Local TDP Ident 192.168.22.1:0
        TCP connection: 192.168.23.1.11004 - 192.168.22.1.711
        State: Oper; PIEs sent/rcvd: 20/20; Downstream
        Up time: 00:14:16
        TDP discovery sources:
          Serial0/0, Src IP addr: 192.168.22.2
        Addresses bound to peer TDP Ident:
          192.168.22.2    192.168.23.1
```

Ingrese este comando para mostrar los vecinos MPLS activos.

Pop_de_LojaNet#sh mpls ldp bindings

```
  tib entry: 192.168.20.0/24, rev 4
        local binding:  tag: 16
        remote binding: tsr: 192.168.23.1:0, tag: 16
  tib entry: 192.168.21.0/30, rev 2
        local binding:  tag: imp-null
```

```
        remote binding: tsr: 192.168.23.1:0, tag: 17
  tib entry: 192.168.22.0/30, rev 6
        local binding:  tag: imp-null
        remote binding: tsr: 192.168.23.1:0, tag: imp-null
  tib entry: 192.168.23.1/32, rev 8
        local binding:  tag: 17
        remote binding: tsr: 192.168.23.1:0, tag: imp-null
```

Ingrese este comando para mostrar los enlaces MPLS que utiliza el protocolo LDP para transportar los paquetes.

CASO DE ESTUDIO 6

CONFIGURACIÓN BÁSICA MPLS DE UN LSR

DESCRIPCIÓN GENERAL Y OBJETIVOS

Este Caso de estudio es el que complementará el Caso de Estudio 5, es decir, aquí realizaremos la configuración del router de Core de la red MPLS que se muestra en en la figura 3.11.

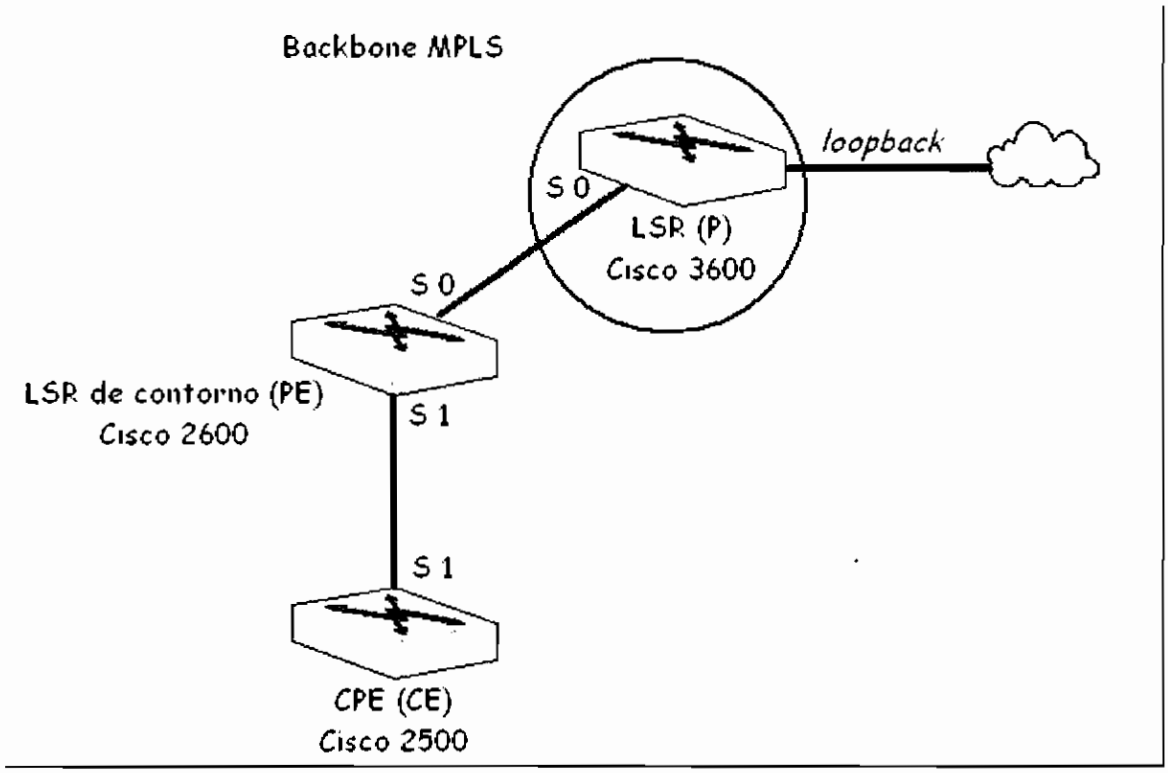


Figura 3.11: Red utilizada en el Caso de Estudio 6

TRABAJO PREPARATORIO Y REQUISITOS

Los requisitos y el trabajo preparatorio son los mismos del Caso de Estudio 5.

CONFIGURACIÓN

ROUTER CE (CISCO 2500)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router CE:

ROUTER CE (Cisco 2500)			
Nombre	LojaNet		
Enable password	loja123		
Console/VTY password	loja		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 1	192.168.21.1/30	DCE
	Ethernet 0	192.168.20.1/24	LAN
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF	192.168.0.0/16	

ROUTER PE (CISCO 2610)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router PE:

ROUTER PE (Cisco 2610)			
Nombre	Pop_de_LojaNet		
Enable password	pop123		
Console/VTY password	poploja		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 0	192.168.22.1/30	DCE
	Serial 1	192.168.21.2/30	DTE
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF	192.168.0.0/16	

ROUTER P (CISCO 3640)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router P:

ROUTER P (Cisco 3640)			
Nombre	Core		
Enable password	core123		
Console/VTY password	core		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 0	192.168.22.2/30	DTE
	Loopback 0	192.168.23.1/32	Simula nube MPLS.
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF	192.168.0.0/16	

Archivo de configuración:

Core

```
Core#show running-config
Building configuration...

Current configuration : 1154 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Core
!
boot-start-marker
boot-end-marker
!
enable password core123
!
no aaa new-model
ip subnet-zero
!
ip cef
ip audit po max-events 100
no ftp-server write-enable
!
interface Loopback0
 ip address 192.168.23.1 255.255.255.255
 no clns route-cache
!
interface Ethernet0/0
```

```
 no ip address
 shutdown
 half-duplex
 no clns route-cache
!
interface TokenRing0/0
 no ip address
 shutdown
 ring-speed 16
 no clns route-cache
!
interface Serial1/0
 ip address 192.168.22.2 255.255.255.252
 tag-switching ip
 no fair
 no clns route-cache
!
interface Serial1/1
 no ip address
 shutdown
 no clns route-cache
!
interface Serial1/2
 no ip address
 shutdown
 no clns route-cache
!
interface Serial1/3
 no ip address
 shutdown
 no clns route-cache
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
ip http server
no ip http secure-server
ip classless
!
line con 0
 password core
 login
line aux 0
line vty 0 4
 password core
 login
!
!
End
Core#
```

Configuración: Core (Ver anexo 3.6.a)

PRUEBAS

La red obtenida al final de la configuración es la mostrada en la figura 3.12.

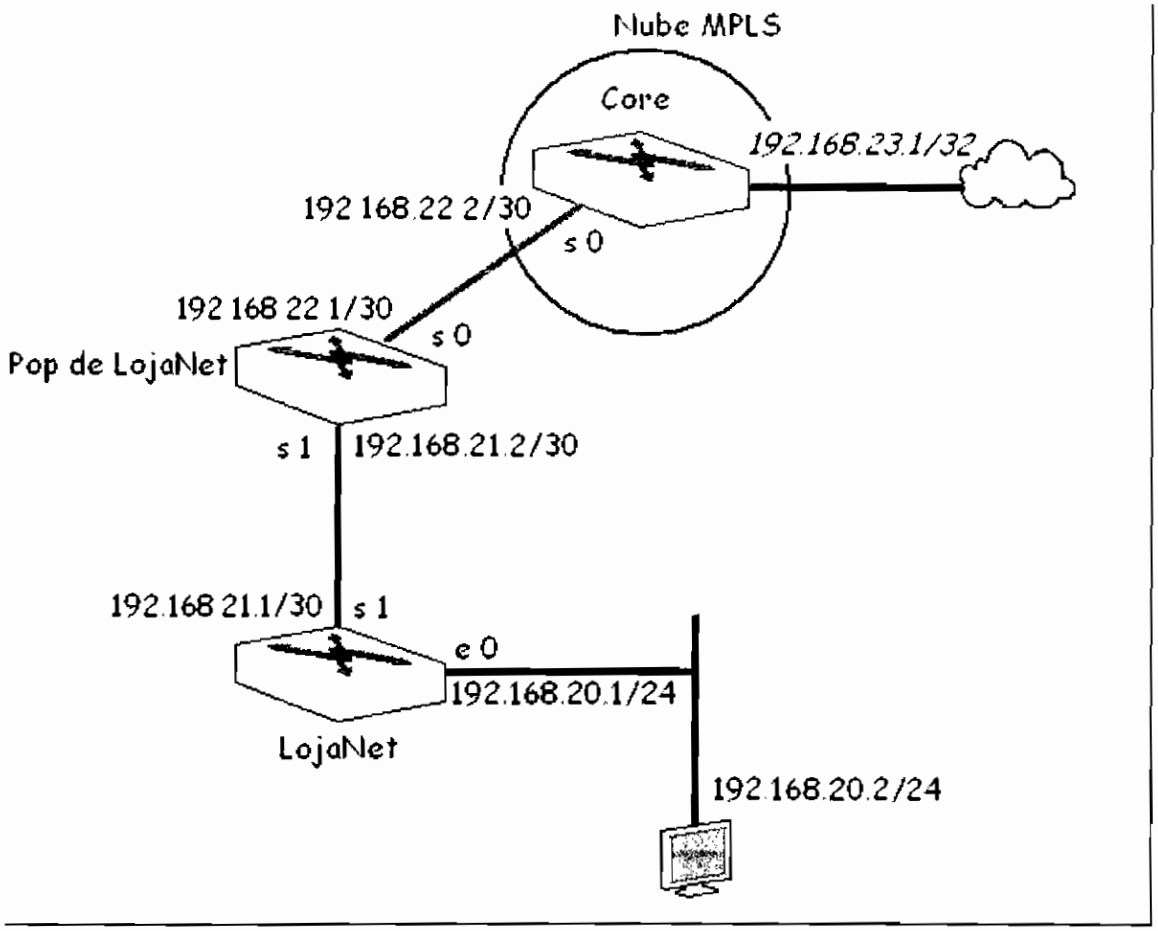


Figura 3.12: Red configurada en el Caso de Estudio 6

Proceso de pruebas:

Como complemento al Caso de Estudio 5 se realizará la verificación de las tablas LFIB para confirmar que el protocolo MPLS está trabajando correctamente. Al configurar la interfaz de loopback 0 del router CORE como ayuda para simular el resto de la nube MPLS y debido a que la característica MPLS no se puede configurar en este tipo de interfaces, el router CORE estaría funcionando como un LSR de contorno y no conmutaría etiquetas sino tan solo realizará acciones pop; el funcionamiento completo se lo verá claramente en los Casos de Estudio 7 y 8 ya que el objetivo de este Caso es únicamente la configuración del router de Core. Note que esta vez se utilizará los comandos mpls/label en lugar de tag/tag para comprobar que no hay diferencia en los resultados obtenidos.

- ***Desde el router CORE:***

Core#sh mpls forwarding-table

```
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
16     16          192.168.20.0/24  0          Se1/0       point2point
17     Pop tag     192.168.21.0/30  0          Se1/0       point2point
```

Core#sh mpls forwarding-table detail

```
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
16     16          192.168.20.0/24  0          Se1/0       point2point
       MAC/Encaps=4/8, MRU=1500, Tag Stack{16}
       0F008847 00010000
       No output feature configured
   Per-packet load-sharing
17     Pop tag     192.168.21.0/30  0          Se1/0       point2point
       MAC/Encaps=4/4, MRU=1504, Tag Stack{}
       0F008847
       No output feature configured
   Per-packet load-sharing
```

Core#sh mpls ldp bindings

```
tib entry: 192.168.20.0/24, rev 6
      local binding:  tag: 16
      remote binding: tsr: 192.168.22.1:0, tag: 16
tib entry: 192.168.21.0/30, rev 8
      local binding:  tag: 17
      remote binding: tsr: 192.168.22.1:0, tag: imp-null
tib entry: 192.168.22.0/30, rev 4
      local binding:  tag: imp-null
      remote binding: tsr: 192.168.22.1:0, tag: imp-null
tib entry: 192.168.23.1/32, rev 2
      local binding:  tag: imp-null
      remote binding: tsr: 192.168.22.1:0, tag: 17
```

Core#sh mpls ldp neighbor

```
   Peer TDP Ident: 192.168.22.1:0; Local TDP Ident 192.168.23.1:0
       TCP connection: 192.168.22.1.711 - 192.168.23.1.11016
       State: Oper; PIEs sent/rcvd: 88/86; Downstream
       Up time: 01:12:56
       TDP discovery sources:
         Serial1/0, Src IP addr: 192.168.22.1
       Addresses bound to peer TDP Ident:
         192.168.22.1    192.168.21.2
```

CASO DE ESTUDIO 7

CONFIGURACIÓN BÁSICA DE UNA NUBE MPLS

DESCRIPCIÓN GENERAL Y OBJETIVOS

En una red MPLS, el enlace PE-CE es muy importante, ya que mediante éste, la red del proveedor y la red del cliente se comunican, y a la vez, es el punto donde se intercambia la información de enrutamiento. En este Caso de Estudio se configurará la red de la empresa ***"CybGonzanama"***, la cual provee servicios de red utilizando la arquitectura MPLS en la ciudad de Loja. En este caso de estudio se configurará una nube MPLS, para luego, complementarla con el caso de Estudio 8, donde se realizará la configuración de un enlace PE-CE lo cual se basará en la configuración de los Casos de Estudio 5 y 6.

La red de la empresa GybGonzanama como se muestra en la figura 3.13 está formada por dos routers Cisco 2600 como LSRs de contorno y un router Cisco 3600 como LSR.

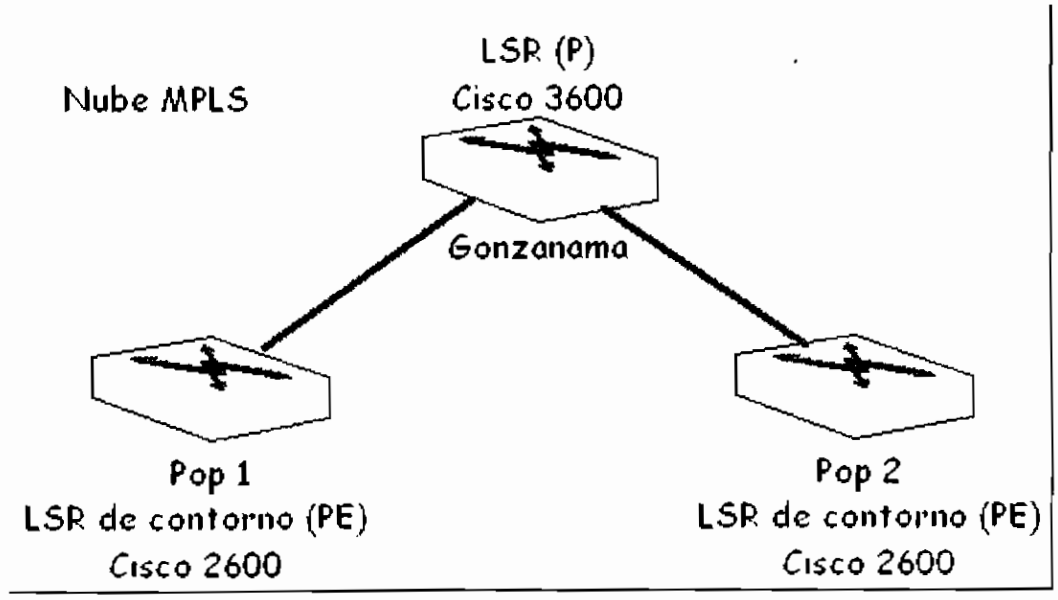


Figura 3.13: Red utilizada en el Caso de Estudio 7

TRABAJO PREPARATORIO

Para desarrollar este Caso de Estudio se deben haber desarrollado los casos de estudio 5 y 6, se debe tener conocimientos básicos sobre el funcionamiento de una red de comunicaciones y los protocolos de enrutamiento como OSPF, también se debe revisar los comandos utilizados para realizar la configuración de una red MPLS, estos son: *ip cef, mpls ip, mpls label protocol, mpls ldp router-id y router ospf.* Será de gran ayuda revisar el numeral 2.1 del Capítulo II

Al final de la configuración se realizarán las pruebas de funcionamiento, las cuales comprenden: pruebas de conectividad, verificar que los procesos de enrutamiento se lleven a cabo correctamente y finalmente comprobar que los paquetes sean etiquetados dentro de la nube MPLS y que dicho protocolo funcione correctamente.

REQUISITOS

La empresa brindará servicios de red utilizando la arquitectura MPLS, por lo tanto, el único requerimiento de la red será brindar conectividad desde el Pop 1 hasta el Pop 2 etiquetando los paquetes que atraviesen dicha red. El protocolo de distribución de etiquetas que se configurará es LDP para notar la diferencia en la configuración de TDP. Cabe indicar que en el funcionamiento global no hay diferencia cuando se utiliza TDP o LDP.

Para las pruebas se utilizará PCs conectados a cada extremo de la red (o interfaces de *loopback*), es decir, uno en la interfaz ethernet del Pop1 y otro en el Pop2 los cuales representarán las redes del cliente.

CONFIGURACIÓN

Todos los comandos ingresados en este caso, están detallados en el Capítulo II.

ROUTER PE (CISCO 2600)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router Pop_1:

ROUTER PE (Cisco 2610)			
Nombre	Pop_1		
Enable password	pop123		
Console/VTY password	pop1		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	loopback 0	10.0.10.1/32	Router-ID
	serial 0	10.0.1.1/24	DCE
	ethernet 0	10.0.3.1/24	LAN
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF	10.0.0.0/16	

Archivo de configuración

Pop_1

```
Pop_1#show running-config
Building configuration...

Current configuration : 1009 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Pop_1
!
boot-start-marker
boot-end-marker
!
enable password pop123
!
memory-size iomem 10
no aaa new-model
ip subnet-zero
!
```

```
!
ip cef
mpls label protocol ldp
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
!
!
interface Loopback0
 ip address 10.0.10.1 255.255.255.255
!
interface Ethernet0/0
 ip address 10.0.3.1 255.255.255.0
 half-duplex
!
interface Serial0/0
 ip address 10.0.1.1 255.255.255.0
 tag-switching ip
 clockrate 56000
 no fair-queue
!
interface Serial0/1
 no ip address
 shutdown
!
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.0.255.255 area 0
!
!
ip http server
ip classless
!
!
!
line con 0
 password pop1
 login
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
 password pop1
 login
 transport preferred all
 transport input all
 transport output all
!
!
end

Pop_1#
```

Configuración: Pop_1 (Ver anexo 3.7.a)

ROUTER PE (CISCO 2610)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router Pop_2:

ROUTER PE (Cisco 2610)			
Nombre	Pop_2		
Enable password	pop123		
Console/VTY password	pop2		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	loopback 0	10.0.10.3/32	Router-ID
	serial 1	10.0.2.2/24	DTE
	loopback 20	10.0.4.1/32	Simula LAN
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF	10.0.0.0/16	

Archivo de configuración:

Pop_2

```
Pop_2#show running-config
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Pop_2
!
boot-start-marker
boot-end-marker
!
enable password pop123
!
no aaa new-model
ip subnet-zero
!
ip cef
mpls label protocol ldp
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
interface Loopback0
 ip address 10.0.10.3 255.255.255.255
!
interface Loopback20
 ip address 10.0.4.1 255.255.255.255
!
interface Ethernet0/0
 no ip address
 shutdown
 half-duplex
!
interface Serial0/0
 no ip address
 shutdown
 no fair-queue
```

```
!
interface Serial0/1
 ip address 10.0.2.2 255.255.255.0
 tag-switching ip
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.0.255.255 area 0
!
ip http server
ip classless
!
line con 0
 password pop2
 login
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
 password pop2
 login
 transport preferred all
 transport input all
 transport output all
!
End

Pop_2#
```

Configuración: Pop_2 (Ver anexo 3.7.b)

ROUTER P (CISCO 3640)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router Gonzanama:

ROUTER P (Cisco 3640)			
Nombre	Gonzanama		
Enable password	gonza123		
Console/VTY password	gonza		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 0	10.0.1.2/24	DTE
	Serial 1	10.0.2.1/24	DCE
	loopback 0	10.0.10.2/32	Router-ID
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF	10.0.0.0/16	

Archivo de configuración

Gonzanama

```
Gonzanama#sh running-config
Building configuration...

Current configuration : 1242 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Gonzanama
!
boot-start-marker
boot-end-marker
!
enable password gonza123
!
no aaa new-model
ip subnet-zero
!
ip cef
ip audit po max-events 100
mpls label protocol ldp
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
interface Loopback0
 ip address 10.0.10.2 255.255.255.255
 no clns route-cache
!
interface Ethernet0/0
 no ip address
 shutdown
 half-duplex
 no clns route-cache
!
interface TokenRing0/0
 no ip address
 shutdown
 ring-speed 16
 no clns route-cache
!
interface Serial1/0
 ip address 10.0.1.2 255.255.255.0
 tag-switching ip
 no clns route-cache
!
interface Serial1/1
 ip address 10.0.2.1 255.255.255.0
 tag-switching ip
 clockrate 56000
 no clns route-cache
!
interface Serial1/2
 no ip address
 shutdown
 no clns route-cache
!
interface Serial1/3
 no ip address
 shutdown
```

```
 no clns route-cache
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.0.255.255 area 0
!
ip http server
no ip http secure-server
ip classless
!
line con 0
 password gonza
 login
line aux 0
line vty 0 4
 password gonza
 login
!
end
Gonzanama#
```

Configuración: Gonzanama (Ver anexo3.7.c)

PRUEBAS

La red obtenida al final de la configuración es la mostrada en la figura 3.14.

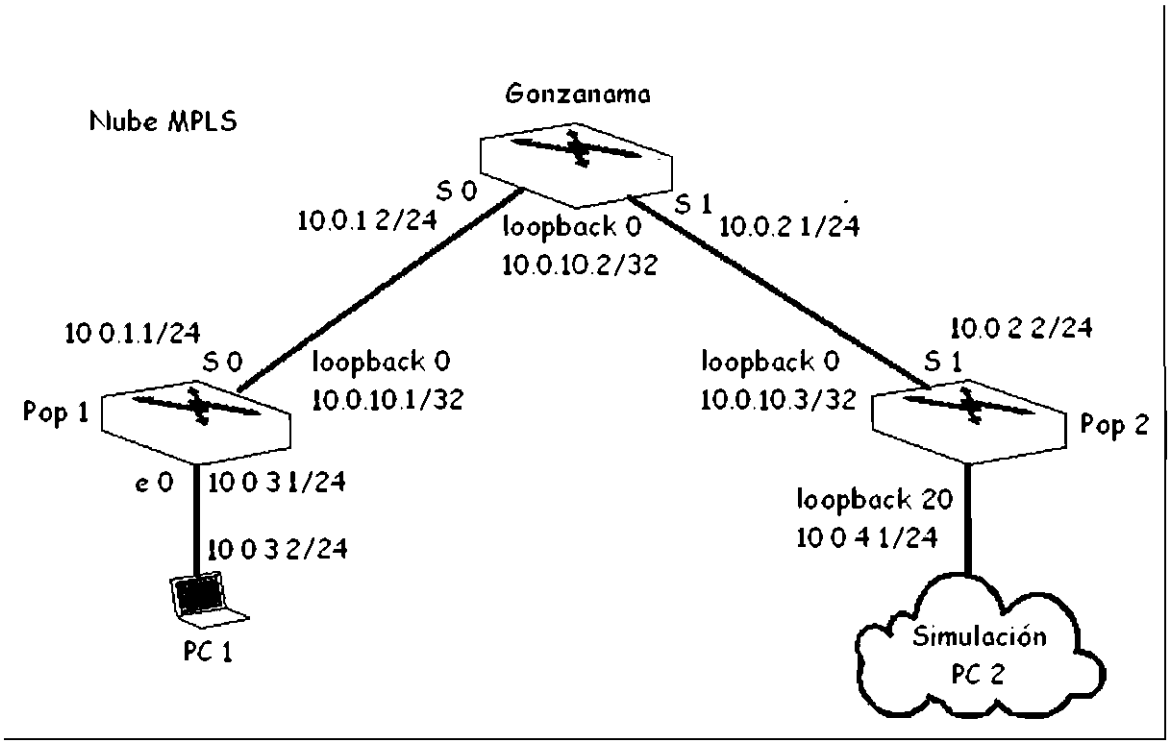


Figura 3.14: Red configurada en el Caso de Estudio 7

Proceso de pruebas:

A continuación se verifica que el proceso OSPF haya iniciado y que esté configurado correctamente (notar en que router es ejecutado el comando):

Gonzanama#

```
*Mar  1 01:11:27.467: %OSPF-5-ADJCHG: Process 1, Nbr 10.1.1.3 on
Serial1/1 from LOADING to FULL, Loading Done
*Mar  1 01:11:33.763: %OSPF-5-ADJCHG: Process 1, Nbr 10.1.1.1 on
Serial1/0 from LOADING to FULL, Loading Done
Gonzanama#
```

Aquí se nota la adyacencia de los vecinos OSPF 10.1.1.3 y 10.1.1.1 e indica que el proceso OSPF se ha iniciado.

Pop_1#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

     10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
C       10.0.10.1/32 is directly connected, Loopback0
O       10.0.10.2/32 [110/782] via 10.0.1.2, 00:13:37, Serial0/0
O       10.0.10.3/32 [110/1563] via 10.0.1.2, 00:13:37, Serial0/0
O       10.0.2.0/24 [110/1562] via 10.0.1.2, 00:13:37, Serial0/0
C       10.0.3.0/24 is directly connected, Ethernet0/0
C       10.0.1.0/24 is directly connected, Serial0/0
O       10.0.4.1/32 [110/1563] via 10.0.1.2, 00:13:37, Serial0/0
```

Gonzanama#sh 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

     10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
O       10.0.10.1/32 [110/782] via 10.0.1.1, 00:15:35, Serial1/0
C       10.0.10.2/32 is directly connected, Loopback0
```

```
O       10.0.10.3/32 [110/782] via 10.0.2.2, 00:15:35, Serial1/1
C       10.0.2.0/24 is directly connected, Serial1/1
O       10.0.3.0/24 [110/791] via 10.0.1.1, 00:15:35, Serial1/0
C       10.0.1.0/24 is directly connected, Serial1/0
O       10.0.4.1/32 [110/782] via 10.0.2.2, 00:15:35, Serial1/1
```

Pop_2#sh 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

     10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
O       10.0.10.1/32 [110/1563] via 10.0.2.1, 00:16:35, Serial0/1
O       10.0.10.2/32 [110/782] via 10.0.2.1, 00:16:35, Serial0/1
C       10.0.10.3/32 is directly connected, Loopback0
C       10.0.2.0/24 is directly connected, Serial0/1
O       10.0.3.0/24 [110/1572] via 10.0.2.1, 00:16:35, Serial0/1
O       10.0.1.0/24 [110/1562] via 10.0.2.1, 00:16:35, Serial0/1
C       10.0.4.1/32 is directly connected, Loopback20
```

Se debe verificar que las rutas estén correctas y que las tres tablas de enrutamiento concuerden.

C:\WINDOWS\system32>ping 10.0.4.1

```
Adaptador Ethernet Conexión de área local          :

        Sufijo de conexión específica DNS :
        Dirección IP. . . . . . . . . . . . : 10.0.3.2
        Máscara de subred . . . . . . . . . : 255.255.255.0
        Puerta de enlace predeterminada    : 10.0.3.1

C:\WINDOWS\system32>ping 10.0.4.1

Haciendo ping a 10.0.4.1 con 32 bytes de datos:

Respuesta desde 10.0.4.1: bytes=32 tiempo=42ms TTL=253
Respuesta desde 10.0.4.1: bytes=32 tiempo=42ms TTL=253
Respuesta desde 10.0.4.1: bytes=32 tiempo=42ms TTL=253
Respuesta desde 10.0.4.1: bytes=32 tiempo=42ms TTL=253

Estadísticas de ping para 10.0.4.1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 42ms, Máximo = 42ms, Media = 42ms

C:\WINDOWS\system32>
```

Finalmente se puede probar la ruta completa enviando un ping desde la PC conectada al Pop_1 con destino a la dirección de loopback 20 del router Pop_2.

Luego se debe verificar la configuración y funcionamiento del protocolo MPLS, para lo cual, se utilizarán los siguientes comandos:

Gonzanama#sh mpls ldp neighbor

```
Peer LDP Ident: 10.0.10.3:0; Local LDP Ident 10.0.10.2:0
        TCP connection: 10.0.10.3.19374 - 10.0.10.2.646
        State: Oper; Msgs sent/rcvd: 19/17; Downstream
        Up time: 00:05:41
        LDP discovery sources:
          Serial1/1, Src IP addr: 10.0.2.2
        Addresses bound to peer LDP Ident:
          10.0.2.2        10.0.10.3       10.0.4.1

Peer LDP Ident: 10.0.10.1:0; Local LDP Ident 10.0.10.2:0
        TCP connection: 10.0.10.1.646 - 10.0.10.2.11159
        State: Oper; Msgs sent/rcvd: 11/11; Downstream
        Up time: 00:01:00
        LDP discovery sources:
          Serial1/0, Src IP addr: 10.0.1.1
        Addresses bound to peer LDP Ident:
          10.0.3.1        10.0.1.1        10.0.10.1
```

Este comando muestra los vecinos que están incluidos en el proceso MPLS.

Gonzanama#sh mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	Pop tag	10.0.3.0/24	5888	Se1/0	point2point
17	Pop tag	10.0.4.1/32	18880	Se1/1	point2point
18	Pop tag	10.0.10.1/32	0	Se1/0	point2point
19	Pop tag	10.0.10.3/32	0	Se1/1	point2point

Gonzanama#sh mpls forwarding-table detail

```
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
16     Pop tag     10.0.3.0/24      14872      Se1/0       point2point
       MAC/Encaps=4/4, MRU=1504, Tag Stack{}
       0F008847
       No output feature configured
   Per-packet load-sharing
17     Pop tag     10.0.4.1/32      27264      Se1/1       point2point
       MAC/Encaps=4/4, MRU=1504, Tag Stack{}
       0F008847
       No output feature configured
   Per-packet load-sharing
18     Pop tag     10.0.10.1/32     0          Se1/0       point2point
       MAC/Encaps=4/4, MRU=1504, Tag Stack{}
       0F008847
       No output feature configured
   Per-packet load-sharing
19     Pop tag     10.0.10.3/32     0          Se1/1       point2point
       MAC/Encaps=4/4, MRU=1504, Tag Stack{}
```

```
        0F008847
        No output feature configured
    Per-packet load-sharing
```

En la LFIB se puede verificar las etiquetas utilizadas en cada interfaz, el tunnel, la interface de salida y los bytes etiquetados que han sido conmutados. Una forma de saber si se están conmutando paquetes etiquetados es observando el contador "Bytes tag switched" éste se incrementa conforme se vayan conmutando los paquetes. En el ejemplo vemos que se han conmutado 14872 paquetes con etiqueta 16 y 27264 con etiqueta 27264.

Gonzanama#sh mpls ldp bindings

```
tib entry: 10.0.1.0/24, rev 25
      local binding:  tag: imp-null
      remote binding: tsr: 10.0.10.3:0, tag: 17
      remote binding: tsr: 10.0.10.1:0, tag: imp-null
tib entry: 10.0.2.0/24, rev 23
      local binding:  tag: imp-null
      remote binding: tsr: 10.0.10.3:0, tag: imp-null
      remote binding: tsr: 10.0.10.1:0, tag: 18
tib entry: 10.0.3.0/24, rev 6
      local binding:  tag: 16
      remote binding: tsr: 10.0.10.3:0, tag: 18
      remote binding: tsr: 10.0.10.1:0, tag: imp-null
tib entry: 10.0.4.1/32, rev 10
      local binding:  tag: 17
      remote binding: tsr: 10.0.10.3:0, tag: imp-null
      remote binding: tsr: 10.0.10.1:0, tag: 19
tib entry: 10.0.10.1/32, rev 12
      local binding:  tag: 18
      remote binding: tsr: 10.0.10.3:0, tag: 19
      remote binding: tsr: 10.0.10.1:0, tag: imp-null
tib entry: 10.0.10.2/32, rev 14
      local binding:  tag: imp-null
      remote binding: tsr: 10.0.10.3:0, tag: 16
      remote binding: tsr: 10.0.10.1:0, tag: 16
tib entry: 10.0.10.3/32, rev 17
      local binding:  tag: 19
      remote binding: tsr: 10.0.10.3:0, tag: imp-null
      remote binding: tsr: 10.0.10.1:0, tag: 17
```

Este comando muestra los enlaces de etiquetas aprendidos por el protocolo LDP y se lo interpreta así: ***a.b.c.d/n*** : prefijo IP y máscara de un destino particular, ***rev***: número de revisión que es usado internamente para administrar la distribución de etiquetas para ese destino, ***local binding:*** etiquetas asignadas por el LSR local, ***remote binding:*** Lista de etiquetas de salida para ese destino aprendido desde otros LSRs, cada item en esta lista identifica el LSR desde el cual la etiqueta

de salida fue aprendida, el LSR es identificado por su identificador LDP.

El comando debug es muy útil para observar el desarrollo del proceso MPLS o cualquier proceso. En este caso vamos a deshabilitar y habilitar una interfaz MPLS con la finalidad de observar el proceso de adyacencia MPLS, para lo cual utilizaremos el comando ***debug tag tdp transport events*** de la siguiente manera:

1.- En el router Pop_1 se deshabilita MPLS en la interfaz serial 0

2.- En el router Gonzanama se introduce el comando:

debug tag tdp transport events interface serial 1/0

3.- Finalmente se habilita MPLS en la interfaz serial 0 del router Pop_1 y se observa en el router Gonzanama el proceso de adyacencia MPLS:

```
Pop_1(config-if)#no mpls ip
Gonzanama#debug tag tdp transport events interface serial 1/0
Pop_1(config-if)#mpls ip

LDP transport events debugging is on for interface Serial1/0
Gonzanama#
*Mar  1 00:23:59.463: ldp: Rcvd ldp hello; Serial1/0, from 10.0.1.1
(10.0.10.1:0), intf_id 0, opt 0xC
*Mar  1 00:23:59.743: ldp: Send ldp hello; Serial1/0, src/dst
10.0.1.2/224.0.0.2, inst_id 0
*Mar  1 00:24:02.503: ldp: ldp conn closed by peer; adj 0x63E2BDAC
     10.0.10.2:11157 <-> 10.0.10.1:646, Serial1/0
*Mar  1 00:24:02.503: ldp: Closing ldp conn 10.0.10.2:11157 <->
10.0.10.1:646, a dj 0x63E2BDAC


*Mar  1 00:24:02.507: ldp: Adj 0x63E2BDAC; state set to closed
*Mar  1 00:24:02.507: %LDP-5-NBRCHG: LDP Neighbor 10.0.10.1:0 is DOWN
*Mar  1 00:24:04.023: ldp: Send ldp hello; Serial1/0, src/dst
10.0.1.2/224.0.0.2, inst_id 0
*Mar  1 00:24:08.271: ldp: Send ldp hello; Serial1/0, src/dst
10.0.1.2/224.0.0, inst_id 0
*Mar  1 00:24:13.203: ldp: Send ldp hello; Serial1/0, src/dst
10.0.1.2/224.0.0.2, inst_id 0


*Mar  1 00:24:15.399: ldp: Rcvd ldp hello; Serial1/0, from 10.0.1.1
(10.0.10.1:0), intf_id 0, opt 0xC
*Mar  1 00:24:15.399: ldp: ldp Hello from 10.0.1.1 (10.0.10.1:0) to
224.0.0.2, opt 0xC
*Mar  1 00:24:15.399: ldp: New adj 0x63E265F0 for 10.0.10.1:0, Serial1/0
*Mar  1 00:24:15.399: ldp:       adj_addr/xport_addr 10.0.1.1/10.0.10.1
*Mar  1 00:24:15.399: ldp: local idb = Serial1/0, holdtime = 15000, peer
```

```
10.01 holdtime = 15000
*Mar  1 00:24:15.399: ldp: Link intvl min cnt = 2, intvl = 5000, idb =
Serial1/0
*Mar  1 00:24:15.403: ldp: Opening ldp conn; adj 0x63E265F0, 10.0.10.2
<-> 10.0.10.1; with normal priority
*Mar  1 00:24:15.427: ldp: Conn failed (TCP activity)!; adj 0x63E265F0,
10.0.1.1, tcb state 0x0
*Mar  1 00:24:15.427: ldp: Closing ldp conn 10.0.10.2:11158 <->
10.0.10.1:646, adj 0x63E265F0
*Mar  1 00:24:15.427: ldp: Adj 0x63E265F0; state set to closed
*Mar  1 00:24:17.007: ldp: Send ldp hello; Serial1/0, , inst_id 0
*Mar  1 00:24:20.259: ldp: Rcvd ldp hello; Serial1/0, from 10.0.1.1
(10.0.10.1:0), intf_id 0, opt 0xC

*Mar  1 00:24:20.259: ldp: ldp Hello from 10.0.1.1 (10.0.10.1:0) to
224.0.0.2, opt 0xC
*Mar  1 00:24:20.259: ldp: New adj 0x63E265F0 for 10.0.10.1:0, Serial1/0
*Mar  1 00:24:20.259: ldp:      adj_addr/xport_addr 10.0.1.1/10.0.10.1
*Mar  1 00:24:20.259: ldp: local idb = Serial1/0, holdtime = 15000, peer
10.0.1. 1 holdtime = 15000
*Mar  1 00:24:20.259: ldp: Link intvl min cnt = 2, intvl = 5000, idb =
Serial1/0
*Mar  1 00:24:20.259: ldp: Opening ldp conn; adj 0x63E265F0, 10.0.10.2
<-> 10.0.10.1; with normal priority
*Mar  1 00:24:20.283: ldp: ldp conn is up; adj 0x63E265F0,
10.0.10.2:11159 <-> 10.0.10.1:646
*Mar  1 00:24:20.335: %LDP-5-NBRCHG: LDP Neighbor 10.0.10.1:0 is UPu
*Mar  1 00:24:21.283: ldp: Send ldp hello; Serial1/0, src/dst
10.0.1.2/224.0.0.2, inst_id 0ndebug al
*Mar  1 00:24:24.767: ldp: Rcvd ldp hello; Serial1/0, from 10.0.1.1
(10.0.10.1:0), intf_id 0, opt 0xC
*Mar  1 00:24:25.467: ldp: Send ldp hello; Serial1/0, src/dst
10.0.1.2/224.0.0.2, inst_id 0l
All possible debugging has been turned off
Gonzanama#
```

En el primer bloque se puede observar que la sesión MPLS se cierra, luego al activar MPLS comienza a funcionar el protocolo Hello y el proceso de adyacencia es inicializado.

CASO DE ESTUDIO 8

CONFIGURACIÓN BÁSICA DE UNA RED MPLS Y DEL SITIO DEL CLIENTE

DESCRIPCIÓN GENERAL Y OBJETIVOS

Una red MPLS puede ser implementada por un proveedor de servicios de red (NSP), para satisfacer las necesidades que sus clientes le exigen. Por lo general una empresa necesita comunicarse con su sucursal ubicada en un área geográfica distinta, es aquí, donde los NSPs intervienen brindando un servicio de red a la empresa-cliente.

En este Caso de Estudio, se configurará la red del NSP y la red del cliente tanto en su matriz como en la sucursal. La empresa ***"MegaNet"*** ubicada en la ciudad de Quito, solicita el servicio de red a ***"TerraNet"***, la cual utiliza la arquitectura MPLS en su backbone. La red completa se muestra en la figura 3.15.

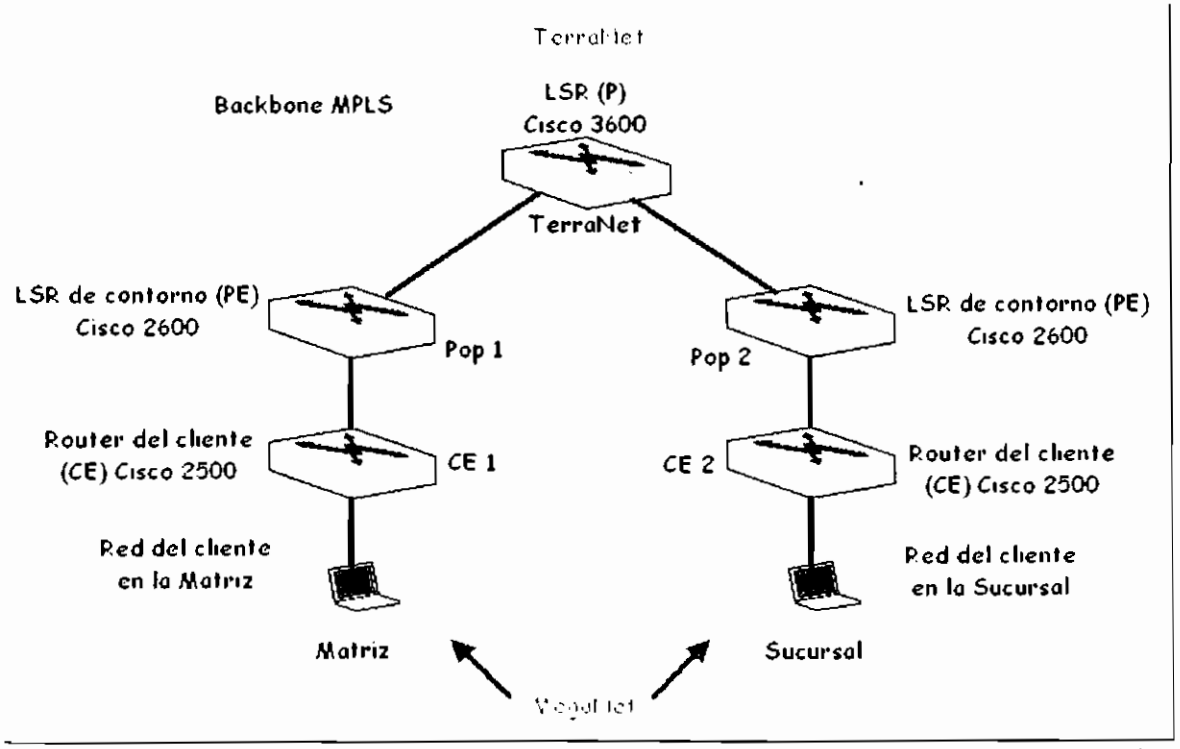


Figura 3.15: Red utilizada en el Caso de Estudio 8

TRABAJO PREPARATORIO

Para desarrollar sin problemas este Caso de Estudio se debe haber completado y comprendido en su totalidad los Casos de Estudio del 1 al 8, ya que a continuación se configurará una red MPLS que servirá para comunicar dos localidades de la empresa MegaNet. Se necesita tener clara la teoría y el funcionamiento del protocolo MPLS tal como se detalla en los casos de estudio anteriores.

REQUISITOS

La red de la empresa MegaNet desea utilizar OSPF como protocolo de enrutamiento, tanto en la matriz como en la sucursal. Posee un router 2500 en la matriz y otro en la sucursal para comunicarse con su NSP (TerraNet). El backbone de TerraNet debe implementar la arquitectura MPLS y dar servicio de red a la empresa MegaNet para interconectar su matriz con la sucursal.

CONFIGURACIÓN

ROUTER CE (CISCO 2500 - MATRIZ)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router de la matriz de la empresa MegaNet:

ROUTER CE (Cisco 2500)			
Nombre	Matriz		
Enable password	matriz123		
Console/VTY password:	matriz		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 0	192.168.20.1/24	DTE
	ethernet 0	10.0.1.1/24	
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF (1)	192.168.0.0/16 10.0.1.0/24	

Archivo de configuración

Matriz

```
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Matriz
!
boot-start-marker
boot-end-marker
!
enable password matriz123
!
no aaa new-model
ip subnet-zero
!
!
interface Ethernet0/0
 ip address 10.0.1.1 255.255.255.0
 half-duplex
!
interface Serial0/0
 ip address 192.168.20.1 255.255.255.0
 no fair-queue
!
interface Ethernet0/1
 no ip address
 shutdown
 half-duplex
!
interface Serial0/1
 no ip address
 shutdown
!
!
!
router ospf 1
 log-adjacency-changes
 network 10.0.1.0 0.0.0.255 area 0
 network 192.168.0.0 0.0.255.255 area 0
!
ip http server
ip classless
!
!
gatekeeper
 shutdown
!
line con 0
 password matriz
 login
line aux 0
line vty 0 4
 password matriz
 login
!
!
End
```

Configuración: Matriz (Ver anexo 3.8.a)

ROUTER CE (CISCO 2500 - SUCURSAL)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router Sucursal de la empresa MegaNet:

ROUTER CE (Cisco 2500)			
Nombre	Sucursal		
Enable password	sucursal123		
Console/VTY password	sucursal		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 0	192.168.23.2/24	DCE
	ethernet 0	172.16.1.1/24	
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF (1)	192.168.0.0/16 172.16.1.0/24	

Archivo de configuración

Sucursal

```
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Sucursal
!
boot-start-marker
boot-end-marker
!
enable password sucursal123
!
no aaa new-model
ip subnet-zero
!
ip cef
ip audit po max-events 100
no ftp-server write-enable
!
interface Loopback30
 ip address 172.16.1.1 255.255.255.0
 no clns route-cache
!
interface Ethernet0/0
 no ip address
 shutdown
```

```
 half-duplex
 no clns route-cache
!
interface Serial0/0
 ip address 192.168.23.2 255.255.255.0
 clockrate 56000
 no fair-queue
 no clns route-cache
!
interface Serial0/1
 no ip address
 shutdown
 no clns route-cache
!
router ospf 1
 log-adjacency-changes
 network 172.16.1.0 0.0.0.255 area 0
 network 192.168.0.0 0.0.255.255 area 0
!
ip http server
no ip http secure-server
ip classless
!
line con 0
 password sucursal
 login
line aux 0
line vty 0 4
 password sucursal
 login
!
end
```

Configuración: Sucursal (Ver anexo 3.8.b)

ROUTER P (CISCO 2610)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router Plaza (PE) de TerraNet al que se conectará la Matriz de la empresa MegaNet:

ROUTER PE (Cisco 2610)			
Nombre	Plaza		
Enable password	plaza123		
Console/VTY password	plaza		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 0	192.168.20.2/24	DCE
	Serial 1	192.168.21.1/24	DCE
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF (1)	192.168.0.0/16	

Archivo de configuración

Plaza

```
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Plaza
!
boot-start-marker
boot-end-marker
!
enable password plaza123
!
memory-size iomem 10
no aaa new-model
ip subnet-zero
!
!
ip cef
mpls label protocol ldp
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
interface Loopback0
 ip address 192.168.10.1 255.255.255.255
!
interface Ethernet0/0
 no ip address
 shutdown
 half-duplex
!
interface Serial0/0
 ip address 192.168.20.2 255.255.255.0
 clockrate 56000
 no fair-queue
!
interface Serial0/1
 ip address 192.168.21.1 255.255.255.0
 tag-switching ip
 clockrate 56000
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
ip http server
ip classless
!
!
!
!
line con 0
 password plaza
 login
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
```

```
 password plaza
 login
 transport preferred all
 transport input all
 transport output all
!
end
```

Configuración: Plaza (Ver anexo 3.8.c)

ROUTER PE (CISCO 2610)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router PE de TerraNet al que se conectará la Sucursal de la empresa MegaNet:

ROUTER PE (Cisco 2610)			
Nombre	Jipiro		
Enable password	jipiro123		
Console/VTY password	jipiro		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 0	192.168.23.1/24	DTE
	Serial 1	192.168.22.2/24	DTE
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	BGP (1)	192.168.0.0/16	

Archivo de configuración

Jipiro

```
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Jipiro
!
boot-start-marker
boot-end-marker
!
enable password jipiro123
!
no aaa new-model
ip subnet-zero
!
!
```

```
ip cef
mpls label protocol ldp
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
!
!
!
interface Loopback0
 ip address 192.168.10.3 255.255.255.255
!
interface Ethernet0/0
 no ip address
 shutdown
 half-duplex
!
interface Serial0/0
 ip address 192.168.23.1 255.255.255.0
 no fair-queue
!
interface Serial0/1
 ip address 192.168.22.2 255.255.255.0
 tag-switching ip
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
ip http server
ip classless
!
!
!
!
line con 0
 password jipiro
 login
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
 password jipiro
 login
 transport preferred all
 transport input all
 transport output all
!
end
```

Configuración: Jipiro (Ver anexo 3.8.d)

ROUTER P (CISCO 3640)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router P del backbone MPLS de la empresa TerraNet:

ROUTER P (Cisco 3640)			
Nombre	TerraNet		
Enable password	terra123		
Console/VTY password	terra		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 0	192.168.21.2/24	DTE
	Serial 1	192.168.22.1/24	DCE
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	BGP (1)	192.168.0.0/16	

Archivo de configuración

TerraNet

```
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname TerraNet
!
boot-start-marker
boot-end-marker
!
enable password terra123
!
no aaa new-model
ip subnet-zero
!
!
!
ip cef
ip audit po max-events 100
mpls label protocol ldp
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
!
!
interface Loopback0
 ip address 192.168.10.2 255.255.255.255
 no clns route-cache
!
!
interface Ethernet0/0
 no ip address
 shutdown
 half-duplex
 no clns route-cache
!
!
```

```
interface TokenRing0/0
 no ip address
 shutdown
 ring-speed 16
 no clns route-cache
!
!
interface Serial1/0
 ip address 192.168.21.2 255.255.255.0
 tag-switching ip
 no fair-queue
 no clns route-cache
!
!
interface Serial1/1
 ip address 192.168.22.1 255.255.255.0
 tag-switching ip
 clockrate 56000
 no clns route-cache
!
!
interface Serial1/2
 no ip address
 shutdown
 no clns route-cache
!
interface Serial1/3
 no ip address
 shutdown
 no clns route-cache
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
ip http server
no ip http secure-server
ip classless
!
line con 0
 password terra
 login
line aux 0
line vty 0 4
 password terra
 login
!
end
```

Configuración: TerraNet (Ver anexo 3.8.e)

PRUEBAS

La red obtenida al final de la configuración es la mostrada en la figura 3.16, a la cual se le realizarán las pruebas que se detallan a continuación y verificarán como actúa el protocolo MPLS y OSPF en un backbone completo

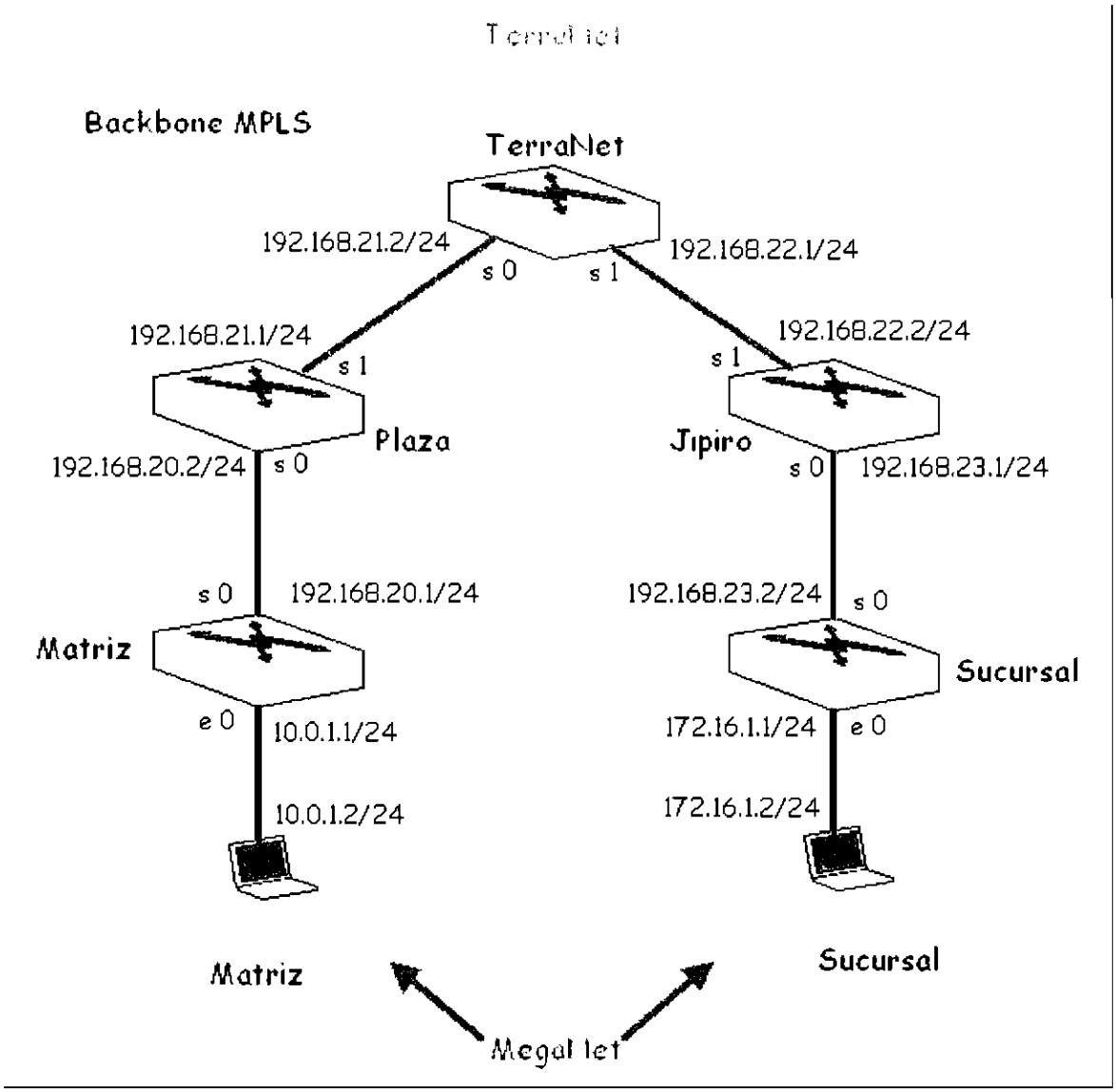


Figura 3.16: Red configurada en el Caso de Estudio 8

Proceso de pruebas:

Los resultados de la verificación se muestran seguidos del comando utilizado. Primero se verificará el funcionamiento del protocolo de enrutamiento OSPF realizando el proceso utilizado en casos de estudio anteriores:

Revisar las tablas de enrutamiento:

```
Matriz#sh 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

     192.168.10.0/32 is subnetted, 3 subnets
O       192.168.10.2 [110/846] via 192.168.20.2, 00:06:38, Serial0/0
O       192.168.10.3 [110/1627] via 192.168.20.2, 00:06:38, Serial0/0
O       192.168.10.1 [110/65] via 192.168.20.2, 00:06:38, Serial0/0
     172.16.0.0/32 is subnetted, 1 subnets
O       172.16.1.1 [110/2408] via 192.168.20.2, 00:06:38, Serial0/0
O    192.168.21.0/24 [110/845] via 192.168.20.2, 00:06:38, Serial0/0
C    192.168.20.0/24 is directly connected, Serial0/0
     10.0.0.0/24 is subnetted, 1 subnets
C       10.0.1.0 is directly connected, Ethernet0/0
O    192.168.23.0/24 [110/2407] via 192.168.20.2, 00:06:38, Serial0/0
O    192.168.22.0/24 [110/1626] via 192.168.20.2, 00:06:46, Serial0/0
```

Plaza#sh 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

     192.168.10.0/32 is subnetted, 3 subnets
O       192.168.10.2 [110/782] via 192.168.21.2, 00:07:21, Serial0/1
O       192.168.10.3 [110/1563] via 192.168.21.2, 00:07:21, Serial0/1
C       192.168.10.1 is directly connected, Loopback0
     172.16.0.0/32 is subnetted, 1 subnets
O       172.16.1.1 [110/2344] via 192.168.21.2, 00:07:21, Serial0/1
C    192.168.21.0/24 is directly connected, Serial0/1
C    192.168.20.0/24 is directly connected, Serial0/0
     10.0.0.0/24 is subnetted, 1 subnets
O       10.0.1.0 [110/791] via 192.168.20.1, 00:07:21, Serial0/0
O    192.168.23.0/24 [110/2343] via 192.168.21.2, 00:07:21, Serial0/1
O    192.168.22.0/24 [110/1562] via 192.168.21.2, 00:07:23, Serial0/1
```

TerraNet#sh 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

     192.168.10.0/32 is subnetted, 3 subnets
C       192.168.10.2 is directly connected, Loopback0
O       192.168.10.3 [110/782] via 192.168.22.2, 00:07:36, Serial1/1
O       192.168.10.1 [110/782] via 192.168.21.1, 00:07:36, Serial1/0
     172.16.0.0/32 is subnetted, 1 subnets
O       172.16.1.1 [110/1563] via 192.168.22.2, 00:07:36, Serial1/1
C    192.168.21.0/24 is directly connected, Serial1/0
O    192.168.20.0/24 [110/1562] via 192.168.21.1, 00:07:36, Serial1/0
     10.0.0.0/24 is subnetted, 1 subnets
O       10.0.1.0 [110/1572] via 192.168.21.1, 00:07:36, Serial1/0
O    192.168.23.0/24 [110/1562] via 192.168.22.2, 00:07:36, Serial1/1
C    192.168.22.0/24 is directly connected, Serial1/1
```

Jipiro#sh 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

     192.168.10.0/32 is subnetted, 3 subnets
O       192.168.10.2 [110/782] via 192.168.22.1, 00:07:51, Serial0/1
C       192.168.10.3 is directly connected, Loopback0
O       192.168.10.1 [110/1563] via 192.168.22.1, 00:07:51, Serial0/1
     172.16.0.0/32 is subnetted, 1 subnets
O       172.16.1.1 [110/782] via 192.168.23.2, 00:07:51, Serial0/0
O    192.168.21.0/24 [110/1562] via 192.168.22.1, 00:07:51, Serial0/1
O    192.168.20.0/24 [110/2343] via 192.168.22.1, 00:07:51, Serial0/1
     10.0.0.0/24 is subnetted, 1 subnets
O       10.0.1.0 [110/2353] via 192.168.22.1, 00:07:51, Serial0/1
C    192.168.23.0/24 is directly connected, Serial0/0
C    192.168.22.0/24 is directly connected, Serial0/1
```

Sucursal#sh 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

     192.168.10.0/32 is subnetted, 3 subnets
O       192.168.10.2 [110/846] via 192.168.23.1, 00:08:25, Serial0/0
```

```
O       192.168.10.3 [110/65] via 192.168.23.1, 00:08:25, Serial0/0
O       192.168.10.1 [110/1627] via 192.168.23.1, 00:08:25, Serial0/0
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.1.0 is directly connected, Loopback30
O    192.168.21.0/24 [110/1626] via 192.168.23.1, 00:08:25, Serial0/0
O    192.168.20.0/24 [110/2407] via 192.168.23.1, 00:08:25, Serial0/0
     10.0.0.0/24 is subnetted, 1 subnets
O       10.0.1.0 [110/2417] via 192.168.23.1, 00:08:25, Serial0/0
C    192.168.23.0/24 is directly connected, Serial0/0
O    192.168.22.0/24 [110/845] via 192.168.23.1, 00:08:27, Serial0/0
```

Para verificar que la red ha convergido completamente debemos comprobar que el ping de extremo ha extremo se exitoso, para lo cual realizamos lo siguiente:

C:\WINDOWS\system32>ipconfig

```
Configuración IP de Windows

Adaptador Ethernet Conexión de área local          :

        Sufijo de conexión específica DNS :
        Dirección IP. . . . . . . . . . . . : 10.0.1.2
        Máscara de subred . . . . . . . . . : 255.255.255.0
        Puerta de enlace predeterminada    : 10.0.1.1
```

C:\WINDOWS\system32>ping 172.16.1.1

```
Haciendo ping a 172.16.1.1 con 32 bytes de datos:

Respuesta desde 172.16.1.1: bytes=32 tiempo=85ms TTL=251
Respuesta desde 172.16.1.1: bytes=32 tiempo=84ms TTL=251
Respuesta desde 172.16.1.1: bytes=32 tiempo=84ms TTL=251
Respuesta desde 172.16.1.1: bytes=32 tiempo=84ms TTL=251

Estadísticas de ping para 172.16.1.1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 84ms, Máximo = 85ms, Media = 84ms
```

C:\WINDOWS\system32>tracert 172.16.1.1

```
Traza a 172.16.1.1 sobre caminos de 30 saltos como máximo.

  1     1 ms    <1 ms    <1 ms  10.0.1.1
  2    25 ms    25 ms    25 ms  192.168.20.2
  3   186 ms   186 ms   186 ms  192.168.21.2
  4   124 ms   124 ms   124 ms  192.168.22.2
  5   121 ms   121 ms   121 ms  172.16.1.1

Traza completa
```

Jipiro#traceroute 10.0.1.2

```
Tracing the route to 10.0.1.2

  1 192.168.22.1 [MPLS: Label 16 Exp 0] 149 msec 157 msec 144 msec
  2 192.168.21.1 [MPLS: Label 16 Exp 0] 76 msec 88 msec 76 msec
  3 192.168.20.1 60 msec 68 msec 64 msec
  4 10.0.1.2 60 msec 64 msec 60 msec
```

Ahora se verificará que el protocolo MPLS esté corriendo sin problemas. Se revisará la tabla LFIB, los enlaces (bindings) MPLS y la tabla CEF que es de donde nace todo el proceso MPLS.

TerraNet#sh mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	16	10.0.1.0/24	33178	Se1/0	point2point
17	Pop tag	192.168.10.1/32	0	Se1/0	point2point
18	Pop tag	192.168.20.0/24	1152	Se1/0	point2point
19	Pop tag	192.168.10.3/32	0	Se1/1	point2point
20	Pop tag	192.168.23.0/24	2371	Se1/1	point2point
21	21	172.16.1.1/32	21212	Se1/1	point2point

Plaza#sh mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	Untagged	10.0.1.0/24	41849	Se0/0	point2point
17	Pop tag	192.168.10.2/32	0	Se0/1	point2point
18	Pop tag	192.168.22.0/24	0	Se0/1	point2point
19	19	192.168.10.3/32	0	Se0/1	point2point
20	20	192.168.23.0/24	0	Se0/1	point2point
21	21	172.16.1.1/32	0	Se0/1	point2point

Jipiro#sh mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	16	10.0.1.0/24	0	Se0/1	point2point
17	17	192.168.10.1/32	0	Se0/1	point2point
18	Pop tag	192.168.10.2/32	0	Se0/1	point2point
19	18	192.168.20.0/24	0	Se0/1	point2point
20	Pop tag	192.168.21.0/24	0	Se0/1	point2point
21	Untagged	172.16.1.1/32	20432	Se0/0	point2point

TerraNet#sh mpls forwarding-table detail

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	16	10.0.1.0/24	75081	Se1/0	point2point
	MAC/Encaps=4/8, MRU=1500, Tag Stack{16}				

```
        0F008847 00010000
        No output feature configured
    Per-packet load-sharing
17      Pop tag     192.168.10.1/32   0          Se1/0      point2point
        MAC/Encaps=4/4, MRU=1504, Tag Stack{}
        0F008847
        No output feature configured
    Per-packet load-sharing
18      Pop tag     192.168.20.0/24   1152       Se1/0      point2point
        MAC/Encaps=4/4, MRU=1504, Tag Stack{}
        0F008847
        No output feature configured
    Per-packet load-sharing
19      Pop tag     192.168.10.3/32   0          Se1/1      point2point
        MAC/Encaps=4/4, MRU=1504, Tag Stack{}
        0F008847
        No output feature configured
    Per-packet load-sharing
20      Pop tag     192.168.23.0/24   2938       Se1/1      point2point
        MAC/Encaps=4/4, MRU=1504, Tag Stack{}
        0F008847
        No output feature configured
    Per-packet load-sharing
21      21          172.16.1.1/32     58840      Se1/1      point2point
        MAC/Encaps=4/8, MRU=1500, Tag Stack{21}
        0F008847 00015000
        No output feature configured
    Per-packet load-sharing
```

Para mostrar información específica con respecto a los enlaces de etiquetas aprendidas mediante el protocol LDP o TDP se debe utilizar el comando *show mpls ip bindings* en el modo privilegiado.

TerraNet#sh mpls ip binding

```
  10.0.1.0/24
        in label:     16
        out label:    16        lsr: 192.168.21.1:0   inuse
        out label:    16        lsr: 192.168.22.2:0
  172.16.1.1/32
        in label:     21
        out label:    21        lsr: 192.168.21.1:0
        out label:    21        lsr: 192.168.22.2:0   inuse
  192.168.10.1/32
        in label:     17
        out label:    imp-null  lsr: 192.168.21.1:0   inuse
        out label:    17        lsr: 192.168.22.2:0
  192.168.10.2/32
        in label:     imp-null
        out label:    17        lsr: 192.168.21.1:0
        out label:    18        lsr: 192.168.22.2:0
  192.168.10.3/32
        in label:     19
        out label:    19        lsr: 192.168.21.1:0
        out label:    imp-null  lsr: 192.168.22.2:0   inuse
  192.168.20.0/24
        in label:     18
        out label:    imp-null  lsr: 192.168.21.1:0   inuse
        out label:    19        lsr: 192.168.22.2:0
```

```
192.168.21.0/24
      in label:     imp-null
      out label:    imp-null  lsr: 192.168.21.1:0
      out label:    20        lsr: 192.168.22.2:0
192.168.22.0/24
      in label:     imp-null
      out label:    18        lsr: 192.168.21.1:0
      out label:    imp-null  lsr: 192.168.22.2:0
192.168.23.0/24
      in label:     20
      out label:    imp-null  lsr: 192.168.22.2:0   inuse
      out label:    20        lsr: 192.168.21.1:0
```

inuse: indica que la etiqueta de salida está en uso para el envío MPLS, esto quiere decir, que la etiqueta está instalada en la tabla de envío MPLS (LFIB.

TerraNet#sh mpls ldp bindings

```
tib entry: 10.0.1.0/24, rev 10
      local binding:  tag: 16
      remote binding: tsr: 192.168.21.1:0, tag: 16
      remote binding: tsr: 192.168.22.2:0, tag: 16
tib entry: 172.16.1.1/32, rev 18
      local binding:  tag: 21
      remote binding: tsr: 192.168.21.1:0, tag: 21
      remote binding: tsr: 192.168.22.2:0, tag: 21
tib entry: 192.168.10.1/32, rev 11
      local binding:  tag: 17
      remote binding: tsr: 192.168.21.1:0, tag: imp-null
      remote binding: tsr: 192.168.22.2:0, tag: 17
tib entry: 192.168.10.2/32, rev 6
      local binding:  tag: imp-null
      remote binding: tsr: 192.168.21.1:0, tag: 17
      remote binding: tsr: 192.168.22.2:0, tag: 18
tib entry: 192.168.10.3/32, rev 14
      local binding:  tag: 19
      remote binding: tsr: 192.168.21.1:0, tag: 19
      remote binding: tsr: 192.168.22.2:0, tag: imp-null
tib entry: 192.168.20.0/24, rev 12
      local binding:  tag: 18
      remote binding: tsr: 192.168.21.1:0, tag: imp-null
      remote binding: tsr: 192.168.22.2:0, tag: 19
tib entry: 192.168.21.0/24, rev 2
      local binding:  tag: imp-null
      remote binding: tsr: 192.168.21.1:0, tag: imp-null
      remote binding: tsr: 192.168.22.2:0, tag: 20
tib entry: 192.168.22.0/24, rev 4
      local binding:  tag: imp-null
      remote binding: tsr: 192.168.21.1:0, tag: 18
      remote binding: tsr: 192.168.22.2:0, tag: imp-null
tib entry: 192.168.23.0/24, rev 16
      local binding:  tag: 20
      remote binding: tsr: 192.168.22.2:0, tag: imp-null
      remote binding: tsr: 192.168.21.1:0, tag: 20
```

Para mostrar las entradas de la tabla FIB (Forwarding Information Base) utilice el comando *show ip cef.*

TerraNet#sh ip cef detail

```
IP CEF with switching (Table Version 20), flags=0x0
  20 routes, 0 reresolve, 0 unresolved (0 old, 0 new), peak 0
  20 leaves, 18 nodes, 21440 bytes, 28 inserts, 8 invalidations
  0 load sharing elements, 0 bytes, 0 references
  universal per-destination load sharing algorithm, id 05937D7E
  3(0) CEF resets, 0 revisions of existing leaves
  Resolution Timer: Exponential (currently 1s, peak 1s)
  0 in-place/0 aborted modifications
  refcounts:  4885 leaf, 4864 node

  Table epoch: 0 (20 entries at this epoch)

Adjacency Table has 4 adjacencies
0.0.0.0/0, version 0, epoch 0, attached, default route handler
0 packets, 0 bytes
  via 0.0.0.0, 0 dependencies
    valid no route adjacency
0.0.0.0/32, version 1, epoch 0, receive
10.0.1.0/24, version 14, epoch 0, cached adjacency to Serial1/0
0 packets, 0 bytes
  tag information set
    local tag: 16
    fast tag rewrite with Se1/0, point2point, tags imposed: {16}
  via 192.168.21.1, Serial1/0, 0 dependencies
    next hop 192.168.21.1, Serial1/0
    valid cached adjacency
    tag rewrite with Se1/0, point2point, tags imposed: {16}
172.16.1.1/32, version 19, epoch 0, cached adjacency to Serial1/1
0 packets, 0 bytes
  tag information set
    local tag: 21
    fast tag rewrite with Se1/1, point2point, tags imposed: {21}
  via 192.168.22.2, Serial1/1, 0 dependencies
    next hop 192.168.22.2, Serial1/1
    valid cached adjacency
    tag rewrite with Se1/1, point2point, tags imposed: {21}
192.168.10.1/32, version 15, epoch 0, cached adjacency to Serial1/0
0 packets, 0 bytes
  tag information set
    local tag: 17
  via 192.168.21.1, Serial1/0, 0 dependencies
    next hop 192.168.21.1, Serial1/0
    valid cached adjacency
    tag rewrite with Se1/0, point2point, tags imposed: {}
192.168.10.2/32, version 13, epoch 0, connected, receive
  tag information set
    local tag: implicit-null
192.168.10.3/32, version 17, epoch 0, cached adjacency to Serial1/1
0 packets, 0 bytes
  tag information set
    local tag: 19
  via 192.168.22.2, Serial1/1, 0 dependencies
    next hop 192.168.22.2, Serial1/1
    valid cached adjacency
    tag rewrite with Se1/1, point2point, tags imposed: {}
192.168.20.0/24, version 16, epoch 0, cached adjacency to Serial1/0
0 packets, 0 bytes
  tag information set
    local tag: 18
  via 192.168.21.1, Serial1/0, 0 dependencies
    next hop 192.168.21.1, Serial1/0
    valid cached adjacency
    tag rewrite with Se1/0, point2point, tags imposed: {}
```

```
192.168.21.0/24, version 8, epoch 0, attached, connected, cached
adjacency to Se
rial1/0
0 packets, 0 bytes
  tag information set
    local tag: implicit-null
  via Serial1/0, 0 dependencies
    valid cached adjacency
192.168.21.0/32, version 5, epoch 0, receive
192.168.21.2/32, version 4, epoch 0, receive
192.168.21.255/32, version 6, epoch 0, receive
192.168.22.0/24, version 12, epoch 0, attached, connected, cached
adjacency to S
erial1/1
0 packets, 0 bytes
  tag information set
    local tag: implicit-null
  via Serial1/1, 0 dependencies
    valid cached adjacency
192.168.22.0/32, version 10, epoch 0, receive
192.168.22.1/32, version 9, epoch 0, receive
192.168.22.255/32, version 11, epoch 0, receive
192.168.23.0/24, version 18, epoch 0, cached adjacency to Serial1/1
0 packets, 0 bytes
  tag information set
    local tag: 20
  via 192.168.22.2, Serial1/1, 0 dependencies
    next hop 192.168.22.2, Serial1/1
    valid cached adjacency
    tag rewrite with Se1/1, point2point, tags imposed: {}
224.0.0.0/4, version 7, epoch 0
0 packets, 0 bytes, Precedence routine (0)
  via 0.0.0.0, 0 dependencies
    next hop 0.0.0.0
    valid drop adjacency
224.0.0.0/24, version 3, epoch 0, receive
255.255.255.255/32, version 2, epoch 0, receive
```

CASO DE ESTUDIO 9

CONFIGURACIÓN VPN - MPLS

DESCRIPCIÓN GENERAL Y OBJETIVOS

Una red privada virtual (VPN) se define como una red en la que la conectividad cliente entre varios sitios se distribuye en una infraestructura compartida con las mismas normas de acceso o seguridad que en una red privada. Con la reciente llegada de actividades de mercadotecnia que rodean al término VPN, desde nuevas tecnologías que soportan las VPN hasta nuevos productos y servicios que permiten habilitar un entorno VPN, se podría pensar que el concepto VPN hace referencia a una importante tecnología. Sin embargo, como suele ser habitual, VPN es un concepto que tiene más de diez años y es muy conocido en el ámbito de los proveedores de servicios*.

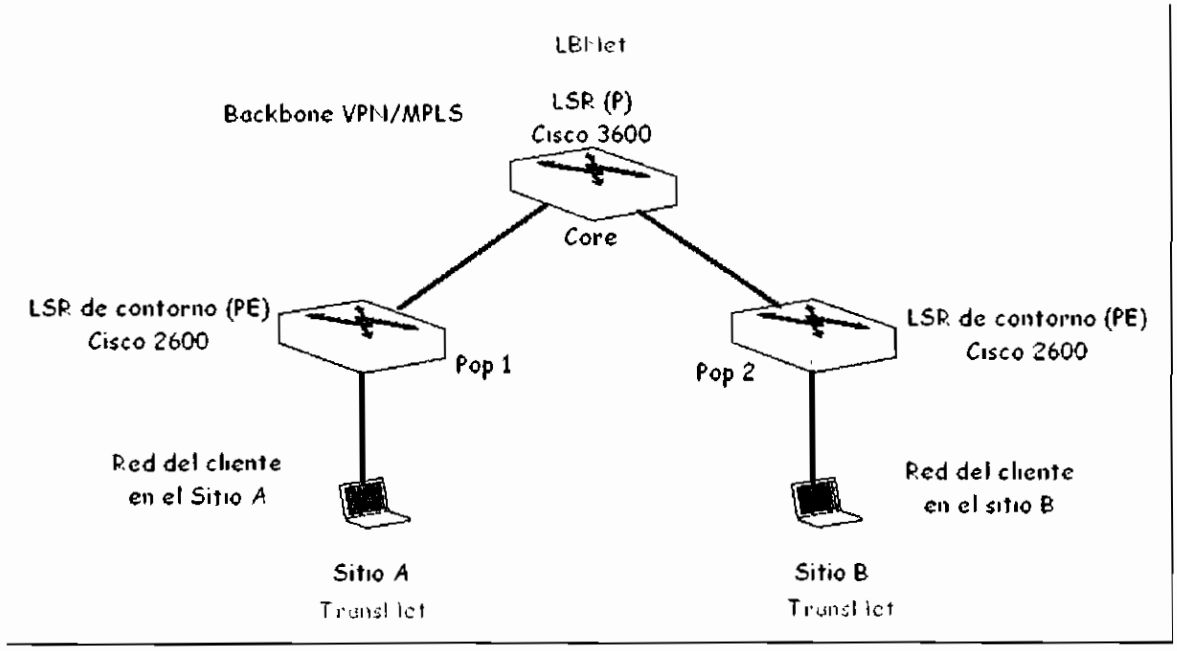


Figura 3.17: Red utilizada en el Caso de Estudio 9

Una de las topologías más simple que se puede suministrar utilizando la arquitectura VPN/MPLS es una intranet entre los diferentes sitios que pertenecen a una misma empresa. En este Caso de Estudio se configurará la red de LBNet

* VPN and MPLS Arquitectures 2000

que se encarga de dar servicio de red a la empresa TransNet. La red ha configurarse se muestra en la figura 3.17.

TRABAJO PREPARATORIO

Como trabajo preparatorio para este Caso se debe tener bien claro cómo funciona una VPN y haber entendido a la perfección el numeral 2.2 del Capítulo II. Sería de gran ayuda tener conocimientos básicos de la configuración de los protocolos de enrutamiento OSPF y BGP ya que éstos interactúan en un backbone MPLS/VPN.

REQUISITOS

La empresa TransNet informa que utilizará OSPF como protocolo de enrutamiento en sus dos redes y solicita a su NSP que le brinde un servicio de VPN. La empresa LBNet debe implementar la arquitectura MPLS en su backbone y satisfacer las necesidades de su cliente TransNet, para lo cual a decidido implementar una arquitectura VPN/MPLS. TransNet dispone de dos routers Cisco 2600 que serán configurados como routers PE y de un Cisco 3640 que hará la función de router de Core. Los sitios de red serán simulados con una PC (o interfaces de *loopback)* en cada lado pero es necesario aclarar que es posible que el cliente disponga de un router en cada sitio para que sean administrados por el cliente, en cualquiera de los dos casos la configuración del backbone VPN/MPLS es la misma. La forma de conectar los equipos se encuentra detallada en los esquemas de configuración.

CONFIGURACIÓN

ROUTER CE (CISCO 2610 – Pop Mariscal)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router PE (Pop1) de la empresa LBNet:

ROUTER PE (Cisco 2610)			
Nombre	Pop_Mariscal		
Enable password	maris123		
Console/VTY password	maris		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 0	192.168.21.1/24	DCE
	Ethernet 0	192.168.20.1/24	LAN
	Loopback 0	192.168.40.1/24	LDP-ID
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF (1)	192.168.0.0/16	
	BGP (100)	192.168.40.3	remoto

Archivo de configuración

Pop_Mariscal

```
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
!
!
hostname Pop_Mariscal
!
!
!
boot-start-marker
boot-end-marker
!
enable password maris123
!
!
!
memory-size iomem 10
no aaa new-model
ip subnet-zero
!
!
ip cef
ip vrf transnet
 rd 100:110
 route-target export 100:1000
 route-target import 100:1000
!
mpls label protocol ldp
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
```

```
!
!
!
interface Loopback0
 ip address 192.168.40.1 255.255.255.255
!
interface Ethernet0/0
 ip vrf forwarding transnet
 ip address 192.168.20.1 255.255.255.0
 half-duplex
!
interface Serial0/0
 ip address 192.168.21.1 255.255.255.0
 tag-switching ip
 clockrate 56000
 no fair-queue
!
interface Serial0/1
 no ip address
 shutdown
!
!
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
!
!
router bgp 100
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 192.168.40.3 remote-as 100
 neighbor 192.168.40.3 update-source Loopback0
 !
 address-family ipv4
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 192.168.40.3 activate
 neighbor 192.168.40.3 route-reflector-client
 neighbor 192.168.40.3 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf transnet
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
!
!
!
ip http server
ip classless
!
line con 0
 password maris
 login
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
```

```
line vty 0 4
 password maris
 login
 transport preferred all
 transport input all
 transport output all
!
!
!
end
```

Configuración: Pop_Mariscal (Ver anexo 3.9.a)

ROUTER CE (CISCO 2610 – Pop Gasca)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router PE (Pop2) de la empresa LBNet:

ROUTER PE (Cisco 2610)			
Nombre	Pop_Gasca		
Enable password	gasca123		
Console/VTY password:	gasca		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 1	192.168.22.2/24	DTE
	Loopback 30	192.168.23.1/24	LAN*
	Loopback 0	192.168.40.3/24	LDP-ID
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSP (1)	192.168.0.0/16	
	BGP (100)	192.168.40.1	remoto

Archivo de configuración

Pop_Gasca

```
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
```

* Mediante la interfaz de loopback 30 se simulará la red en el sitio Gasca de la empresa TransNet.

```
!
hostname Pop_Gasca
!
boot-start-marker
boot-end-marker
!
enable password gasca123
!
no aaa new-model
ip subnet-zero
!
!
ip cef
ip vrf transnet
 rd 100:110
 route-target export 100:1000
 route-target import 100:1000
!
mpls label protocol ldp
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
interface Loopback0
 ip address 192.168.40.3 255.255.255.255
!
interface Loopback30
 ip vrf forwarding transnet
 ip address 192.168.23.1 255.255.255.0
!
interface Ethernet0/0
 no ip address
 shutdown
 half-duplex
!
interface Serial0/0
 no ip address
 shutdown
 no fair-queue
!
interface Serial0/1
 ip address 192.168.22.2 255.255.255.0
 tag-switching ip
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
router bgp 100
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 192.168.40.1 remote-as 100
 neighbor 192.168.40.1 update-source Loopback0
 !
 address-family ipv4
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 192.168.40.1 activate
 neighbor 192.168.40.1 route-reflector-client
 neighbor 192.168.40.1 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf transnet
```

```
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
!
ip http server
ip classless
!
line con 0
 password gasca
 login
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
 password gasca
 login
 transport preferred all
 transport input all
 transport output all
!
end
```

Configuración: Pop_Gasca (Ver anexo 3.9.b)

ROUTER CE (CISCO 3640 – CORE)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router P (Core) de la empresa LBNet:

ROUTER P (Cisco 3640)			
Nombre	Vicentina		
Enable password	vicen123		
Console/VTY password	vicen		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 1	192.168.21.2/24	DTE
	Serial 0	192.168.22.1/24	DCE
	loopback 0	192.168.40.2/32	
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF (1)	192.168.0.0/16	

Archivo de configuración

Vicentina

```
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Vicentina
!
boot-start-marker
boot-end-marker
!
enable password vicen123
!
no aaa new-model
ip subnet-zero
!
ip cef
ip audit po max-events 100
mpls label protocol ldp
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
interface Loopback0
 ip address 192.168.40.2 255.255.255.255
 no clns route-cache
!
interface Ethernet0/0
 no ip address
 shutdown
 half-duplex
 no clns route-cache
!
interface TokenRing0/0
 no ip address
 shutdown
 ring-speed 16
 no clns route-cache
!
interface Serial1/0
 ip address 192.168.22.1 255.255.255.0
 tag-switching ip
 clockrate 56000
 no fair-queue
 no clns route-cache
!
interface Serial1/1
 ip address 192.168.21.2 255.255.255.0
 tag-switching ip
 no clns route-cache
!
interface Serial1/2
 no ip address
 shutdown
 no clns route-cache
!
interface Serial1/3
 no ip address
 shutdown
 no clns route-cache
!
router ospf 1
 log-adjacency-changes
```

```
 network 192.168.0.0 0.0.255.255 area 0
!
ip http server
no ip http secure-server
ip classless
!
line con 0
 password vicen
 login
line aux 0
line vty 0 4
 password vicen
 login
!
end
```

Configuración: Vicentina (Ver anexo 3.9.c)

PRUEBAS

La red obtenida al final de la configuración es la mostrada en la figura 3.18.

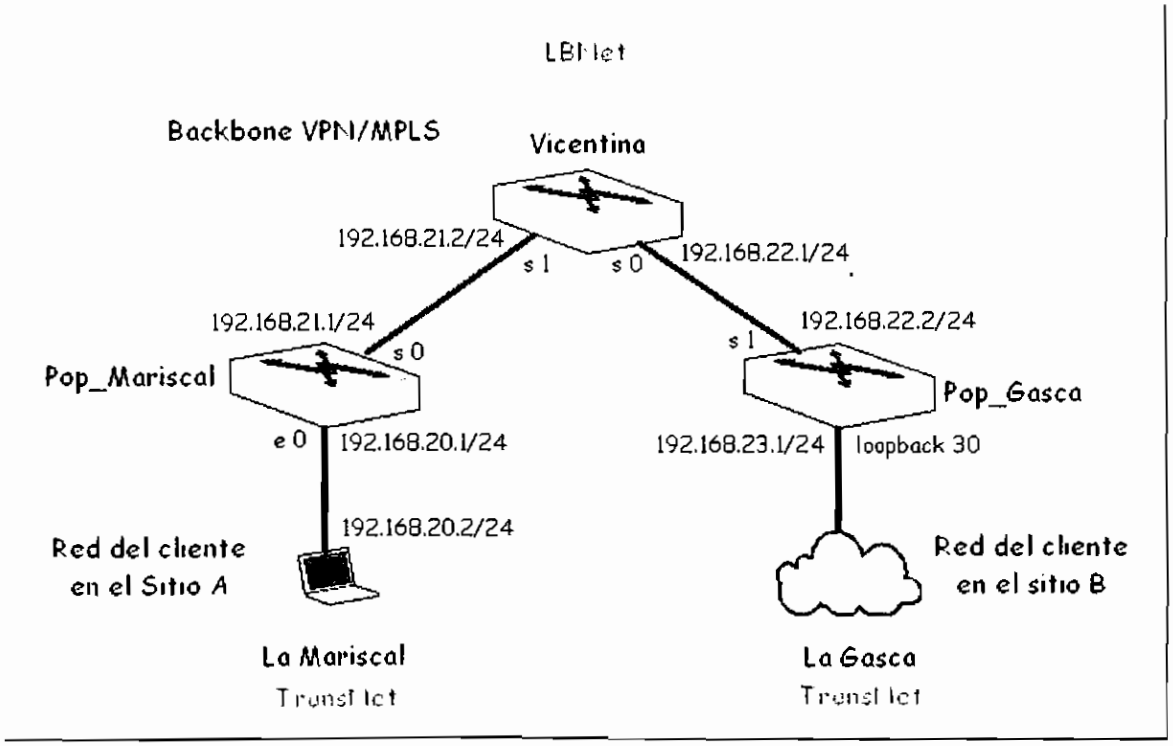


Figura 3.18: Red configurada en el Caso de Estudio 9

Proceso de pruebas:

Los resultados de la verificación se muestran seguidos del comando utilizado. Primero se verificará el protocolo de enrutamiento.

Vicentina#sh 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

     192.168.40.0/32 is subnetted, 2 subnets
O       192.168.40.1 [110/782] via 192.168.21.1, 00:02:27, Serial1/1
C       192.168.40.2 is directly connected, Loopback0
C    192.168.21.0/24 is directly connected, Serial1/1
O    192.168.20.0/24 [110/791] via 192.168.21.1, 00:02:27, Serial1/1
     192.168.23.0/32 is subnetted, 1 subnets
O       192.168.23.1 [110/782] via 192.168.22.2, 00:02:27, Serial1/0
C    192.168.22.0/24 is directly connected, Serial1/0
```

La configuración MPLS/VPN se la comprobará utilizando los comandos *show mpls forwarding-table* y *show ip route vrf.* Nótese la etiqueta que aparece aomo *agrégate* ésta es la utilizada para etiquetar los paquetes que utilizan la VPN.

Vicentina#sh mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	Untagged	192.168.23.1/32	1080	Se1/0	point2point
17	Pop tag	192.168.40.1/32	432	Se1/1	point2point

Pop_Mariscal#sh mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	Pop tag	192.168.22.0/24	0	Se0/0	point2point
17	16	192.168.23.1/32	0	Se0/0	point2point
18	Pop tag	192.168.40.2/32	0	Se0/0	point2point
19	**Aggregate**	**192.168.20.0/24[V]**	**0**	\	

Para mostrar la tabla de enrutamiento IP asociada con una instancia (vrf) VPN se debe utilizar el comando *show ip route vrf* y para mostrar la información de una VPNv4 desde la base de datos BGP se utiliza el comando *show ip bgp vpnv4*:

Pop_Mariscal#sh ip route vrf transnet

```
Routing Table: transnet
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

C    192.168.20.0/24 is directly connected, Ethernet0/0
```

Pop_Mariscal#show ip bgp vpnv4 vrf transnet labels

```
Network          Next Hop      In label/Out label
Route Distinguisher: 100:110 (transnet)
   192.168.20.0     0.0.0.0         19/aggregate(transnet)
```

CASO DE ESTUDIO 10

CONFIGURACIÓN DE UN BACKBONE VPN – MPLS QUE DA SERVICIO A REDES DE DOS CLIENTES

DESCRIPCIÓN GENERAL Y OBJETIVOS

La característica VPN, cuando es usada con MPLS, permite que varios sitios se interconecten transparentemente a través de la red de un proveedor de servicios. Un NSP puede soportar varias VPNs diferentes, cada una de ellas aparece como una red privada, separada de las demás. Cada sitio en una VPN envía paquetes IP a otro sitio en la misma VPN.

El objetivo de este Caso de Estudio es configurar y comprobar el funcionamiento de la red del NSP que da servicio de red privada virtual a dos clientes con dos redes en sitios distantes. El diagrama de red se muestra en la figura 3.19.

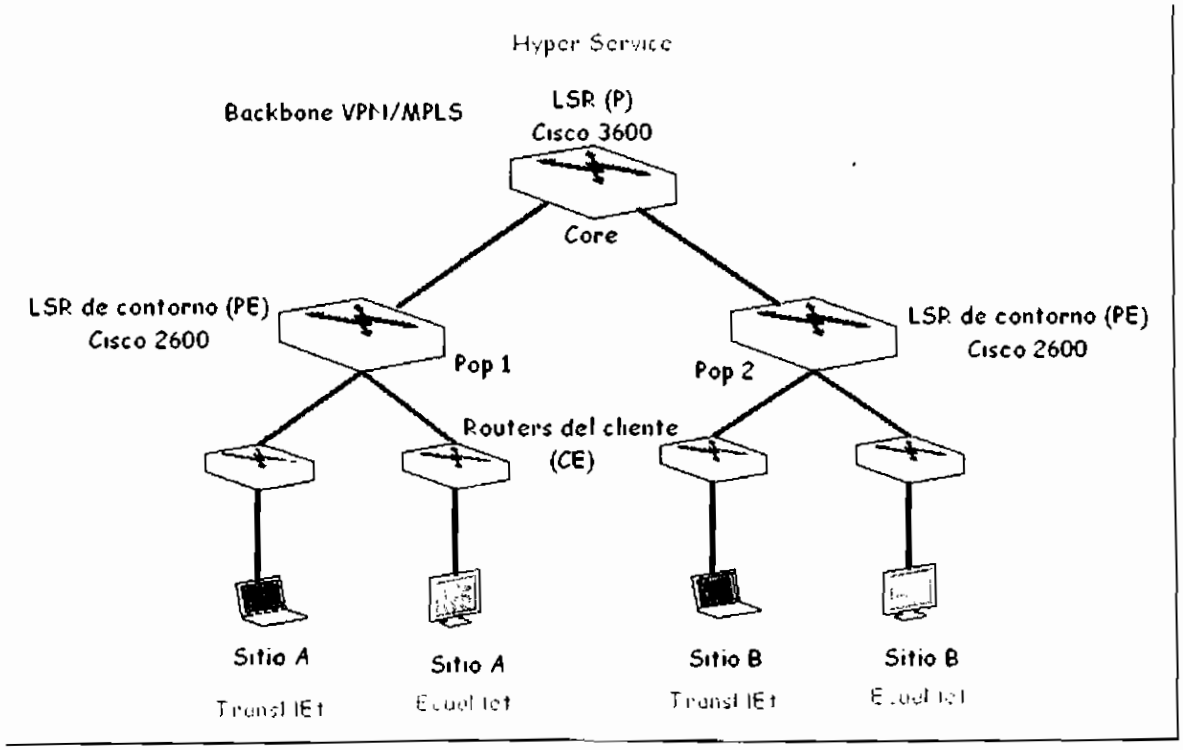


Figura 3.19: Red utilizada en el Caso de Estudio 10

TRABAJO PREPARATORIO

Como trabajo preparatorio para este Caso se debe tener bien claro cómo funciona una VPN y haber entendido a la perfección el numeral 2.2 del Capítulo II. Sería de gran ayuda tener conocimientos básicos de la configuración de los protocolos de enrutamiento OSPF y BGP ya que éstos interactúan en un backbone MPLS/VPN. Además se deben estudiar los comandos utilizados para la verificación del funcionamiento del protocolo BGP y VPN, estos son: *sh ip bgp vpnv4 vrf* y *sh ip route vrf.*

REQUISITOS

Hyperservice es la encargada de dar servicio de red privada virtual a TransNet y EcuaNet, las mismas que utilizarán OSPF como protocolo de enrutamiento y dispone de dos routers Cisco 2500 que se utilizarán como CEs. En este caso las redes de TecRed y EcuaNet serán simuladas con interfaces de *loopback* ya que el resultado en la simulación es exactamente como si tuviéramos routers. Se debe señalar que es recomendable utilizar un router por cada cliente y localidad.

Hyperservice implementará la arquitectura VPN/MPLS en su backbone y dispone de dos routers Cisco 2610 que serán utilizados como PE y un Cisco 3640 como P.

CONFIGURACIÓN

Se configurará dos routers Cisco 2610 como routers LSR de contorno (PE) y un router Cisco 3640 como LSR (P), las redes de los clientes se simularán con las interfaces de *loopback* 30 y 50 para TransNet y 40 y 60 para Ecuanet. Por tratarse de un complemento del Caso de Estudio 9 se utilizará un backbone de características similares.

ROUTER PE (CISCO 2610 – Pop_Mariscal)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router PE el cual actuará como LSR de contorno en el Sitio A de los clientes:

ROUTER PE (Cisco 2610)			
Nombre	Pop_Mariscal		
Enable password	maris123		
Console/VTY password	maris		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 0	192.168.21.1/24	DCE
	Ethernet 0	192.168.12.1/24	LAN*
	Loopback 0	192.168.10.1/32	Router-ID
	Loopback 30	192.168.20.1/24	TransNet
	Loopback 40	192.168.24.1/24	EcuaNet
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF (1)	192.168.0.0/16	
	BGP (100)	192.168.10.3	neighbor

Archivo de configuración

Pop_Mariscal

```
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Pop_Mariscal
!
boot-start-marker
boot-end-marker
!
enable password maris123
!
no aaa new-model
ip subnet-zero
!
!
!
```

* Será utilizada sólo para fines de administración.

```
ip vrf transnet
 rd 1:100
 route-target export 1:1000
 route-target import 1:1000
!
ip cef
ip audit po max-events 100
mpls label protocol ldp
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
!
interface Loopback0
 ip address 192.168.10.1 255.255.255.255
 no clns route-cache
!
interface Ethernet0/0
 no ip address
 shutdown
 half-duplex
 no clns route-cache
!
interface Serial0/0
 ip vrf forwarding transnet
 ip address 192.168.20.2 255.255.255.0
 no clns route-cache
!
interface Ethernet0/1
 no ip address
 shutdown
 half-duplex
 no clns route-cache
!
interface Serial0/1
 ip address 192.168.21.1 255.255.255.0
 tag-switching ip
 no clns route-cache
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
router bgp 1
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 192.168.10.3 remote-as 1
 neighbor 192.168.10.3 update-source Loopback0
 !
 address-family ipv4
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 192.168.10.3 activate
 neighbor 192.168.10.3 route-reflector-client
 neighbor 192.168.10.3 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf transnet
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
!
```

```
ip http server
no ip http secure-server
ip classless
!
!
line con 0
 password maris
 login
line aux 0
line vty 0 4
 password maris
 login
!
!
end
```

Configuración: Pop_Mariscal (Ver anexo 3.10.a)

ROUTER PE (CISCO 2610 – Pop_Gasca)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router PE PE el cual actuará como LSR de contorno en el Sitio B de los clientes:

ROUTER PE (Cisco 2610)			
Nombre	Pop_Gasca		
Enable password	gasca123		
Console/VTY password	gasca		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 1	192.168.22.2/24	DTE
	Loopback 0	192.168.10.3/32	Router-ID
	Loopback 50	192.168.23.1/24	TransNet
	Loopback 60	192.168.24.1/24	EcuaNet
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF (1)	192.168.0.0/16	
	BGP (100)	192.168.10.1	neighbor

Archivo de configuración

Pop_Gasca

```
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
```

```
!
hostname Pop_Gasca
!
boot-start-marker
boot-end-marker
!
enable password gasca123
!
no aaa new-model
ip subnet-zero
!
ip vrf transnet
 rd 1:100
 route-target export 1:1000
 route-target import 1:1000
!
ip cef
ip audit po max-events 100
mpls label protocol ldp
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
interface Loopback0
 ip address 192.168.10.3 255.255.255.255
 no clns route-cache
!
interface Ethernet0/0
 no ip address
 shutdown
 half-duplex
 no clns route-cache
!
interface Serial0/0
 ip address 192.168.22.2 255.255.255.0
 tag-switching ip
 no fair-queue
 no clns route-cache
!
interface Serial0/1
 ip vrf forwarding transnet
 ip address 192.168.23.1 255.255.255.0
 clockrate 56000
 no clns route-cache
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
router bgp 1
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 192.168.10.1 remote-as 1
 neighbor 192.168.10.1 update-source Loopback0
 !
 address-family ipv4
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 192.168.10.1 activate
 neighbor 192.168.10.1 route-reflector-client
 neighbor 192.168.10.1 send-community extended
 exit-address-family
 !
```

```
 address-family ipv4 vrf transnet
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
!
ip http server
no ip http secure-server
ip classless
!
!
line con 0
 password gasca
 login
line aux 0
line vty 0 4
 password gasca
 login
!
end
```

Configuración: Pop_Gasca (Ver anexo 3.10.b)

ROUTER PE (CISCO 3640 – Vicentina)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router P que hace la función de Core de la red de Hypersevice:

ROUTER P (Cisco 3640)			
Nombre	Vicentina		
Enable password	vicen123		
Console/VTY password	vicen		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 0	192.168.22.1/24	DCE
	Serial 1	192.168.21.2/24	DTE
	Loopback 0	192.168.10.2/32	Router-ID
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF (1)	192.168.0.0/16	

Archivo de configuración

Vicentina

```
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
```

```
!
hostname Vicentina
!
boot-start-marker
boot-end-marker
!
enable password vicen123
!
no aaa new-model
ip subnet-zero
!
ip cef
ip audit po max-events 100
mpls label protocol ldp
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
interface Loopback0
 ip address 192.168.10.2 255.255.255.255
 no clns route-cache
!
interface Ethernet0/0
 no ip address
 shutdown
 half-duplex
 no clns route-cache
!
interface TokenRing0/0
 no ip address
 shutdown
 ring-speed 16
 no clns route-cache
!
interface Serial1/0
 ip address 192.168.21.2 255.255.255.0
 tag-switching ip
 clockrate 56000
 no fair-queue
 no clns route-cache
!
interface Serial1/1
 ip address 192.168.22.1 255.255.255.0
 tag-switching ip
 clockrate 56000
 no clns route-cache
!
interface Serial1/2
 no ip address
 shutdown
 no clns route-cache
!
interface Serial1/3
 no ip address
 shutdown
 no clns route-cache
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
ip http server
no ip http secure-server
ip classless
!
!
!
```

```
line con 0
 password vicen
 login
line aux 0
line vty 0 4
 password vicen
 login
!
!
end
```

Configuración: Vicentina (Ver anexo 3.10.c)

PRUEBAS

La red obtenida al final de la configuración es la mostrada en la figura 3.20. El proceso de pruebas se lo realiza a continuación:

Proceso de pruebas:

Debido a que los resultados que se obtendrán del router Pop_Mariscal y Pop_Gasca son muy similares se los enfoca en el router Pop_Gasca para evitar duplicación de información innecesaria. Los resultados de la verificación se muestran seguidos del comando utilizado. Notar en que router es ejecutado el comando.

Primero se obtendrá la información de las interfaces del router:

Pop_Gasca#sh ip interface brief

```
Interface      IP-Address      OK? Method Status                Protocol
Ethernet0/0    unassigned      YES unset  administratively down down
Serial0/0      unassigned      YES unset  administratively down down
Serial0/1      192.168.22.2    YES SLARP  up                    up
Loopback0      192.168.10.3    YES manual up                    up
Loopback50     192.168.23.1    YES manual up                    up
Loopback60     192.168.25.1    YES manual up                    up
```

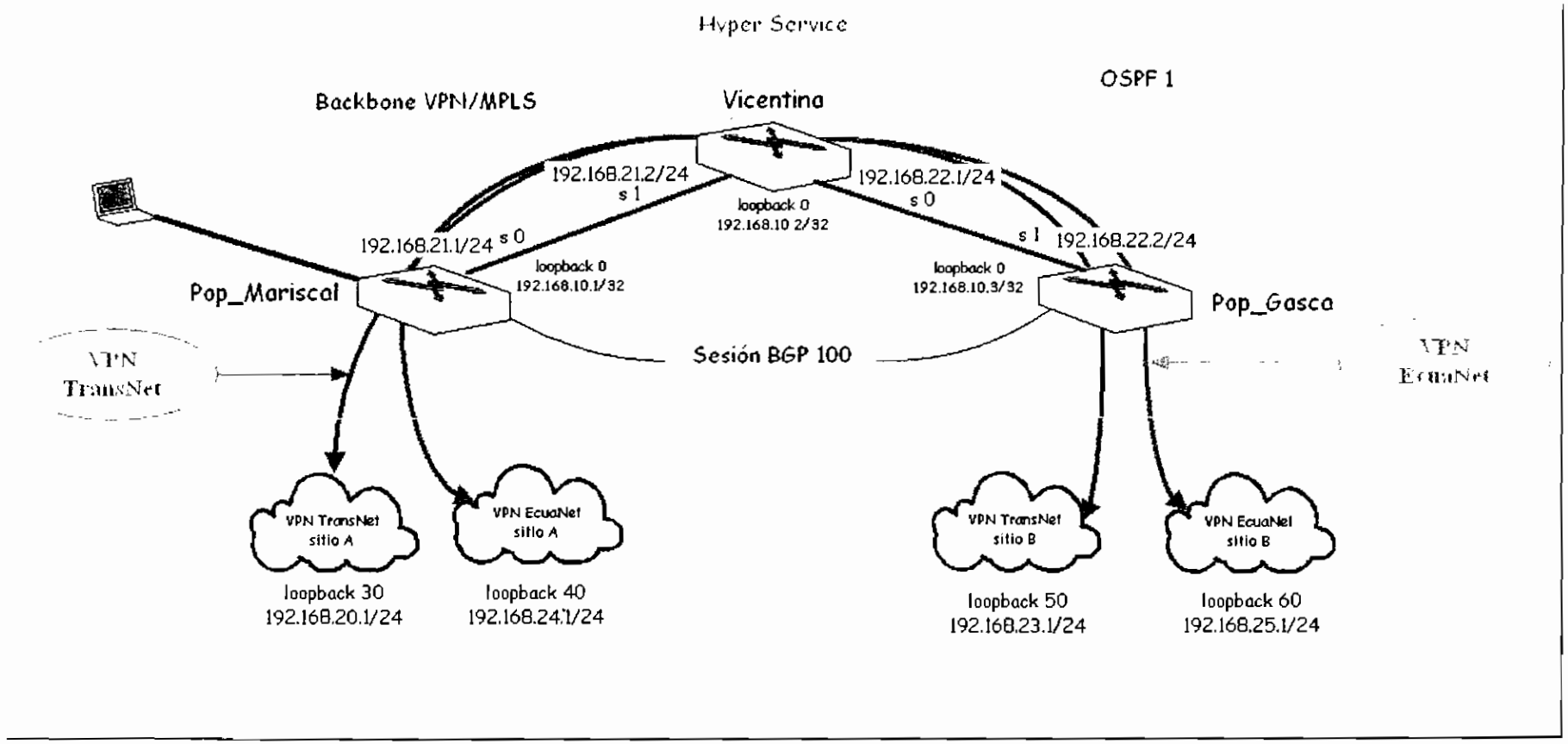


Figura 3.20: Red configurada en el Caso de Estudio 10

Verificar las rutas aprendidas mediante el proceso OSPF:

Pop_Gasca#sh 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

O    192.168.12.0/24 [110/1572] via 192.168.22.1, 00:20:59, Serial0/1
     192.168.10.0/32 is subnetted, 3 subnets
O       192.168.10.2 [110/782] via 192.168.22.1, 00:20:59, Serial0/1
C       192.168.10.3 is directly connected, Loopback0
O       192.168.10.1 [110/1563] via 192.168.22.1, 00:20:59, Serial0/1
O    192.168.21.0/24 [110/1562] via 192.168.22.1, 00:20:59, Serial0/1
C    192.168.22.0/24 is directly connected, Serial0/1
```

Verificar la tabla de enrutamiento para las VPNs de los clientes TransNet y EcuaNet:

Pop_Gasca#sh ip route vrf transnet

```
Routing Table: transnet
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

B    192.168.25.0/24 is directly connected, 00:18:30, Loopback60
B    192.168.24.0/24 [200/0] via 192.168.10.1, 00:19:30
B    192.168.20.0/24 [200/0] via 192.168.10.1, 00:19:30
C    192.168.23.0/24 is directly connected, Loopback50
```

Pop_Gasca#sh ip route vrf ecuanet

```
Routing Table: ecuanet
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

C    192.168.25.0/24 is directly connected, Loopback60
B    192.168.24.0/24 [200/0] via 192.168.10.1, 00:19:36
B    192.168.20.0/24 [200/0] via 192.168.10.1, 00:19:36
B    192.168.23.0/24 is directly connected, 00:18:36, Loopback50
```

Verificar las etiquetas asociadas a las entradas de la tabla de enrutamiento VPN BGP:

```
Pop_Gasca#sh ip bgp vpnv4 vrf transnet labels
```

```
Network          Next Hop      In label/Out label
Route Distinguisher: 100:110 (transnet)
   192.168.20.0     192.168.10.1    nolabel/18
   192.168.23.0     0.0.0.0         20/aggregate(transnet)
   192.168.24.0     192.168.10.1    nolabel/19
   192.168.25.0     0.0.0.0         nolabel/aggregate(ecuanet)
```

```
Pop_Gasca#sh ip bgp vpnv4 vrf ecuanet labels
```

```
   Network          Next Hop      In label/Out label
Route Distinguisher: 100:120 (ecuanet)
   192.168.20.0     192.168.10.1    nolabel/18
   192.168.23.0     0.0.0.0         nolabel/aggregate(transnet)
   192.168.24.0     192.168.10.1    nolabel/19
   192.168.25.0     0.0.0.0         21/aggregate(ecuanet)
```

Verificar la tabla de envío o LFIB. Se debe tener en cuenta que la columna *Local tag* significa etiqueta de entrada pero solo para mensajes con etiqueta que llegan al nodo, caso contrario no considerar su valor:

```
Pop_Gasca#sh mpls forwarding-table detail
```

```
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface

16     16          192.168.10.1/32  0          Se0/1       point2point
       MAC/Encaps=4/8, MRU=1500, Tag Stack{16}
       0F008847 00010000
       No output feature configured
    Per-packet load-sharing

17     Pop tag     192.168.10.2/32  0          Se0/1       point2point
       MAC/Encaps=4/4, MRU=1504, Tag Stack{}
       0F008847
```

```
        No output feature configured
    Per-packet load-sharing

18      17          192.168.12.0/24   0          Se0/1       point2point
        MAC/Encaps=4/8, MRU=1500, Tag Stack{17}
        0F008847 00011000
        No output feature configured
    Per-packet load-sharing

19      Pop tag     192.168.21.0/24   0          Se0/1       point2point
        MAC/Encaps=4/4, MRU=1504, Tag Stack{}
        0F008847
        No output feature configured
    Per-packet load-sharing

20      Aggregate   192.168.23.0/24[V]    \
                                      648
        MAC/Encaps=0/0, MRU=0, Tag Stack{}
        VPN route: transnet
        No output feature configured
    Per-packet load-sharing

21      Aggregate   192.168.25.0/24[V]    \
                                      0
        MAC/Encaps=0/0, MRU=0, Tag Stack{}
        VPN route: ecuanet
        No output feature configured
    Per-packet load-sharing
```

Para comprobar conectividad y el número de etiqueta que se está utilizando se debe ejecutar el comando *traceroute* de la siguiente manera:

```
Pop_Gasca#traceroute 192.168.10.1

Type escape sequence to abort.
Tracing the route to 192.168.10.1

  1 192.168.22.1 [MPLS: Label 16 Exp 0] 44 msec 48 msec 40 msec
  2 192.168.21.1 44 msec 48 msec *
```

```
Pop_Gasca#traceroute vrf transnet 192.168.20.1

Type escape sequence to abort.
Tracing the route to 192.168.20.1

  1 192.168.22.1 [MPLS: Labels 16/18 Exp 0] 100 msec 112 msec 100 msec
  2 192.168.20.1 48 msec 52 msec *
```

```
Pop_Gasca#traceroute vrf ecuanet 192.168.24.1

Type escape sequence to abort.
Tracing the route to 192.168.24.1

  1 192.168.22.1 [MPLS: Labels 16/19 Exp 0] 101 msec 113 msec 100 msec
  2 192.168.24.1 48 msec 52 msec *
```

En los routers Pop_Mariscal y Vicentina basta con verificar la LFIB:

Pop_Marisacal#sh mpls forwarding-table detail

```
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
16     Pop tag     192.168.10.2/32  0          Se0/0       point2point
       MAC/Encaps=4/4, MRU=1504, Tag Stack{}
       0F008847
       No output feature configured
   Per-packet load-sharing
17     Pop tag     192.168.22.0/24  0          Se0/0       point2point
       MAC/Encaps=4/4, MRU=1504, Tag Stack{}
       0F008847
       No output feature configured
   Per-packet load-sharing
18     Aggregate   192.168.20.0/24[V]    \
                                    1056
       MAC/Encaps=0/0, MRU=0, Tag Stack{}
       VPN route: transnet
       No output feature configured
   Per-packet load-sharing
19     Aggregate   192.168.24.0/24[V]    \
                                    176
       MAC/Encaps=0/0, MRU=0, Tag Stack{}
       VPN route: ecuanet
       No output feature configured
   Per-packet load-sharing
20     18          192.168.10.3/32  0          Se0/0       point2point
       MAC/Encaps=4/8, MRU=1500, Tag Stack{18}
       0F008847 00012000
       No output feature configured
   Per-packet load-sharing
```

Vicentina#sh mpls forwarding-table detail

```
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
16     Pop tag     192.168.10.1/32  993        Se1/1       point2point
       MAC/Encaps=4/4, MRU=1504, Tag Stack{}
       0F008847
       No output feature configured
   Per-packet load-sharing
17     Pop tag     192.168.12.0/24  29488      Se1/1       point2point
       MAC/Encaps=4/4, MRU=1504, Tag Stack{}
       0F008847
       No output feature configured
   Per-packet load-sharing
18     Pop tag     192.168.10.3/32  2753       Se1/0       point2point
       MAC/Encaps=4/4, MRU=1504, Tag Stack{}
       0F008847
       No output feature configured
   Per-packet load-sharing
```

CASO DE ESTUDIO 11

CONFIGURACIÓN QoS EN UN BACKBONE MPLS

DESCRIPCIÓN GENERAL Y OBJETIVOS

En este caso de estudio se realizará la configuración de la característica QoS que se puede implementar junto con la arquitectura MPLS. Para esto se utilizará un backbone MPLS similar al configurado en el Caso de Estudio 7. La red a configurarse se muestra en la figura 3.21. Hay algunas formas de implementar la característica QoS en un backbone MPLS, en este proyecto implementaremos QoS utilizando *Commited Access Rate* (CAR) ya que es la que soportan los routers Cisco 2610 utilizados.

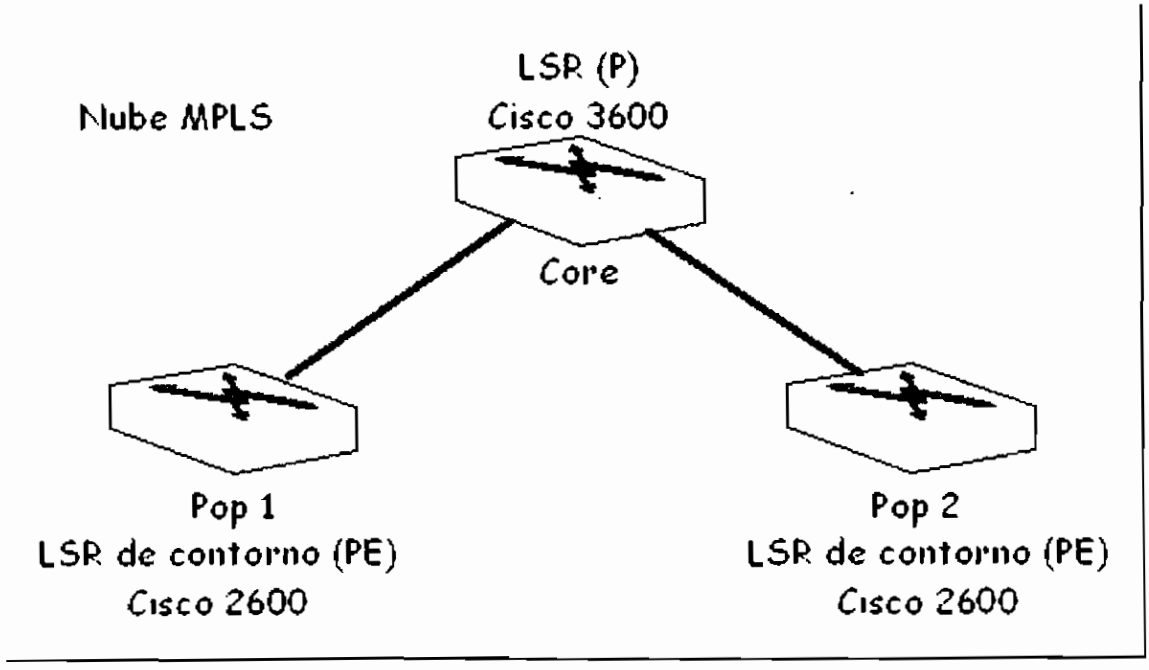


Figura 3.21: Red utilizada en el Caso de Estudio 11

TRABAJO PREPARATORIO

Como trabajo preparatorio para este Caso se debe tener bien claro cómo funciona una red MPLS y también el significado de Calidad de Servicio (QoS). Se debe haber realizado y comprendido el Caso de Estudio 7. Se deben estudiar los comandos utilizados para configurar la característica QoS en un backbone MPLS estos son: *access-list rate-limit y rate-limit.*

REQUISITOS

La empresa brindará servicios de red utilizando la arquitectura MPLS, y también ofrecer Calidad de Servicio por lo tanto, el requerimiento de la red será brindar conectividad desde el Pop 1 hasta el Pop 2 etiquetando los paquetes que atraviesen dicha red tomando en cuenta su precedencia IP. Para las pruebas se utilizará PCs conectados a cada extremo de la red, es decir, uno en la interfaz ethernet (o interfaces de *loopback*) del Pop1 y otro en el Pop2 los cuales representarán las redes del cliente.

CONFIGURACIÓN

La red de la empresa GybGonzanama como se muestra en la figura 3.21 está formada por dos routers Cisco 2600 como LSRs de contorno y un router Cisco 3600 como LSR. La característica QoS se implementa configurando las interfaces de entrada en los LSR de contorno de acuerdo a lo requerido y no es necesaria una configuración adicional en el router LSR (P) ya que estos se encargan únicamente de la conmutación de etiquetas y toman decisiones de QoS revisando los bits EXP de la etiqueta MPLS.

ROUTER PE (CISCO 2600)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router Pop_1 (PE):

ROUTER PE (Cisco 2610)			
Nombre	Pop_1		
Enable password	pop1123		
Console/VTY password	pop1		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 0	192.168.20.1/24	DCE
	ethernet 0	192.168.19.1/24	LAN
	loopback 0	192.168.10.1/32	Router-ID
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF (1)	192.168.0.0/16	

Archivo de configuración

Pop_1

```
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Pop_1
!
boot-start-marker
boot-end-marker
!
enable password pop1123
!
memory-size iomem 10
no aaa new-model
ip subnet-zero
!
ip cef
mpls label protocol ldp
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
interface Loopback0
 ip address 192.168.10.1 255.255.255.255
!
interface Ethernet0/0
 ip address 192.168.19.1 255.255.255.0
 rate-limit input access-group rate-limit 30 8000 8000 8000 conform-action set-
mpls-exp-imposition-transmit 1 exceed-action set-mpls-exp-imposition-transmit 0
 half-duplex
!
interface Serial0/0
 ip address 192.168.20.1 255.255.255.0
 tag-switching ip
 clockrate 56000
 no fair-queue
!
interface Serial0/1
```

```
 no ip address
 shutdown
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
ip http server
ip classless
!
access-list rate-limit 30 0
!
line con 0
 password pop1
 login
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
 password pop1
 login
 transport preferred all
 transport input all
 transport output all
!
end
```

Configuración: Pop_1 (Ver anexo 3.11.a)

ROUTER PE (CISCO 2610)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router Pop_2 (PE):

ROUTER PE (Cisco 2610)			
Nombre	Pop_2		
Enable password	pop2123		
Console/VTY password	pop2		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Loopback 20	192.168.22.1/24	LAN
	Serial 1	192.168.21.2/24	DTE
	Loopback 0	192.168.10.3/32	Router-ID
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF (1)	192.168.0.0/16	

Archivo de configuración

Pop_2

```
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Pop_2
!
boot-start-marker
boot-end-marker
!
enable password pop2123
!
no aaa new-model
ip subnet-zero
!
!
ip cef
mpls label protocol ldp
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
!
!
!
interface Loopback0
 ip address 192.168.10.3 255.255.255.255
!
interface Loopback20
 ip address 192.168.22.1 255.255.255.0
 rate-limit input access-group rate-limit 20 8000 8000 8000 conform-action set-
mpls-exp-imposition-transmit 5 exceed-action set-mpls-exp-imposition-transmit 0
!
interface Ethernet0/0
 no ip address
 rate-limit input access-group rate-limit 20 8000 8000 8000 conform-action set-
mpls-exp-imposition-transmit 5 exceed-action set-mpls-exp-imposition-transmit 0
 shutdown
 half-duplex
!
interface Serial0/0
 no ip address
 shutdown
 no fair-queue
!
interface Serial0/1
 ip address 192.168.21.2 255.255.255.0
 tag-switching ip
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
ip http server
ip classless
!
!
access-list rate-limit 20 5
!
!
```

```
line con 0
 password pop2
 login
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
 password pop2
 login
 transport preferred all
 transport input all
 transport output all
!
end
```

Configuración: Pop_2 (Ver anexo 3.11.b)

ROUTER P (CISCO 3640)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router Gonzanama (P):

ROUTER P (Cisco 3640)			
Nombre	Gonzanama		
Enable password	gonza123		
Console/VTY password	gonza		
jlaleyInterfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 0	192.168.20.2/24	DTE
	Serial 1	192.168.21.1/24	DCE
	Loopback 0	192.168.10.2/32	Router-ID
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF (1)	192.168.0.0/16	

Archivo de configuración

Gonzanama

```
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
```

```
!
hostname Gonzanama
!
boot-start-marker
boot-end-marker
!
enable password gonza123
!
no aaa new-model
ip subnet-zero
!
!
!
ip cef
ip audit po max-events 100
mpls label protocol ldp
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
!
!
!
!
interface Loopback0
 ip address 192.168.10.2 255.255.255.255
 no clns route-cache
!
interface Ethernet0/0
 no ip address
 shutdown
 half-duplex
 no clns route-cache
!
interface TokenRing0/0
 no ip address
 shutdown
 ring-speed 16
 no clns route-cache
!
interface Serial1/0
 ip address 192.168.20.2 255.255.255.0
 tag-switching ip
 no fair-queue
 no clns route-cache
!
interface Serial1/1
 ip address 192.168.21.1 255.255.255.0
 tag-switching ip
 clockrate 56000
 no clns route-cache
!
interface Serial1/2
 no ip address
 shutdown
 no clns route-cache
!
interface Serial1/3
 no ip address
 shutdown
 no clns route-cache
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
ip http server
```

```
no ip http secure-server
ip classless
!
line con 0
 password gonza
 login
line aux 0
line vty 0 4
 password gonza
 login
!
end
```

Configuración: Gonzanama (Ver anexo 3.11.c)

PRUEBAS

La red obtenida al final de la configuración es la mostrada en la figura 3.22. Las pruebas se las debe realizar como se muestra a continuación:

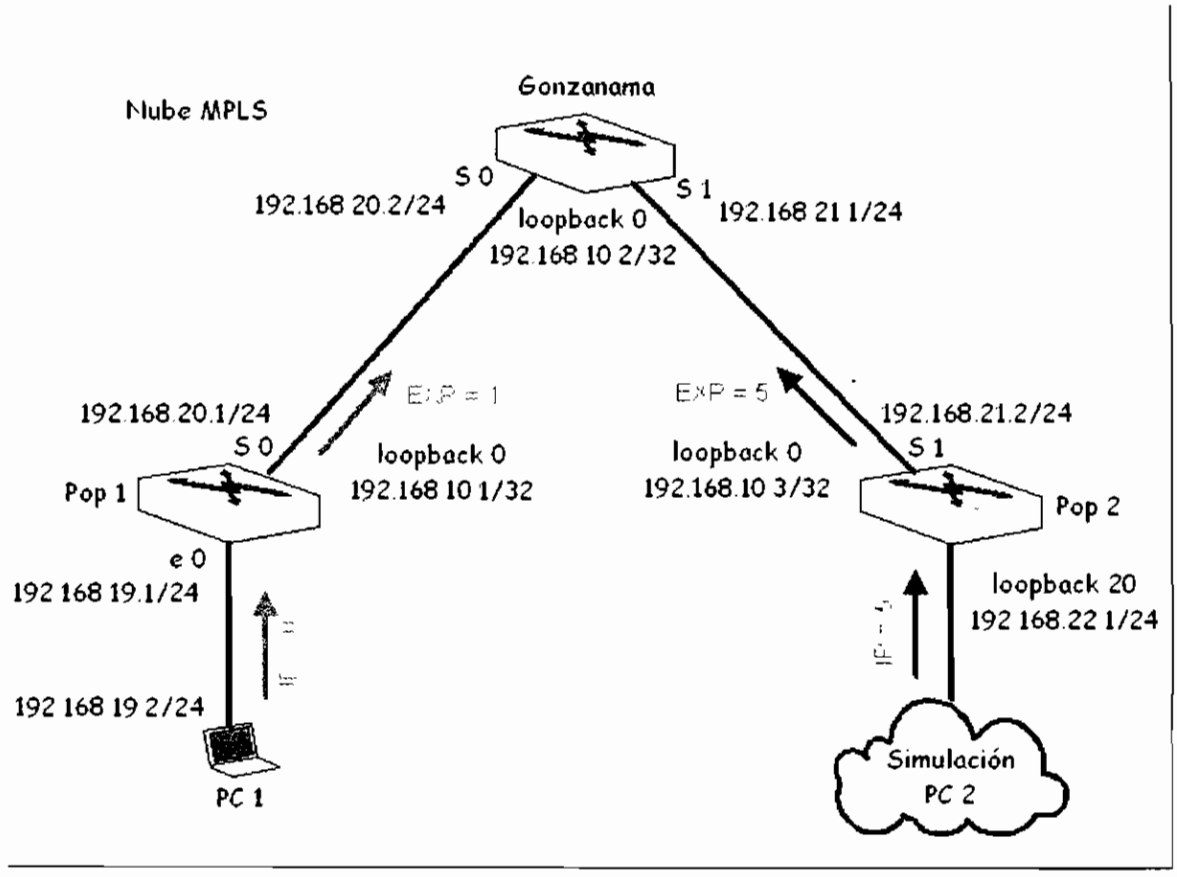


Figura 3.22: Red configurada en el Caso de Estudio 11

Proceso de pruebas:

Los resultados de la verificación se muestran seguidos del comando utilizado. Notar en que router es ejecutado el comando.

Primero se verificará que las rutas se hayan aprendido correctamente mediante el proceso OSPF:

Gonzanama#sh 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

     192.168.10.0/32 is subnetted, 3 subnets
C       192.168.10.2 is directly connected, Loopback0
O       192.168.10.3 [110/782] via 192.168.21.2, 00:01:06, Serial1/1
O       192.168.10.1 [110/782] via 192.168.20.1, 00:01:06, Serial1/0
C    192.168.21.0/24 is directly connected, Serial1/1
C    192.168.20.0/24 is directly connected, Serial1/0
     192.168.22.0/32 is subnetted, 1 subnets
O       192.168.22.1 [110/782] via 192.168.21.2, 00:01:06, Serial1/1
O    192.168.19.0/24 [110/791] via 192.168.20.1, 00:01:08, Serial1/0
```

Verificar las tablas LFIB para comprobar que el proceso MPLS se esté llevando a cabo:

Pop_1#sh mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	Pop tag	192.168.10.2/32	0	Se0/0	point2point
17	Pop tag	192.168.21.0/24	0	Se0/0	point2point
18	18	192.168.10.3/32	0	Se0/0	point2point
19	19	192.168.22.1/32	0	Se0/0	point2point

Pop_2#sh mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	Untagged	192.168.10.1/32	0	Se0/1	point2point
17	Untagged	192.168.10.2/32	0	Se0/1	point2point
18	Untagged	192.168.19.0/24	0	Se0/1	point2point
19	Untagged	192.168.20.0/24	0	Se0/1	point2point

Gonzanama#sh mpls forwarding-table detail

```
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
16     Pop tag     192.168.10.1/32  0          Se1/0       point2point
       MAC/Encaps=4/4, MRU=1504, Tag Stack{}
       0F008847
       No output feature configured
    Per-packet load-sharing
17     Pop tag     192.168.19.0/24  0          Se1/0       point2point
       MAC/Encaps=4/4, MRU=1504, Tag Stack{}
       0F008847
       No output feature configured
    Per-packet load-sharing
18     Untagged    192.168.10.3/32  0          Se1/1       point2point
       MAC/Encaps=0/0, MRU=1504, Tag Stack{}
       No output feature configured
    Per-packet load-sharing
19     Untagged    192.168.22.1/32  256        Se1/1       point2point
       MAC/Encaps=0/0, MRU=1504, Tag Stack{}
       No output feature configured
    Per-packet load-sharing
```

Para verificar la configuración de QoS se utiliza dos comandos: *sh access-lists rate-limit* muestra el número de la lista o listas de acceso tipo rate-limit y el valor de precedencia IP de los paquetes que serán filtrados y *sh inteface rate-limit* muestra la cantidad de paquetes que han sido filtrados por la lista de acceso y el valor del campo EXP de la cabecera MPLS que se le ha asignado. Los resultados al ejecutar estos dos comandos en los routers Pop_1 y Pop_2 se muestran a continuación:

Pop_1#sh access-lists rate-limit

```
Rate-limit access list 30
    3
```

Pop_1#sh interfaces rate-limit

```
Ethernet0/0
  Input
    matches: access-group rate-limit 30
      params:  8000 bps, 8000 limit, 8000 extended limit
      conformed 0 packets, 0 bytes; action: set-mpls-exp-imposition-
transmit 1
      exceeded 0 packets, 0 bytes; action: set-mpls-exp-imposition-
transmit 0
      last packet: 901654ms ago, current burst: 0 bytes
      last cleared 00:04:02 ago, conformed 0 bps, exceeded 0 bps
```

Pop_2#show access-lists rate-limit

```
Rate-limit access list 20
    5
```

Pop_2#show interfaces rate-limit

```
Loopback20
  Input
    matches: access-group rate-limit 20
      params:  8000 bps, 8000 limit, 8000 extended limit
      conformed 0 packets, 0 bytes; action: set-mpls-exp-imposition-
transmit 5
      exceeded 0 packets, 0 bytes; action: set-mpls-exp-imposition-
transmit 0
      last packet: 3240263ms ago, current burst: 0 bytes
      last cleared 00:24:03 ago, conformed 0 bps, exceeded 0 bps
```

Los paquetes que ingresen por la interfaz Ethernet 0 del router *Pop_1* con precedencia IP = 3 su cabecera MPLS tendrá el Campo EXP = 1 y en el router *Pop_2* sucederá algo similar IP = 5 MPLS = 5.

Se puede realizar una prueba adicional cambiando la lista de acceso para que filtre paquetes con precedencia IP = 0 como por ejemplo los paquetes que utiliza el comando *ping* y hacer que éstos viajen por el backbone MPLS con el valor del campo EXP = 1, para esto se realizará lo siguiente:

Pop_1(config)#access-list rate-limit 30 0

Pop_1#sh access-lists rate-limit

```
Rate-limit access list 30
    0
```

Pop_1#sh interfaces rate-limit

```
Ethernet0/0
  Input
    matches: access-group rate-limit 30
      params:  8000 bps, 8000 limit, 8000 extended limit
      conformed 236 packets, 16162 bytes; action: set-mpls-exp-
imposition-transm
it 1
      exceeded 0 packets, 0 bytes; action: set-mpls-exp-imposition-
transmit 0
      last packet: 12ms ago, current burst: 0 bytes
      last cleared 00:26:47 ago, conformed 0 bps, exceeded 0 bps
```

Nótese que en el campo *conformed* se tienen 236 paquetes procesados cuya cabecera MPLS tienen el valor del campo EXP igual a 1.

CASO DE ESTUDIO 12

CONFIGURACIÓN QoS EN UN BACKBONE VPN/MPLS

DESCRIPCIÓN GENERAL Y OBJETIVOS

En este caso de estudio se realizará la configuración de la característica QoS que se puede implementar junto con las arquitecturas MPLS y VPN. Para esto se utilizará un backbone VPN/MPLS similar al configurado en el Caso de Estudio 9 para completarlo configurando la característica QoS.

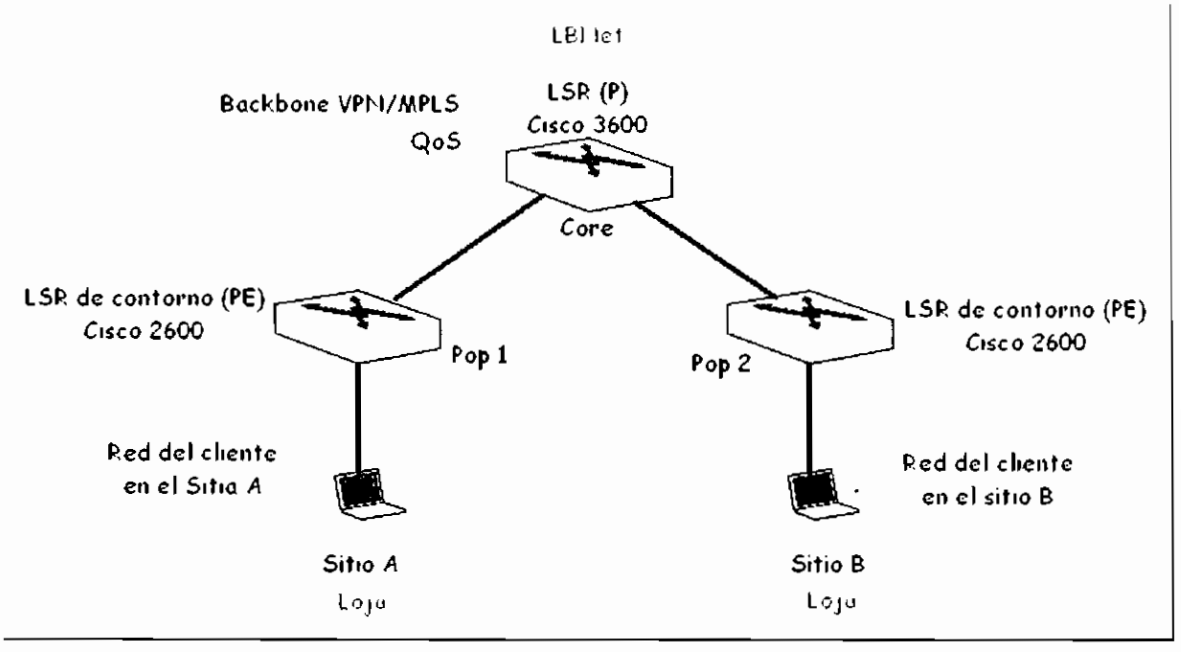


Figura 3.23: Red utilizada en el Caso de Estudio 12

Una de las topologías que se puede suministrar utilizando la arquitectura VPN/MPLS es una intranet entre los diferentes sitios que pertenecen a una misma empresa. En este Caso de Estudio configuraremos la red de LBNet que se encarga de dar servicio VPN/MPLS y QoS de red a la empresa LojaNet. La red a configurarse se muestra en la figura 3.23.

TRABAJO PREPARATORIO

Como trabajo preparatorio para este Caso se debe tener bien claro cómo funciona una red VPN/MPLS y también el significado de Calidad de Servicio (QoS). Se debe haber realizado y comprendido el Caso de Estudio 9. Se deben estudiar los comandos utilizados para configurar la característica QoS en un backbone VPN/MPLS estos son: *access-list rate-limit y rate-limit.*

REQUISITOS

La empresa LojaNet informa que utilizará OSPF como protocolo de enrutamiento en sus dos redes y solicita a su NSP que le brinde un servicio de VPN y QoS. La empresa LBNet debe implementar la arquitectura MPLS en su backbone y satisfacer las necesidades de su cliente, para lo cual ha decidido implementar una arquitectura VPN/MPLS. LojaNet dispone de dos routers Cisco 2600 que serán configurados como routers PE y de un Cisco 3640 que hará la función de router de Core. Los sitios de red serán simulados con una PC (o interfaces de *loopback)* en cada lado pero es necesario aclarar que es posible que el cliente disponga de un router en cada sitio para que sean administrados por el cliente, en cualquiera de los dos casos la configuración del backbone VPN/MPLS es la misma.

CONFIGURACIÓN

A continuación se realizará la configuración de los routers de la empresa LBNet que está formada por un router Cisco 3640 que funcionará como LSR y dos routers Cisco 2610 que trabajarán como LSRs de contorno.

ROUTER CE (CISCO 2610 – Pop Mariscal)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router PE (Pop1) de la empresa LBNet:

ROUTER PE (Cisco 2610)			
Nombre	Pop_Mariscal		
Enable password	maris123		
Console/VTY password	Maris		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 0	192.168.20.1/24	DCE
	Ethernet 0	192.168.15.1/24	LAN
	Loopback 0	192.168.10.1/32	Router-ID
	Loopback 20	192.168.19.1/24	VPN Loja
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF (1)	192.168.0.0/16	
	BGP (10)	192.168.10.3	neighbor

Archivo de configuración

Pop_Mariscal

```
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Pop_Mariscal
!
boot-start-marker
boot-end-marker
!
enable password maris123
!
memory-size iomem 10
no aaa new-model
ip subnet-zero
!
!
ip cef
ip vrf loja
 rd 10:100
 route-target export 10:1000
 route-target import 10:1000
!
mpls label protocol ldp
mpls ldp explicit-null
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
!
!
!
```

```
interface Loopback0
 ip address 192.168.10.1 255.255.255.255
!
interface Loopback20
 ip vrf forwarding loja
 ip address 192.168.19.1 255.255.255.0
 rate-limit input access-group rate-limit 20 8000 8000 8000 conform-action set-
mpls-exp-imposition-transmit 3 exceed-action set-mpls-exp-imposition-transmit 3
!
interface Ethernet0/0
 ip address 192.168.15.1 255.255.255.0
 half-duplex
!
interface Serial0/0
 ip address 192.168.20.1 255.255.255.0
 tag-switching ip
 clockrate 56000
 no fair-queue
!
interface Serial0/1
 no ip address
 shutdown
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
router bgp 10
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 192.168.10.3 remote-as 10
 neighbor 192.168.10.3 update-source Loopback0
 !
 address-family ipv4
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 192.168.10.3 activate
 neighbor 192.168.10.3 route-reflector-client
 neighbor 192.168.10.3 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf loja
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
!
ip http server
ip classless
!
!
access-list rate-limit 20 3
!
!
line con 0
 password maris
 login
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
```

```
line vty 0 4
 password maris
 login
 transport preferred all
 transport input all
 transport output all
!
end
```

Configuración: Pop_Mariscal (Ver anexo 3.12.a)

ROUTER CE (CISCO 2610 – Pop Gasca)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router PE (Pop2) de la empresa LBNet:

ROUTER PE (Cisco 2610)			
Nombre	Pop_Gasca		
Enable password	gasca123		
Console/VTY password:	gasca		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 1	192.168.21.2/24	DTE
	Loopback 0	192.168.10.3/32	Router-ID
	Loopback 20	192.168.22.1/24	VPN Loja
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF (1)	192.168.0.0/16	
	BGP (10)	192.168.10.1	neighbor

Archivo de configuración

Pop_Gasca

```
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Pop_Gasca
!
boot-start-marker
boot-end-marker
```

```
!
enable password gasca123
!
no aaa new-model
ip subnet-zero
!
!
!
ip cef
ip vrf loja
 rd 10:100
 route-target export 10:1000
 route-target import 10:1000
!
!
mpls label protocol ldp
mpls ldp explicit-null
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
interface Loopback0
 ip address 192.168.10.3 255.255.255.255
!
interface Loopback20
 ip vrf forwarding loja
 ip address 192.168.22.1 255.255.255.0
 rate-limit input access-group rate-limit 20 8000 8000 8000 conform-action set-
mpls-exp-imposition-transmit 3 exceed-action set-mpls-exp-imposition-transmit 3
!
interface Ethernet0/0
 no ip address
 shutdown
 half-duplex
!
interface Serial0/0
 no ip address
 shutdown
 no fair-queue
!
interface Serial0/1
 ip address 192.168.21.2 255.255.255.0
 tag-switching ip
!
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
!
router bgp 10
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 192.168.10.1 remote-as 10
 neighbor 192.168.10.1 update-source Loopback0
 !
 address-family ipv4
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 192.168.10.1 activate
 neighbor 192.168.10.1 route-reflector-client
 neighbor 192.168.10.1 send-community extended
 exit-address-family
```

```
 !
 address-family ipv4 vrf loja
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
!
ip http server
ip classless
!
access-list rate-limit 40 3
!
line con 0
 password gasca
 login
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
 password gasca
 login
 transport preferred all
 transport input all
 transport output all
!
end
```

Configuración: Pop_Gasca (Ver anexo 3.12.b)

ROUTER CE (CISCO 3640 – CORE)

Esquema de configuración:

A continuación se muestra el esquema de configuración del router P (Core) de la empresa LBNet:

ROUTER P (Cisco 3640)			
Nombre	Vicentina		
Enable password	vicen123		
Console/VTY password	vicen		
Interfaces	*Número*	*IP / Máscara*	*Tipo*
	Serial 0	192.168.20.2/24	DTE
	Serial 1	192.168.21.1/24	DCE
	Loopback 0	192.168.10.2/32	Router-ID
Protocolo de enrutamiento	*Protocolo*	*Redes*	
	OSPF (1)	192.168.0.0/16	

Archivo de configuración

Vicentina

```
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Vicentina
!
boot-start-marker
boot-end-marker
!
enable password vicen123
!
no aaa new-model
ip subnet-zero
!
ip cef
ip audit po max-events 100
mpls label protocol ldp
mpls ldp explicit-null
tag-switching tdp router-id Loopback0
no ftp-server write-enable
!
interface Loopback0
 ip address 192.168.10.2 255.255.255.255
 no clns route-cache
!
interface Ethernet0/0
 no ip address
 shutdown
 half-duplex
 no clns route-cache
!
interface TokenRing0/0
 no ip address
 shutdown
 ring-speed 16
 no clns route-cache
!
interface Serial1/0
 ip address 192.168.20.2 255.255.255.0
 tag-switching ip
 no fair-queue
 no clns route-cache
!
interface Serial1/1
 ip address 192.168.21.1 255.255.255.0
 tag-switching ip
 clockrate 56000
 no clns route-cache
!
interface Serial1/2
 no ip address
 shutdown
 no clns route-cache
!
interface Serial1/3
 no ip address
 shutdown
 no clns route-cache
!
router ospf 1
```

```
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
ip http server
no ip http secure-server
ip classless
!
line con 0
 password vicen
 login
line aux 0
line vty 0 4
 password vicen
 login
!
!
end
```

Configuración: Vicentina (Ver anexo 3.12.c)

PRUEBAS

La red obtenida al final de la configuración es la mostrada en la figura 3.24. Las pruebas se las realiza a continuación:

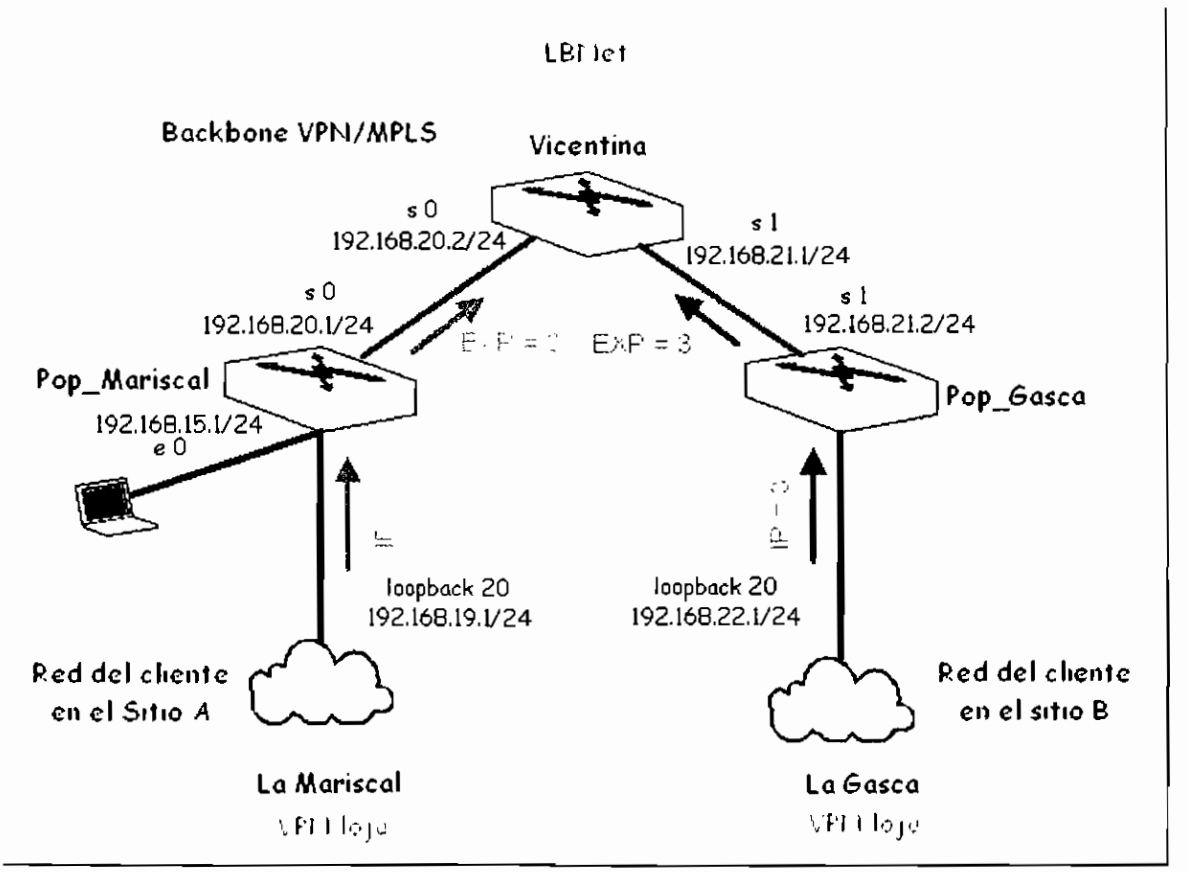


Figura 3.24: Red configurada en el Caso de Estudio 12

Proceso de pruebas:

Los resultados de la verificación se muestran seguidos del comando utilizado. Notar en que router se ejecuta el comando.

Primero se debe verificar que los protocolos de enrutamiento estén trabajando correctamente utilizando los siguientes comandos:

Vicentina#sh 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

     192.168.10.0/32 is subnetted, 3 subnets
C       192.168.10.2 is directly connected, Loopback0
O       192.168.10.3 [110/782] via 192.168.21.2, 00:01:34, Serial1/1
O       192.168.10.1 [110/782] via 192.168.20.1, 00:01:34, Serial1/0
C    192.168.21.0/24 is directly connected, Serial1/1
C    192.168.20.0/24 is directly connected, Serial1/0
     192.168.22.0/32 is subnetted, 1 subnets
O       192.168.22.1 [110/782] via 192.168.21.2, 00:01:34, Serial1/1
     192.168.19.0/32 is subnetted, 1 subnets
O       192.168.19.1 [110/782] via 192.168.20.1, 00:01:35, Serial1/0
```

Pop_Mariscal#sh 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

C    192.168.15.0/24 is directly connected, Ethernet0/0
     192.168.10.0/32 is subnetted, 3 subnets
O       192.168.10.2 [110/782] via 192.168.20.2, 00:03:08, Serial0/0
O       192.168.10.3 [110/1563] via 192.168.20.2, 00:03:08, Serial0/0
C       192.168.10.1 is directly connected, Loopback0
O    192.168.21.0/24 [110/1562] via 192.168.20.2, 00:03:08, Serial0/0
C    192.168.20.0/24 is directly connected, Serial0/0
```

Verificar que el protocolo MPLS esté trabajando correctamente y que ha asignado etiquetas a todas las rutas, esto se lo realiza ejecutando los siguientes comandos:

Pop_Mariscal#sh mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	0	192.168.10.2/32	0	Se0/0	point2point
17	16	192.168.10.3/32	0	Se0/0	point2point
18	0	192.168.21.0/24	0	Se0/0	point2point
19	Aggregate	192.168.19.0/24[V]	\ 0		

Pop_Gasca#sh mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	17	192.168.10.1/32	0	Se0/1	point2point
17	0	192.168.10.2/32	0	Se0/1	point2point
18	0	192.168.20.0/24	0	Se0/1	point2point
19	Aggregate	192.168.22.0/24[V]	\ 528		
20	18	192.168.15.0/24	0	Se0/1	point2point

Vicentina#sh mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	0	192.168.10.3/32	2053	Se1/1	point2point
17	0	192.168.10.1/32	2609	Se1/0	point2point
18	0	192.168.15.0/24	272	Se1/0	point2point

Aggregate: indica que esa etiqueta a sido asignada para una VPN que se especifica mas adelante con el comando ***sh ip bgp vpnv4 vrf loja labels***

Verificar que el protocolo BGP esté enrutando la VPN Loja y que el protocolo MPLS ha asignado una etiqueta para dicha VPN, esto se lo realiza ejecutando los siguientes comandos:

Pop_Mariscal#sh ip route vrf loja

```
Routing Table: loja
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

B    192.168.22.0/24 [200/0] via 192.168.10.3, 00:07:52
C    192.168.19.0/24 is directly connected, Loopback20
```

Pop_Mariscal#sh ip bgp vpnv4 vrf loja labels

```
   Network          Next Hop      In label/Out label
Route Distinguisher: 10:100 (loja)
   192.168.19.0     0.0.0.0         19/aggregate(loja)
   192.168.22.0     192.168.10.3    nolabel/19
```

Para verificar conectividad entre los routers se utilizará el comando *traceroute* de la siguiente manera:

Pop_Mariscal#traceroute 192.168.10.3

```
Type escape sequence to abort.
Tracing the route to 192.168.10.3

  1 192.168.20.2 [MPLS: Label 16 Exp 0] 88 msec 88 msec 88 msec
  2 192.168.21.2 36 msec 36 msec *
```

Pop_Mariscal#traceroute vrf loja 192.168.22.1

```
Type escape sequence to abort.
Tracing the route to 192.168.22.1

  1 192.168.20.2 [MPLS: Labels 16/19 Exp 0] 92 msec 92 msec 92 msec
  2 192.168.22.1 36 msec 40 msec *
```

Nótese que se puede apreciar la etiqueta utilizada para llegar al destino especificado en el comando.

Finalmente se verificará que la configuración de QoS utilizando dos comandos: *sh access-lists rate-limit* muestra el número de la lista o listas de acceso tipo rate-limit y el valor de precedencia IP de los paquetes que serán filtrados y *sh inteface rate-limit* muestra la cantidad de paquetes que han sido filtrados por la lista de acceso y el valor del campo EXP de la cabecera MPLS que se le ha asignado.

Pop_Mariscal#sh access-lists rate-limit

```
Rate-limit access list 20
    3
```

Pop_Mariscal#sh interfaces rate-limit

```
Loopback20
  Input
    matches: access-group rate-limit 20
      params:  8000 bps, 8000 limit, 8000 extended limit
      conformed 0 packets, 0 bytes; action: set-mpls-exp-imposition-
transmit 3
      exceeded 0 packets, 0 bytes; action: set-mpls-exp-imposition-
transmit 3
      last packet: 2840965ms ago, current burst: 0 bytes
      last cleared 00:00:59 ago, conformed 0 bps, exceeded 0 bps
```

Pop_Gasca#sh access-lists rate-limit

```
Rate-limit access list 40
    3
```

Pop_Gasca#sh interfaces rate-limit

```
Loopback20
  Input
    matches: access-group rate-limit 20
      params:  8000 bps, 8000 limit, 8000 extended limit
      conformed 0 packets, 0 bytes; action: set-mpls-exp-imposition-
transmit
      exceeded 0 packets, 0 bytes; action: set-mpls-exp-imposition-
transmit
      last packet: 3027158ms ago, current burst: 0 bytes
      last cleared 00:00:40 ago, conformed 0 bps, exceeded 0 bps
```

CASO DE ESTUDIO 13

FAMILIARIZACIÓN CON LA HERRAMIENTA DE SIMULACIÓN "OPEN SIMMPLS" (SIMULACIÓN 1)

DESCRIPCIÓN GENERAL Y OBJETIVOS

El objetivo de este Caso de Estudio es familiarizarse con la herramienta de simulación Open SimMPLS, para lo cual se realizará un reconocimiento de las funciones y opciones de la aplicación.

TRABAJO PREPARATORIO Y REQUISITOS

Como trabajo preparatorio se debe revisar el portal web de la aplicación http://patanegra.unex.es/opensimmpls y notar sus principales características.

DESARROLLO

En este caso de estudio se configurará un PC con el fin de que la aplicación "Open SimMPLS" funcione correctamente, para lo cual se debe realizar lo siguiente:

1. Requerimientos que el PC debe cumplir.- los requerimientos mínimos y recomendados que el PC debe cumplir se listan a continuación:

Requisitos mínimos:

- Procesador a 300 MHz.
- 64 MB. de memoria RAM.
- 30 MB. de espacio libre en disco.
- Java ® Runtime Enviroment 1.4.2 instalado y configurado.

Requisitos recomendados:

- Procesador a 1,5 GHz.

- 512 MB. de memoria RAM.
- 1 GB. de espacio libre en disco.
- Java ® Runtime Enviroment 1.4.2 instalado y configurado.

Sistemas probados con éxito (sin dificultades):

- **Windows 98 SE** en un Intel Pentium II 300 MHz, RAM de 64 MB.
- **Windows 2000 pro** en un AMD Athlon Thunderbird 800 MHz, RAM de 512 MB.
- **Windows Me** en un AMD Athlon 1 GHz, RAM de 256 MB.
- **GNU/Linux Red Hat 8.x** en un AMD Athlon Thunderbird 800 MHz, RAM de 512 MB.
- **GNU/Linux Red Hat 9.x** en un AMD Athlon Thunderbird 800 MHz, RAM de 512 MB.
- **GNU/Linux LinEx 3.x** en un AMD Athlon Thunderbird 800 MHz, RAM de 512 MB.
- **GNU/Linux LinEx 3.x** en un AMD Athlon 1 GHz, RAM de 256 MB.
- **GNU/Linux RedHat Fedora** en un portatil Intel Cetrino 2,5 GHz, RAM de 512 MB.

Sistemas probados con éxito (con dificultades):

- **GNU/Linux Debian Woody** en Sun Sparc Ultra 5 (64 bits), RAM de 256 MB.

Sistemas donde no funciona:

- **MSDOS**, sobre cualquier plataforma i386.
- **PC-DOS**, sobre cualquier plataforma i386.

2. Instalación de la aplicación en el PC.- para instalar la aplicación en el PC se debe copiar el archivo ejecutable "openSimMPLS.jar" en cualquier directorio del PC de preferencia en c:\SimMPLS. Este archivo puede ser descargado gratuitamente del portal http://patanegra.unex.es/opensimmpls y aproximadamente tiene un tamaño de 2.3 MB.

3. Ejecución de la aplicación.- para iniciar la aplicación se ejecuta el archivo openSimMPLS.jar y aparece la siguiente pantalla:

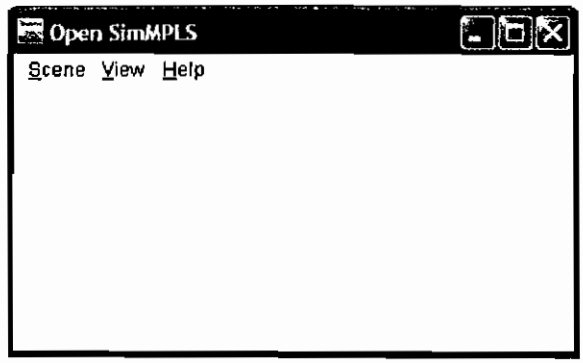


Figura 3.25: Pantalla principal del Open SimMPLS 1.0

Para crear un nuevo escenario se selecciona en la barra de herramientas "Scene" y luego la opción "New".

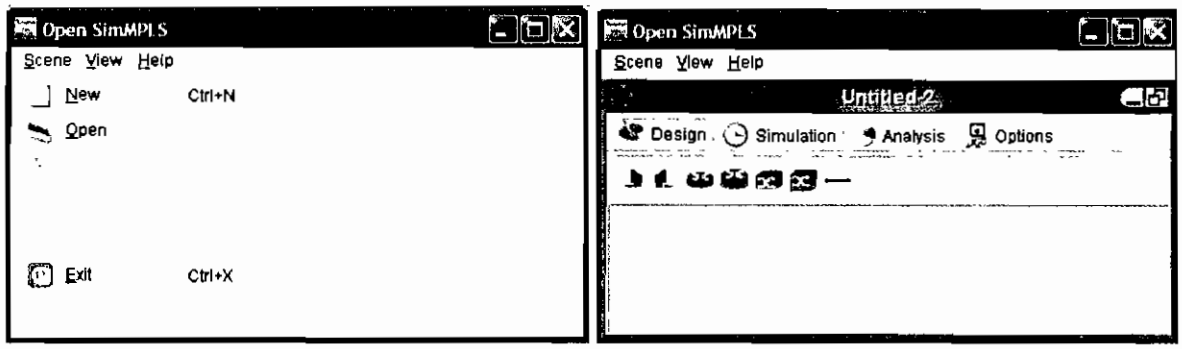


Figura 3.26: Creación de un nuevo escenario.

Luego se procede a insertar cualquiera de los siete tipos de elementos disponibles en la barra para configurar nuestro escenario estos son: de izquierda a derecha: traffic source, traffic sink, LER, active LER, LSR, active LSR y link. Recuerde que cada elemento requiere de un nombre al momento de insertarlo y se puede configurar o cambiar sus características de acuerdo a lo que se requiera. (LER y LSR activos son utilizados en ambientes donde se configuren GoS)

Se utilizará un escenario preconfigurado para utilizar las características de Simulación y Análisis que Open SimMPLS nos ofrece, para esto se simulará el

ejemplo "ReencaminamientoTradicional.osm" que puede ser descargado gratuitamente en el portal http://patanegra.unex.es/opensimmpls y se muestra a continuación:

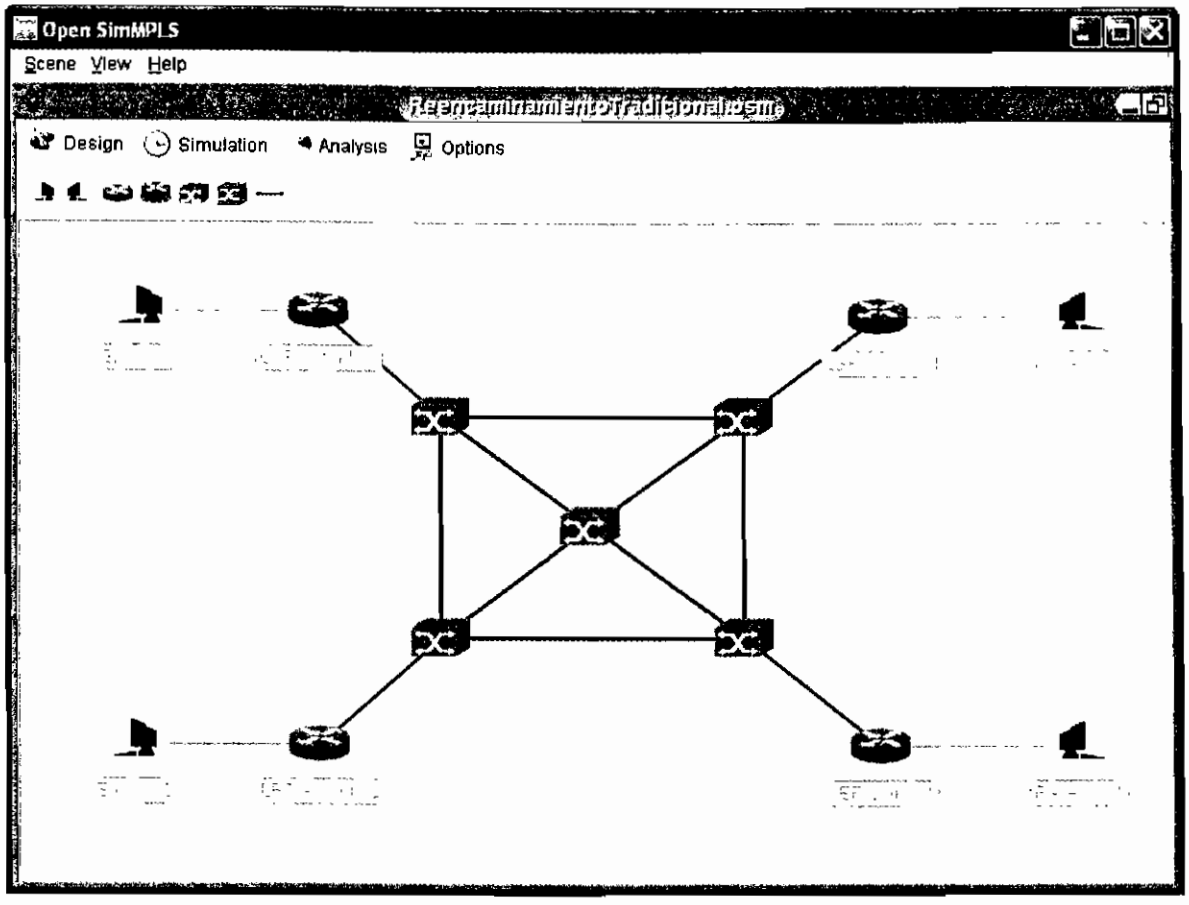


Figura 3.27: Ejemplo preconfigurado "Reencaminamiento Tradicional".

En el panel de Design se puede configurar las características de cada elemento, modificar el escenario, crear o eliminar conexiones y elementos, etc. Las características principales que se puede configurar se listan a continuación:

traffic source:

Name: Nombre del elemento.

traffic destination: Destino del tráfico generado.

Generate statistics for this: Genera una gráfica con las estadísticas de tráfico de este elemento.

traffic sink:

Name: Nombre del elemento.

Generate statistics for this: Genera una gráfica con las estadísticas de tráfico de este elemento.

LER:

Name: Nombre del elemento.

Switching power: Capacidad de conmutación de paquetes del LER.

Incoming buffer size: Tamaño del buffer de entrada.

Generate statistics for this: Genera una gráfica con las estadísticas de tráfico de este elemento.

LSR:

Name: Nombre del elemento.

Switching power: Capacidad de conmutación de paquetes del LSR.

Incoming buffer size: Tamaño del buffer de entrada.

Generate statistics for this: Genera una gráfica con las estadísticas de tráfico de este elemento.

link:

Name: Nombre del elemento.

Outside-left: Punto de conexión izquierdo.

Outside-right: Punto de conexión derecho.

Link speed: Velocidad del enlace.

Generate statistics for this: Genera una gráfica con las estadísticas de tráfico de este elemento.

SIMULACIÓN

A continuación se configurará el LSR del centro de la nube MPLS para que genere un gráfico de estadísticas de tráfico, para esto, en el panel Design con click derecho en el LSR central se escoge Properties y en la pestaña Fast chequeamos la opción de generar estadísticas. Luego se pasará al panel Simulation y se empezará a simular.

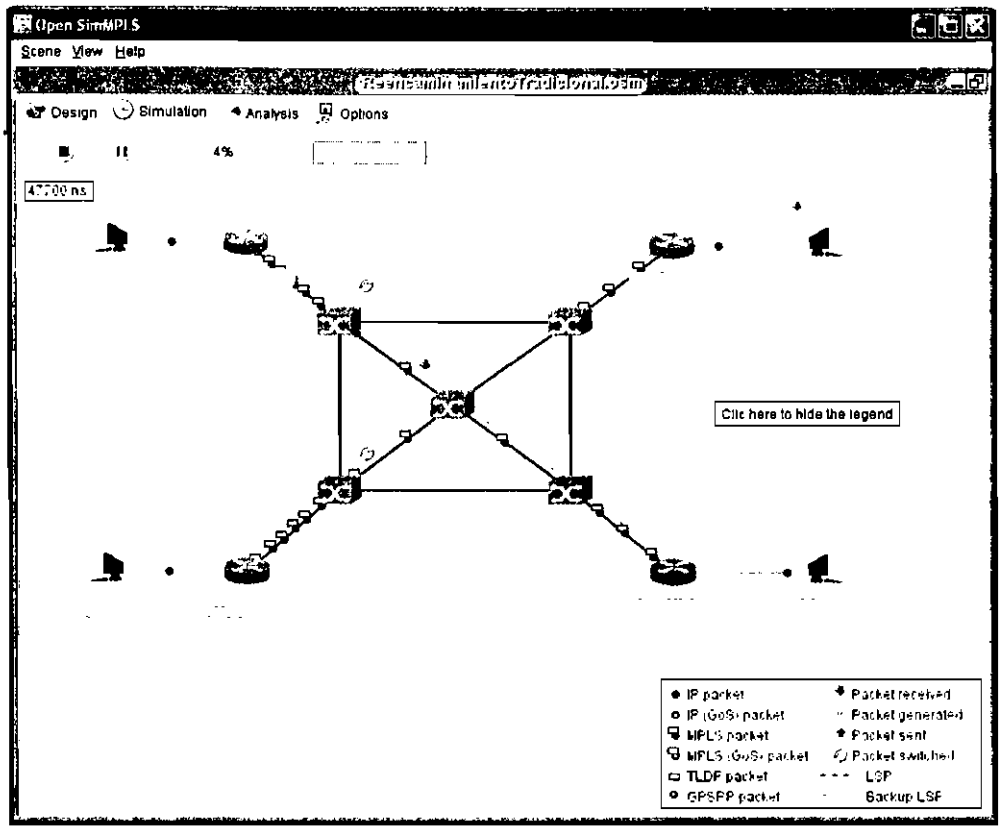


Figura 3.28: Simulación "Reencaminamiento Tradicional".

Finalmente en el panel Analysis se observará gráficamente las estadísticas del tráfico que pasa a través del LSR central para esto se selecciona LSR 5 en el menú de la sección Analysis y se observa las siguientes gráficas:

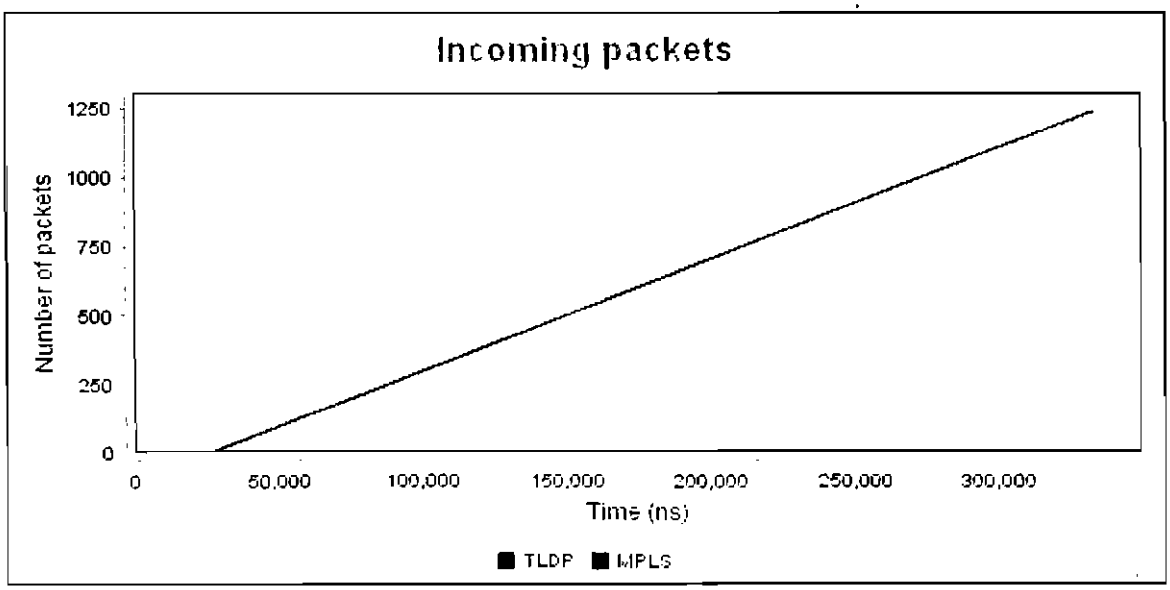


Figura 3.29: Paquetes de entrada "Reencaminamiento Tradicional".

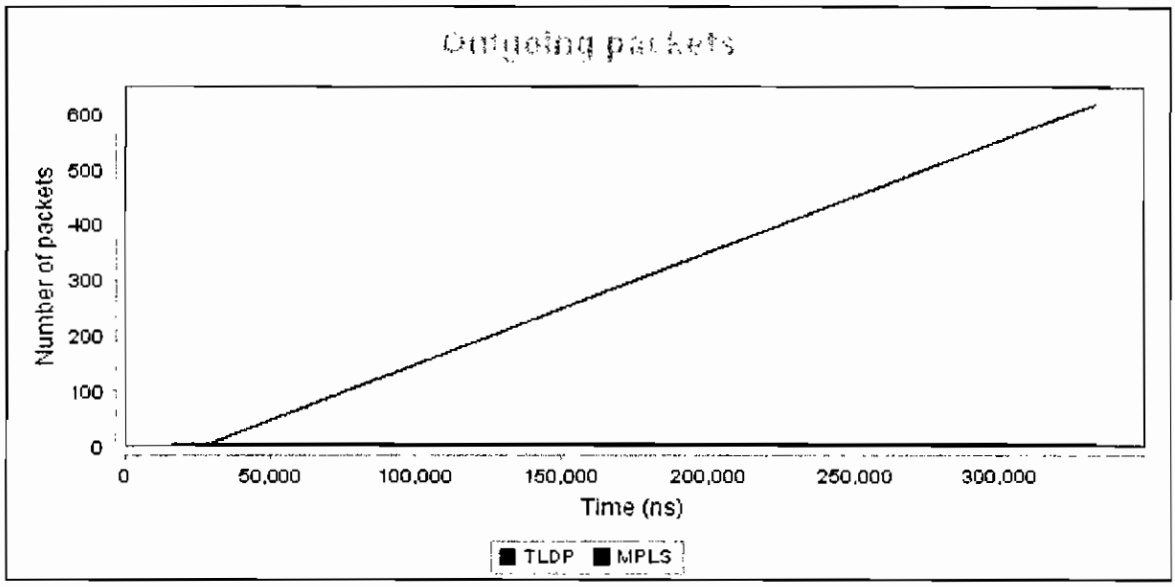


Figura 3.30: Paquetes de salida "Reencaminamiento Tradicional".

Como se observa en las gráficas, el tráfico de entrada y salida es constante debido a que el generador de tráfico está configurado con tráfico de esa característica y la conmutación se realiza sin problemas, es decir, a cada paquete que es recibido se le cambia la etiqueta según la tabla LFIB, es conmutado y enviado por la interfaz de salida.

CASO DE ESTUDIO 14

SIMULACIÓN DE LA PÉRDIDA DE UN ENLACE DE LA NUBE MPLS Y LA REESTRUTURACIÓN DEL LSP CON MPLS TRADICIONAL (SIMULACIÓN 2)

DESCRIPCIÓN GENERAL Y OBJETIVOS

El objetivo es observar el comportamiento de la red MPLS al momento de perder un enlace y como se produce el reestructuramiento de un LSP con MPLS tradicional al producirse dicha pérdida.

TRABAJO PREPARATORIO Y REQUISITOS

El trabajo preparatorio es realizar el Caso de Estudio 13 y los requisitos del PC se detallan en el mismo caso de estudio.

DESARROLLO

1. Elaboración del escenario.- Para el desarrollo de este caso de estudio, se debe elaborar el escenario del la figura ce.14.a, el mismo que tiene las siguientes características:

Emisores:

Traffic: rate = 10240 Mbps, constante, payload = 618 octetos

LERs:

LER features: very high range LER

LSRs:

LSR features: very high range LSR

Links:

Link speed: todos los links tienen la velocidad configurada en too fast excepto el enlace entre el LSR 2 y el LSR 3 que tiene un delay de 3677 ns.

Los LSRs y el LER 2 están configurados para que generen las gráficas de estadísticas de tráfico.

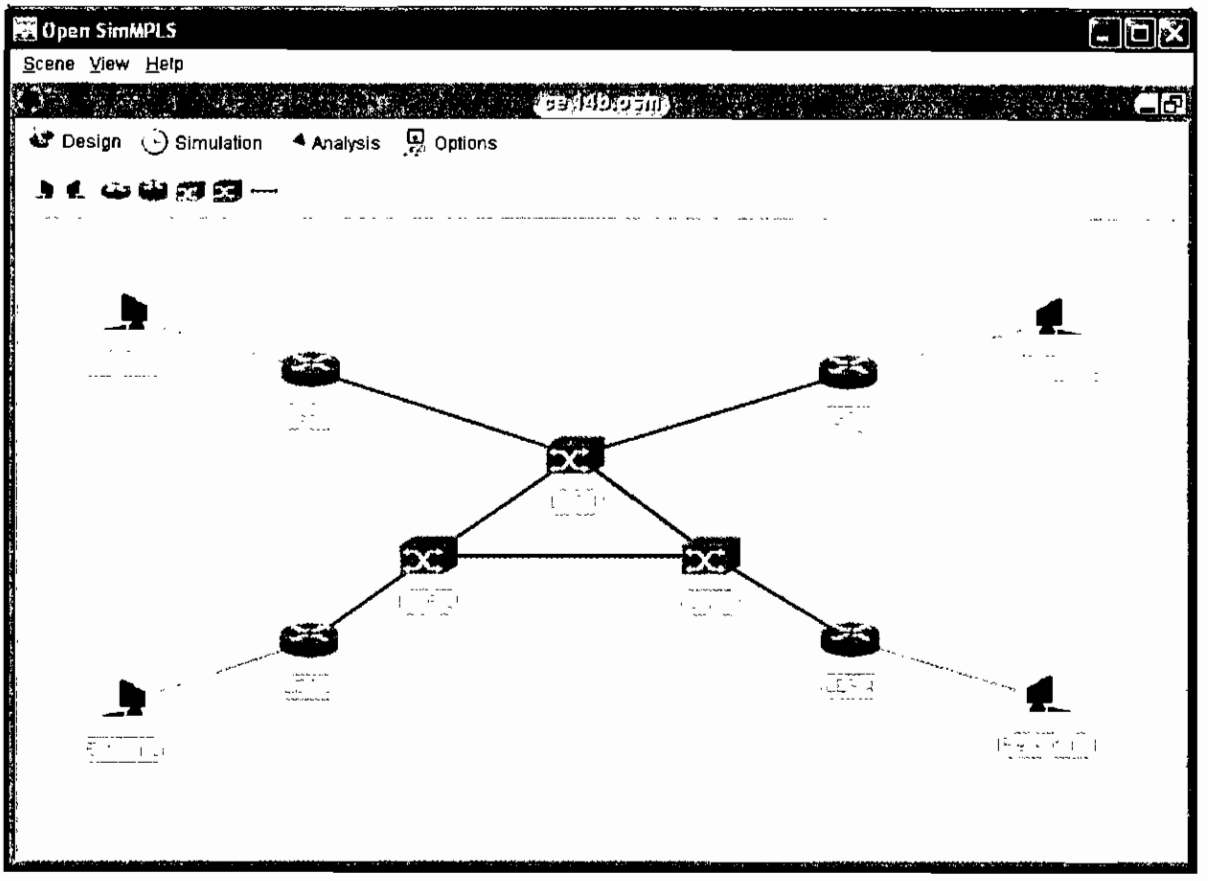


Figura 3.31: Red utilizada en el Caso de Estudio 14

SIMULACIÓN

Para realizar la simulación se debe seguir los siguientes pasos:

- Iniciar la simulación desde el panel Simulation:
  Recuerde que en el panel Options puede variar los parámetros de simulación como tiempo total de simulación y características de las gráficas de tráfico.

  NOTA: Las capacidades de conmutación de los LERs y LSRs, los retardos de los enlaces y los tiempos de simulación deben tener concordancia para observar correctamente los resultados.

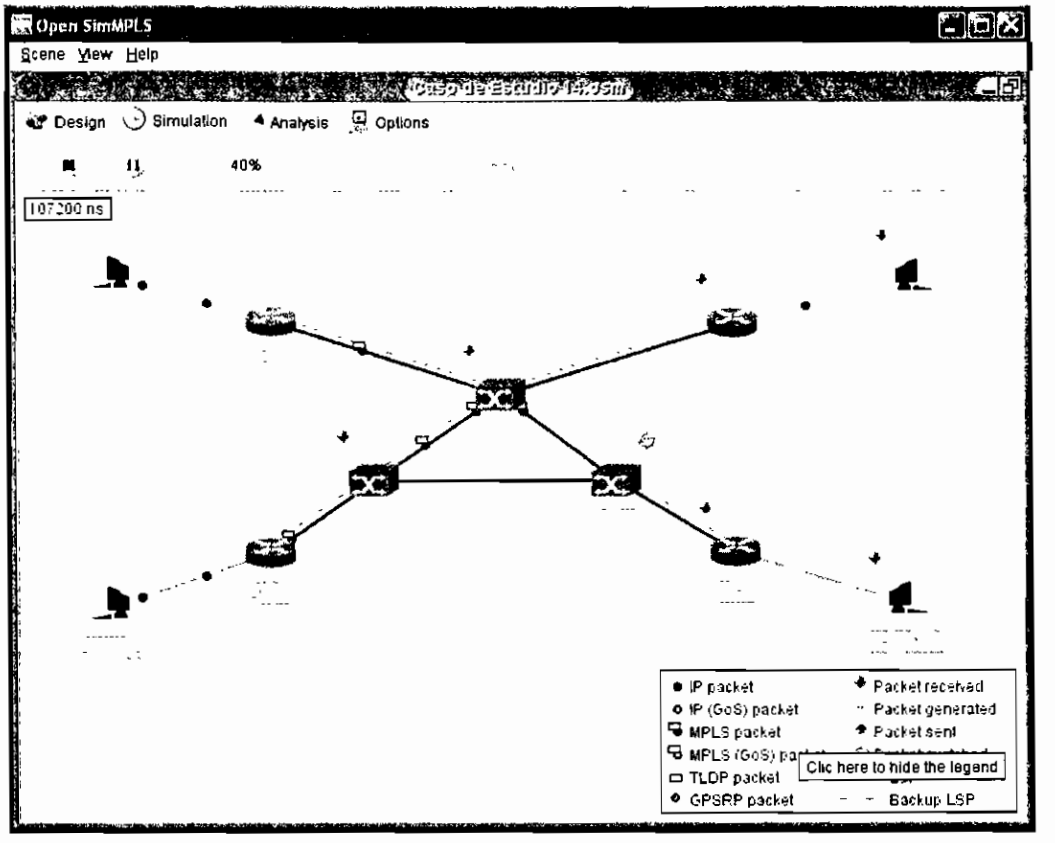


Figura 3.32: Simulación 1 Caso de Estudio 14

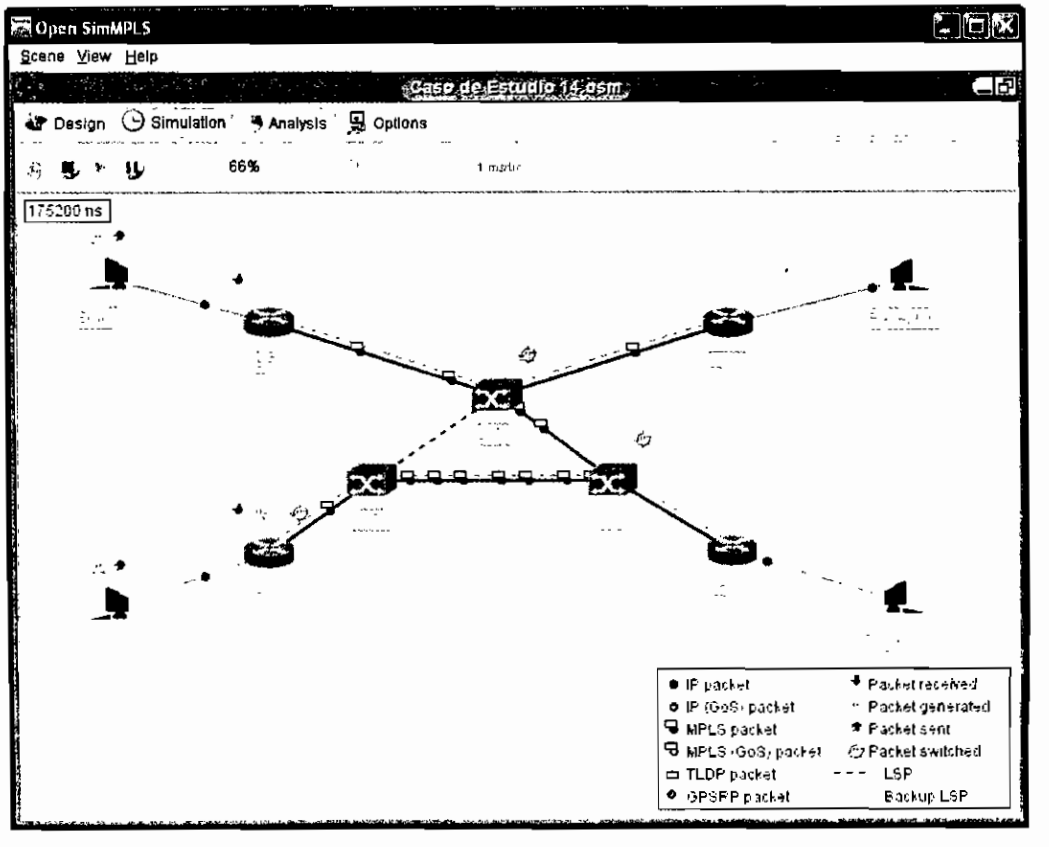


Figura 3.33: Simulación 2 Caso de Estudio 14

- Una vez establecidos los LSPs simule la caída de un enlace haciendo click sobre el mismo como se muestra en la figura ce.14.c y note como los paquetes LDP son enviados nuevamente para reestructurar el LSP y al finalizar este proceso los paquetes son enviados por el nuevo LSP.

- Finalmente se observa la gráfica del LSR 2 y se nota que a los 120 ns aprox. el enlace se cayó y no se tenía tráfico y se comenzaron a tener paquetes descartados ya que no se tenía un LSP activo como se muestra en la figura 3.34:

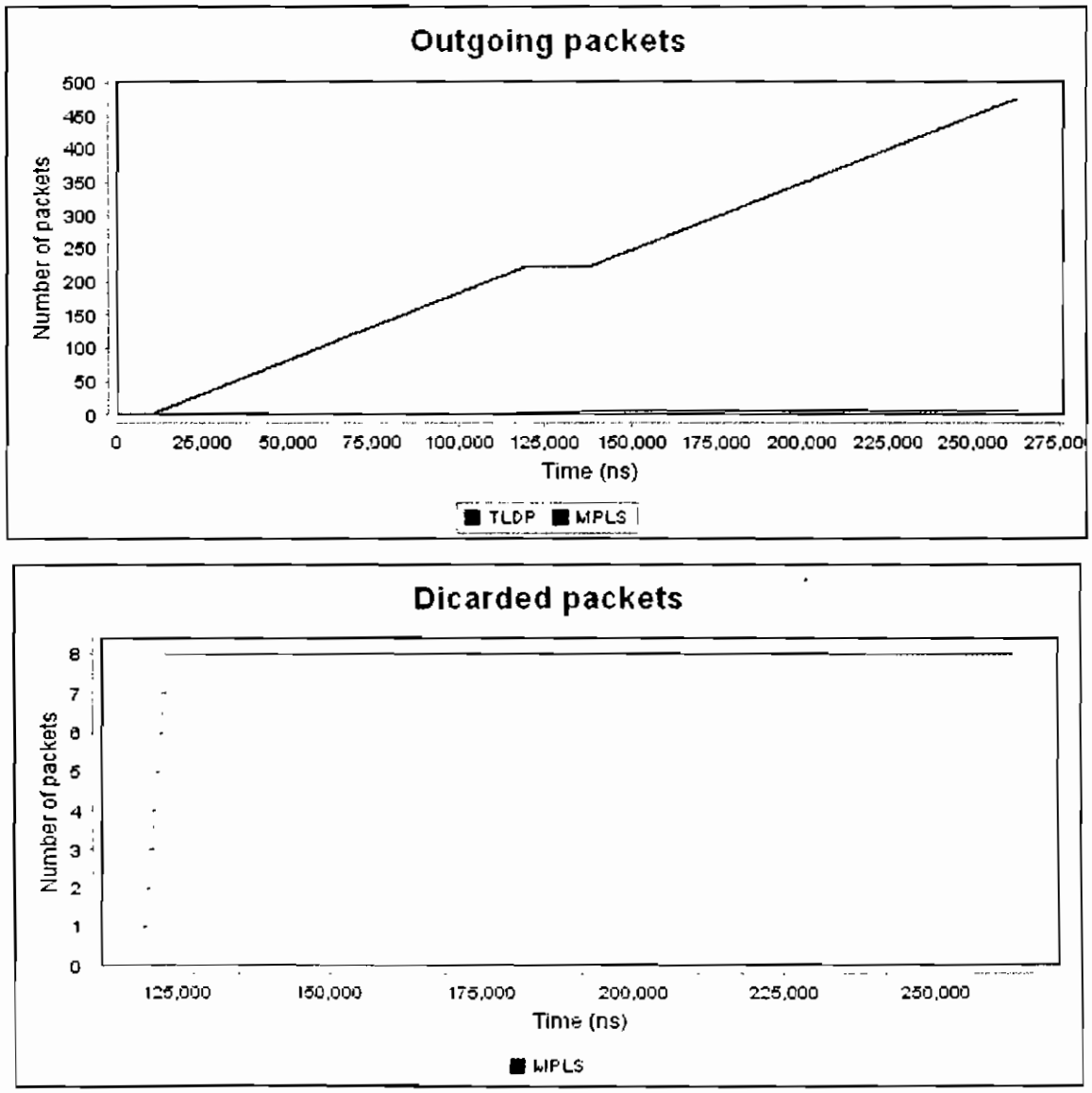


Figura 3.34: Paquetes de salida y descartados en el LSR 2

CASO DE ESTUDIO 15

SIMULACIÓN DE UN LSR CONGESTIONADO EN UN BACKBONE MPLS TRADICIONAL (SIMULACIÓN 3)

DESCRIPCIÓN GENERAL Y OBJETIVOS

El objetivo es observar el comportamiento de la red MPLS al momento que un LSR se congestiona debido a no poder procesar la cantidad de tráfico recibido con MPLS tradicional.

TRABAJO PREPARATORIO Y REQUISITOS

El trabajo preparatorio es realizar el Caso de Estudio 14 y los requisitos del PC se detallan en el caso de estudio 13.

DESARROLLO

1. Elaboración del escenario.- Para el desarrollo de este caso de estudio, se debe elaborar el escenario de la figura 3.35, el mismo que tiene las siguientes características:

Emisores:

Traffic: rate = 10240 Mbps, variable.

LERs:

LER features: very high range LER

LSRs:

LSR features: very high range LSR, excepto el LSR 4 (972 Mbps switching)

Links:

Link speed: todos los links tienen la velocidad configurada en too fast excepto los enlaces entre los LSR 1-LSR 3, LSR 3-LSR 6, LSR 2-LSR 5 y LSR 5-LSR 7 que tiene un delay de 3677 ns.

El LSR 4 está configurado para que genere las gráficas de estadísticas de tráfico.

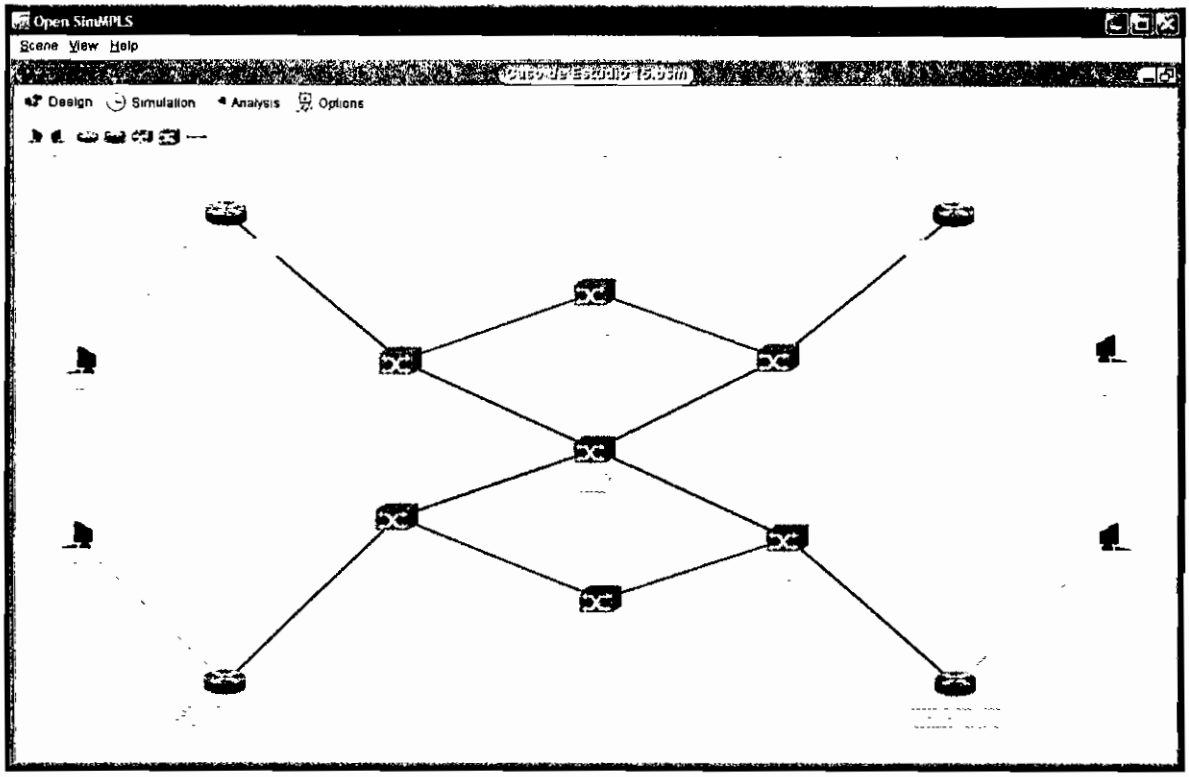


Figura 3.35: Red utilizada en el Caso de Estudio 15

SIMULACIÓN

Para realizar la simulación se debe seguir los siguientes pasos:

- Iniciar la simulación desde el panel Simulation:
  Recuerde que en el panel Options puede variar los parámetros de simulación como tiempo total de simulación y características de las gráficas de tráfico.

  NOTA: Las capacidades de conmutación de los LERs y LSRs, los retardos de los enlaces y los tiempos de simulación deben tener concordancia para observar correctamente los resultados.

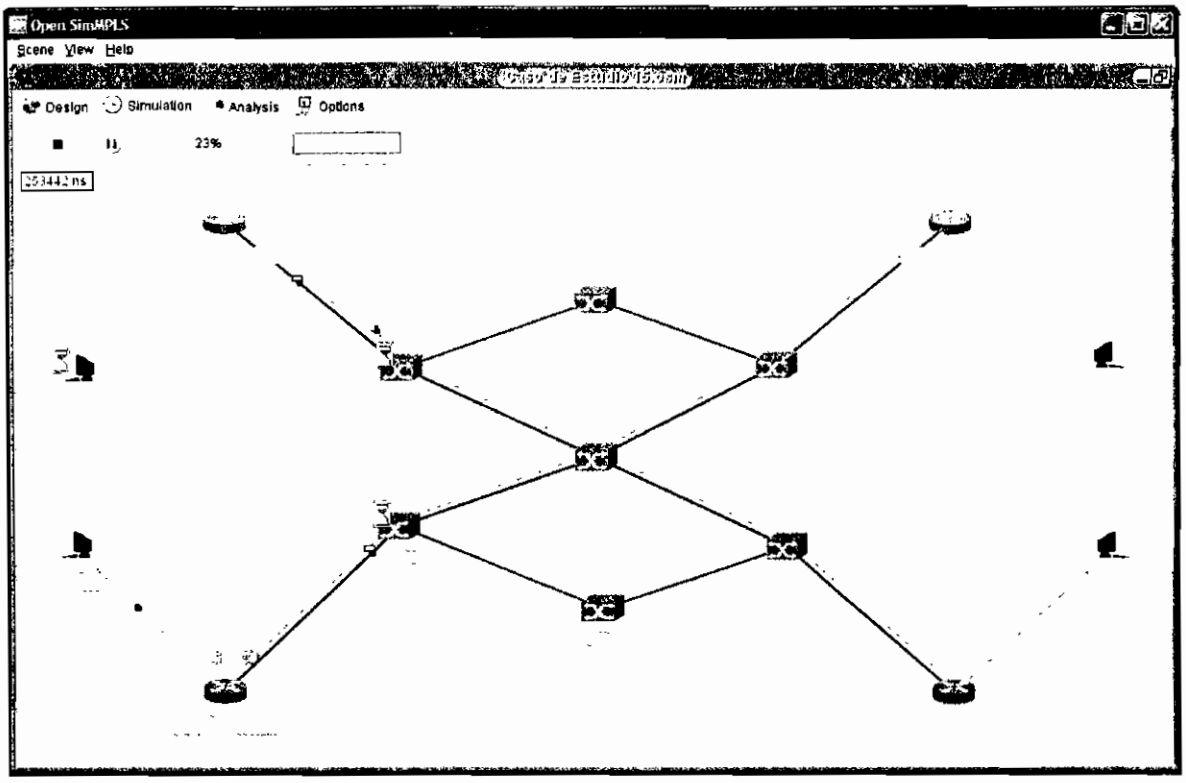


Figura 3.36: Simulación 1 Caso de Estudio 15

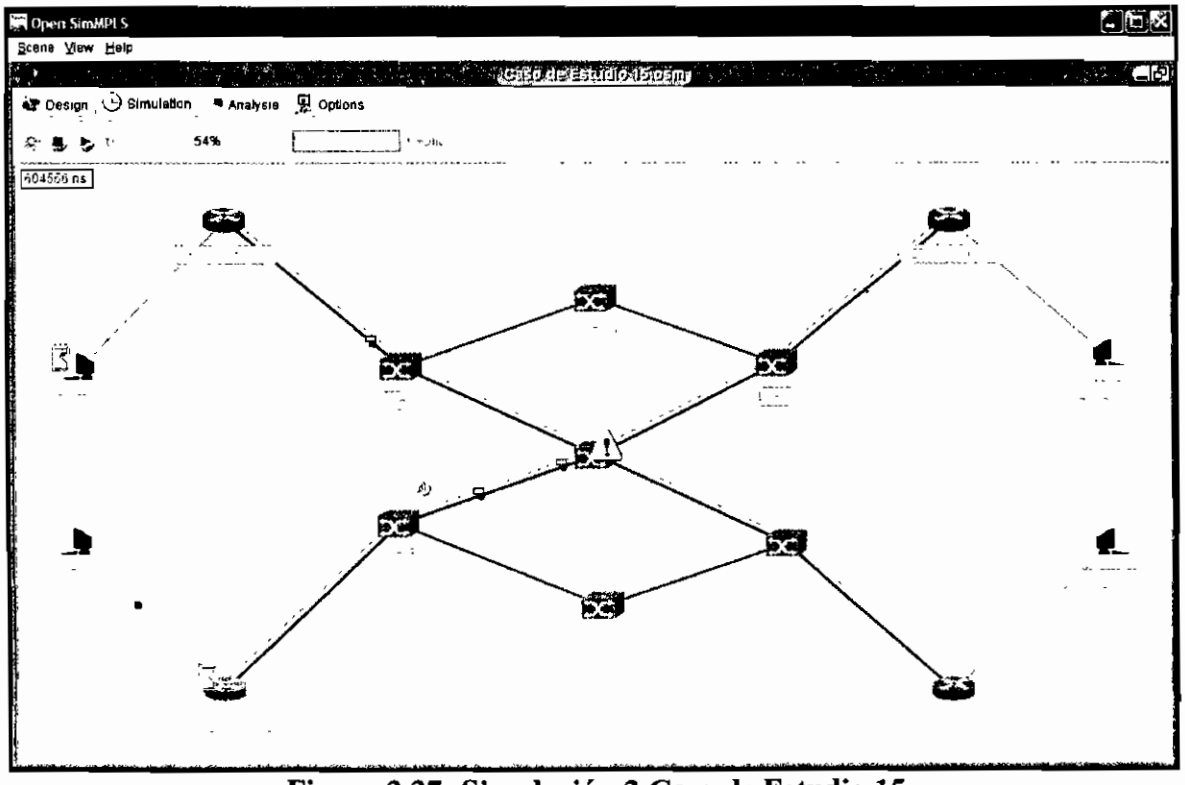


Figura 3.37: Simulación 2 Caso de Estudio 15

- Luego de iniciar la simulación (figura 3.36) se crearán los LSPs, pero los paquetes al atravesar el LSR 4 lo harán con lentitud ya que el poder de conmutación de dicho LSR es menor a la velocidad con que el tráfico es creado, esto se diseñó con el fin de congestionar el LSR 4. En la figura 3.37 se nota como se congestiona el LSR 4 cuando el icono cambia a color rojo.

- Finalmente se observa la gráfica del LSR 4 y se nota que se congestiona a los 500.000 ns y se comienzan a descartar paquetes debido a que el LSR no tiene la capacidad para procesarlos y el buffer de entrada ya está saturado:

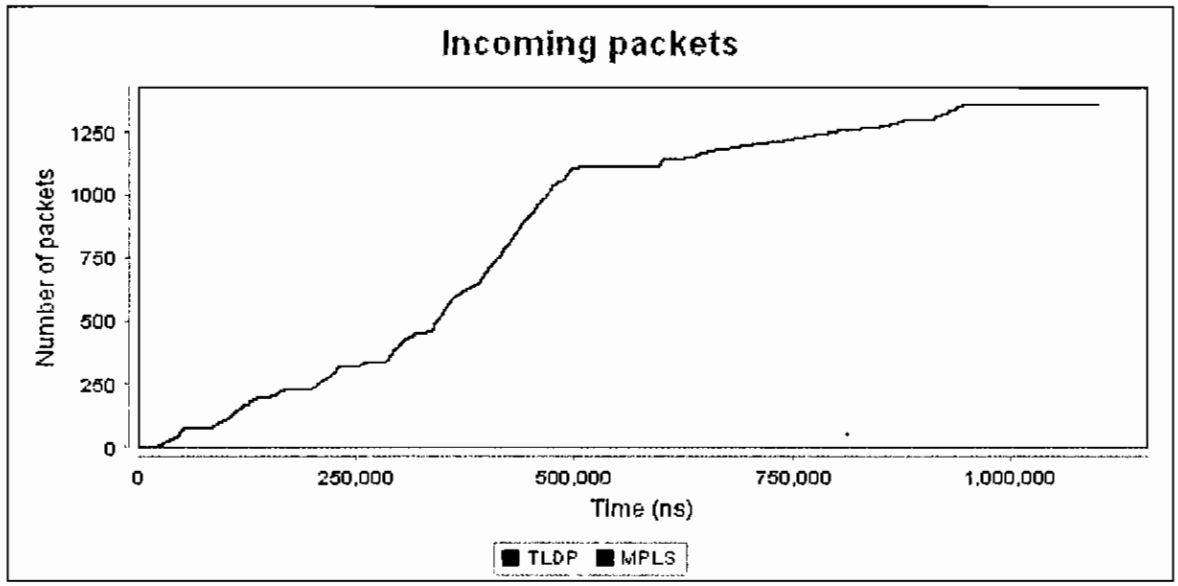


Figura 3.38: Paquetes de entrada en el LSR 4

Comparando las figuras 3.38 y 3.39 se puede notar como el tráfico de entrada es superior al de salida y los paquetes descartados son confirmados por la figura 3.40.

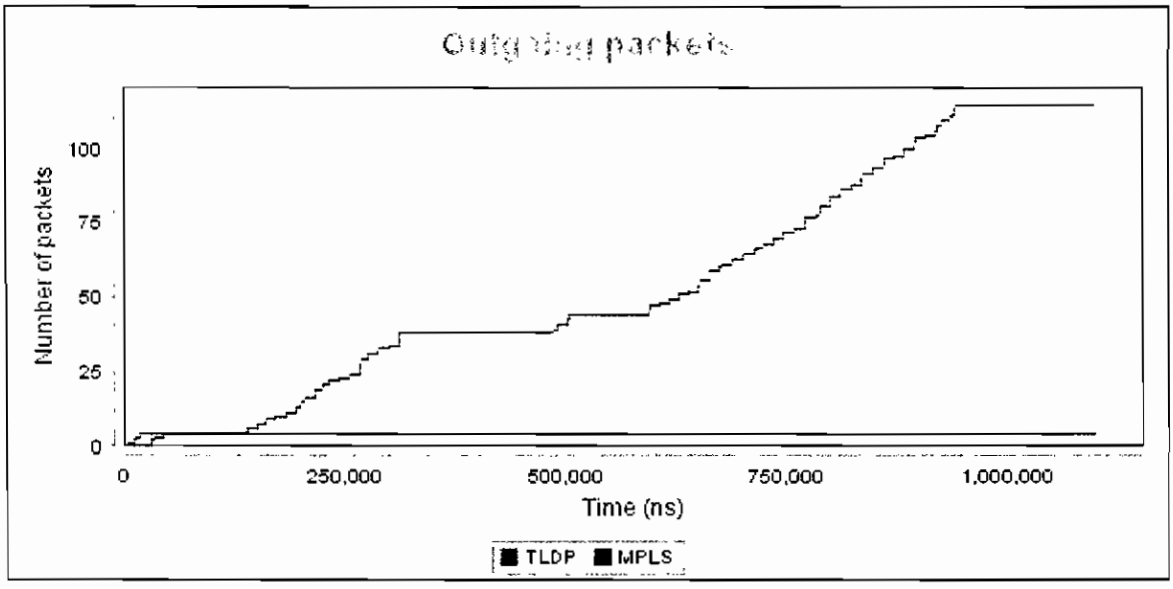


Figura 3.39: Paquetes de salida en el LSR 4

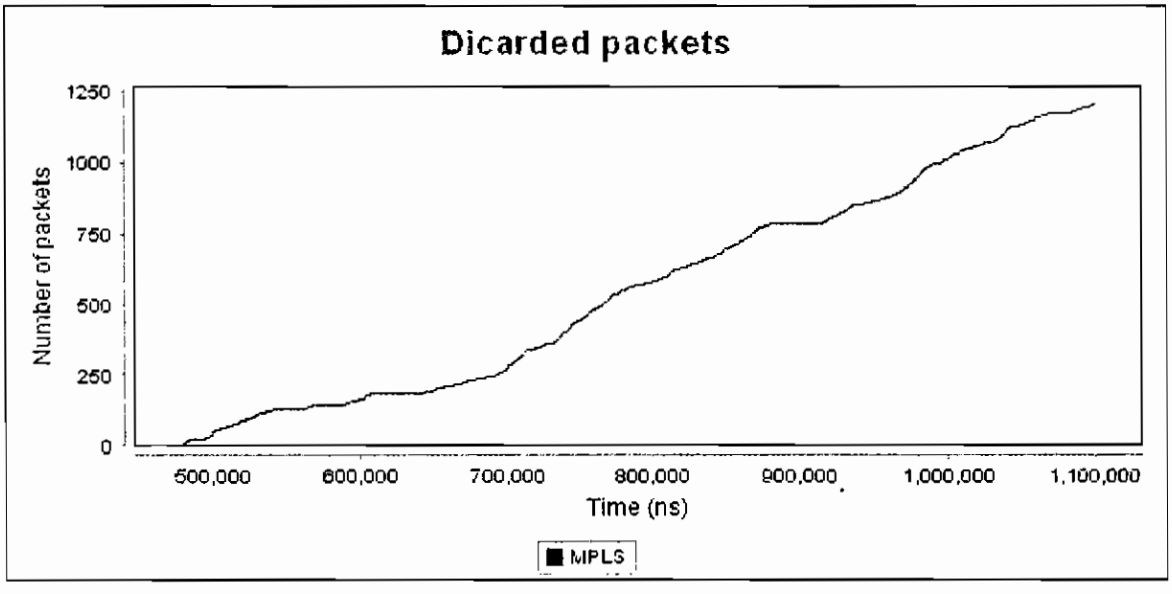


Figura 3.40: Paquetes descartados en el LSR 4

CAPÍTULO IV

4 CONCLUSIONES Y RECOMENDACIONES

- La arquitectura MPLS se ha convertido en la solución más prometedora a las nuevas necesidades de las redes de backbone que tienen el enorme reto de gestionar redes cada vez más complejas y extensas, con una mayor gama de servicios y con creciente demanda de ancho de banda, calidad y garantías.

- Una de las características importantes de MPLS es ser multiprotocolo esto significa que soporta varios protocolos, tales como ATM y Frame Relay, lo cual fortalece a la arquitectura mucho más ya que puede ser utilizada por diferentes tipos de redes que utilizan protocolos distintos.

- MPLS es el último paso en la evolución de las tecnologías de conmutación multinivel (o conmutación IP). La idea básica de separar lo que es el envío de los datos (mediante el algoritmo de intercambio de etiquetas) de los procedimientos de enrutamiento estándar IP, ha llevado a un acercamiento de los niveles 3 y 2, con el consiguiente beneficio en cuanto a rendimiento y flexibilidad de esta arquitectura.

- MPLS abre a los proveedores IP la oportunidad de ofrecer nuevos servicios que no son posibles con las técnicas actuales de enrutamiento IP (típicamente limitadas a enrutar por dirección de destino).

- Además de poder hacer ingeniería de tráfico IP, MPLS permite mantener clases de servicio y soporta con gran eficacia la creación de VPNs. Por todo ello, MPLS aparece ahora como la gran promesa y esperanza para poder mantener el ritmo actual de crecimiento de la Internet.

- La arquitectura MPLS es independiente del protocolo de Capa Red (Capa 3), por lo que, MPLS funciona igual de bien sin tomar en cuenta el protocolo de Capa 3 utilizado incluyendo IP e IPX.

- MPLS funciona con DiffServ y RSVP para brindar característica QoS adicionales.

- La conmutación de etiquetas es mucho más rápida que el enrutamiento IP debido al hecho de que en MPLS se procesa una cabecera de 4 bytes mucho menor comparada con la cabecera IPv4 que varía de 20 a 60 bytes.

- Si bien el tiempo de convergencia de una red MPLS es mayor al de una red que utiliza técnicas de enrutamiento comunes, luego de dicha convergencia el rendimiento de la red MPLS es mucho mayor al de cualquier otra red.

- La configuración de MPLS en equipos Cisco es sencilla pero debe ser llevada a cabo con un gran cuidado ya que podemos encontrar inconvenientes en el orden en el cual ingresamos los comandos, esto se detalla en el Capítulo II.

- Debido a que la arquitectura MPLS no requiere de un equipo extra en la red del cliente hace que ésta se coloque como una de las soluciones más convenientes al momento de requerir una red de backbone de un mayor rendimiento.

- Al momento de empezar a configurar los equipos se debe asegurar que la versión del IOS de Cisco sea la correcta y soporte los comandos que se debe utilizar. En el portal www.cisco.com se puede consultar y comparar las características de los diferentes IOS disponibles.

- Para comenzar a configurar un equipo se debe borrar el archivo de configuración para asegurar de no tener problemas posteriores por duplicación de configuración o características de configuración no deseadas como un protocolo de enrutamiento que no se necesite.

- El uso de herramientas de simulación es importante ya que de esta manera se puede tener una idea del funcionamiento de ciertas configuraciones de red que son imposibles de configurar físicamente en un laboratorio.

REFERENCIAS BIBLIOGRÁFICAS

[1] PEPELNJAK, Ivan; GUICHARD, Jim, "Arquitecturas MPLS y VPN". 1° Edición. Pearson Educación. Madrid. 2003.

[2] MILLER, Bruce; STEWART, Elliott "MPLS Conformance and Performance Testing". Ixia. 2004.

[3] CISCO. "Cisco MPLS Controller Software Configuration Guide". http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t4/ft_cos4t.htm#96788

[4] CISCO. "Cisco MPLS Class of Service Enhancements". http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120st/120st10/10st_cos.htm

[5] CISCO. "Command Reference". http://www.cisco.com/univercd/home/home.htm

[6] Donoso Meisel Yezid, Multidifusión IP sobre MPLS sin y con QoS. http://eia.udg.es/~atm/bcds/pdf/yezid_02.pdf

[7] José Barberá, MPLS: Una arquitectura de backbone para la Internet del siglo XXI. http://www.rediris.es/rediris/boletin/53/enfoque1.html

[8] Juniper, http://www.juniper.net/solutions/literature/white_papers/200012.pdf

[9] Cisco, Cisco MPLS AutoBandwidth Allocator, http://www.cisco.com/swa/c/cdc.css

[10] Cisco, Cisco IOS® MPLS Virtual Private LAN Service (VPLS), http://www.cisco.com/en/US/products/sw/iosswrel/ps1839/products_command_reference_chapter09186a00800b3510.html

[11] MPLScon. http://www.mplsrc.com/

[12] IETF, Multiprotocol Label Switching. http://www.ietf.org/html.charters/mpls-charter.html

[13] Universidad Carlos III, Una tecnología para la construcción de protocolos. http://gsyc.escet.urjc.es/simple_com/phd-thesis-es/node20.html

[14] Red-MPLS, Workshop MPLS Girona, 28 de marzo de 2003, http://red-mpls.udg.es/programa-ws-28-03-2003.html

[15] Alfonso Ariza Quintana, Encaminamiento en redes orientada a flujo con imprecisión de datos. http://red-mpls.udg.es/presentaciones/ariza_girona.pdf

[16] Tomás P. de Miguel, Redes Privadas Virtuales y MPLS, http://red-mpls.udg.es/presentaciones/rpv_mpls.pdf

[17] Escuela Politécnica de Cáceres, España, 2004, http://patanegra.unex.es/opensimmpls

[18] Tomás P. de Miguel, Redes Privadas Virtuales y MPLS, http://icadc.cordis.europa.eu.int/fep-cgi/srchidadb?CALLER=ES_NEWS&ACTION=D&SESSION=&RCN=25109

CAPITULO II

CONFIGURACIÓN BÁSICA DE LA NUBE MPLS.

RESULTADOS DE LA CONFIGURACIÓN.

Anexo 2.1.3.a: Configuración LSR_P

ROUTER LSR_P (Cisco 3640)

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname LSR_P
LSR_P(config)#enable password lsrp
LSR_P(config)#line console 0
LSR_P(config-line)#password lsrp
LSR_P(config-line)#login
LSR_P(config-line)#exit
LSR_P(config)#line vty 0 4
LSR_P(config-line)#password lsrp
LSR_P(config-line)#login
LSR_P(config-line)#exit
LSR_P(config)#exit
LSR_P#
LSR_P(config)#interface serial 1/0
LSR_P(config-if)#ip address 192.168.1.2 255.255.255.252
LSR_P(config-if)#no shutdown
LSR_P(config-if)#exit
LSR_P(config)#interface serial 1/1
LSR_P(config-if)#ip address 192.168.1.5 255.255.255.252
LSR_P(config-if)#clock rate 56000
LSR_P(config-if)#no shutdown
LSR_P(config-if)#exit
LSR_P(config)#exit
LSR_P(config)#
LSR_P(config)#
LSR_P(config)#router ospf 1
LSR_P(config-router)#network 192.168.1.0 0.0.0.255 area 0
LSR_P(config-router)#exit
LSR_P(config)#
LSR_P(config)#
LSR_P(config)#ip cef
LSR_P(config)#
LSR_P(config)#
LSR_P(config)#mpls ip
LSR_P(config)#
LSR_P(config)#
LSR_P(config)#interface serial 1/0
LSR_P(config-if)#mpls ip
LSR_P(config-if)#exit
LSR_P(config)#
LSR_P(config)#
LSR_P(config)#interface serial 1/1
LSR_P(config-if)#mpls ip
LSR_P(config-if)#exit
LSR_P(config)#
LSR_P(config)#interface loopback 0
LSR_P(config-if)#ip address 192.168.1.100 255.255.255.255
LSR_P(config-if)#no shutdown
LSR_P(config-if)#exit
```

```
LSR_P(config)#mpls label protocol ldp
LSR_P(config)#
LSR_P(config)#mpls ldp router-id loopback 0
LSR_P(config)#
LSR_P(config)#exit
LSR_P#
```

Anexo 2.1.3.b: Configuración: LSR_PE_1

ROUTER LSR_PE_1 (Cisco 2610)

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname LSR_PE_1
LSR_PE_1(config)#enable password lsr1
LSR_PE_1(config)#line console 0
LSR_PE_1(config-line)#password lsr1
LSR_PE_1(config-line)#login
LSR_PE_1(config-line)#exit
LSR_PE_1(config)#line vty 0 4
LSR_PE_1(config-line)#password lsr1
LSR_PE_1(config-line)#login
LSR_PE_1(config-line)#exit
LSR_PE_1(config)#exit
LSR_PE_1#
LSR_PE_1#
*Mar  1 00:36:38.743: %SYS-5-CONFIG_I: Configured from console by console
LSR_PE_1#
LSR_PE_1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
LSR_PE_1(config)#interface serial 0/0
LSR_PE_1(config-if)#ip address 192.168.1.1 255.255.255.252
LSR_PE_1(config-if)#clock rate 56000
LSR_PE_1(config-if)#no shutdown
LSR_PE_1(config-if)#exit
LSR_PE_1(config)#exit
LSR_PE_1#
*Mar  1 00:37:26.886: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
*Mar  1 00:37:27.888: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial0/0,
 changed state to up
*Mar  1 00:37:28.000: %SYS-5-CONFIG_I: Configured from console by console
LSR_PE_1#
LSR_PE_1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
LSR_PE_1(config)#
LSR_PE_1(config)#router ospf 1
LSR_PE_1(config-router)#network 192.168.1.0 0.0.0.255 area 0
LSR_PE_1(config-router)#exit
LSR_PE_1(config)#
*Mar  1 00:38:24.998: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.5 on Serial0/0
fr
om LOADING to FULL, Loading Done
LSR_PE_1(config)#
LSR_PE_1(config)#ip cef
LSR_PE_1(config)#
LSR_PE_1(config)#
LSR_PE_1(config)#mpls ip
LSR_PE_1(config)#
LSR_PE_1(config)#
```

```
LSR_PE_1(config)#interface serial 0/0
LSR_PE_1(config-if)#mpls ip
LSR_PE_1(config-if)#exit
LSR_PE_1(config)#
LSR_PE_1(config)#interface loopback 0
LSR_PE_1(config-if)#ip address 192.168.1.101 255.255.255.255
LSR_PE_1(config-if)#no shutdown
LSR_PE_1(config-if)#exit
LSR_PE_1(config)#
LSR_PE_1(config)#
LSR_PE_1(config)#mpls label protocol ldp
LSR_PE_1(config)#
LSR_PE_1(config)#
LSR_PE_1(config)#mpls ldp router-id loopback 0
LSR_PE_1(config)#
LSR_PE_1(config)#
LSR_PE_1(config)#exit
LSR_PE_1#
LSR_PE_1#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
LSR_PE_1#
```

Anexo 2.1.3.c: Configuración: LSR_PE_2

ROUTER LSR_PE_2 (Cisco 2610)

```
Router>enable
Router#
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#host
Router(config)#hostname LSR_PE_2
LSR_PE_2(config)#enable password lsr2
LSR_PE_2(config)#line console 0
LSR_PE_2(config-line)#password lsr2
LSR_PE_2(config-line)#login
LSR_PE_2(config-line)#exit
LSR_PE_2(config)#line vty 0 4
LSR_PE_2(config-line)#password lsr2
LSR_PE_2(config-line)#login
LSR_PE_2(config-line)#exit
LSR_PE_2(config)#exit
LSR_PE_2(config)#
LSR_PE_2(config)#interface serial 0/1
LSR_PE_2(config-if)#ip address 192.168.1.6 255.255.255.252
LSR_PE_2(config-if)#no shutdown
LSR_PE_2(config-if)#exit
LSR_PE_2(config)#
LSR_PE_2(config)#
LSR_PE_2(config)#router ospf 1
LSR_PE_2(config-router)#network 192.168.1.0 0.0.0.255 area 0
LSR_PE_2(config-router)#exit
LSR_PE_2(config)#
LSR_PE_2(config)#
*Mar  1 00:35:20.058: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.10.10 on
Serial0/1
from LOADING to FULL, Loading Done
LSR_PE_2(config)#
LSR_PE_2(config)#
LSR_PE_2(config)#ip cef
```

```
LSR_PE_2(config)#
LSR_PE_2(config)#
LSR_PE_2(config)#mpls ip
LSR_PE_2(config)#
LSR_PE_2(config)#interface serial 0/1
LSR_PE_2(config-if)#mpls ip
LSR_PE_2(config-if)#exit
LSR_PE_2(config)#
LSR_PE_2(config)#
LSR_PE_2(config)#interface loopback 0
LSR_PE_2(config-if)#ip address 192.168.1.102 255.255.255.255
LSR_PE_2(config-if)#no shutdown
LSR_PE_2(config-if)#exit
LSR_PE_2(config)#
LSR_PE_2(config)#
LSR_PE_2(config)#mpls label protocol ldp
LSR_PE_2(config)#
LSR_PE_2(config)#
LSR_PE_2(config)#mpls ldp router-id loopback 0
LSR_PE_2(config)#
LSR_PE_2(config)#exit
LSR_PE_2#
*Mar  1 00:37:44.523: %SYS-5-CONFIG_I: Configured from console by console
LSR_PE_2#
LSR_PE_2#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]

LSR_PE_2#sh run
```

CAPITULO II

CONFIGURACIÓN DE VPNs SOBRE MPLS.

RESULTADOS DE LA CONFIGURACIÓN.

Anexo 2.2.3.a: Configuración Router_P

ROUTER Router_P (Cisco 3640)

```
Router>
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname Router_P
Router_P(config)#enable password p
Router_P(config)#line console 0
Router_P(config-line)#password p
Router_P(config-line)#login
Router_P(config-line)#exit
Router_P(config)#line vty 0 4
Router_P(config-line)#password p
Router_P(config-line)#login
Router_P(config-line)#exit
Router_P(config)#
Router_P(config)#
Router_P(config)#interface serial 1/0
Router_P(config-if)#ip address 10.1.13.2 255.255.255.0
Router_P(config-if)#mpls ip
Router_P(config-if)#no shutdown
Router_P(config-if)#exit
Router_P(config)#
Router_P(config)#
Router_P(config)#
*Mar  1 00:04:39.923: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
*Mar  1 00:04:40.923: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial1/0, changed state to up
Router_P(config)#
Router_P(config)#interface serial 1/1
Router_P(config-if)#ip address 10.1.23.2 255.255.255.0
Router_P(config-if)#clock rate 56000
Router_P(config-if)#mpls ip
Router_P(config-if)#no shutdown
Router_P(config-if)#
*Mar  1 00:05:28.591: %LINK-3-UPDOWN: Interface Serial1/1, changed state to up
*Mar  1 00:05:29.591: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial1/1, changed state to up
Router_P(config-if)#exit
Router_P(config)#
Router_P(config)#
Router_P(config)#router ospf 1
Router_P(config-router)#network 10.1.0.0 0.0.255.255 area 0
Router_P(config-router)#exit
Router_P(config)#
Router_P(config)#
Router_P(config)#ip cef
Router_P(config)#mpls ip
Router_P(config)#
Router_P(config)#
Router_P(config)#interface loopback 0
```

```
Router_P(config-if)#ip address 10.1.1.3 255.255.255.255
Router_P(config-if)#exit
Router_P(config)#exit
Router_P(config)#
Router_P(config)#
Router_P(config)#mpls label protocol ldp
Router_P(config)#
Router_P(config)#
Router_P(config)#mpls ldp router-id loopback 0
Router_P(config)#
Router_P(config)#
Router_P#wr
Building configuration...
Router_P(config)#
```

Anexo 2.2.3.b: Configuración: Router_PE1

ROUTER Router_PE1 (Cisco 2610)

```
Router>
Router>enable
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname Router_PE1
Router_PE1(config)#enable password pe1
Router_PE1(config)#line console 0
Router_PE1(config-line)#password pe1
Router_PE1(config-line)#login
Router_PE1(config-line)#exit
Router_PE1(config)#line vty 0 4
Router_PE1(config-line)#password pe1
Router_PE1(config-line)#login
Router_PE1(config-line)#exit
Router_PE1(config)#
Router_PE1(config)#
Router_PE1(config)#interface serial 0/0
Router_PE1(config-if)#ip address 10.1.13.1 255.255.255.0
Router_PE1(config-if)#clock rate 56000
Router_PE1(config-if)#mpls ip
Router_PE1(config-if)#no shutdown
Router_PE1(config-if)#exit
Router_PE1(config)#interface ethernet 0/0
Router_PE1(config-if)#ip address 10.1.14.1 255.255.255.0
Router_PE1(config-if)#no shutdown
Router_PE1(config-if)#exit
Router_PE1(config)#
Router_PE1(config)#
Router_PE1(config)#interface loopback 0
Router_PE1(config-if)#ip address 10.1.1.1 255.255.255.255
Router_PE1(config-if)#exit
Router_PE1(config)#
Router_PE1(config)#
Router_PE1(config)#ip cef
Router_PE1(config)#mpls ip
Router_PE1(config)#
Router_PE1(config)#
Router_PE1(config)#ip vrf cliente_a
Router_PE1(config-vrf)#rd 100:110
Router_PE1(config-vrf)#route-target export 100:1000
Router_PE1(config-vrf)#route-target import 100:1000
Router_PE1(config-vrf)#exit
```

```
Router_PE1(config)#
Router_PE1(config)#
Router_PE1(config)#ip vrf cliente_b
Router_PE1(config-vrf)#rd 100:120
Router_PE1(config-vrf)#route-target export 100:2000
Router_PE1(config-vrf)#route-target import 100:2000
Router_PE1(config-vrf)#exit
Router_PE1(config)#
Router_PE1(config)#
Router_PE1(config)#interface loopback 10
Router_PE1(config-if)#ip vrf forwarding cliente_a
Router_PE1(config-if)#ip address 10.10.1.1 255.255.255.0
Router_PE1(config-if)#exit
Router_PE1(config)#
Router_PE1(config)#
Router_PE1(config)#interface loopback 20
Router_PE1(config-if)#ip vrf forwarding cliente_b
Router_PE1(config-if)#ip address 10.20.1.1 255.255.255.0
Router_PE1(config-if)#exit
Router_PE1(config)#
Router_PE1(config)#
Router_PE1(config)#ip cef
Router_PE1(config)#mpls ip
Router_PE1(config)#mpls label protocol ldp
Router_PE1(config)#mpls ldp router-id loopback 0
Router_PE1(config)#
Router_PE1(config)#
Router_PE1(config)#router ospf 1
Router_PE1(config-router)#network 10.1.0.0 0.0.255.255 area 0
Router_PE1(config-router)#exit
Router_PE1(config)#
Router_PE1(config)#
Router_PE1(config)#router bgp 100
Router_PE1(config-router)#no bgp default ipv4-unicast
Router_PE1(config-router)#redistribute connected
Router_PE1(config-router)#neighbor 10.1.1.2 remote-as 100
Router_PE1(config-router)#neighbor 10.1.1.2 update-source loopback 0
Router_PE1(config)#
Router_PE1(config)#
Router_PE1(config-router)#address-family vpnv4
Router_PE1(config-router-af)#neighbor 10.1.1.2 activate
Router_PE1(config-router-af)#neighbor 10.1.1.2 route-reflector-client
Router_PE1(config-router-af)#neighbor 10.1.1.2 send-community extended
Router_PE1(config-router-af)#exit-address-family
Router_PE1(config)#
Router_PE1(config)#
Router_PE1(config-router)#address-family ipv4 vrf cliente_a
Router_PE1(config-router-af)#redistribute connected
Router_PE1(config-router-af)#exit-address-family
Router_PE1(config)#
Router_PE1(config)#
Router_PE1(config-router)#address-family ipv4 vrf cliente_b
Router_PE1(config-router-af)#redistribute connected
Router_PE1(config-router-af)#exit-address-family
Router_PE1(config-router)#exit
Router_PE1(config)#exit
Router_PE1#wr
Building configuration...
[OK]
Router_PE1#
```

Anexo 2.2.3.c: Configuración: Router_PE2

ROUTER Router_PE2 (Cisco 2610)

```
Router>
Router>enable
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname Router_PE2
Router_PE2(config)#enable password pe2
Router_PE2(config)#line console 0
Router_PE2(config-line)#password pe2
Router_PE2(config-line)#login
Router_PE2(config-line)#exit
Router_PE2(config)#line vty 0 4
Router_PE2(config-line)#password pe2
Router_PE2(config-line)#login
Router_PE2(config-line)#exit
Router_PE2(config)#
Router_PE2(config)#
Router_PE2(config)#interface serial 0/1
Router_PE2(config-if)#ip address 10.1.23.1 255.255.255.0
Router_PE2(config-if)#mpls ip
Router_PE2(config-if)#no shutdown
Router_PE2(config-if)#exit
Router_PE2(config)#
Router_PE2(config)#
Router_PE2(config)#interface loopback 0
Router_PE2(config-if)#ip address 10.1.1.2 255.255.255.255
Router_PE2(config-if)#exit
Router_PE2(config)#
Router_PE2(config)#
Router_PE2(config)#ip cef
Router_PE2(config)#mpls ip
Router_PE2(config)#
Router_PE2(config)#
Router_PE2(config)#ip vrf cliente_a
Router_PE2(config-vrf)#rd 100:110
Router_PE2(config-vrf)#route-target export 100:1000
Router_PE2(config-vrf)#route-target import 100:1000
Router_PE2(config-vrf)#exit
Router_PE2(config)#
Router_PE2(config)#
Router_PE2(config)#ip vrf cliente_b
Router_PE2(config-vrf)#rd 100:120
Router_PE2(config-vrf)#route-target export 100:2000
Router_PE2(config-vrf)#route-target import 100:2000
Router_PE2(config-vrf)#exit
Router_PE2(config)#
Router_PE2(config)#
Router_PE2(config)#interface loopback 10
Router_PE2(config-if)#ip vrf forwarding cliente_a
Router_PE2(config-if)#ip address 10.10.2.1 255.255.255.0
Router_PE2(config-if)#exit
Router_PE2(config)#
Router_PE2(config)#
Router_PE2(config)#interface loopback 20
Router_PE2(config-if)#ip vrf forwarding cliente_b
Router_PE2(config-if)#ip address 10.20.2.1 255.255.255.0
Router_PE2(config-if)#exit
Router_PE2(config)#
Router_PE2(config)#
Router_PE2(config)#ip cef
Router_PE2(config)#mpls ip
Router_PE2(config)#mpls label protocol ldp
```

```
Router_PE2(config)#mpls ldp router-id loopback 0
Router_PE2(config)#
Router_PE2(config)#
Router_PE2(config)#router ospf 1
Router_PE2(config-router)#network 10.1.0.0 0.0.255.255 area 0
Router_PE2(config-router)#exit
Router_PE2(config)#
Router_PE2(config)#
Router_PE2(config)#router bgp 100
Router_PE2(config-router)#no bgp default ipv4-unicast
Router_PE2(config-router)#redistribute connected
Router_PE2(config-router)#neighbor 10.1.1.1 remote-as 100
Router_PE2(config-router)#neighbor 10.1.1.1 update-source loopback 0
Router_PE2(config)#
Router_PE2(config)#
Router_PE2(config-router)#address-family vpnv4
Router_PE2(config-router-af)#neighbor 10.1.1.1 activate
Router_PE2(config-router-af)#neighbor 10.1.1.1 route-reflector-client
Router_PE2(config-router-af)#neighbor 10.1.1.1 send-community extended
Router_PE2(config-router-af)#exit-address-family
Router_PE2(config)#
Router_PE2(config)#
Router_PE2(config-router)#address-family ipv4 vrf cliente_a
Router_PE2(config-router-af)#redistribute connected
Router_PE2(config-router-af)#exit-address-family
Router_PE2(config)#
Router_PE2(config)#
Router_PE2(config-router)#address-family ipv4 vrf cliente_b
Router_PE2(config-router-af)#redistribute connected
Router_PE2(config-router-af)#exit-address-family
Router_PE2(config-router)#exit
Router_PE2(config)#exit
Router_PE2#wr
Building configuration...
[OK]
Router_PE2#
```

CAPITULO II

CONFIGURACIÓN DE CoS PARA BRINDAR QoS A LA RED MPLS.

RESULTADOS DE LA CONFIGURACIÓN.

Anexo 2.3.3.a: Configuración P_QoS

ROUTER P_QoS (Cisco 3640)

```
Router>
Router>enable
Router#configure terminal
Router(config)#
Router(config)#hostname P_QoS
P_QoS(config)#
P_QoS(config)#enable password p
P_QoS(config)#line console 0
P_QoS(config-line)#password p
P_QoS(config-line)#login
P_QoS(config-line)#exit
P_QoS(config)#line vty 0 4
P_QoS(config-line)#password p
P_QoS(config-line)#login
P_QoS(config-line)#exit
P_QoS(config)#
P_QoS(config)#
P_QoS(config)#interface serial 1/0
P_QoS(config-if)#ip address 192.168.30.1 255.255.255.0
P_QoS(config-if)#mpls ip
P_QoS(config-if)#no shutdown
P_QoS(config-if)#exit
P_QoS(config)#
P_QoS(config)#interface serial 1/1
P_QoS(config-if)#ip address 192.168.10.2 255.255.255.0
P_QoS(config-if)#mpls ip
P_QoS(config-if)#no shutdown
P_QoS(config-if)#exit
P_QoS(config)#
P_QoS(config)#interface loopback 0
P_QoS(config-if)#ip address 192.168.1.3 255.255.255.255
P_QoS(config-if)#exit
P_QoS(config)#
P_QoS(config)#
P_QoS(config)#ip cef
P_QoS(config)#mpls ip
P_QoS(config)#
P_QoS(config)#mpls label protocol ldp
P_QoS(config)#mpls ldp router-id loopback 0
P_QoS(config)#
P_QoS(config)#
P_QoS(config)#router ospf 1
P_QoS(config-router)#network 192.168.0.0 255.255.0.0 area 0
P_QoS(config-router)#log-adjacency-changes
P_QoS(config-router)#exit
P_QoS(config)#exit
P_QoS#wr
Building configuration...
[OK]
P_QoS#
```

Anexo 2.3.3.b: Configuración: PE1_QoS

ROUTER PE1_QoS (Cisco 2610)

```
Router>
Router>enable
Router#configure terminal
Router(config)#
Router(config)#hostname PE1_QoS
PE1_QoS(config)#enable password pe1
PE1_QoS(config)#line console 0
PE1_QoS(config-line)#password pe1
PE1_QoS(config-line)#login
PE1_QoS(config-line)#exit
PE1_QoS(config)#line vty 0 4
PE1_QoS(config-line)#password pe1
PE1_QoS(config-line)#login
PE1_QoS(config-line)#exit
PE1_QoS(config)#
PE1_QoS(config)#
PE1_QoS(config)#interface serial 0/0
PE1_QoS(config-if)#ip address 192.168.30.2 255.255.255.0
PE1_QoS(config-if)#clock rate 56000
PE1_QoS(config-if)#mpls ip
PE1_QoS(config-if)#no shutdown
PE1_QoS(config-if)#exit
PE1_QoS(config)#
PE1_QoS(config)#
PE1_QoS(config)#interface loopback 0
PE1_QoS(config-if)#ip address 192.168.1.2 255.255.255.255
PE1_QoS(config-if)#exit
PE1_QoS(config)#
PE1_QoS(config)#interface ethernet 0/0
PE1_QoS(config-if)#ip address 192.168.40.1 255.255.255.0
PE1_QoS(config-if)#no shutdown
PE1_QoS(config-if)#exit
PE1_QoS(config)#
PE1_QoS(config)#
PE1_QoS(config)#ip cef
PE1_QoS(config)#mpls ip
PE1_QoS(config)#mpls label protocol ldp
PE1_QoS(config)#mpls ldp router-id loopback 0
PE1_QoS(config)#
PE1_QoS(config)#
PE1_QoS(config)#router ospf 1
PE1_QoS(config-router)#network 192.168.0.0 0.0.255.255 area 0
PE1_QoS(config-router)#exit
PE1_QoS(config)#
PE1_QoS(config)#access-list rate-limit 24 4
PE1_QoS(config)#interface ethernet 0/0
PE1_QoS(config-if)# rate-limit input access-group rate-limit 24 8000 8000 8000
conform-action set-mpls-exp-imposition-transmit 4 exceed-action set-mpls-exp-
imposition-transmit 0
PE1_QoS#
```

Anexo 2.3.3.c: Configuración: PE2_QoS

ROUTER PE2_QoS (Cisco 2610)

```
Router>
```

```
Router>enable
Router#configure  terminal
Router(config)#hostname PE2_QoS
PE2_QoS(config)#enable password pe2
PE2_QoS(config)#line console 0
PE2_QoS(config-line)#password pe2
PE2_QoS(config-line)#login
PE2_QoS(config-line)#exit
PE2_QoS(config)#line vty 0 4
PE2_QoS(config-line)#password pe2
PE2_QoS(config-line)#login
PE2_QoS(config-line)#exit
PE2_QoS(config)#
PE2_QoS(config)#
PE2_QoS(config)#interface serial 0/1
PE2_QoS(config-if)#ip address 192.168.10.2 255.255.255.0
PE2_QoS(config-if)#no shutdown
PE2_QoS(config-if)#exit
PE2_QoS(config)#
PE2_QoS(config)#
PE2_QoS(config)#interface loopback 0
PE2_QoS(config-if)#ip address 192.168.1.1 255.255.255.255
PE2_QoS(config-if)#exit
PE2_QoS(config)#
PE2_QoS(config)#ip cef
PE2_QoS(config)#mpls ip
PE2_QoS(config)#
PE2_QoS(config)#interface serial 0/1
PE2_QoS(config-if)#mpls ip
PE2_QoS(config-if)#exit
PE2_QoS(config)#
PE2_QoS(config)#
PE2_QoS(config)#mpls label protocol ldp
PE2_QoS(config)#mpls ldp router-id loopback 0
PE2_QoS(config)#
PE2_QoS(config)#router ospf 1
PE2_QoS(config-router)#network 192.168.0.0 0.0.255.255 area 0
PE2_QoS(config-router)#log-adjacency-changes
PE2_QoS(config-router)#exit
PE2_QoS(config)#exit
PE2_QoS#
```

CAPITULO III

CASO DE ESTUDIO 1

Anexo 3.1.a: Configuración KIMERA

KIMERA

```
Press Enter to Start

Router>
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

* Configuración del nombre del router:
Router(config)#hostname Kimera

* Configuración de claves de acceso:
Kimera(config)#enable password kimera123
Kimera(config)#line console 0
Kimera(config-line)#password kimera
Kimera(config-line)#login
Kimera(config-line)#exit
Kimera(config)#line vty 0 4
Kimera(config-line)#password kimera
Kimera(config-line)#login
Kimera(config-line)#exit
* Nótese la importancia de configurar una clave para el
* acceso mediante telnet e introducir el comando login,
* caso contrario este acceso sería imposible.

* Configuración de las interfaces:
Kimera(config)#interface serial 0
Kimera(config-if)#ip address 10.0.1.1 255.255.0.0
Kimera(config-if)#no shutdown
Kimera(config-if)#exit
Kimera(config)#exit

* Es necesario grabar la configuración en la memoria NVRAM
* del router para evitar contratiempos:
Kimera#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
```

Anexo 3.1.b: Configuración: TEKRA

TEKRA

```
Router>
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

* Configuración del nombre del router:
```

```
Router(config)#hostname Tekra

* Configuración de claves de acceso:
Tekra(config)#enable password tekra123
Tekra(config)#line console 0
Tekra(config-line)#password tekra
Tekra(config-line)#login
Tekra(config-line)#exit
Tekra(config)#line vty 0 4
Tekra(config-line)#password tekra
Tekra(config-line)#login
Tekra(config-line)#exit

* Configuración de las interfaces (nótese que el router
* tekra es el DCE por eso se debe configurar en éste el
* reloj):
Tekra(config)#interface serial 0
Tekra(config-if)#ip address 10.0.1.2 255.255.0.0
Tekra(config-if)#clock rate 56000
Tekra(config-if)#no shutdown
Tekra(config-if)#exit
Tekra(config)#exit

* Es necesario grabar la configuración en la memoria NVRAM
* del router para evitar contratiempos:
Tekra#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
```

CAPITULO III

CASO DE ESTUDIO 2

Anexo 3.2.a: Configuración ROCAFUERTE

ROCAFUERTE

```
* Configuración inicial:
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname Rocafuerte
Rocafuerte(config)#enable password rocafuerte123
Rocafuerte(config)#line console 0
Rocafuerte(config-line)#password rocafuerte
Rocafuerte(config-line)#login
Rocafuerte(config-line)#exit
Rocafuerte(config)#line vty 0 4
Rocafuerte(config-line)#password rocafuerte
Rocafuerte(config-line)#login
Rocafuerte(config-line)#exit
Rocafuerte(config)#interface serial 0
Rocafuerte(config-if)#ip address 192.168.10.1 255.255.255.0
Rocafuerte(config-if)#clock rate 56000
Rocafuerte(config-if)#no shutdown
%LINK-3-UPDOWN: Interface Serial0, changed state to up
%LINK-3-UPDOWN: Interface Serial0, changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed state to down
Rocafuerte(config-if)#exit
Rocafuerte(config)#interface ethernet 0
Rocafuerte(config-if)#ip address 10.0.0.1 255.0.0.0
Rocafuerte(config-if)#no shutdown
%LINK-3-UPDOWN: Interface Ethernet0, changed state to up
Rocafuerte(config-if)#exit

* Configuración del protocolo de enrutamiento:
Rocafuerte(config)#router rip
Rocafuerte(config-router)#version 2
Rocafuerte(config-router)#network 192.168.10.0
Rocafuerte(config-router)#network 10.0.0.0
Rocafuerte(config-router)#exit
Rocafuerte(config)#exit

* Es necesario grabar la configuración en la memoria NVRAM
* del router para evitar contratiempos:
Rocafuerte#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
```

Anexo 3.2.b: Configuración ROLDOS

ROLDOS

```
* Configuración inicial:
Router>enable
```

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname Roldos
Roldos(config)#enable password roldos123
Roldos(config)#line console 0
Roldos(config-line)#password roldos
Roldos(config-line)#login
Roldos(config-line)#exit
Roldos(config)#line vty 0 4
Roldos(config-line)#password roldos
Roldos(config-line)#login
Roldos(config-line)#exit
Roldos(config)#interface serial 0
Roldos(config-if)#ip address 192.168.10.2 255.255.255.0
Roldos(config-if)#no shutdown
%LINK-3-UPDOWN: Interface Serial0, changed state to up
Roldos(config-if)#exit
Roldos(config)#interface ethernet 0
Roldos(config-if)#ip address 172.20.0.1 255.255.0.0
Roldos(config-if)#no shutdown
%LINK-3-UPDOWN: Interface Ethernet0, changed state to up
Roldos(config-if)#exit

* Configuración del protocolo de enrutamiento:
Roldos(config)#router rip
Roldos(config-router)#version 2
Roldos(config-router)#network 192.168.10.0
Roldos(config-router)#network 172.20.0.0
Roldos(config-router)#exit
Roldos(config)#exit

* Es necesario grabar la configuración en la memoria NVRAM
* del router para evitar contratiempos:
Roldos#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
```

CAPITULO III

CASO DE ESTUDIO 3

Anexo 3.3.a: Configuración MATRIZ

MATRIZ

```
Router>
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
Router(config)#hostname Matriz
Matriz(config)#enable password matriz123
Matriz(config)#line console 0
Matriz(config-line)#password matriz
Matriz(config-line)#login
Matriz(config-line)#exit
Matriz(config)#line vty 0 4
Matriz(config-line)#password matriz
Matriz(config-line)#login
Matriz(config-line)#exit
Matriz(config)#
Matriz(config)#interface serial 0
Matriz(config-if)#ip address 192.168.10.1 255.255.255.252
Matriz(config-if)#no shutdown
Matriz(config-if)#exit
Matriz(config)#
Matriz(config)#
*Mar  1 00:39:52.891: %LINK-3-UPDOWN: Interface Serial0/0, changed state to
down
Matriz(config)#
Matriz(config)#interface ethernet 0/0
Matriz(config-if)#ip add 10.0.0.1 255.0.0.0
Matriz(config-if)#no shutdown
Matriz(config-if)#exit
Matriz(config)#
Matriz(config)#router ospf 1
Matriz(config-router)#network 192.168.10.0 0.0.0.255 area 0
Matriz(config-router)#network 10.0.0.0 0.255.255.255 area 0
Matriz(config-router)#log-adjacency-changes
Matriz(config-router)#exit
Matriz(config)#exit
Matriz#wr
Building configuration...
[OK]
Matriz#
```

Anexo 3.3.b: Configuración POP 1

POP 1

```
Router>
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname Pop_1
Pop_1(config)#enable password pop1123
```

```
Pop_1(config)#line console 0
Pop_1(config-line)#password pop1
Pop_1(config-line)#login
Pop_1(config-line)#
Pop_1(config)#line vty 0 5
Pop_1(config-line)#password pop1
Pop_1(config-line)#login
Pop_1(config-line)#exit
Pop_1(config)#
Pop_1(config)#
Pop_1(config)#interface serial 0/0
Pop_1(config-if)#ip address 192.168.10.2 255.255.255.252
Pop_1(config-if)#clock rate 56000
Pop_1(config-if)#no shutdown
Pop_1(config-if)#exit
Pop_1(config)#
*Mar  1 00:22:42.829: %LINK-3-UPDOWN: Interface Serial0/0, changed state to
down
Pop_1(config)#
Pop_1(config)#interface serial 0/1
Pop_1(config-if)#ip address 192.168.10.5 255.255.255.252
Pop_1(config-if)#clock rate 56000
Pop_1(config-if)#no shutdown
Pop_1(config-if)#exit
Pop_1(config)#
*Mar  1 00:23:10.903: %LINK-3-UPDOWN: Interface Serial0/1, changed state to up
*Mar  1 00:23:11.905: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial0/1, changed state to up
Pop_1(config)#
Pop_1(config)#router ospf 1
Pop_1(config-router)#network 192.168.10.0 0.0.0.255 area 0
Pop_1(config-router)#log-adjacency-changes
Pop_1(config-router)#exit
Pop_1(config)#
Pop_1(config)#router bgp 1
Pop_1(config-router)#network 192.168.10.0
Pop_1(config-router)#network 192.168.10.4
Pop_1(config-router)#redistribute ospf 1
Pop_1(config-router)#neighbor 192.168.10.10 remote-as 1
Pop_1(config-router)#exit
Pop_1(config)#exit
Pop_1#wr
Building configuration...
[OK]
Pop_1#
```

Anexo 3.3.c: Configuración CORE

CORE

```
Router>
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname Core
Core(config)#enable password core123
Core(config)#line console 0
Core(config-line)#password core
Core(config-line)#login
Core(config-line)#exit
Core(config)#line vty 0 4
```

```
Core(config-line)#password core
Core(config-line)#login
Core(config-line)#exit
Core(config)#
Core(config)#
Core(config)#interface serial 1/0
Core(config-if)#ip address 192.168.10.6 255.255.255.252
Core(config-if)#no shutdown
Core(config-if)#exit
Core(config)#
Core(config)#
*Mar  1 00:17:57.403: %LINK-3-UPDOWN: Interface Serial1/0, changed state to down
Core(config)#interface serial 1/1
Core(config-if)#ip address 192.168.10.9 255.255.255.252
Core(config-if)#clock rate 56000
Core(config-if)#no shutdown
Core(config-if)#exit
Core(config)#
Core(config)#router ospf 1
Core(config-router)#network 192.168.10.0 0.0.0.255 area 0
Core(config-router)#log-adjacency-changes
Core(config-router)#exit
Core(config)#exit
*Mar  1 00:18:32.891: %LINK-3-UPDOWN: Interface Serial1/1, changed state to up
*Mar  1 00:18:33.891: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1,
changed state to up
```

Anexo 3.3.d: Configuración POP 2

POP 2

```
Router>
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname Pop_2
Pop_2(config)#enable password pop2123
Pop_2(config)#line console 0
Pop_2(config-line)#password pop2
Pop_2(config-line)#login
Pop_2(config-line)#exit
Pop_2(config)#line vty 0 4
Pop_2(config-line)#password pop2
Pop_2(config-line)#login
Pop_2(config-line)#exit
Pop_2(config)#
Pop_2(config)#
Pop_2(config)#interface serial 0/0
Pop_2(config-if)#ip address 192.168.10.13 255.255.255.252
Pop_2(config-if)#no shutdown
Pop_2(config-if)#exit
*Mar  1 00:32:42.961: %LINK-3-UPDOWN: Interface Serial0/0, changed state to down
Pop_2(config)#interface serial 0/1
Pop_2(config-if)#ip address 192.168.10.10 255.255.255.252
Pop_2(config-if)#no shutdown
Pop_2(config-if)#exit
Pop_2(config)#
Pop_2(config)#router ospf 1
Pop_2(config-router)#network 192.168.10.0 0.0.0.255 area 0
Pop_2(config-router)#log-adjacency-changes
Pop_2(config-router)#
Pop_2(config)#router bgp 1
```

```
Pop_2(config-router)#network 192.168.10.8
Pop_2(config-router)#network 192.168.10.12
Pop_2(config-router)#neighbor 192.169.10.5 remote-as 1
Pop_2(config-router)#redistribute ospf 1
Pop_2(config-router)#exit
Pop_2(config)#exit
Pop_2#wr
Building configuration...
*Mar  1 00:33:50.688: %SYS-5-CONFIG_I: Configured from console by console
*Mar  1 00:33:54.334: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.10.9 on Serial0/1 f
rom LOADING to FULL, Loading Done
```

Anexo 3.3.e: Configuración SUCURSAL

SUCURSAL

```
Router>
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname Sucursal
Sucursal(config)#enable password sucursal123
Sucursal(config)#line console 0
Sucursal(config-line)#password sucursal
Sucursal(config-line)#login
Sucursal(config-line)#exit
Sucursal(config)#line vty 0 4
Sucursal(config-line)#password sucursal
Sucursal(config-line)#login
Sucursal(config-line)#exit
Sucursal(config)#
Sucursal(config)#
Sucursal(config)#interface serial 0/0
Sucursal(config-if)#ip add 192.168.10.14 255.255.255.252
Sucursal(config-if)#clock rate 56000
Sucursal(config-if)#no shutdown
Sucursal(config-if)#exit
Sucursal(config)#
*Mar  1 00:04:36.947: %LINK-3-UPDOWN: Interface Serial0/0, changed state to down
Sucursal(config)#
Sucursal(config)#interface loopback 0
Sucursal(config-if)#ip add 172.20.0.1 255.255.0.0
Sucursal(config-if)#exit
Sucursal(config-if)#
*Mar  1 00:05:36.611: %LINK-3-UPDOWN: Interface Ethernet0/0, changed state to up
Sucursal(config-if)#
Sucursal(config-if)#exit
Sucursal(config)#
Sucursal(config)#router ospf 1
Sucursal(config-router)#network 192.168.10.0 0.0.0.255 area 0
Sucursal(config-router)#network 172.20.0.0 0.0.255.255 area 0
Sucursal(config-router)#log-adjacency-changes
Sucursal(config-router)#exit
Sucursal(config)#exit
Sucursal#
Sucursal#wr
Building configuration...
*Mar  1 00:06:39.627: %SYS-5-CONFIG_I: Configured from console by console[OK]
Sucursal#
```

CAPITULO III

CASO DE ESTUDIO 5

Anexo 3.5.a: Configuración Pop_de_LojaNet

Pop_de_LojaNet

```
Router>
Router>enable
Router#configure terminal
Router(config)#hostname Pop_de_LojaNet
Pop_de_LojaNet(config)#enable password pop123
Pop_de_LojaNet(config)#line console 0
Pop_de_LojaNet(config-line)#password poploja
Pop_de_LojaNet(config-line)#login
Pop_de_LojaNet(config-line)#exit
Pop_de_LojaNet(config)#line vty 0 4
Pop_de_LojaNet(config-line)#password poploja
Pop_de_LojaNet(config-line)#login
Pop_de_LojaNet(config-line)#exit
Pop_de_LojaNet(config)#
Pop_de_LojaNet(config)#
Pop_de_LojaNet(config)#ip cef
Pop_de_LojaNet(config)#mpls ip
Pop_de_LojaNet(config)#
Pop_de_LojaNet(config)#
Pop_de_LojaNet(config)#interface s0/0
Pop_de_LojaNet(config-if)#ip address 192.168.22.1 255.255.255.252
Pop_de_LojaNet(config-if)#clock rate 56000
Pop_de_LojaNet(config-if)#mpls ip
Pop_de_LojaNet(config-if)#no shutdown
Pop_de_LojaNet(config-if)#
Pop_de_LojaNet(config-if)#
*Mar  1 00:41:45.947: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
*Mar  1 00:41:46.949: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial0/0,
 changed state to up
Pop_de_LojaNet(config-if)#
Pop_de_LojaNet(config-if)#exit
Pop_de_LojaNet(config)#interface s0/1
Pop_de_LojaNet(config-if)#ip address 192.168.21.2 255.255.255.252
Pop_de_LojaNet(config-if)#no shutdown
Pop_de_LojaNet(config-if)#
*Mar  1 00:42:55.585: %LINK-3-UPDOWN: Interface Serial0/1, changed state to up
*Mar  1 00:42:56.587: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial0/1,
 changed state to up
Pop_de_LojaNet(config-if)#
Pop_de_LojaNet(config-if)#
Pop_de_LojaNet(config-if)#exit
Pop_de_LojaNet(config)#
Pop_de_LojaNet(config)#
Pop_de_LojaNet(config)#router ospf 1
Pop_de_LojaNet(config-router)#network 192.168.0.0 0.0.255.255 area 0
Pop_de_LojaNet(config-router)#exit
Pop_de_LojaNet(config)#
Pop_de_LojaNet(config-router)#
Pop_de_LojaNet(config-router)#exit
Pop_de_LojaNet(config)#exit
Pop_de_LojaNet#
```

CAPITULO III

CASO DE ESTUDIO 6

Anexo 3.6.a: Configuración Core

Core

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname Core
Core(config)#enable password core123
Core(config)#line console 0
Core(config-line)#password core
Core(config-line)#login
Core(config-line)#exit
Core(config)#line vty 0 4
Core(config-line)#password core
Core(config-line)#login
Core(config-line)#exit
Core(config)#
Core(config)#
Core(config)#interface serial 1/0
Core(config-if)#ip address 192.168.22.2 255.255.255.252
Core(config-if)#mpls ip
Core(config-if)#no shutdown
Core(config-if)#exit
*Mar  1 00:14:38.331: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
*Mar  1 00:14:39.331: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial1/0,
 changed state to up
Core(config)#
Core(config)#interface loopback 0
Core(config-if)#ip address 192.168.23.1 255.255.255.255
Core(config-if)#exit
Core(config)#
Core(config)#mpls ip
Core(config)#
Core(config)#ip cef
Core(config)#
Core(config)#
Core(config)#router ospf 1
Core(config-router)#network 192.168.0.0 0.0.255.255 area 0
Core(config-router)#exit
Core(config)#exit
Core#wr
Core#
```

CAPITULO III

CASO DE ESTUDIO 7

Anexo 3.7.a: Configuración Pop_1

Pop_1

```
Router>
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname Pop_1
Pop_1(config)#enable password pop123
Pop_1(config)#line console 0
Pop_1(config-line)#password pop1
Pop_1(config-line)#login
Pop_1(config-line)#exit
Pop_1(config)#line vty 0 4
Pop_1(config-line)#password pop1
Pop_1(config-line)#login
Pop_1(config-line)#exit
Pop_1(config)#
Pop_1(config)#
Pop_1(config)#ip cef
Pop_1(config)#
Pop_1(config)#mpls ip
Pop_1(config)#
Pop_1(config)#
Pop_1(config)#interface serial 0/0
Pop_1(config-if)#ip address 10.0.1.1 255.255.255.0
Pop_1(config-if)#clock rate 56000
Pop_1(config-if)#mpls ip
Pop_1(config-if)#no shutdown
Pop_1(config-if)#
Pop_1(config-if)#exit
Pop_1(config)#
Pop_1(config)#
Pop_1(config)#interface ethernet 0/0
Pop_1(config-if)#ip address 10.0.3.1 255.255.255.0
Pop_1(config-if)#no shutdown
Pop_1(config-if)#
Pop_1(config-if)#
Pop_1(config-if)#exit
Pop_1(config)#
Pop_1(config)#
Pop_1(config)#interface loopback 0
Pop_1(config-if)#ip address 10.0.10.1 255.255.255.255
Pop_1(config-if)#exit
Pop_1(config)#
Pop_1(config)#
Pop_1(config)#mpls label protocol ldp
Pop_1(config)#
Pop_1(config)#
Pop_1(config)#mpls ldp router-id loopback 0
Pop_1(config)#
Pop_1(config)#
Pop_1(config)#router ospf 1
Pop_1(config-router)#network 10.0.0.0 0.0.255.255 area 0
Pop_1(config-router)#exit
Pop_1(config)#
Pop_1(config)#exit
```

```
Pop_1#
Pop_1#wr
```

Anexo 3.7.b: Configuración Pop_2

Pop_2

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname Pop_2
Pop_2(config)#enable password pop123
Pop_2(config)#line console 0
Pop_2(config-line)#password pop2
Pop_2(config-line)#login
Pop_2(config-line)#exit
Pop_2(config)#line vty 0 4
Pop_2(config-line)#password pop2
Pop_2(config-line)#login
Pop_2(config-line)#exit
Pop_2(config)#
Pop_2(config)#
Pop_2(config)#ip cef
Pop_2(config)#
Pop_2(config)#
Pop_2(config)#mpls ip
Pop_2(config)#
Pop_2(config)#
Pop_2(config)#interface serial 0/1
Pop_2(config-if)#ip address 10.0.2.2 255.255.255.0
Pop_2(config-if)#mpls ip
Pop_2(config-if)#no shutdown
Pop_2(config-if)#
Pop_2(config-if)#
Pop_2(config-if)#
Pop_2(config)#interface loopback 0
Pop_2(config-if)#ip address 10.0.10.3 255.255.255.255
Pop_2(config-if)#exit
Pop_2(config)#
Pop_2(config)#
Pop_2(config)#interface loopback 20
Pop_2(config-if)#ip address 10.0.4.1 255.255.255.255
Pop_2(config-if)#exit
Pop_2(config)#
Pop_2(config)#
Pop_2(config)#mpls label protocol ldp
Pop_2(config)#mpls ldp router-id loopback 0
Pop_2(config)#
Pop_2(config)#
Pop_2(config)#router ospf 1
Pop_2(config-router)#network 10.0.0.0 0.0.255.255 area 0
Pop_2(config-router)#exit
Pop_2(config)#
Pop_2#wr
Building configuration...
[OK]
Pop_2#
```

Anexo 3.7.c: Configuración Gonzanama

Gonzanama

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname Gonzanama
Gonzanama(config)#enable password gonza123
Gonzanama(config)#line console 0
Gonzanama(config-line)#password gonza
Gonzanama(config-line)#login
Gonzanama(config-line)#exit
Gonzanama(config)#line vty 0 4
Gonzanama(config-line)#password gonza
Gonzanama(config-line)#login
Gonzanama(config-line)#exit
Gonzanama(config)#
Gonzanama(config)#ip cef
Gonzanama(config)#mpls ip
Gonzanama(config)#
Gonzanama(config)#interface serial 1/0
Gonzanama(config-if)#ip address 10.0.1.2 255.255.255.0
Gonzanama(config-if)#mpls ip
Gonzanama(config-if)#no shutdown
Gonzanama(config-if)#exit
Gonzanama(config)#
Gonzanama(config)#interface serial 1/1
Gonzanama(config-if)#ip address 10.0.2.1 255.255.255.0
Gonzanama(config-if)#clock rate 56000
Gonzanama(config-if)#mpls ip
Gonzanama(config-if)#no shutdown
Gonzanama(config-if)#exit
Gonzanama(config)#
Gonzanama(config)#
Gonzanama(config)#interface loopback 0
Gonzanama(config-if)#ip address 10.0.10.2 255.255.255.255
Gonzanama(config-if)#exit
Gonzanama(config)#
Gonzanama(config)#mpls label protocol ldp
Gonzanama(config)#mpls ldp router-id loopback 0
Gonzanama(config)#
Gonzanama(config)#router ospf 1
Gonzanama(config-router)#network 10.0.0.0 0.0.255.255 area 0
Gonzanama(config-router)#exit
Gonzanama(config)#exit
Gonzanama#
Gonzanama#wr
Building configuration...
[OK]
Gonzanama#
```

CAPITULO III

CASO DE ESTUDIO 8

Anexo 3.8.a: Configuración Matriz

Matriz

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname Matriz
Matriz(config)#enable password matriz123
Matriz(config)#line console 0
Matriz(config-line)#password matriz
Matriz(config-line)#login
Matriz(config-line)#exit
Matriz(config)#line vty 0 4
Matriz(config-line)#password matriz
Matriz(config-line)#login
Matriz(config-line)#exit
Matriz(config)#
Matriz(config)#interface ethernet 0/0
Matriz(config-if)#ip address 10.0.1.1 255.255.255.0
Matriz(config-if)#no shutdown
Matriz(config-if)#
Matriz(config-if)#exit
*Mar  1 00:06:08.875: %LINK-3-UPDOWN: Interface Ethernet0/0, changed state to
up
*Mar  1 00:06:09.875: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Ethernet0/0, changed state to up
Matriz(config)#
Matriz(config)#interface serial 0/0
Matriz(config-if)#ip address 192.168.20.1 255.255.255.0
Matriz(config-if)#no shutdown
Matriz(config-if)#exit
Matriz(config)#
*Mar  1 00:08:14.235: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
*Mar  1 00:08:15.235: %LINEPROTO-5-UPDOWN: L changed state to up
Matriz(config)#
*Mar  1 00:08:36.867: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial0/0, changed state to down
Matriz(config)#
Matriz(config)#router ospf 1
Matriz(config-router)#network 192.168.0.0 0.0.255.255 area 0
Matriz(config-router)#network 10.0.1.0 0.0.0.255 area 0
Matriz(config-router)#log-adjacency-changes
Matriz(config-router)#exit
Matriz(config)#exit
Matriz#wr
Building configuration...
[OK]
Matriz#
```

Anexo 3.8.b: Configuración Sucursal

Sucursal

```
Router>
Router>enable
```

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname Sucursal
Sucursal(config)#enable password sucursal123
Sucursal(config)#line console 0
Sucursal(config-line)#password sucursal
Sucursal(config-line)#login
Sucursal(config-line)#exit
Sucursal(config)#line vty 0 4
Sucursal(config-line)#password sucursal
Sucursal(config-line)#login
Sucursal(config-line)#exit
Sucursal(config)#
Sucursal(config)#interface serial 0/0
Sucursal(config-if)#ip address 192.168.23.2 255.255.255.0
Sucursal(config-if)#clock rate 56000
Sucursal(config-if)#no shutdown
Sucursal(config-if)#exit
Sucursal(config)#
*Mar  1 00:40:38.411: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
*Mar  1 00:40:39.411: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial0/0, changed state to up
Sucursal(config)#
Sucursal(config)#interface loopback 30
Sucursal(config-if)#ip address 172.16.1.1 255.255.255.0
Sucursal(config-if)#exit
Sucursal(config)#
Sucursal(config)#router ospf 1
Sucursal(config-router)#network 192.168.0.0 0.0.255.255 area 0
Sucursal(config-router)#network 172.16.1.0 0.0.0.255 area 0
Sucursal(config-router)#log-adjacency-changes
Sucursal(config-router)#
Sucursal(config-router)#exit
Sucursal(config)#exit
Sucursal#wr
Building configuration...
[OK]
Sucursal#
```

Anexo 3.8.c: Configuración Plaza

Plaza

```
Router>
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname Plaza
Plaza(config)#enable password plaza123
Plaza(config)#line console 0
Plaza(config-line)#password plaza
Plaza(config-line)#login
Plaza(config-line)#exit
Plaza(config)#line vty 0 4
Plaza(config-line)#password plaza
Plaza(config-line)#login
Plaza(config-line)#exit
Plaza(config)#
Plaza(config)#interface serial 0/0
Plaza(config-if)#ip address 192.168.20.2 255.255.255.0
Plaza(config-if)#clock rate 56000
Plaza(config-if)#no shutdown
```

```
Plaza(config-if)#exit
Plaza(config)#
*Mar  1 00:15:56.046: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
*Mar  1 00:15:57.048: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial0/0, changed state to up
Plaza(config)#
Plaza(config)#interface serial 0/1
Plaza(config-if)#ip address 192.168.21.1 255.255.255.0
Plaza(config-if)#clock rate 56000
Plaza(config-if)#mpls ip
Plaza(config-if)#no shutdown
Plaza(config-if)#exit
Plaza(config)#
*Mar  1 00:16:27.117: %LINK-3-UPDOWN: Interface Serial0/1, changed state to up
*Mar  1 00:16:28.119: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial0/1, changed state to up
Plaza(config)#
Plaza(config)#interface loopback 0
Plaza(config-if)#ip address 192.168.10.1 255.255.255.255
Plaza(config-if)#exit
Plaza(config)#
Plaza(config)#ip cef
Plaza(config)#
Plaza(config)#mpls ip
Plaza(config)#mpls label protocol ldp
Plaza(config)#mpls ldp router-id loopback 0
Plaza(config)#
Plaza(config)#router ospf 1
Plaza(config-router)#network 192.168.0.0 0.0.255.255 area 0
Plaza(config-router)#log-adjacency-changes
Plaza(config-router)#exit
Plaza(config)#exit
Plaza#wr
Building configuration...
*Mar  1 00:18:18.003: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.20.1 on Serial0/0
f
rom LOADING to FULL, Loading Done
[OK]
Plaza#
```

Anexo 3.8.d: Configuración Jipiro

Jipiro

```
Router>
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname Jipiro
Jipiro(config)#enable password jipiro123
Jipiro(config)#line console 0
Jipiro(config-line)#password jipiro
Jipiro(config-line)#login
Jipiro(config-line)#exit
Jipiro(config)#line vty 0 4
Jipiro(config-line)#password jipiro
Jipiro(config-line)#login
Jipiro(config-line)#exit
Jipiro(config)#
Jipiro(config)#
Jipiro(config)#ip cef
Jipiro(config)#
```

```
Jipiro(config)#mpls ip
Jipiro(config)#
Jipiro(config)#interface serial 0/0
Jipiro(config-if)#ip address 192.168.23.1 255.255.255.0
Jipiro(config-if)#no shutdown
Jipiro(config-if)#exit
Jipiro(config)#
*Mar  1 00:32:58.174: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
*Mar  1 00:32:59.175: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial0/0, changed state to up
Jipiro(config)#interface serial 0/1
Jipiro(config-if)#ip address 192.168.22.2 255.255.255.0
Jipiro(config-if)#mpls ip
Jipiro(config-if)#no shutdown
Jipiro(config-if)#
*Mar  1 00:33:19.280: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial0/0, changed state to down
*Mar  1 00:33:19.280: %TDP-5-INFO: Default-IP-Routing-Table: TDP ID
removedexit
Jipiro(config)#
Jipiro(config)#interface loopback 0
Jipiro(config-if)#ip address 192.168.10.3 255.255.255.255
Jipiro(config-if)#exit
Jipiro(config)#
Jipiro(config)#mpls label protocol ldp
Jipiro(config)#mpls ldp router-id loopback 0
Jipiro(config)#
Jipiro(config)#router ospf 1
Jipiro(config-router)#network 192.168.0.0 0.0.255.255 area 0
Jipiro(config-router)#log-adjacency-changes
Jipiro(config-router)#exit
Jipiro(config)#exit
*Mar  1 00:34:36.435: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.10.2 on Serial0/1
f
rom LOADING to FULL, Loading Done
Jipiro#wr
Building configuration...
*Mar  1 00:34:37.993: %SYS-5-CONFIG_I: Configured from console by console
*Mar  1 00:34:48.967: %LDP-5-NBRCHG: LDP Neighbor 192.168.21.2:0 is UP
[OK]
Jipiro#
```

Anexo 3.8.e: Configuración TerraNet

TerraNet

```
Router>
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname TerraNet
TerraNet(config)#enable password terra123
TerraNet(config)#line console 0
TerraNet(config-line)#password terra
TerraNet(config-line)#login
TerraNet(config-line)#exit
TerraNet(config)#line vty 0 4
TerraNet(config-line)#password terra
TerraNet(config-line)#login
TerraNet(config-line)#exit
TerraNet(config)#
TerraNet(config)#ip cef
TerraNet(config)#
```

```
TerraNet(config)#mpls ip
TerraNet(config)#
TerraNet(config)#interface serial 1/0
TerraNet(config-if)#ip address 192.168.21.2 255.255.255.0
TerraNet(config-if)#mpls ip
TerraNet(config-if)#no shutdown
TerraNet(config-if)#exit
TerraNet(config)#
TerraNet(config)#interface serial 1/1
TerraNet(config-if)#ip address 192.168.22.1 255.255.255.0
TerraNet(config-if)#clock rate 56000
TerraNet(config-if)#mpl
TerraNet(config-if)#no shutdown
TerraNet(config-if)#exit
TerraNet(config)#
TerraNet(config)#
*Mar  1 00:25:58.639: %LINK-3-UPDOWN: Interface Serial1/1, changed state to up
*Mar  1 00:25:59.639: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial1/1, changed state to up
TerraNet(config)#
TerraNet(config)#interface loopback 0
TerraNet(config-if)#ip add 192.168.10.2 255.255.255.255
TerraNet(config-if)#exit
TerraNet(config)#
TerraNet(config)#mpls label protocol ldp
TerraNet(config)#
*Mar  1 00:26:55.551: %LDP-5-NBRCHG: LDP Neighbor 192.168.21.1:0 is UPloo
TerraNet(config)#mpls ldp router-id loopback 0
TerraNet(config)#
TerraNet(config)#router ospf 1
TerraNet(config-router)#network 192.168.0.0 0.0.255.255 area 0
TerraNet(config-router)#log-adjacency-changes
TerraNet(config-router)#exit
*Mar  1 00:27:58.267: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.10.1 on Serial1/0
f
rom LOADING to FULL, Loading Doneexit
TerraNet(config)#exit
TerraNet#
Building configuration...
[OK]
TerraNet#
```

CAPITULO III

CASO DE ESTUDIO 9

Anexo 3.9.a: Configuración Pop-Mariscal

Pop_Mariscal

```
Router>
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname Pop_Mariscal
Pop_Mariscal(config)#enable password maris123
Pop_Mariscal(config)#line console 0
Pop_Mariscal(config-line)#password maris
Pop_Mariscal(config-line)#login
Pop_Mariscal(config-line)#exit
Pop_Mariscal(config)#line vty 0 4
Pop_Mariscal(config-line)#password maris
Pop_Mariscal(config-line)#login
Pop_Mariscal(config-line)#exit
Pop_Mariscal(config)#
Pop_Mariscal(config)#
Pop_Mariscal(config)#ip cef
Pop_Mariscal(config)#
Pop_Mariscal(config)#mpls ip
Pop_Mariscal(config)#
Pop_Mariscal(config)#
Pop_Mariscal(config)#interf ace serial 0/0
Pop_Mariscal(config-if)#ip address 192.168.21.1 255.255.255.0
Pop_Mariscal(config-if)#clock rate 56000
Pop_Mariscal(config-if)#mpls ip
Pop_Mariscal(config-if)#no shutdown
Pop_Mariscal(config-if)#
*Mar  1 00:40:26.036: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
*Mar  1 00:40:27.038: %LINEPROTO-5-UPDOWN: Line protocol o
 changed state to up
Pop_Mariscal(config-if)#
Pop_Mariscal(config)#interface loopback 0
Pop_Mariscal(config-if)#ip address 192.168.40.1 255.255.255.255
Pop_Mariscal(config-if)#exit
Pop_Mariscal(config)#
Pop_Mariscal(config)#ip vrf transnet
Pop_Mariscal(config-vrf)#rd 100:110
Pop_Mariscal(config-vrf)#route-target export 100:1000
Pop_Mariscal(config-vrf)#route-target import 100:1000
Pop_Mariscal(config-vrf)#exit
Pop_Mariscal(config)#
Pop_Mariscal(config)#interface ethernet 0/0
Pop_Mariscal(config-if)#ip vrf forwarding transnet
Pop_Mariscal(config-if)#ip address 192.168.20.1 255.255.255.0
Pop_Mariscal(config-if)#exit
Pop_Mariscal(config)#
Pop_Mariscal(config)#
Pop_Mariscal(config)#ip cef
Pop_Mariscal(config)#mpls ip
Pop_Mariscal(config)#
Pop_Mariscal(config)#
Pop_Mariscal(config)#mpls label protocol ldp
Pop_Mariscal(config)#mpls ldp router-id loopback 0
Pop_Mariscal(config)#
```

```
Pop_Mariscal(config)#
Pop_Mariscal(config)#router ospf 1
Pop_Mariscal(config-router)#network 192.168.0.0 0.0.255.255 area 0
Pop_Mariscal(config-router)#exit
Pop_Mariscal(config)#
Pop_Mariscal(config)#
Pop_Mariscal(config)#router bgp 100
Pop_Mariscal(config-router)#
*Mar  1 00:54:04.145: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.22.1 on Serial0/0
f
rom LOADING to FULL, Lo
Pop_Mariscal(config-router)#no bgp default ipv4-unicast
*Mar  1 00:54:36.498: %LDP-5-NBRCHG: LDP Neighbor 192.168.22.1
Pop_Mariscal(config-router)#redistribute connected
Pop_Mariscal(config-router)#
Pop_Mariscal(config-router)#neighbor 192.168.40.3 remote-as 100
Pop_Mariscal(config-router)#neighbor 192.168.40.3 update-source loopback 0
Pop_Mariscal(config-router)#
Pop_Mariscal(config-router)#
Pop_Mariscal(config-router)#address-family vpnv4
Pop_Marisca(config-router-af)#neighbor 192.168.40.3 activate
Pop_Marisca(config-router-af)#neighbor 192.168.40.3 route-reflector-client
Pop_Marisca(config-router-af)#neighbor 192.168.40.3 send-community extended
Pop_Marisca(config-router-af)#exit-address-family
Pop_Mariscal(config-router)#
Pop_Mariscal(config-router)#address-family ipv4 vrf transnet
Pop_Marisca(config-router-af)#redistribute connected
Pop_Marisca(config-router-af)#exit-address-family
Pop_Mariscal(config-router)#
Pop_Mariscal(config-router)#exit
Pop_Mariscal(config)#exit
Pop_Mariscal#
Pop_Mariscal#
Pop_Mariscal#wr
Building configuration...
[OK]
Pop_Mariscal#
```

Anexo 3.9.b: Configuración Pop_Gasca

Pop_Gasca

```
Router>
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname Pop_Gasca
Pop_Gasca(config)#enable password gasca123
Pop_Gasca(config)#line console 0
Pop_Gasca(config-line)#password gasca
Pop_Gasca(config-line)#login
Pop_Gasca(config-line)#exit
Pop_Gasca(config)#line vty 0 4
Pop_Gasca(config-line)#password gasca
Pop_Gasca(config-line)#login
Pop_Gasca(config-line)#exit
Pop_Gasca(config)#
Pop_Gasca(config)#
Pop_Gasca(config)#ip cef
Pop_Gasca(config)#
Pop_Gasca(config)#mpls ip
Pop_Gasca(config)#
```

```
Pop_Gasca(config)#interf ace serial 0/1
Pop_Gasca(config-if)#ip address 192.168.22.2 255.255.255.0
Pop_Gasca(config-if)#mpls ip
Pop_Gasca(config-if)#no shutdown
Pop_Gasca(config-if)#
Pop_Gasca(config-if)#
Pop_Gasca(config)#interface loopback 0
Pop_Gasca(config-if)#ip address 192.168.40.3 255.255.255.255
Pop_Gasca(config-if)#exit
Pop_Gasca(config)#
Pop_Gasca(config)#
Pop_Gasca(config)#
Pop_Gasca(config)#
Pop_Gasca(config)#ip vrf transnet
Pop_Gasca(config-vrf)#rd 100:110
Pop_Gasca(config-vrf)#route-target export 100:1000
Pop_Gasca(config-vrf)#route-target import 100:1000
Pop_Gasca(config-vrf)#exit
Pop_Gasca(config)#
Pop_Gasca(config)#
Pop_Gasca(config)#interface loopback 30
Pop_Gasca(config-if)#ip vrf forwarding transnet
Pop_Gasca(config-if)#ip address 192.168.23.1 255.255.255.0
Pop_Gasca(config-if)#exit
Pop_Gasca(config)#
Pop_Gasca(config)#
Pop_Gasca(config)#mpls label protocol ldp
*Mar  1 01:25:57.982: %LDP-5-NBRCHG: LDP Neighbor 192.168.22.1:0 is
Pop_Gasca(config)#mpls ldp router-id loopback 0
Pop_Gasca(config)#
Pop_Gasca(config)#router ospf 1
Pop_Gasca(config-router)#network 192.168.0.0 0.0.255.255 area 0
Pop_Gasca(config-router)#exit
Pop_Gasca(config)#
*Mar  1 01:27:22.072: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.22.1 on Serial0/1
f
rom LOADING to FULL, Loading Don
Pop_Gasca(config)#
Pop_Gasca(config)#router bgp 100
Pop_Gasca(config-router)#no bgp default ipv4-unicast
Pop_Gasca(config-router)#redistribute connected
Pop_Gasca(config-router)#neighbor 192.168.40.1 remote-as 100
Pop_Gasca(config-router)#neighbor 192.168.40.1 update-source loopback 0
Pop_Gasca(config-router)#
Pop_Gasca(config-router)#address-family vpnv4
Pop_Gasca(config-router-af)#neighbor 192.168.40.1 activate
Pop_Gasca(config-router-af)#neighbor 192.168.40.1 route-reflector-client
Pop_Gasca(config-router-af)#neighbor 192.168.40.1 send-community ext
Pop_Gasca(config-router-af)#exit-address-family
Pop_Gasca(config-router)#
Pop_Gasca(config-router)#
Pop_Gasca(config-router)#address-family ipv4 vrf transnet
Pop_Gasca(config-router-af)#redistribute connected
Pop_Gasca(config-router-af)#exit-address-family
Pop_Gasca(config-router)#
Pop_Gasca(config-router)#exit
Pop_Gasca(config)#exit
Pop_Gasca#
Building configuration...
[OK]
Pop_Gasca#
```

Anexo 3.9.c: Configuración Vicentina

Vicentina

```
Router>
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname Vicentina
Vicentina(config)#enable password vicen123
Vicentina(config)#line console 0
Vicentina(config-line)#password vicen
Vicentina(config-line)#login
Vicentina(config-line)#exit
Vicentina(config)#line vty 0 4
Vicentina(config-line)#password vicen
Vicentina(config-line)#login
Vicentina(config-line)#exit
Vicentina(config)#
Vicentina(config)#
Vicentina(config)#interface serial 1/0
Vicentina(config-if)#ip address 192.168.22.1 255.255.255.0
Vicentina(config-if)#clock rate 56000
Vicentina(config-if)#mpls ip
Vicentina(config-if)#no shutdown
Vicentina(config-if)#exit
*Mar  1 00:11:32.023: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
*Mar  1 00:11:33.023: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial1/0, changed state to up
Vicentina(config)#
Vicentina(config)#
Vicentina(config)#ip cef
Vicentina(config)#
Vicentina(config)#mpls ip
Vicentina(config)#
Vicentina(config)#interface serial 1/1
Vicentina(config-if)#ip address 192.168.21.2 255.255.255.0
Vicentina(config-if)#mpls ip
Vicentina(config-if)#no shutdown
Vicentina(config-if)#exit
Vicentina(config)#
*Mar  1 00:13:31.271: %LINK-3-UPDOWN: Interface Serial1/1, changed state to up
*Mar  1 00:13:32.271: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial1/1, changed state to up
Vicentina(config)#
Vicentina(config)#
Vicentina(config)#router ospf 1
Vicentina(config-router)#network 192.168.0.0 0.0.255.255 area 0
Vicentina(config-router)#exit
Vicentina(config)#
Vicentina(config)#
Vicentina(config)#interface loopback 0
Vicentina(config-if)#ip address 192.168.40.2 255.255.255.255
Vicentina(config-if)#exit
Vicentina(config)#
Vicentina(config)#mpls label protocol ldp
Vicentina(config)#mpls ldp router-id loopback 0
Vicentina(config)#exit
Vicentina#wr
Building configuration...
[OK]
Vicentina#
```

CAPITULO III

CASO DE ESTUDIO 10

Anexo 3.10.a: Configuración Pop-Mariscal

Pop_Mariscal

```
Router>
Router>enable
Router#configure terminal
Router(config)#hostname Pop_Marisacal
Pop_Marisacal(config)#enable password maris123
Pop_Marisacal(config)#line console 0
Pop_Marisacal(config-line)#password maris
Pop_Marisacal(config-line)#login
Pop_Marisacal(config-line)#exit
Pop_Marisacal(config)#line vty 0 4
Pop_Marisacal(config-line)#password maris
Pop_Marisacal(config-line)#login
Pop_Marisacal(config-line)#exit
Pop_Marisacal(config)#
Pop_Marisacal(config)#
Pop_Marisacal(config)#ip cef
Pop_Marisacal(config)#
Pop_Marisacal(config)#mpls ip
Pop_Marisacal(config)#
Pop_Marisacal(config)#interface loopback 0
Pop_Marisacal(config-if)#ip address 192.168.10.1 255.255.255.255
Pop_Marisacal(config)#
Pop_Marisacal(config)#interface serial 0/0
Pop_Marisacal(config-if)#ip address 192.168.21.1 255.255.255.0
Pop_Marisacal(config-if)#clock rate 56000
Pop_Marisacal(config-if)#mpls ip
Pop_Marisacal(config-if)#no shutdown
Pop_Marisacal(config-if)#exit
Pop_Marisacal(config)#
Pop_Marisacal(config)#
Pop_Marisacal(config)#interface ethernet 0/0
Pop_Marisacal(config-if)#ip address 192.168.12.1 255.255.255.0
Pop_Marisacal(config-if)#no shutdown
Pop_Marisacal(config-if)#exit
Pop_Marisacal(config)#
Pop_Marisacal(config)#
Pop_Marisacal(config)#ip vrf transnet
Pop_Marisacal(config-vrf)#rd 100:110
Pop_Marisacal(config-vrf)#route-target export 100:1000
Pop_Marisacal(config-vrf)#route-target import 100:1000
Pop_Marisacal(config-vrf)#exit
Pop_Marisacal(config)#
Pop_Marisacal(config)#ip vrf ecuanet
Pop_Marisacal(config-vrf)#rd 100:120
Pop_Marisacal(config-vrf)#route-target export 100:1000
Pop_Marisacal(config-vrf)#route-target import 100:1000
Pop_Marisacal(config-vrf)#exit
Pop_Marisacal(config)#
Pop_Marisacal(config)#
Pop_Marisacal(config)#interface loopback 30
Pop_Marisacal(config-if)#ip vrf forwarding transnet
Pop_Marisacal(config-if)#ip address 192.168.20.1 255.255.255.0
Pop_Marisacal(config-if)#exit
Pop_Marisacal(config)#
```

```
Pop_Marisacal(config)#interface loopback 40
Pop_Marisacal(config-if)#ip vrf forwarding ecuanet
Pop_Marisacal(config-if)#ip address 192.168.24.1 255.255.255.0
Pop_Marisacal(config-if)#exit
Pop_Marisacal(config)#
Pop_Marisacal(config)#mpls label protocol ldp
Pop_Marisacal(config)#
Pop_Marisacal(config)#mpls ldp router-id loopback 0
Pop_Marisacal(config)#
Pop_Marisacal(config)#
Pop_Marisacal(config)#router ospf 1
Pop_Marisacal(config-router)#network 192.168.0.0 0.0.255.255 area 0
Pop_Marisacal(config-router)#log-adjacency-changes
Pop_Marisacal(config-router)#exit
Pop_Marisacal(config)#
Pop_Marisacal(config)#router bgp 100
Pop_Marisacal(config-router)#no bgp default ipv4-unicast
Pop_Marisacal(config-router)#redistribute connected
Pop_Marisacal(config-router)# neighbor 192.168.10.3 remote-as 100
Pop_Marisacal(config-router)#neighbor 192.168.10.3 update-source loopback 0
Pop_Marisacal(config-router)#
Pop_Marisacal(config-router)#address-family vpnv4
Pop_Marisacal(config-router-af)#neighbor 192.168.10.3 activate
Pop_Marisacal(config-router-af)#neighbor 192.168.10.3 route-reflector-client
Pop_Marisacal(config-router-af)#neighbor 192.168.10.3 send-community extended
Pop_Marisacal(config-router-af)#exit-address-family
Pop_Marisacal(config-router)#
Pop_Marisacal(config-router)#address-family ipv4 vrf transnet
Pop_Marisacal(config-router-af)#redistribute connected
Pop_Marisacal(config-router-af)#exit-address-family
Pop_Marisacal(config-router)#
Pop_Marisacal(config-router)#address-family ipv4 vrf ecuanet
Pop_Marisacal(config-router-af)#redistribute connected
Pop_Marisacal(config-router-af)#exit-address-family
Pop_Marisacal(config-router)#exit
Pop_Marisacal(config)#exit
Pop_Marisacal#
Pop_Marisacal#wr
Building configuration...
[OK]
Pop_Marisacal#
```

Anexo 3.10.b: Configuración Pop_Gasca

Pop_Gasca

```
Router>
Router>enable
Router#configure terminal
Router(config)#hostname Pop_Gasca
Pop_Gasca(config)#enable password gasca123
Pop_Gasca(config)#line console 0
Pop_Gasca(config-line)#password gasca
Pop_Gasca(config-line)#login
Pop_Gasca(config-line)#exit
Pop_Gasca(config)#line vty 0 4
Pop_Gasca(config-line)#password gasca
Pop_Gasca(config-line)#login
Pop_Gasca(config-line)#exit
Pop_Gasca(config)#
Pop_Gasca(config)#ip cef
Pop_Gasca(config)#
```

```
Pop_Gasca(config)#mpls ip
Pop_Gasca(config)#
Pop_Gasca(config)#interface loopback 0
Pop_Gasca(config-if)#ip address 192.168.10.3 255.255.255.255
Pop_Gasca(config-if)#exit
Pop_Gasca(config)#
Pop_Gasca(config)#interface serial 0/1
Pop_Gasca(config-if)#ip address 192.168.22.2 255.255.255.0
Pop_Gasca(config-if)#mpls ip
Pop_Gasca(config-if)#no shutdown
Pop_Gasca(config-if)#exit
Pop_Gasca(config)#
Pop_Gasca(config)#ip vrf transnet
Pop_Gasca(config-vrf)#rd 100:110
Pop_Gasca(config-vrf)#route-target export 100:1000
Pop_Gasca(config-vrf)#route-target import 100:1000
Pop_Gasca(config-vrf)#exit
Pop_Gasca(config)#
Pop_Gasca(config)#ip vrf ecuanet
Pop_Gasca(config-vrf)#rd 100:120
Pop_Gasca(config-vrf)#route-target export 100:1000
Pop_Gasca(config-vrf)#route-target import 100:1000
Pop_Gasca(config-vrf)#exit
Pop_Gasca(config)#
Pop_Gasca(config)#
Pop_Gasca(config)#interface loopback 50
Pop_Gasca(config-if)#ip vrf forwarding transnet
Pop_Gasca(config-if)#ip address 192.168.23.1 255.255.255.0
Pop_Gasca(config-if)#exit
Pop_Gasca(config)#
Pop_Gasca(config)#interface loopback 60
Pop_Gasca(config-if)#ip vrf forwarding ecuanet
Pop_Gasca(config-if)#ip address 192.168.25.1 255.255.255.0
Pop_Gasca(config-if)#exit
Pop_Gasca(config)#
Pop_Gasca(config)#
Pop_Gasca(config)#mpls label protocol ldp
Pop_Gasca(config)#mpls ldp router-id loopback 0
Pop_Gasca(config)#
Pop_Gasca(config)#
Pop_Gasca(config)#router ospf 1
Pop_Gasca(config-router)#network 192.168.0.0 0.0.255.255 area 0
Pop_Gasca(config-router)#log-adjacency-changes
Pop_Gasca(config-router)#exit
Pop_Gasca(config)#
Pop_Gasca(config)#router bgp 100
Pop_Gasca(config-router)#no bgp default ipv4-unicast
Pop_Gasca(config-router)#redistribute connected
Pop_Gasca(config-router)#neighbor 192.168.10.1 remote-as 100
Pop_Gasca(config-router)#neighbor 192.168.10.1 update-source loopback 0
Pop_Gasca(config-router)#
Pop_Gasca(config-router)#
Pop_Gasca(config-router)#address-family vpnv4
Pop_Gasca(config-router-af)#neighbor 192.168.10.1 activate
Pop_Gasca(config-router-af)#neighbor 192.168.10.1 route-reflector-client
Pop_Gasca(config-router-af)#neighbor 192.168.10.1 send-community extended
Pop_Gasca(config-router-af)#exit-address-family
Pop_Gasca(config-router)#
Pop_Gasca(config-router)#
Pop_Gasca(config-router)#address-family ipv4 vrf transnet
Pop_Gasca(config-router-af)#redistribute connected
Pop_Gasca(config-router-af)#exit-address-family
Pop_Gasca(config-router)#
Pop_Gasca(config-router)#
Pop_Gasca(config-router)#address-family ipv4 vrf ecuanet
Pop_Gasca(config-router-af)#redistribute connected
```

```
Pop_Gasca(config-router-af)#exit-address-family
Pop_Gasca(config-router)#exit
Pop_Gasca(config)#exit
Pop_Gasca#
```

Anexo 3.10.c: Configuración Vicentina

Vicentina

```
Router>
Router>enable
Router#configure terminal
Router(config)#hostname Vicentina
Vicentina(config)#enable password vicen123
Vicentina(config)#line console 0
Vicentina(config-line)#password vicen
Vicentina(config-line)#login
Vicentina(config-line)#exit
Vicentina(config)#line vty 04
Vicentina(config-line)#line vty 0 4
Vicentina(config-line)#password vicen
Vicentina(config-line)#login
Vicentina(config-line)#exit
Vicentina(config)#
Vicentina(config)#
Vicentina(config)#ip cef
Vicentina(config)#
Vicentina(config)#mpls ip
Vicentina(config)#
Vicentina(config)#
Vicentina(config)#interface loopback 0
Vicentina(config-if)#ip address 192.168.10.2 255.255.255.255
Vicentina(config-if)#exit
Vicentina(config)#
Vicentina(config)#interface serial 1/0
Vicentina(config-if)#ip address 192.168.22.1 255.255.255.0
Vicentina(config-if)#clock rate 56000
Vicentina(config-if)#mpls ip
Vicentina(config-if)#no shutdown
Vicentina(config-if)#exit
Vicentina(config)#
Vicentina(config)#
Vicentina(config)#interface serial 1/1
Vicentina(config-if)#ip address 192.168.21.2 255.255.255.0
Vicentina(config-if)#mpls ip
Vicentina(config-if)#no shutdown
Vicentina(config-if)#exit
Vicentina(config)#
Vicentina(config)#
Vicentina(config)#mpls label protocol ldp
Vicentina(config)#
Vicentina(config)#mpls ldp router-id loopback 0
Vicentina(config)#
Vicentina(config)#router ospf 1
Vicentina(config-router)#network 192.168.0.0 0.0.255.255 area 0
Vicentina(config-router)#log-adjacency-changes
Vicentina(config-router)#exit
Vicentina(config)#exit
Vicentina#
Vicentina#wr
Building configuration...
[OK]
Vicentina#
```

CAPITULO III

CASO DE ESTUDIO 11

Anexo 3.11.a: Configuración Pop-1

Pop_1

```
Router>
Router>enable
Router#configure terminal
Router(config)#hostname Pop_1
Pop_1(config)#enable password pop1123
Pop_1(config)#line console 0
Pop_1(config-line)#password pop1
Pop_1(config-line)#login
Pop_1(config-line)#exit
Pop_1(config)#line vty 0 4
Pop_1(config-line)#password pop1
Pop_1(config-line)#login
Pop_1(config-line)#exit
Pop_1(config)#
Pop_1(config)#
Pop_1(config)#ip cef
Pop_1(config)#
Pop_1(config)#mpls
Pop_1(config)#
Pop_1(config)#interface serial 0/0
Pop_1(config-if)#ip address 192.168.20.1 255.255.255.0
Pop_1(config-if)#mpls ip
Pop_1(config-if)#no shutdown
Pop_1(config-if)#exit
Pop_1(config)#
Pop_1(config)#interface ethernet 0/0
Pop_1(config-if)#ip address 192.168.19.1 255.255.255.0
Pop_1(config-if)#no shutdown
Pop_1(config-if)#exit
Pop_1(config)#
Pop_1(config)#
Pop_1(config)#interface loopback 0
Pop_1(config-if)#ip address 192.168.10.1 255.255.255.255
Pop_1(config-if)#exit
Pop_1(config)#
Pop_1(config)#mpls label protocol ldp
Pop_1(config)#mpls ldp router-id loopback
Pop_1(config)#
Pop_1(config)#
Pop_1(config)#router ospf 1
Pop_1(config-router)#network 192.168.0.0 0.0.255.255 area 0
Pop_1(config-router)#log-adjacency-changes
Pop_1(config-router)#exit
Pop_1(config)#
Pop_1(config)#
Pop_1(config)#
Pop_1(config)#access-list rate-limit 30 3
Pop_1(config)#interface ethernet 0/0
Pop_1(config-if)#rat
Pop_1(config-if)#rate-limit input access-group rate-limit 30 8000 8000 8000
conform-action set-mpls-exp-imposition-transmit 1 exceed-action set-mpls-exp-
imposition-transmit 0
Pop_1(config-if)#
Pop_1(config-if)#exit
```

```
Pop_1(config)#exit
Pop_1#wr
Building configuration...
[OK]
Pop_1#
```

Anexo 3.11.b: Configuración Pop_2

Pop_2

```
Router>
Router>enable
Router#configure terminal
Router(config)#hostname Pop_2
Pop_2(config)#enable password pop2123
Pop_2(config)#line console 0
Pop_2(config-line)#password pop2
Pop_2(config-line)#login
Pop_2(config-line)#exit
Pop_2(config)#line vty 0 4
Pop_2(config-line)#password pop2
Pop_2(config-line)#login
Pop_2(config-line)#exit
Pop_2(config)#
Pop_2(config)#
Pop_2(config)#interface loopback 0
Pop_2(config-if)#ip address 192.168.10.3 255.255.255.255
Pop_2(config-if)#exit
Pop_2(config)#
Pop_2(config)#interface serial 0/1
Pop_2(config-if)#ip address 192.168.21.2 255.255.255.0
Pop_2(config-if)#mpls ip
Pop_2(config-if)#no shutdown
Pop_2(config-if)#exit
Pop_2(config)#
Pop_2(config)#interface loopback 20
Pop_2(config-if)#ip address 192.168.22.1 255.255.255.0
Pop_2(config-if)#exit
Pop_2(config)#
Pop_2(config)#ip cef
Pop_2(config)#
Pop_2(config)#mpls ip
Pop_2(config)#
Pop_2(config)#router ospf 1
Pop_2(config-router)#network 192.168.0.0 0.0.255.255 area 0
Pop_2(config-router)#log-adjacency-changes
Pop_2(config-router)#exit
Pop_2(config)#
Pop_2(config)#
Pop_2(config)#access-list rate-limit 20 5
Pop_2(config)#
Pop_2(config)#interface loopback 20
Pop_2(config-if)#rate-limit input access-group rate-limit 20 8000 8000 8000
conform-action set-mpls-exp-imposition-transmit 5 exceed-action set-mpls-exp-
imposition-transmit 0
Pop_2(config-if)#end
Pop_2#
Pop_2#wr
Building configuration...
[OK]
Pop_2#
```

Anexo 3.11.c: Configuración Gonzanama

Gonzanamá

```
Router>enable
Router#configure terminal
Router(config)#hostname Gonzanama
Gonzanama(config)#enable password gonza123
Gonzanama(config)#line console 0
Gonzanama(config-line)#password gonza
Gonzanama(config-line)#login
Gonzanama(config-line)#exit
Gonzanama(config)#line vty 0 4
Gonzanama(config-line)#password gonza
Gonzanama(config-line)#login
Gonzanama(config-line)#exit
Gonzanama(config)#
Gonzanama(config)#
Gonzanama(config)#ip cef
Gonzanama(config)#
Gonzanama(config)#mpls ip
Gonzanama(config)#
Gonzanama(config)#interface serial 1/0
Gonzanama(config-if)#ip address 192.168.20.2 255.255.255.0
Gonzanama(config-if)#clock rate 56000
Gonzanama(config-if)#mpls ip
Gonzanama(config-if)#no shutdown
Gonzanama(config-if)#exit
Gonzanama(config)#
Gonzanama(config)#
Gonzanama(config)#interface serial 1/1
Gonzanama(config-if)#ip address 192.168.21.1 255.255.255.0
Gonzanama(config-if)#clock rate 56000
Gonzanama(config-if)#mpls ip
Gonzanama(config-if)#no shutdown
Gonzanama(config-if)#exit
Gonzanama(config)#
Gonzanama(config)#
Gonzanama(config)#interface loopback 0
Gonzanama(config-if)#ip address 192.168.10.2 255.255.255.255
Gonzanama(config-if)#exit
Gonzanama(config)#
Gonzanama(config)#mpls label protocol ldp
Gonzanama(config)#mpls ldp router-id loopback 0
Gonzanama(config)#
Gonzanama(config)#
Gonzanama(config)#router ospf 1
Gonzanama(config-router)#network 192.168.0.0 0.0.255.255 area 0
Gonzanama(config-router)#log-adjacency-changes
Gonzanama(config-router)#exit
Gonzanama(config)#exit
Gonzanama#
Gonzanama#wr
Building configuration...
 [OK]
Gonzanama#
```

CAPITULO III

CASO DE ESTUDIO 12

Anexo 3.12.a: Configuración Pop-Mariscal

Pop_Mariscal

```
Router>
Router>enable
Router#configure terminal
Router(config)#hostname Pop_Mariscal
Pop_Mariscal(config)#enable password maris123
Pop_Mariscal(config)#line console 0
Pop_Mariscal(config-line)#password maris
Pop_Mariscal(config-line)#login
Pop_Mariscal(config-line)#exit
Pop_Mariscal(config)#line vty 0 4
Pop_Mariscal(config-line)#password maris
Pop_Mariscal(config-line)#login
Pop_Mariscal(config-line)#exit
Pop_Mariscal(config)#
Pop_Mariscal(config)#
Pop_Mariscal(config)#interface loopback 0
Pop_Mariscal(config-if)#ip address 192.168.10.1 255.255.255.255
Pop_Mariscal(config-if)#exit
Pop_Mariscal(config)#interface serial 0/0
Pop_Mariscal(config-if)#ip address 192.168.20.1 255.255.255.0
Pop_Mariscal(config-if)#clock rate 56000
Pop_Mariscal(config-if)#mpls ip
Pop_Mariscal(config-if)#no shutdown
Pop_Mariscal(config-if)#exit
Pop_Mariscal(config)#
Pop_Mariscal(config)#
Pop_Mariscal(config)#interface loopback 20
Pop_Mariscal(config-if)#ip address 192.168.19.1 255.255.255.0
Pop_Mariscal(config-if)#exit
Pop_Mariscal(config)#
Pop_Mariscal(config)#
Pop_Mariscal(config)#ip cef
Pop_Mariscal(config)#
Pop_Mariscal(config)#mpls ip
Pop_Mariscal(config)#
Pop_Mariscal(config)#mpls label protocol ldp
Pop_Mariscal(config)#
Pop_Mariscal(config)#mpls ldp router-id loopback 0
Pop_Mariscal(config)#
Pop_Mariscal(config)#
Pop_Mariscal(config)#router ospf 1
Pop_Mariscal(config-router)#network 192.168.0.0 0.0.255.255 area 0
Pop_Mariscal(config-router)#log-adjacency-changes
Pop_Mariscal(config-router)#exit
Pop_Mariscal(config)#exit
Pop_Mariscal#
Pop_Mariscal#
Pop_Mariscal(config)#ip vrf loja
Pop_Mariscal(config-vrf)#rd 10:100
Pop_Mariscal(config-vrf)#route-target export 10:1000
Pop_Mariscal(config-vrf)#route-target import 10:1000
Pop_Mariscal(config-vrf)#exit
Pop_Mariscal(config)#
Pop_Mariscal(config)#router bgp 10
```

```
Pop_Mariscal(config-router)#no bgp default ipv4-unicast
Pop_Mariscal(config-router)#redistribute connected
Pop_Mariscal(config-router)#neighbor 192.168.10.3 remote-as 10
Pop_Mariscal(config-router)#neighbor 192.168.10.3 update-source loopback 0
Pop_Mariscal(config-router)#
Pop_Mariscal(config-router)#
Pop_Mariscal(config-router)#address-family vpnv4
Pop_Mariscal(config-router-af)#neighbor 192.168.10.3 activate
Pop_Mariscal(config-router-af)#neighbor 192.168.10.3 route-reflector-client
Pop_Mariscal(config-router-af)#neighbor 192.168.10.3 send-community extended
Pop_Mariscal(config-router-af)#exit-address-family
Pop_Mariscal(config-router)#
Pop_Mariscal(config-router)#address-family ipv4 vrf loja
Pop_Mariscal(config-router-af)#redistribute connected
Pop_Mariscal(config-router-af)#exit-address-family
Pop_Mariscal(config-router)#bgp log-neighbor-changes
Pop_Mariscal(config-router)#
Pop_Mariscal(config-router)#exit
Pop_Mariscal(config)#
Pop_Mariscal(config)#
Pop_Mariscal(config)#access-list rate-limit 20 3
Pop_Mariscal(config)#
Pop_Mariscal(config)#
Pop_Mariscal(config)#interface loopback 20
Pop_Mariscal(config-if)#
Pop_Mariscal(config-if)#rate-limit input access-group rate-limit 20 8000 8000
8000 conform-action set-mpls-exp-imposition-transmit 3 exceed-action set-mpls-
exp-imposition-transmit 3
Pop_Mariscal(config-if)#exit
Pop_Mariscal(config)#exit
Pop_Mariscal#
```

Anexo 3.12.b: Configuración Pop_Gasca

Pop_Gasca

```
Router>
Router>enable
Router#configure terminal
Router(config)#hostname Pop_Gasca
Pop_Gasca(config)#enable password gasca123
Pop_Gasca(config)#line console 0
Pop_Gasca(config-line)#password gasca
Pop_Gasca(config-line)#login
Pop_Gasca(config-line)#exit
Pop_Gasca(config)#line vty 0 4
Pop_Gasca(config-line)# password gasca
Pop_Gasca(config-line)#login
Pop_Gasca(config-line)#exit
Pop_Gasca(config)#
Pop_Gasca(config)#
Pop_Gasca(config)#interface loopback 0
Pop_Gasca(config-if)#ip address 192.168.10.3 255.255.255.255
Pop_Gasca(config-if)#exit
Pop_Gasca(config)#
Pop_Gasca(config)#interface serial 0/1
Pop_Gasca(config-if)#ip address 192.168.21.2 255.255.255.0
Pop_Gasca(config-if)#mpls ip
Pop_Gasca(config-if)#no shutdown
Pop_Gasca(config-if)#exit
Pop_Gasca(config)#
Pop_Gasca(config)#
Pop_Gasca(config)#interface loopback 20
Pop_Gasca(config-if)#ip address 192.168.22.1 255.255.255.0
```

```
Pop_Gasca(config-if)#exit
Pop_Gasca(config)#
Pop_Gasca(config)#
Pop_Gasca(config)#ip cef
Pop_Gasca(config)#
Pop_Gasca(config)#mpls ip
Pop_Gasca(config)#
Pop_Gasca(config)#mpls label protocol ldp
Pop_Gasca(config)#
Pop_Gasca(config)#mpls ldp router-id loopback 0
Pop_Gasca(config)#
Pop_Gasca(config)#
Pop_Gasca(config)#router ospf 1
Pop_Gasca(config-router)#network 192.168.0.0 0.0.255.255 area 0
Pop_Gasca(config-router)#log-adjacency-changes
Pop_Gasca(config-router)#exit
Pop_Gasca(config)#
Pop_Gasca(config)#ip vrf loja
Pop_Gasca(config-vrf)#rd 10:100
Pop_Gasca(config-vrf)#route-target export 10:1000
Pop_Gasca(config-vrf)#route-target import 10:1000
Pop_Gasca(config-vrf)#exit
Pop_Gasca(config)#
Pop_Gasca(config)#router bgp 10
Pop_Gasca(config-router)#no bgp default ipv4-unicast
Pop_Gasca(config-router)#redistribute connected
Pop_Gasca(config-router)#neighbor 192.168.10.1 remote-as 10
Pop_Gasca(config-router)#neighbor 192.168.10.1 update-source loopback 0
Pop_Gasca(config-router)#
Pop_Gasca(config-router)#
Pop_Gasca(config-router)#address-family vpnv4
Pop_Gasca(config-router-af)#neighbor 192.168.10.1 activate
Pop_Gasca(config-router-af)#neighbor 192.168.10.1 route-reflector-client
Pop_Gasca(config-router-af)#neighbor 192.168.10.1 send-community extended
Pop_Gasca(config-router-af)#exit-address-family
Pop_Gasca(config-router)#
Pop_Gasca(config-router)#address-family ipv4 vrf loja
Pop_Gasca(config-router-af)#redistribute connected
Pop_Gasca(config-router-af)#exit-address-family
Pop_Gasca(config-router)#bgp log-neighbor-changes
Pop_Gasca(config-router)#exit
Pop_Gasca(config)#
Pop_Gasca(config)#access-list rate-limit 40 3
Pop_Gasca(config)#
Pop_Gasca(config)#interface loopback 20
Pop_Gasca(config-if)# rate-limit input access-group rate-limit 20 8000 8000
8000 conform-action set-mpls-exp-imposition-transmit 3 exceed-action set-mpls-
exp-imposition-transmit 3
Pop_Gasca(config-if)#exit
Pop_Gasca(config)#exit
Pop_Gasca#
```

Anexo 3.12.c: Configuración Vicentina

Vicentina

```
Router>
Router>enable
Router#configure terminal
Router(config)#hostname Vicentina
Vicentina(config)#enable password vicen123
Vicentina(config)#line console 0
Vicentina(config-line)#password vicen
```

```
Vicentina(config-line)#login
Vicentina(config-line)#exit
Vicentina(config)#line vty 0 4
Vicentina(config-line)#password vicen
Vicentina(config-line)#login
Vicentina(config-line)
Vicentina(config)#
Vicentina(config)#
Vicentina(config)#interface loopback 0
Vicentina(config-if)#ip address 192.168.10.2 255.255.255.255
Vicentina(config-if)#exit
Vicentina(config)#
Vicentina(config)#interface serial 1/0
Vicentina(config-if)#ip address 192.168.20.2 255.255.255.0
Vicentina(config-if)#mpls ip
Vicentina(config-if)#no shutdown
Vicentina(config-if)#exit
Vicentina(config)#
Vicentina(config)#
Vicentina(config)#interface serial 1/1
Vicentina(config-if)#ip address 192.168.21.1 255.255.255.0
Vicentina(config-if)#clock rate 56000
Vicentina(config-if)#mpls ip
Vicentina(config-if)#no shutdown
Vicentina(config-if)#exit
Vicentina(config)#
Vicentina(config)#
Vicentina(config)#ip cef
Vicentina(config)#
Vicentina(config)#mpls ip
Vicentina(config)#
Vicentina(config)#mpls label protocol ldp
Vicentina(config)#
Vicentina(config)#mpls ldp router-id loopback 0
Vicentina(config)#
Vicentina(config)#
Vicentina(config)#router ospf 1
Vicentina(config-router)#network 192.168.0.0 0.
Vicentina(config-router)#log-adjacency-changes
Vicentina(config-router)#exit
Vicentina(config)#exit
Vicentina#
```

Glosario

Aggregate Route-based IP Switching (ARIS).- Es desarrollado por IBM y es muy similar a Tag switching de Cisco. En ARIS la distribución de etiquetas comienza en el enrutador de salida y se propaga de forma ordenada hacia el enrutador de entrada.

ASN.- Número de sistema autónomo

ATM.- Asynchronous Transfer Mode, Modo de Transferencia Asíncrona, es una tecnología de telecomunicación desarrollada para hacer frente a la gran demanda de capacidad de transmisión para servicios y aplicaciones

BGP.- Border Gateway Protocol, el protocolo BGP se ha constituido como el principal protocolo de encaminamiento externo utilizado en Internet.

CAR.- Commited Access Rate.

CE.- Customer Edge, es el router que está administrado por el cliente y que se conecta al PE

CEF.- Es una tecnología de conmutación avanzada de Capa 3. CEF optimiza el rendimiento y escalabilidad de redes cuyo tráfico es de carácter dinámico y poseen una topología dispersa tales como aplicaciones basadas en web y sesiones interactivas.

Cell Switching Router (CSR).- Fue desarrollado por Toshiba y presentado a la IETF en 1994. Fue una de las primeras propuestas para utilizar protocolos IP para controlar infraestructura ATM. CRS se ha desarrollado en redes comerciales y académicas en Japón.

Componente de control (control component).- Construye y mantiene la tabla de envío para el nodo a utilizar. Trabaja junto con los componentes de control de otros nodos para distribuir información de enrutamiento de forma consistente, también asegura que se utilicen los procedimientos locales adecuados para la creación de la tabla de envío.

Componente de envío (forwarding component).- Lleva al cabo el envío del paquete basándose en información de la tabla de envío (mantenida por el ruteador).

Conmutación (switching).- Es generalmente utilizado para describir la transferencia de datos de un puerto de entrada a un puerto de

salida donde la selección del puerto de salida esta basado en información de la capa 2 (ej: VPI/VCI en ATM).

Conmutación de Etiqueta (Label Switching).- Es una forma avanzada de envío de paquetes la cual reemplaza el algoritmo de envío convencional por un algoritmo mas eficiente de intercambio de etiqueta.

CPE.- Customer Premises Equipment, Equipo Terminal del Cliente

Enlaces de etiquetas.- El componente de control es el responsable de la creación y mantenimiento de la información de envío de etiquetas (a lo que nos referimos como enlaces).

Enrutamiento (routing).- Proceso para encontrar una ruta hacia un host destino. El enrutamiento en redes de gran tamaño es muy complejo dada la gran cantidad de destinos intermedios potenciales que debe atravesar un paquete antes de llegar al host destino.

Etiqueta (label).- Es un identificador corto de longitud fija de significado local el cual es utilizado para identificar un FEC. La etiqueta que se coloca en un paquete particular representa el FEC al cual el paquete es asignado.

FEC.- Forwarding Equivalence Class, una clase equivalente de envío se define como un conjunto de paquetes que se envían de la misma manera, por la misma ruta y con el mismo tratamiento. Nombre que se le da al tráfico que se encamina bajo una etiqueta. Subconjunto de paquetes tratados del mismo modo por el conmutador.

FIB.- Forwarding Information Base, es la tabla de envío IP utilizado por un router para tomar decisiones de envío.

Frame Relay.- Se define, oficialmente, como un servicio portador RDSI de banda estrecha en modo de paquetes, y ha sido especialmente adaptado para velocidades de hasta 2.048 Mbps, aunque nada le impide superarlas. Trabaja en el nivel de enlace de datos del modelo OSI, aunque también posee funcionalidad de nivel de red. Es utilizado para conectar distintas LANs entre si de una manera rapida y eficiente.

IANA.- Internet Assigned Number Authority. La Agencia de Asignación de Números Internet era el antiguo registro central de los protocolos Internet, como puertos, números de protocolo y empresa, opciones y códigos. Fue sustituido en 1998 por ICANN.

IETF.- Fuerza de Trabajo para la Ingeniería de Internet.

IGP.- Internal Gateway Protocol (IGP, protocolo de pasarela interno) hace referencia a los protocolos usados dentro de un sistema autónomo. Los protocolos IGP más utilizados son RIP, OSPF y IS-IS. Internal Gateway Protocol (IGP, protocolo de pasarela interno) IGP es un protocolo que genera tablas de enrutamiento dentro de un sistema autónomo.

IP switching.- Desarrollado por Ipsilon, se anunció en 1996. El objetivo básico de IP switching fue el de integrar conmutadores ATM de una manera eficiente (eliminando el plano de control ATM). Ipsilon utilizó la presencia de tráfico para controlar el establecimiento de una etiqueta.

LDP.- Label Distribution Protocol, es usado en conjunto con los protocolos de enrutamiento estandarizados de capa Red para distribuir información de etiquetado entre dispositivos en una red de conmutación de etiquetas.

LER.- Label Edge Router: elemento que inicia o termina el túnel (pone y quita cabeceras).

LFIB.- Label Forwarding Information Base, es la tabla de envío de etiquetas.

LSP.- Label Switched Path, nombre genérico de un camino MPLS (para cierto tráfico o FEC).

LSR de contorno.- Es un router que básicamente realiza dos funciones: imposición de etiquetas y la determinación de etiquetas en el contorno de la red

LSR.- Label Switching Router, se encarga de la conmutación de etiquetas en el corazón de la red.

MPLS.- Multiprotocol Label Switching, arquitectura basada en el concepto de conmutación de etiquetas creada para proporcionar circuitos virtuales en las redes IP.

NAT.- Network Address Translation, Traducción de Dirección de Red, es un estándar creado por la Internet Engineering Task Force (IETF) el cual utiliza una o más direcciones IP para conectar varios computadores a otra red (normalmente a Internet), los cuales tiene una dirección IP completamente distinta (normalmente una IP no válida de Internet definida por el RFC 1918). Por lo tanto, se puede utilizar para dar salida a redes públicas a computadores que se encuentran con direccionamiento privado o para proteger máquinas públicas

NLRI.- Network Layer Reachability Information

NSP.- Network Service Provider.

OSPF.- Open Shortest Path First, protocolo de enrutamiento que basa su funcionalidad sobre la técnica de estado-enlace.

P.- Representa el router que se encuentra en el núcleo de la red del Proveedor (Provider).

PE.- Provider Edge, es un LSR de contorno el cual es administrado por el proveedor de servicios de red. No existe diferencia entre un LSR de contorno y un PE.

PVC.- Switched Virtual Circuit, Circuito Virtual Conmutado, es un tipo de conexión utilizado en Frame Relay.

PVP.- Permanent Virtual Circuit, Circuito Virtual Permanente, es un tipo de conexión utilizado en Frame Relay.

QoS.- Quality of Service, Calidad de Servicio, es la medida del desempeño de un sistema de transmisión que refleja su calidad de transmisión y la disponibilidad de servicio. Es el conjunto de procedimientos que permiten a las aplicaciones de red, solicitar y recibir niveles de servicio en ancho de banda, propagación y variaciones de retardo (jitter)

Retención Liberal.- Cuando un router recibe enlaces de etiquetas para la misma FEC desde diferentes vecinos, todos los enlaces son retenidos. sólo se usarán algunos de esos enlaces de etiquetas; esto se basa en el siguiente salto actual para la FEC tal como se encuentra en la tabla de enrutamiento del LSR.

RIP.- Routing Information Protocol, Protocolo de Información de Enrutamiento. Es un protocolo de pasarela interior o IGP (Internet Gateway Protocol) utilizado por los routers aunque también pueden actuar en equipos, para intercambiar información acerca de redes IP.

Sistema Autónomo.- Es más que un conjunto de subredes, y el hardware asociado, administradas por una única autoridad, de forma que en ellas se puede implementar un algoritmo de encaminamiento independiente de los considerados en otro sistemas autónomos. Son los conocidos como protocolos de encaminamiento internos o IGP.

SLA.- Service Level Agreement, Acuerdo de Nivel de Servicio (ANS) en español Protocolo plasmado normalmente en un documento de carácter legal por el que una compañía que presta un servicio a otra se

compromete a prestar el mismo bajo unas determinadas condiciones y con unas prestaciones mínimas.

Spoofing.- Esquema que usan los routers de Cisco para hacer que un host trate a una interfaz como si estuviera funcionando y soportando una sesión. El router hace spoofing de respuestas a mensajes de actividad del host para convencer a ese host de que la sesión continúa. El spoofing resulta útil en entornos de enrutamiento como DDR, en el cual un enlace de conmutación de circuito se desconecta cuando no existe tráfico que se deba enviar a través del enlace, a fin de ahorrar gastos por llamadas pagadas. Ver también DDR. 2. La acción de un paquete que ilegalmente dice provenir de una dirección desde la cual en realidad no se lo ha enviado. El spoofing está diseñado para contrarrestar los mecanismos de seguridad de la red por ejemplo, filtros y listas de acceso.

Tabla de envío (forwarding table).- Es un conjunto de campos en una tabla, los cuales proporcionan la información que ayuda al componente de envío a realizar su función de conmutación. La tabla de envío debe asociar cada paquete con un campo (tradicionalmente la dirección destino).

Tag switching.- Es la tecnología de conmutación de etiquetas desarrollada por Cisco Systems. A diferencia de las dos soluciones anteriores, tag switching es una técnica la cual no requiere de flujo de tráfico para la creación de tablas de etiqueta en un enrutador. En lugar de esto utilizaba protocolos de enrutamiento IP para determinar el siguiente brinco.

VCI.- Virtual Circuit Identifier, Identificador de circuito virtual (16 bits)

VPI.- Virtual Path Identifier, Identificador de camino virtual (8 bits)

VPN-IPv4.- Es la combinación de una dirección IPv4 y el distintivo de ruta.

VRF.- Virtual Routing and Forwarding, cada VPN es asociada con una o más instancias de enrutamineto o envío VPN, una vrf está formada de una tabla de enrutamiento IP, a tabla derivada CEF y un conjunto de interfaces que utilizan la tabla de envío

WAN.- WAN es un acrónimo de Wide Area Network que en inglés significa (red de área amplia). Un ejemplo de este tipo de redes sería rediris, la misma internet o cualquier red en que no esté en un mismo edificio todos sus miembros (sobre la distancia hay discusión posible). Opera en la capa física y de enlace del modelo de referencia OSI. A nivel de alcance, esta red abarca desde unos 100km (País) hasta llegar incluso a 1000km (Continente).

X.25.- Red de conmutación de paquetes basada en el protocolo HDLC proveniente de IBM. Establece mecanismos de direccionamiento entre usuarios, negociación de características de comunicación, técnicas de recuperación de errores. Los servicios públicos de conmutación de paquetes admiten numerosos tipos de estaciones de distintos fabricantes. Por lo tanto, es de la mayor importancia definir la interfaz entre el equipo del usuario final y la red.