



## **DESARROLLO DE UN ENTORNO MPLS BASADO EN GNS3**

**María Rodríguez González**

**Tutor: Víctor Miguel Sempere Paya**

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Curso 2018-19

Valencia, 10 de septiembre de 2019



## Resumen

Creación y configuración de un entorno MPLS usando el programa de simulación GNS3. Se tratan los principales conceptos de las redes MPLS: distribución de etiquetas, Ingeniería de Tráfico y VPN.

Se detallan los pasos de instalación y configuración del simulador y resto de programas a usar y el desarrollo de cada red, junto con una serie de experimentos para monitorizar su funcionamiento e identificar los paquetes que circulan por las redes.

## Resum

Creació i configuració d'un entorn MPLS usant el programa de simulació GNS3. En el projecte es tracten els principals conceptes de les xarxes MPLS: distribució d'etiquetes, Enginyeria de Tràfic i VPN.

Es detallen els passos d'instal·lació i configuració del simulador i resta de programes a usar i el desenvolupament de cada xarxa, juntament amb una sèrie d'experiments per a monitorar el seu funcionament i identificar els paquets que circulen per les xarxes.

## Abstract

Creation and configuration of an MPLS environment using the GNS3 simulation program. The main concepts of MPLS networks are discussed in the project: label distribution, Traffic Engineering and VPN.

The installation and configuration of the simulator and the development of each network are detailed, together with a series of experiments to monitor the correct operation and identify the packages that circulate through the networks.



## Índice

Capítulo 1.	Objetivos del trabajo .....	1
Capítulo 2.	Softwares necesarios .....	2
2.1	Introducción .....	2
2.2	GNS3.....	2
2.2.1	Descarga e instalación.....	2
2.2.2	Interfaz gráfica y configuración de elementos .....	6
2.2.3	Wireshark. Captura de tráfico .....	9
2.3	Máquina Virtual .....	10
2.3.1	Configuración.....	10
Capítulo 3.	Práctica 1 “Simulación Básica de una red MPLS”.....	12
3.1	Introducción .....	12
3.2	Etiqueta MPLS .....	12
3.3	Elementos MPLS.....	13
3.3.1	Forwarding Equivalence Class (FEC).....	13
3.3.2	Label Switched Path (LSP) .....	13
3.3.3	Label Switch Routers (LSR) .....	13
3.3.4	Label Edge Routers (LER).....	13
3.4	Distribución de etiquetas.....	14
3.5	Objetivos .....	16
3.6	Elementos necesarios .....	16
3.7	Topología de red.....	17
3.8	Configuración de la red.....	18
3.8.1	Asignación de direcciones.....	19
3.8.2	Protocolo OSPF.....	19
3.8.3	Configuración MPLS .....	20
3.8.4	Verificación final.....	21
3.9	Actividades propuestas.....	23
Capítulo 4.	Práctica 2 “Simulación de una red MPLS-TE” .....	32
4.1	Introducción .....	32
4.2	MPLS-TE .....	32
4.3	Routing basado en restricciones (CBR) .....	33
4.4	RSVP-TE.....	33
4.5	Fast Reroute .....	34



4.6	Objetivos .....	34
4.7	Topología de red.....	34
4.8	Configuración.....	35
4.8.1	Añadir direcciones y OSPF .....	35
4.8.2	Configuración de TE .....	35
4.8.3	Establecimiento túneles .....	36
4.8.4	Verificación final.....	38
4.9	Actividades propuestas.....	40
Capítulo 5.	Práctica 3 “Simulación Básica de una red MPLS VPN” .....	45
5.1	Introducción .....	45
5.2	Componentes y arquitectura L3VPN .....	45
5.2.1	Customer Edge (CE) .....	45
5.2.2	Provider Edge (PE).....	45
5.2.3	Provider (P) .....	45
5.2.4	Virtual Routing Forwarding (VRF).....	46
5.2.5	Route Distinguishers (RD) .....	46
5.2.6	Route Targets (RT).....	46
5.3	MP-BGP .....	47
5.4	Propagación de rutas y envío de paquetes en MPLS VPN.....	47
5.5	Objetivos .....	48
5.6	Topología de red.....	48
5.7	Configuración.....	49
5.7.1	Asignación de direcciones.....	49
5.7.2	MPLS .....	49
5.7.3	MP-BGP .....	50
5.7.4	VRF .....	51
5.7.5	Routing CE-PE.....	52
5.7.6	Redistribución rutas.....	52
5.7.7	Verificación final.....	53
5.8	Actividades propuestas.....	54
Capítulo 6.	Bibliografía.....	59



## Capítulo 1. Objetivos del trabajo

El objetivo principal de este trabajo es el desarrollo de redes MPLS en un entorno simulado. El trabajo se dividirá en tres prácticas orientadas a la parte del protocolo MPLS de la asignatura “Redes Públicas de Transporte”.

Estas prácticas serán realizadas con el simulador GNS3, para que los alumnos puedan desarrollar las redes de una forma más avanzada, detallada y visual, aprovechando las herramientas que proporciona dicho simulador, y afianzar los conceptos impartidos en las clases teóricas.

Los objetivos específicos que se detallan en el trabajo son:

- Configurar y comprender el funcionamiento del simulador GNS3, Wireshark y de la máquina virtual VMware.
- Diseñar y configurar redes básicas MPLS en GNS3.
- Estudiar la arquitectura y comprobar el funcionamiento de MPLS.
- Establecer y estudiar el protocolo LDP con diversas pruebas.
- Implementar la Ingeniería de Tráfico en una red MPLS en GNS3.
- Establecer y estudiar el protocolo RSVP.
- Diseñar y configurar redes MPLS VPN de nivel 3 en GNS3.
- Establecer y estudiar el protocolo BGP.

## Capítulo 2. Softwares necesarios

### 2.1 Introducción

En estas prácticas se manejarán varios softwares de uso ampliamente extendidos en los entornos de virtualización, diseño y simulación de redes, como son GNS3, Wireshark y VMware.

**GNS3** es un simulador gráfico de red libre y de código abierto, bajo licencia GPLv3, el cual permite diseñar topologías de red complejas de alta calidad y realizar simulaciones sobre las mismas.

Para la ejecución de simulaciones, GNS3 está compuesto por diferentes módulos de entre los que destacan:

- Dynamips: emulador de IOS que permite a los usuarios ejecutar imágenes binarias de IOS de Cisco Systems, como routers o switches.
- Dynagen: front-end basado en texto para Dynamips.
- Qemu y VirtualBox/VMWare: uso de máquinas virtuales.
- VPCS: emuladores de PC con funciones básicas de networking.

La principal ventaja de GNS3 frente a otros softwares de simulación, como puede ser el Packet Tracer de Cisco, es que los equipos de red simulados disponen de todas las funcionalidades de un equipo real, ya que ejecuta el mismo firmware que utilizaríamos en un equipo físico. Lo que nos permitirá diseñar una topología de red simulada lo más parecida posible a una situación real sin tener la necesidad de tener acceso a ningún equipo físico.

Su principal inconveniente es que consume bastantes recursos y puede verse afectado por las limitaciones del ordenador en el que se ejecuta, por lo que no es aconsejable ejecutarlo en PCs con menos de 8 GB de RAM.

**Wireshark** es un sniffer o analizador de protocolos con interfaz gráfica que captura el tráfico de una red. Utilizado para analizar y solucionar problemas en redes de comunicaciones, para desarrollo de software y protocolos, y como herramienta didáctica. Al igual que GNS3, Wireshark es un software libre con licencia GPL.

Algunas de sus características más importantes son:

- Captura datos de red y lee datos almacenados en un archivo (una captura previa).
- Interfaz con gran flexibilidad.
- Alta capacidad de filtrado.
- Compatibilidad con más de 480 protocolos.

**VMware WorkStation Player** es un software libre de virtualización que permite crear y ejecutar de una forma sencilla máquinas virtuales con distintos sistemas operativos en un mismo PC.

Es una opción mejor frente a otros softwares de máquinas virtuales, como VirtualBox, debido a su mayor velocidad y al soporte de virtualización anidada.

### 2.2 GNS3

#### 2.2.1 Descarga e instalación

Al ser GNS3 un software libre, se puede descargar únicamente desde su página web oficial (<https://www.gns3.com/>.) Para poder realizar la descarga se tendrá que registrar previamente de manera gratuita, si no se había hecho anteriormente.

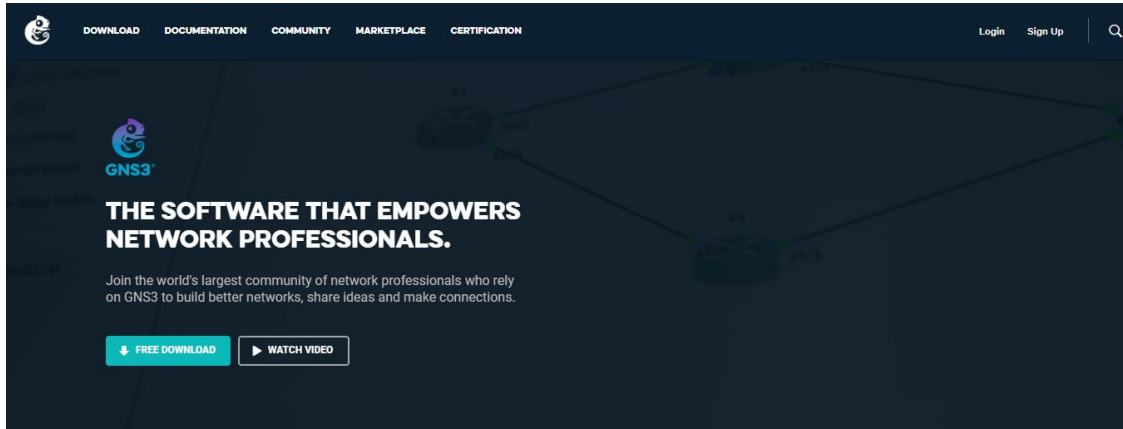


Figura 1. Página inicial GNS3

Los requerimientos de hardware mínimos son:

- Sistema Operativo: Windows 7 (64 bit) o superior, Apple MAC OS o Linux.
- Procesador: 2 o más núcleos.
- Memoria: 4 GB RAM.
- Almacenamiento: 200 MB de espacio disponible para la instalación en Windows.

Haga click sobre *Free Download* y accederá a la pantalla de Sign Up para realizar el registro. Si ya tiene un usuario y contraseña, introdúzcalos en la pestaña de Login y tendrá acceso a la pantalla de descargas.

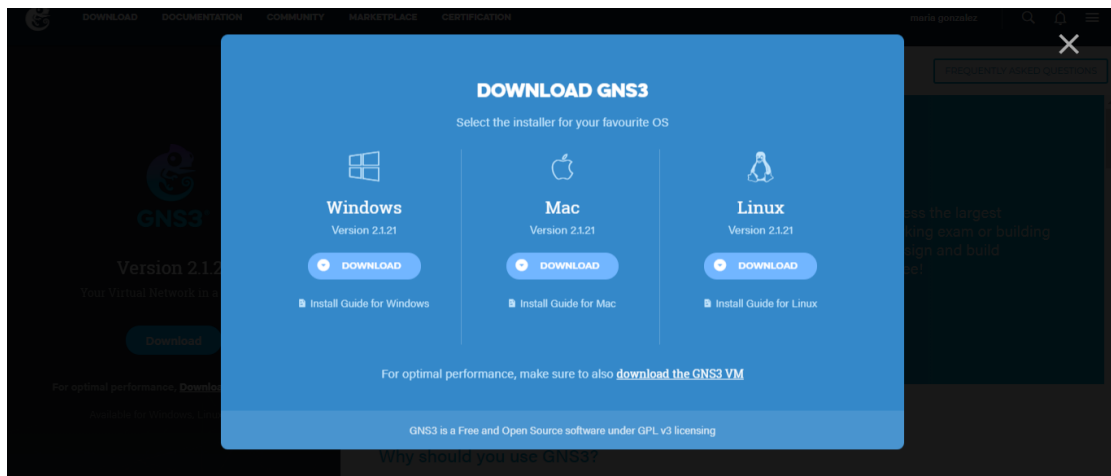


Figura 2. Página descarga GNS3

Seleccione la versión para su sistema operativo, en este caso se trabajará sobre Windows 10 con la versión de GNS3 2.1.21.

Presionando sobre Download, comenzará la descarga de un fichero denominado *GNS3-2.1.21-all-in-one-regular*. Una vez completada la descarga del fichero, ejecútelo haciendo doble click sobre el mismo y accederá a la siguiente pantalla de Setup.



Figura 3. Setup

Presione el botón de Next y acepte la licencia. Vuelva a presionar Next y aparecerá la pantalla de selección de componentes.

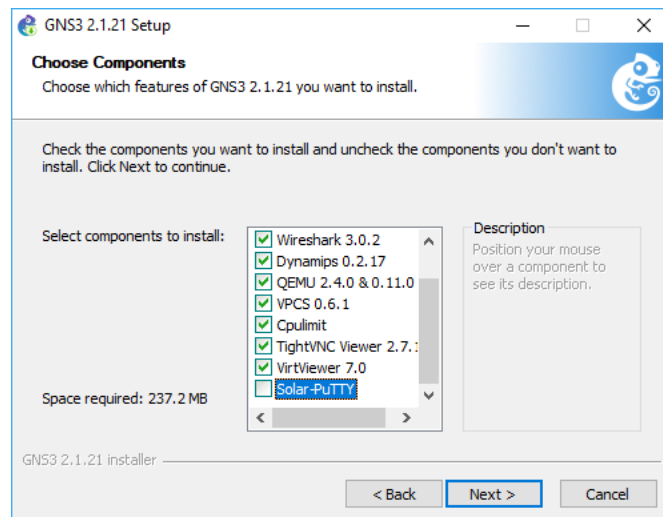


Figura 4. Componentes del Setup

Seleccione todos los componentes a excepción de Solar-PuTTY. Este componente emula al terminal de un equipo. En este caso es mejor usar el termina por defecto de GNS3.

Vuelva a hacer click sobre Next, seleccione el directorio donde desee instalar el programa y presione *Install*.

Una vez se haya completado la instalación haga click en Next. En la siguiente pantalla, seleccione que no desea instalar el Solarwinds Standard, una colección de herramientas avanzadas de red que no serán necesarias para realizar nuestras redes.



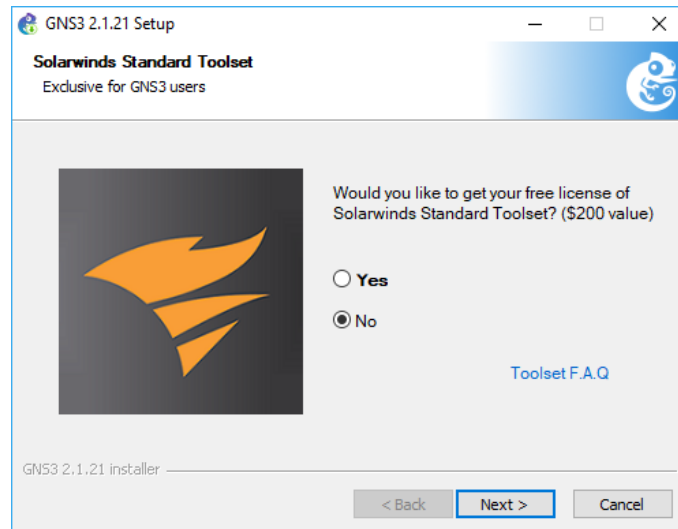


Figura 5. Componentes del Setup

Por último, presione Finish y se iniciará GNS3.

La primera vez que se ejecuta GNS3 aparece un *Setup Wizard*.

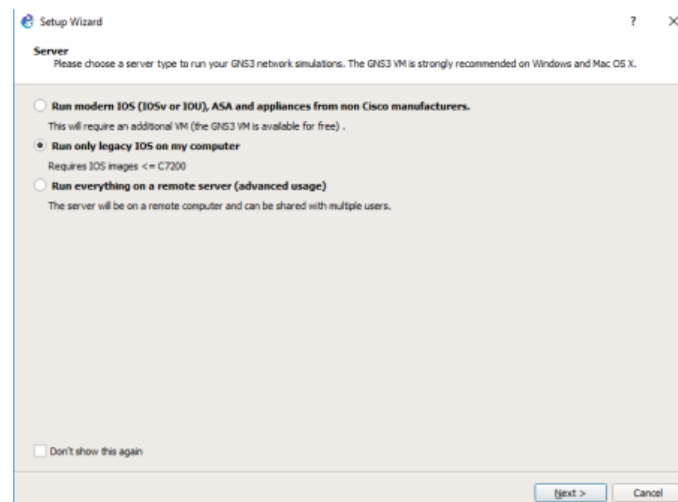


Figura 6. Elección servidor

Se tendrá que escoger el servidor sobre el que se ejecutará la simulación de la red. Al ser la primera vez que ejecutamos el programa, escogeremos la segunda opción, ya que el resto de las opciones requiere la instalación de una máquina virtual, la cual se instalará posteriormente. Se podrá cambiar esta opción.

En la pantalla de *Local server configuration*, seleccione en el desplegable Host binding la dirección IP del equipo donde se ha instalado el software.

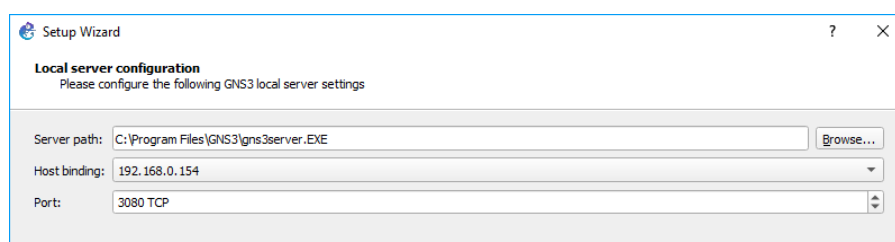


Figura 7. Configuración servidor local

Si no ha habido ningún problema en la conexión, se mostrará en pantalla “*Connection to local server successful*” y podrá continuar.

En la siguiente pantalla presione *Finish* y habrá terminado la instalación.

A continuación, le aparecerá la pantalla de plantillas para acceder dispositivos, presione en *Cancel*, para acceder directamente al programa. Posteriormente aprenderá a añadir los dispositivos al espacio de trabajo.

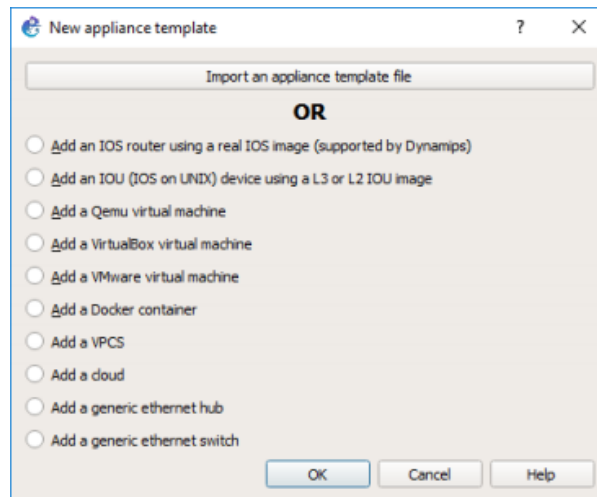


Figura 8. Plantillas

Presione también *Cancel* sobre la ventana activa Project

### 2.2.2 Interfaz gráfica y configuración de elementos

Primero cree un nuevo proyecto, en menú *File – New blank Project* o presionando el primer icono de la izquierda de la barra de herramientas superior.



Figura 9. Barra de herramientas

Se abrirá la ventana del nuevo proyecto, asígnele un nombre y seleccione el directorio para guardarlo.

Seguidamente aparecerá la ventana de GNS3 con su área de trabajo.

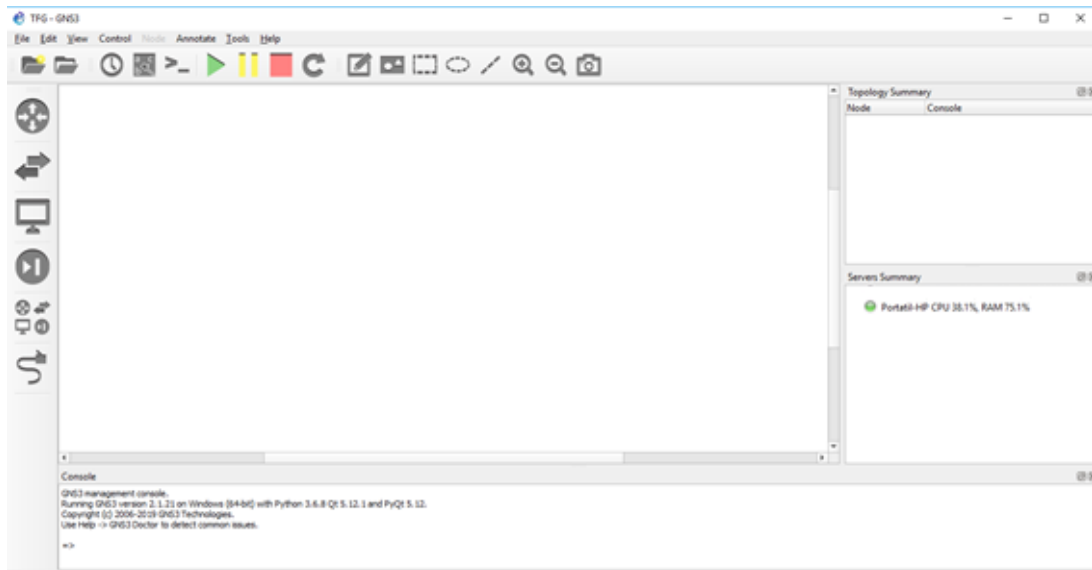


Figura 10. Ventana principal

La ventana principal de GNS3 se divide en varias subventanas:

- Área de trabajo: donde se crean las topologías.
- Topology Summary: información de la topología creada, detallando el nodo y el estado en el que se encuentra,
- Servers Summary: servidores y su estado.
- Panel de la consola: muestra los mensajes, avisos y errores del simulador.
- Barra de elementos: situada en la zona izquierda, muestra los distintos tipos de nodos o equipos disponibles que se podrán añadir a la topología de red de su proyecto. De arriba-abajo podemos seleccionar routers, switches, dispositivos finales como VPCS, elementos de seguridad, todos los dispositivos y conexiones. Haciendo click sobre cada uno de ellos, aparece un desplegable donde podrá seleccionar todos los dispositivos disponibles en cada categoría.

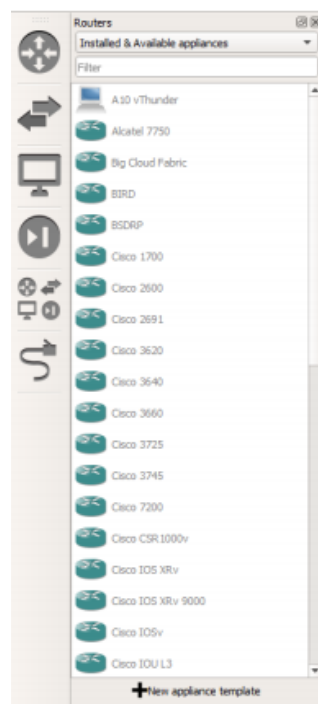


Figura 11. Barra de elementos y desplegable

Se tiene disponible desde los pequeños routers Cisco de la serie 1700 hasta los routers más avanzados para montaje en rack de la serie 7200. Es recomendable usar los modelos c3640, c3660, c3725, c3745y c7200.

Para insertar el primer elemento de la red, acceda a la barra de elementos y haga click sobre el primer icono, *Routers*. Para este caso seleccionaremos el modelo Cisco 7200, el cual nos permitirá un mayor número de accesos FastEthernet para la red que crearemos posteriormente y es el único que admite ciertos protocolos que usaremos. Pinche sobre él y arrástrelo a la zona de trabajo.

Si es la primera vez que utiliza este dispositivo, aparecerá la pantalla de configuración.

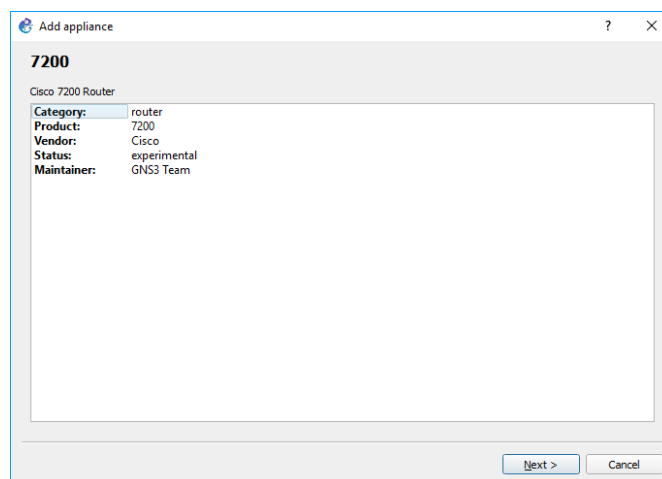


Figura 12. Añadir elemento

Pulse sobre *Next* y seguidamente seleccione el servidor donde se simulará el dispositivo. Al no estar todavía instalada la máquina virtual, únicamente aparecerá la opción de que el elemento corra sobre el servidor local.

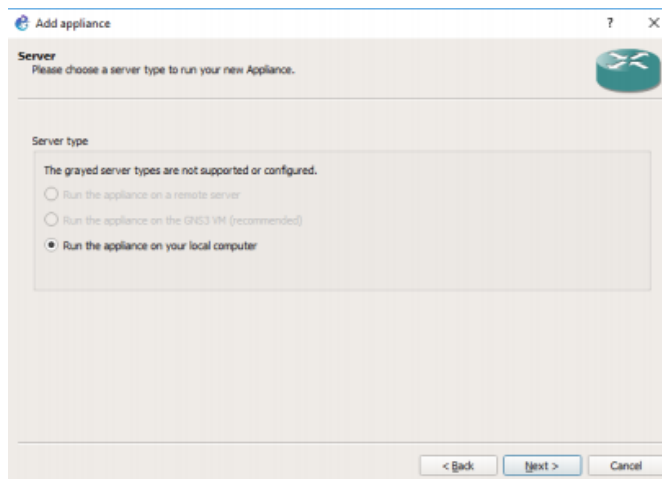


Figura 13. Servidor del dispositivo

En la siguiente pantalla se le solicita el fichero de la ROM de IOS que ejecutará el router. Por defecto aparece seleccionada la versión con la que la comunidad ha desarrollado su emulación. Si no dispone de dicha versión, o quiere utilizar una versión diferente, para ello debería de hacer click sobre *Create a new version*, asignarle un nombre y cargar la versión de IOS para el router, que desee utilizar.

Para cargar la ROM, haga click sobre el nombre del fichero, presione *Import* y seleccione el fichero IOS con el que desea trabajar.

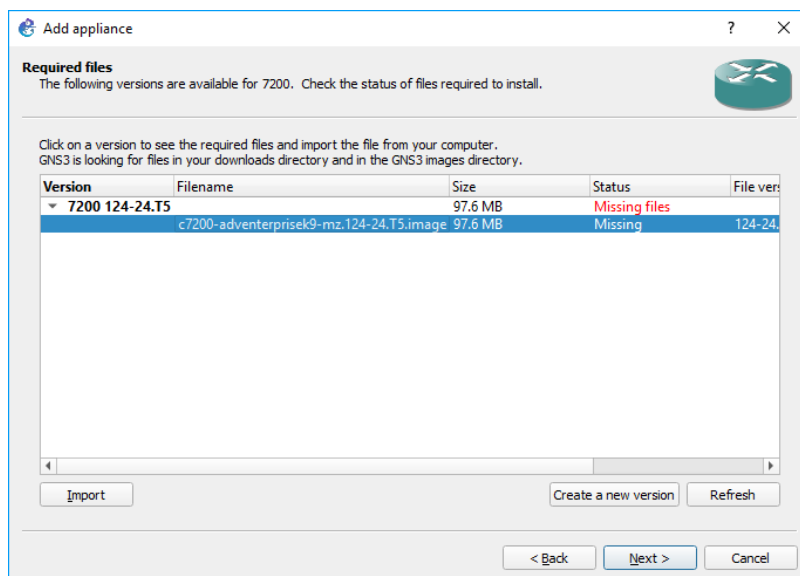


Figura 14. Archivos requeridos

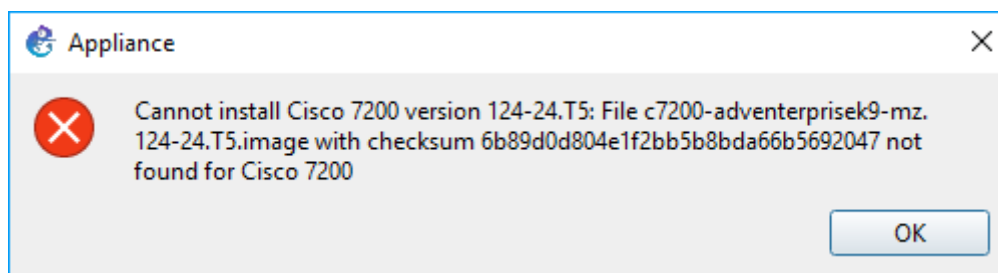


Figura 15. Fallo de versión

La versión que disponemos no es exactamente igual al que tiene predeterminado el dispositivo, no coinciden los hashes. Por lo que crearemos una nueva versión, en *Create new version*, con el nombre del fichero de la imagen, “c7200-a3jk9s-mz.124-25g” e importando nuevamente la ROM.

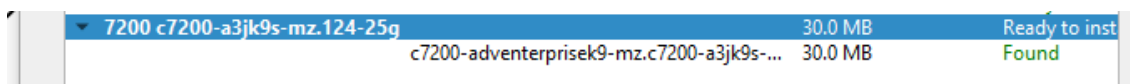


Figura 16. Nueva versión del dispositivo

Haga click sobre la nueva versión y luego seleccione *Next*.

Le aparecerá una pantalla pidiéndole confirmación para instalar la ROM seleccionada, responda que *sí*.

Posteriormente se abrirá una ventana mostrando un resumen de las características del dispositivo, pulse en *Next* y en la siguiente sobre *Finish*. A partir de este momento el modelo de router estará instalado y disponible para ser utilizado en la red.

### 2.2.3 Wireshark. Captura de tráfico

Con *GNS3* se puede capturar el tráfico que circula por los enlaces virtuales de una topología usando el software Wireshark. Haciendo click con el botón derecho sobre un enlace de la red iniciará la captura en tiempo real y la mostrará en su interfaz gráfica.

Además, permitirá filtrar ese tráfico según el protocolo que se elija, desplegar paquetes y realizar gráficas, entre otras opciones.

## 2.3 Máquina Virtual

### 2.3.1 Configuración

En GNS3 se puede utilizar el propio equipo donde está instalado el programa como servidor, pero es necesario utilizar una máquina virtual para poder añadir equipos preconfigurados a través de *appliances* (ficheros con extensión gns3a) de un forma rápida y sencilla.

Para activar la VM vaya al menú *Edit – Preferences*.

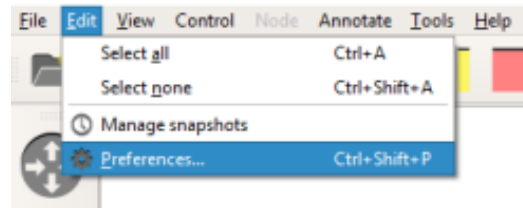


Figura 17. Preferencias

En el panel izquierdo, pinche en el apartado *GNS3 VM*. Haciendo click sobre el recuadro *Enable the GNS3 VM* y seleccione del desplegable el software de virtualización que tiene instalado en su equipo, en este caso *VMware Workstation Player*.

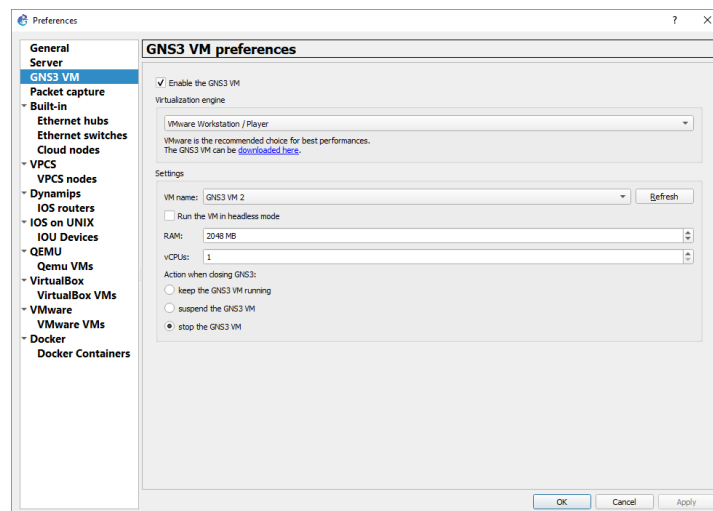


Figura 18. Preferencias VM

La imagen de la máquina virtual se podrá descargar desde la web oficial de *GNS3* (<https://www.gns3.com/>). La versión debe coincidir con la versión del *GNS3*. Descomprímala y haga doble click sobre la misma para importarla a *VMware*.

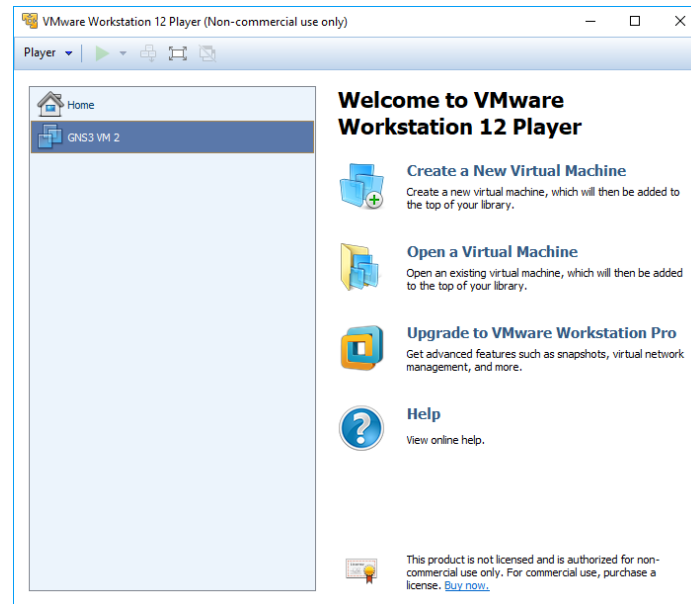


Figura 19. VMware

Una vez realizados los pasos anteriores reinicie el *GNS3* para activar la VM. Cuando arranque de nuevo le aparecerá una pantalla similar a la siguiente. Es la VM donde se ejecuta el servidor de GNS3.

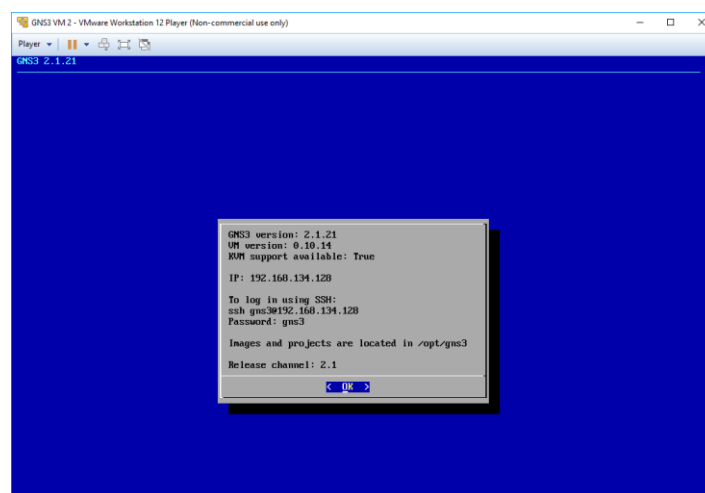


Figura 20. Pantalla principal VM

Observe que en la ventana *Server Summary* de GNS3 aparece la instancia del servidor VM.

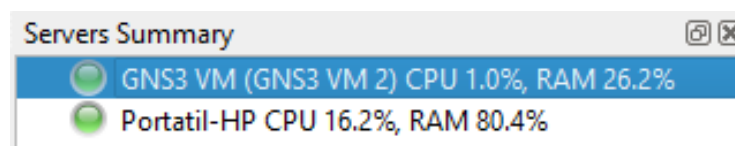


Figura 21. Servers Summary

Cada vez que inicie el simulador, tendrá que esperar a que también lo haga la máquina virtual para abrir o crear un proyecto.

## Capítulo 3. Práctica 1 “Simulación Básica de una red MPLS”

### 3.1 Introducción

En la década de los 90, según incrementaban los tamaños de las redes y aparecían nuevas aplicaciones de audio y video streaming, los proveedores de servicios exigían mejores prestaciones y recursos, por lo que era necesario buscar una alternativa al encapsulado único en IP.

Se introdujo ATM (Asynchronous Transfer Model) en la capa 2 (capa de enlace) de las redes. Este modelo de IP sobre ATM utilizaba el encaminamiento de nivel 3 de los routers, con conmutadores de nivel 2 funcionando con etiquetas y ofrecía un incremento del ancho de banda y del rendimiento, pero era difícil de integrar al basarse en dos tecnologías distintas y de escalar por el aumento de adyacencias según aumentaban las redes.

Posteriormente, aparecieron otras soluciones que intentaban integrar ATM con encaminamiento IP en un único router, utilizando protocolos IP (de enrutamiento y reenvío) para distribuir etiquetas. Estos protocolos no eran compatibles entre sí y necesitaban de infraestructuras ATM.

En 1997, un grupo de investigadores de CISCO establecieron un sistema basado en la conmutación de etiquetas llamado MPLS. De esta forma, los routers examinarían las etiquetas para realizar el proceso de enrutamiento y evitarían mirar continuamente las tablas de routing IP, proporcionando una mayor velocidad y efectividad al proceso. [RFC 3031]

**MPLS** (Multiprotocol Label Switching) es una tecnología de conmutación de tráfico por etiquetas cuyo encapsulado se sitúa entre las capas 2 y 3, siendo independiente del protocolo de la capa de red (L3) usado.

Separa completamente la parte de encaminamiento, la cual es lenta y compleja, de la parte de conmutación en el reenvío de paquetes, que es más rápida y simple.

Los routers calculan todas las rutas mediante protocolos de enrutamiento (en estas prácticas utilizaremos OSPF y BGP), a partir de los cuales construyen las tablas de encaminamiento. Usando esas tablas de routing y protocolos de distribución de etiquetas, establecen etiquetas MPLS y caminos virtuales o LSP por donde irán los paquetes. Estos caminos discurren por dos tipos de nodos por los que está compuestas las redes MPLS: LER y LSR.

Sus principales aplicaciones son funciones de Ingeniería de Tráfico (TE), servicios de VPNs, técnicas de QoS y Policy Routing.

### 3.2 Etiqueta MPLS

La cabecera MPLS de 32 bits es introducida entre las cabeceras de capa 3 y 2, en los paquetes entrantes de la red MPLS. Las etiquetas van encapsuladas dentro de dichas cabeceras, tienen valor local al router MPLS y cambian tras cada salto (swap) siendo eliminada al llegar al router frontera (Pop).

La etiqueta es examinada y comparada con las tablas de enrutamiento, para saber a dónde reenviar el paquete, por lo que no se examina la dirección de destino. De esta forma se consigue una mayor velocidad en el enrutamiento y se disminuye los tiempos de retardo y jitter.



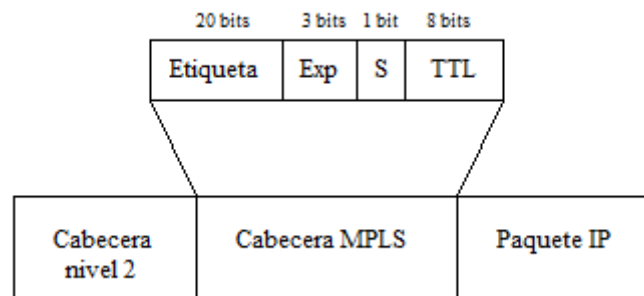


Figura 22. Formato de la cabecera MPLS

Como se puede observar, la cabecera se divide en 4 campos:

- Etiqueta: valor numérico de la etiqueta.
- Exp: identifica la clase de servicio (CoS).
- S: referente a la pila de etiquetas. Posee valor 1 o 0 según si hay 1 o más etiquetas apiladas.
- TTL: tiempo de vida del paquete antes de ser descartado por la red.

### 3.3 Elementos MPLS

#### 3.3.1 Forwarding Equivalence Class (FEC)

Conjunto de paquetes de un mismo flujo que entran en la red, reciben la misma etiqueta y circulan por el mismo camino con igual prioridad y tratamiento.

#### 3.3.2 Label Switched Path (LSP)

Camino que siguen los paquetes pertenecientes a un determinado FEC. Están formados por uno o varios LSR y son unidireccionales, transmiten tráfico en un único sentido.

Son creados por protocolos de distribución de etiquetas y se pueden establecer de dos maneras: Punto a punto o manualmente (explícita).

#### 3.3.3 Label Switch Routers (LSR)

Elemento que conmuta etiquetas. Dos tipos de nodos: **LSR Core (LSR)** situados en el núcleo de la red MPLS y **LSR Edge (LER)** o routers frontera.

El LSR recibe paquetes etiquetados, les intercambia la etiqueta (label swapping) y reenvía al siguiente LSR, según la información de las tablas LIB y LFIB.

- **LIB**: tabla de rutas que se actualiza según los protocolos de routing y es obtenida mediante el LDP.
- **LFIB**: tabla que asocia etiquetas con sus destinos o rutas y el interfaz de salida del router, indicando si tiene que poner o quitar etiqueta.

Otra función de los LSRs, es el mantenimiento de la tabla **RIB** (Routing Information Base) creada por el protocolo de enrutamiento usado.

#### 3.3.4 Label Edge Routers (LER)

Routers situados en el borde de la red MPLS, que asignan o eliminan etiquetas de los paquetes según la información que lleven. Ingress si es de entrada y Egress si es de salida.

Realizan las mismas funciones que un LSR, y también recibe, analiza y envía paquetes IP eliminando etiquetas MPLS.

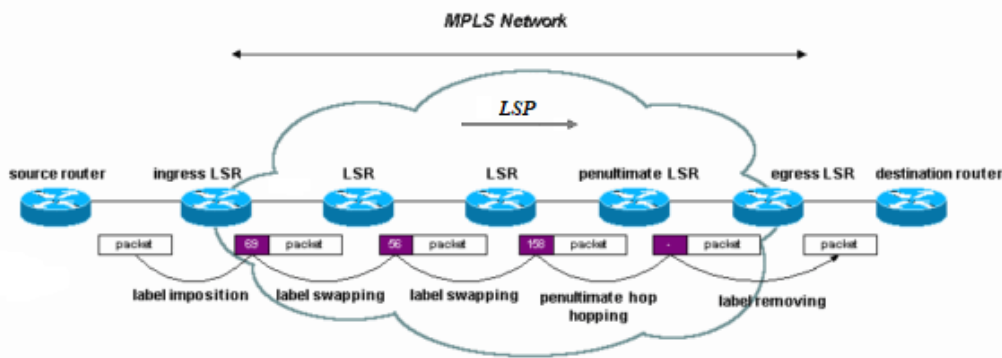


Figura 23. Topología red MPLS

### 3.4 Distribución de etiquetas

En MPLS es necesario un mecanismo o protocolo que distribuya etiquetas entre los nodos de la red y que establezca un LSP para un FEC específico por donde el LER de entrada reenviará los paquetes entrantes hacia ese FEC.

Para la asociación de etiquetas un LSR puede usar dos técnicas:

- Bajo demanda: un LSR solicita explícitamente una asociación de etiquetas a su siguiente salto o vecino downstream.
- No solicitado: no existe ninguna petición. Un LSR anuncia a todos los vecinos independientemente de sus posiciones para un particular FEC.

Existen varios mecanismos para la distribución:

- LDP (Label Distribution Protocol): protocolo de distribución de etiquetas basado en el enrutamiento IP.
- CD-LDP: protocolo derivado de LDP basado en restricciones de QoS.
- RSVP-TE (RSVP Traffic Engineering): protocolo de señalización y reserva de recursos que soporta Ingeniería de Tráfico.
- MP-BGP.

En el caso de la primera práctica, utilizaremos LDP.

**LDP** es un protocolo que establece y mantiene asociaciones de etiquetas para un LSP asociado a un FEC. Mediante este protocolo los LSRs intercambian información para alcanzar otros nodos y las etiquetas usadas para ello.

Las sesiones LDP se establecen entre parejas de LSRs (LDP Peers). Para ello, el LDP trata de descubrir peers mediante el envío de un mensaje "Hello" (multicast 224.0.0.2) utilizando el puerto UDP 646.

Una vez hayan sido descubiertos dos LSRs vecinos, realizarán un proceso de negociación para el establecimiento de la sesión LDP entre ellos. Usando el puerto TCP 646 y aportando fiabilidad a la red.

Ambos routers intercambian mensajes de inicialización y mapas de etiquetas tras recibir el primer "KeepAlive". Estos mensajes son temporizadores enviados para monitorizar la sesión LDP y mantener la conexión activa.

Cuando las sesiones LDP han sido establecidas, comienza la distribución de etiquetas y se crean los caminos (LSP) escogidos por el protocolo de encaminamiento (OSPF en nuestro caso).

Los LSRs anuncian las direcciones de sus interfaces con mensajes "Address", o retiran las ya anunciadas con "Address Withdraw". Tras estos mensajes, se envían entre ellos "Label Request"

para solicitar el mapeado de un FEC (un FEC puede ser una IP de un LSR) y responden con “Label Mapping”, anunciando el mapeado de una etiqueta al FEC.

Al distribuir las etiquetas junto a los prefijos o direcciones IP, los routers construyen las tablas LIB y FIB.

Los mensajes LDP se pueden clasificar en cuatro tipos:

- **Descubrimiento:** son enviados periódicamente para indicar la presencia de LSRs mediante mensajes UDP de “Hello”.
- **Sesión:** establecen y mantienen la sesión LDP entre peers. En este tipo se encuentran los mensajes de establecimiento TCP, Inicialización y KeepAlive.
- **Anuncio:** informan a su vecino sobre la distribución de etiquetas a los FEC. A este grupo pertenecen los mensajes Address y Label Mapping
- **Notificación:** informan a un LDP peer de su estado o de un error.

En la siguiente figura se observa cómo se establece la sesión LDP y se clasifican sus mensajes explicado anteriormente.

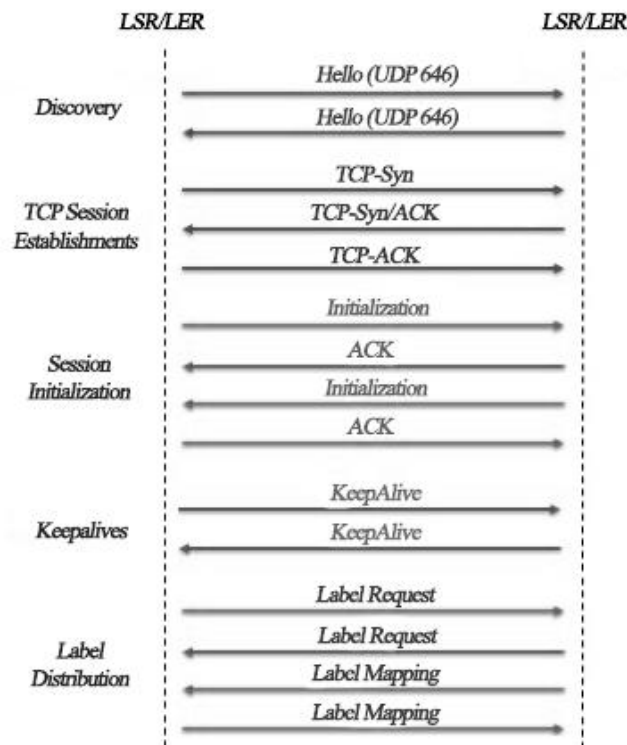


Figura 24. Operaciones LDP

### 3.5 Objetivos

El objetivo principal de la primera práctica es familiarizarse con el protocolo MPLS realizando una configuración básica de una red MPLS en un entorno simulado (GNS3).

Posteriormente se realizarán una serie de actividades con las herramientas que nos proporciona GNS3 y Wireshark para visualizar el correcto funcionamiento y distinguir los conceptos de MPLS explicados en la teoría anteriormente: tablas MPLS, campos de las cabeceras, establecimiento de sesión LDP y distribución de etiquetas.

### 3.6 Elementos necesarios

Para realizar la práctica crearemos una red L3 MPLS formada por 6 routers c7200 y 2 PCs virtuales en GNS3.

Los routers de la serie Cisco 7200, ya instalados previamente, nos aportarán flexibilidad con un mayor número de interfaces y permitirán aplicar MPLS.

Desplazaremos los routers y los VPCS desde sus correspondientes grupos en la barra de elementos al área de trabajo. Haciendo doble click encima de cada nodo o botón derecho y *Configure*, se abrirá la ventana de configuración de sus propiedades.

En la pestaña *General* escribiremos el nombre de cada nodo, evitando usar el comando *hostname* en la ventana de comandos. En *Memories and Disks* se puede cambiar el tamaño de la memoria RAM y NVRAM, en nuestro caso la dejaremos como está, y deshabilitaremos la opción del borrado automático de la memoria y los archivos de disco.

En *Slots* elegiremos los tipos y la cantidad de puertos que llevará el router. El slot 0 es el adaptador principal, pudiendo elegir entre 1 o puertos FastEthernet. El resto de slots es equivalente a añadir una tarjeta de expansión con puertos a un router.

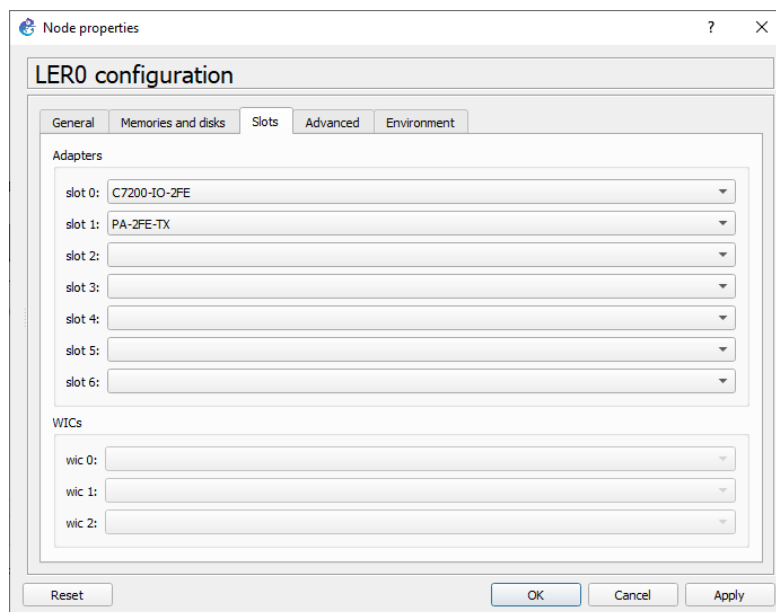


Figura 25. Ventana Slots

En el Slot principal seleccionamos 2 FastEthernet y añadiremos otro adaptador de puerto con 1 o 2 interfaces FE más dependiendo de la topología. Los routers C7200 también permiten en los slots adicionales puertos serie (PA-8T), GigaEthernet (PA-GE) o ATM (PA-A1) entre otros.

Una vez configurado, haremos click en *Apply* y *OK*. Cuando todos los routers estén configurados, procederemos a enlazarlos con el botón “*Add a link*” de la barra de elementos. Clickando en cada nodo seleccionamos los puertos a unir.

Además, la barra de herramientas de GNS3 permite mostrar en el área de trabajo los identificadores de los interfaces, escribir las direcciones red o cualquier dato necesario, dibujar figuras geométricas, hacer capturas de pantalla, etc.

### 3.7 Topología de red

La red a diseñar es la siguiente:

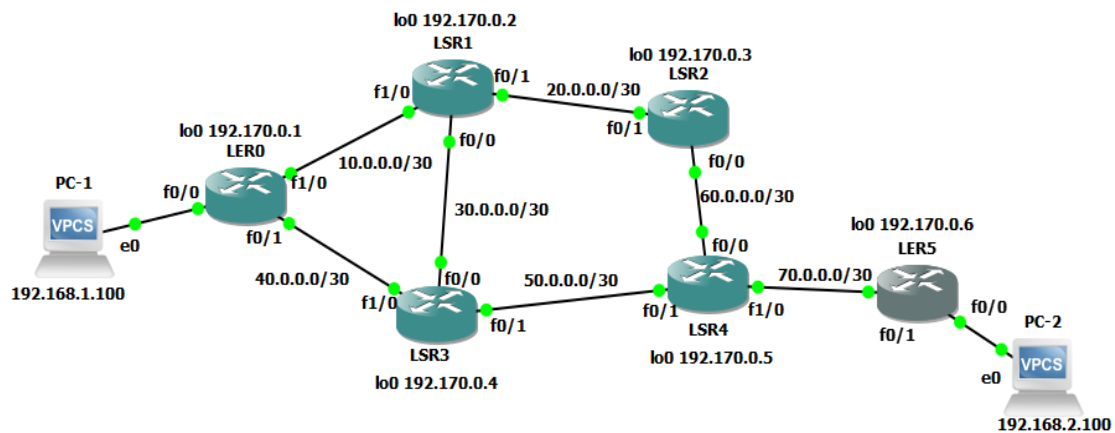


Figura 26. Diagrama de la red MPLS

A continuación se muestran las direcciones IP y máscaras de red pertenecientes a cada interfaz de los distintos equipos de la red.

Dispositivo	Interfaz	Dirección IP	Máscara de red	Gateway predeterminado
LER0	Lo0	192.170.0.1	255.255.255.255	N/A
	Fa0/0	192.168.1.1	255.255.255.0	N/A
	Fa0/1	40.0.0.1	255.255.255.252	N/A
	Fa1/0	10.0.0.1	255.255.255.252	N/A
LSR1	Lo0	192.170.0.2	255.255.255.255	N/A
	Fa0/0	30.0.0.1	255.255.255.252	N/A
	Fa0/1	20.0.0.1	255.255.255.252	N/A
	Fa1/0	10.0.0.2	255.255.255.252	N/A
LSR2	Lo0	192.170.0.3	255.255.255.255	N/A
	Fa0/0	60.0.0.1	255.255.255.252	N/A
	Fa0/1	20.0.0.2	255.255.255.252	N/A
LSR3	Lo0	192.170.0.4	255.255.255.255	N/A
	Fa0/0	30.0.0.2	255.255.255.252	N/A
	Fa0/1	50.0.0.1	255.255.255.252	N/A
	Fa1/0	40.0.0.2	255.255.255.252	N/A
LSR4	Lo0	192.170.0.5	255.255.255.255	N/A
	Fa0/0	60.0.0.2	255.255.255.252	N/A
	Fa0/1	50.0.0.2	255.255.255.252	N/A
	Fa1/0	70.0.0.2	255.255.255.252	N/A
LER5	Lo0	192.170.0.6	255.255.255.255	N/A
	Fa0/0	192.168.2.1	255.255.255.0	N/A
	Fa0/1	70.0.0.1	255.255.255.252	N/A
PC-1	NIC	192.168.1.100	255.255.255.0	192.168.1.1
PC-2	NIC	192.168.2.100	255.255.255.0	192.168.2.1

Tabla 1. Tabla de direccionamiento

### 3.8 Configuración de la red

Vamos a realizar la configuración de los distintos equipos que conforman la red.

Una vez esté la red inicializada (todas las luces de color verde), haga doble click en cada uno de los equipos para abrir el terminal donde se introducirán los comandos de la configuración. Aparecerá directamente en modo privilegiado (#), si no fuese así ejecute el comando *enable*.

Como el proyecto es nuevo y los equipos acaban de ser instalados en la red, no es necesario borrar sus configuraciones. En el caso de que la red se diese ya creada, se tendría que borrar cualquier configuración existente en los routers con el comando *delete nvram:startup-config*, para poder



asegurarnos de que no haya interferencias con la configuración a realizar. Y posteriormente reiniciar la red con el botón *reload*.

Antes de comenzar con la asignación de las direcciones de los routers, introduciremos unos comandos para evitar que aparezcan mensajes inesperados en pantalla desplazando los comandos que estamos escribiendo o revisando.

```
LSR1#configure terminal
LSR1(config)#line console 0
LSR1(config-line)#logging synchronous
```

### 3.8.1 Asignación de direcciones

Procederemos con la propia configuración de los equipos. En primer lugar, asignaremos las direcciones ip y sus máscaras de cada uno de los interfaces junto con las direcciones de loopback de los routers. Los loopback son interfaces de red virtual usados para identificar a cada nodo ante cualquier protocolo, como OSPF, LDP o BGP.

Para el LSR3 los comandos son los siguientes:

```
LSR3#configure terminal
LSR3(config)#interface loopback0
LSR3(config-if)#ip address 192.170.0.4 255.255.255.255
LSR3(config)#interface fa0/0
LSR3(config-if)#ip add 40.0.0.2 255.255.255.252
LSR3(config-if)#no shutdown
LSR3(config-if)#int fa1/0
LSR3(config-if)#ip add 50.0.0.1 255.255.255.252
LSR3(config-if)#no shutdown
LSR3(config-if)#int fa0/1
LSR3(config-if)#ip add 30.0.0.2 255.255.255.252
LSR3(config-if)#no shutdown
LSR3(config-if)#exit
```

Repita estos pasos en todos los routers con sus respectivas direcciones.

Para los dos PCs virtuales habrá que introducir la dirección ip, su máscara y puerta de enlace o Gateway. Posteriormente guardaremos su configuración.

```
PC-1> ip 192.168.1.100 255.255.255.255 192.168.1.1
PC-1> save
```

### 3.8.2 Protocolo OSPF

Una vez estén todas las direcciones asignadas, configuraremos el protocolo de routing OSPF, siendo uno de los protocolos más extendidos y sencillos de aplicar por su escalabilidad.

Para activar el protocolo OSPF de manera global, se usará el comando **“router ospf id proceso”**. La id del proceso es una variable que identifica al proceso en ejecución dentro del CISCO IOS. Aunque no es necesario que la variable sea idéntica en todos los routers de la red, utilizaremos la misma para que sea más simple.

Con el comando “**Network dir\_ip wildcard area num\_area**” habilitaremos el protocolo en todos los interfaces del router, añadiendo las redes a las que pertenecen. La máscara Wildcard es la inversa de una máscara de red. Los bits que están a cero son los bits de la dirección de red que se tomarán en cuenta y los bits puestos a uno (255) no. El argumento “area num\_area” indica el área OSPF a la que van a pertenecer los interfaces. Es recomendable utilizar como número de área el 0 y que sea el mismo para todos.

```
LER0(config)#router ospf 1
LER0(config-router)#network 10.0.0.0 0.0.0.3 area 0
LER0(config-router)#network 40.0.0.0 0.0.0.3 area 0
LER0(config-router)#network 192.170.0.1 0.0.0.0 area 0
LER0(config-router)#network 192.168.1.0 0.0.0.255 area 0
```

Cuando todos los equipos de la red tengan configurado el protocolo OSPF, deberíamos poder acceder a todos los equipos conectados a la red desde cualquier otro dispositivo de ésta. Para comprobarlo haga ping o traceroute desde los diferentes equipos y compruebe que tiene acceso a los interfaces y servidores de la red.

### 3.8.3 Configuración MPLS

Seguidamente comenzaremos con la configuración MPLS. En los dispositivos CISCO es necesario habilitar la función CEF (Cisco Express Forwarding) con la que el router construye las tablas FIB y LFIB. Se activa con “**ip cef**” en configuración global.

FIB es una tabla organizada de modo diferente que la tabla de enrutamiento, y es la que se utiliza para definir a qué interfaz se debe reenviar el paquete.

Habrá que habilitar el proceso MPLS a nivel global, indicar el protocolo de distribución de etiquetas (LDP) y establecer el rango de etiquetas en cada router con el comando “**mpls label range rango**”. En la siguiente tabla podemos ver los rangos de etiquetas que asignará cada router a los paquetes:

Router	Rango
LER0	16-99
LSR1	100-199
LSR2	200-299
LSR3	300-399
LSR4	400-499
LER5	500-599

Tabla 2. Rango de etiquetas a limitar

El rango de etiquetas 0-15 se encuentra reservado.

Posteriormente, también habrá que habilitar MPLS de manera individual en cada uno de los interfaces pertenecientes a la red, sin contar los loopback. Si “**mpls ip**” no se habilitase en los interfaces, las adyacencias no se formarían.

Los comandos para habilitar MPLS en el router LSR4 son:



```
LSR4(config)#ip cef
LSR4(config)#mpls ip
LSR4(config)#mpls label protocol ldp
LSR4(config)#mpls label range 400 499
LSR4(config)#int fa0/0
LSR4(config-if)#mpls ip
LSR4(config-if)#int fa0/1
LSR4(config-if)#mpls ip
LSR4(config-if)#int fa1/0
LSR4(config-if)#mpls ip
```

Repetir los pasos anteriores en cada router de la red.

Introduciendo “*show mpls interfaces*” podemos comprobar si se ha habilitado MPLS correctamente en cada uno de los interfaces y si se encuentran operativos.

```
LER0#sh mpls interfaces
Interface          IP          Tunnel  Operational
FastEthernet0/0    Yes (ldp)   No      Yes
FastEthernet0/1    Yes (ldp)   No      Yes
FastEthernet1/0    Yes (ldp)   No      Yes
```

Figura 27. Interfaces MPLS

Las características y parámetros de la sesión LDP se consultan con “*show mpls ldp parameters*”.

```
LER0#show mpls ldp parameters
Protocol version: 1
Downstream label generic region: min label: 16; max label: 99
Session hold time: 180 sec; keep alive interval: 60 sec
Discovery hello: holdtime: 15 sec; interval: 5 sec
Discovery targeted hello: holdtime: 90 sec; interval: 10 sec
Downstream on Demand max hop count: 255
Downstream on Demand Path Vector Limit: 255
LDP for targeted sessions
LDP initial/maximum backoff: 15/120 sec
LDP loop detection: off
```

Figura 28. Parámetros sesión LDP

El router mantiene otro temporizador, a parte del “*KeepAlive*”, para cada sesión establecida. Es reiniciado cuando llega un nuevo mensaje “*Hello*” y si el tiempo expira se cierra la sesión LDP.

### 3.8.4 Verificación final

Para verificar si la red está correctamente configurada utilizaremos diversos comandos:

“*Show mpls ldp discovery*” muestra en pantalla la información del LDP y del descubrimiento de los vecinos, como el identificativo del router (direcciones de loopback) y los interfaces.

```
LSR1#sh mpls ldp discovery
Local LDP Identifier:
192.170.0.2:0
Discovery Sources:
Interfaces:
  FastEthernet0/0 (ldp): xmit/rcv
    LDP Id: 192.170.0.4:0
  FastEthernet0/1 (ldp): xmit/rcv
    LDP Id: 192.170.0.3:0
  FastEthernet1/0 (ldp): xmit/rcv
    LDP Id: 192.170.0.1:0
```

Figura 29. Adyacencias LDP

“*Show mpls ldp neighbor*” proporciona información sobre el estado de las conexiones LDP establecidas con los peer vecinos.

```
LSR3#sh mpls ldp neighbor
Peer LDP Ident: 192.170.0.1:0; Local LDP Ident 192.170.0.4:0
TCP connection: 192.170.0.1.646 - 192.170.0.4.58258
State: Oper; Msgs sent/rcvd: 38/38; Downstream
Up time: 00:17:42
LDP discovery sources:
  FastEthernet1/0, Src IP addr: 40.0.0.1
Addresses bound to peer LDP Ident:
  192.168.1.1    192.170.0.1    40.0.0.1    10.0.0.1
Peer LDP Ident: 192.170.0.2:0; Local LDP Ident 192.170.0.4:0
TCP connection: 192.170.0.2.646 - 192.170.0.4.39966
State: Oper; Msgs sent/rcvd: 37/38; Downstream
Up time: 00:17:36
LDP discovery sources:
  FastEthernet0/0, Src IP addr: 30.0.0.1
Addresses bound to peer LDP Ident:
  30.0.0.1    192.170.0.2    20.0.0.1    10.0.0.2
Peer LDP Ident: 192.170.0.5:0; Local LDP Ident 192.170.0.4:0
TCP connection: 192.170.0.5.48665 - 192.170.0.4.646
State: Oper; Msgs sent/rcvd: 37/37; Downstream
Up time: 00:17:26
LDP discovery sources:
  FastEthernet0/1, Src IP addr: 50.0.0.2
Addresses bound to peer LDP Ident:
  60.0.0.2    192.170.0.5    50.0.0.2    70.0.0.2
```

Figura 30. Conexiones Peer LDP

En este caso el LSR3 ha establecido sesión TCP y LDP con sus vecinos LER0, LSR1 y LSR5.

Con “*show mpls forwarding table*” podemos consultar la tabla LFIB de cada router y con “*show mpls ldp bindings*” la tabla LIB.

Por último, y una vez se haya verificado el correcto funcionamiento de la red, procederemos a guardar las configuraciones de los routers introduciendo el comando “*copy running-config startup-config*”.

Al igual que sucede con los equipos reales, GNS3 no guarda las configuraciones de sus equipos automáticamente, solo las topologías, por lo que esta función es necesaria si queremos volver a utilizar la red configurada o se produce cualquier imprevisto con el simulador.

```
LSR2#copy running-config startup-config
Destination filename [startup-config]?
```

### 3.9 Actividades propuestas

#### 1. Ejecute el comando traceroute al PC-2 desde LER0. Comente la ruta creada.

Con el comando traceroute se muestra la ruta que siguen los paquetes enviados. El camino que se ha creado es la ruta más corta calculada por el protocolo IGP establecido.

En dicha ruta se observa el uso de dos etiquetas.

```
LER0#traceroute 192.168.2.100

Type escape sequence to abort.
Tracing the route to 192.168.2.100

 1 40.0.0.2 [MPLS: Label 309 Exp 0] 104 msec 96 msec 88 msec
 2 50.0.0.2 [MPLS: Label 409 Exp 0] 96 msec 88 msec 96 msec
 3 70.0.0.1 96 msec 88 msec 84 msec
 4 192.168.2.100 144 msec 104 msec 76 msec
LER0#
```

Figura 31. Traceroute a PC-2

- 1) LER0 envía tramas con la etiqueta MPLS 309 para alcanzar LSR3.
- 2) LSR3 reemplaza la etiqueta 309 por la 409 y envía paquetes con dicha etiqueta.
- 3) LSR4 retira la etiqueta 409 por ser el **PHP** (Penultime Hop Popping). El siguiente LSR (LER5) tiene el destino directamente conectado, por lo que no se utiliza etiqueta. Se usa para evitar consultas innecesarias en la tabla LIB.

#### 2. Mediante los comandos “show mpls forwarding table” y “show mpls ldp bindings” y la topología de la red, construya las distintas tablas de MPLS (RIB, FIB, LIB y LFIB) del router LER0 de manera similar a las actividades realizadas en las clases teóricas.

\* Las etiquetas pueden variar cada vez que se inicializa el router.

La tabla RIB se realiza fácilmente siguiendo la topología y la tabla de direccionamiento de la red, añadiendo las distintas redes a las que no está directamente conectado el LER0 y el siguiente salto por dónde reenviará los paquetes dirigidos a dichas redes.

RED	SALTO
20.0.0.0/30	LSR1
30.0.0.0/30	LSR1
	LSR3
50.0.0.0/30	LSR3
60.0.0.0/30	LSR1
	LSR3
70.0.0.0/30	LSR3
192.168.2.0/24	LSR3

Tabla 3. RIB

“Show mpls ldp bindings” es usado para visualizar los datos de la tabla LIB. (No se tendrán en cuenta las direcciones loopback)

```
LER0#show mpls ldp bindings
tib entry: 10.0.0.0/30, rev 6
  local binding: tag: imp-null
  remote binding: tsr: 192.170.0.2:0, tag: imp-null
  remote binding: tsr: 192.170.0.4:0, tag: 301
tib entry: 20.0.0.0/30, rev 10
  local binding: tag: 16
  remote binding: tsr: 192.170.0.2:0, tag: imp-null
  remote binding: tsr: 192.170.0.4:0, tag: 300
tib entry: 30.0.0.0/30, rev 12
  local binding: tag: 17
  remote binding: tsr: 192.170.0.2:0, tag: imp-null
  remote binding: tsr: 192.170.0.4:0, tag: imp-null
tib entry: 40.0.0.0/30, rev 4
  local binding: tag: imp-null
  remote binding: tsr: 192.170.0.2:0, tag: 100
  remote binding: tsr: 192.170.0.4:0, tag: imp-null
tib entry: 50.0.0.0/30, rev 16
  local binding: tag: 19
  remote binding: tsr: 192.170.0.4:0, tag: imp-null
  remote binding: tsr: 192.170.0.2:0, tag: 103
tib entry: 60.0.0.0/30, rev 18
  local binding: tag: 20
  remote binding: tsr: 192.170.0.2:0, tag: 104
  remote binding: tsr: 192.170.0.4:0, tag: 305
tib entry: 70.0.0.0/30, rev 25
  local binding: tag: 23
  remote binding: tsr: 192.170.0.4:0, tag: 307
  remote binding: tsr: 192.170.0.2:0, tag: 107
tib entry: 192.168.1.0/24, rev 2
  local binding: tag: imp-null
  remote binding: tsr: 192.170.0.2:0, tag: 101
  remote binding: tsr: 192.170.0.4:0, tag: 302
tib entry: 192.168.2.0/24, rev 28
  local binding: tag: 25
  remote binding: tsr: 192.170.0.4:0, tag: 309
  remote binding: tsr: 192.170.0.2:0, tag: 109
```

Figura 32. Tabla LIB

RED	LSR	TAG
20.0.0.0/30	LOCAL	16
	LSR1	-
30.0.0.0/30	LOCAL	17
	LSR1	-
	LSR3	-
50.0.0.0/30	LOCAL	19
	LSR3	-
60.0.0.0/30	LOCAL	20
	LSR1	104
	LSR3	305
70.0.0.0/30	LOCAL	23
	LSR3	307
192.168.2.0/24	LOCAL	25
	LSR3	309

Tabla 4. LIB

*Implicit-Null* aparece en los nodos conectados directamente e indica que el paquete es reenviado con prefijo IP y no con etiqueta MPLS.

*Local binding* se refiere a cómo el router (en este caso el LER0) quiere que se etiqueten los paquetes que van hacia una red en particular cuando pasan por él.

*Remote binding* se refiere a cómo el próximo salto quiere que sean etiquetados los paquetes que van hacia una red en particular cuando vayan a pasar por él.

La tabla FIB es la tabla de rutas basándose en la función CEF.

RED	SALTO	TAG
20.0.0.0/30	LSR1	-
30.0.0.0/30	LSR1	-
	LSR3	-
50.0.0.0/30	LSR3	-
60.0.0.0/30	LSR1	104
	LSR3	305
70.0.0.0/30	LSR3	307
192.168.2.0/24	LSR3	309

Tabla 5. FIB

La tabla LFIB se consulta con “*show mpls forwarding-table*”:

```
LER0#show mpls forwarding-table
Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC  or Tunnel Id    switched   interface
16     Pop tag    20.0.0.0/30    0          Fa1/0     10.0.0.2
17     Pop tag    30.0.0.0/30    0          Fa0/1     40.0.0.2
      Pop tag    30.0.0.0/30    0          Fa1/0     10.0.0.2
18     Pop tag    192.170.0.2/32 0          Fa1/0     10.0.0.2
19     Pop tag    50.0.0.0/30    0          Fa0/1     40.0.0.2
20     305       60.0.0.0/30    0          Fa0/1     40.0.0.2
      104       60.0.0.0/30    0          Fa1/0     10.0.0.2
21     105       192.170.0.3/32 0          Fa1/0     10.0.0.2
22     Pop tag    192.170.0.4/32 0          Fa0/1     40.0.0.2
23     307       70.0.0.0/30    0          Fa0/1     40.0.0.2
24     308       192.170.0.5/32 0          Fa0/1     40.0.0.2
25     309       192.168.2.0/24 0          Fa0/1     40.0.0.2
26     310       192.170.0.6/32 0          Fa0/1     40.0.0.2
```

Figura 33. Tabla LFIB

“Outgoing tag or VC” se refiere a la acción realizada por dicho router. Si aparece el valor de una etiqueta, la acción es un swap o intercambio de etiqueta. Pop si elimina la etiqueta.

TAG	ACTION	SALTO
16	POP	LSR3
17	POP	LSR1
		LSR3
19	POP	LSR3
20	104	LSR1
	305	LSR3
23	307	LSR3
25	309	LSR3

Tabla 6. LFIB

3. Utilizando Wireshark y las funcionalidades que nos permite GNS3, monitorizaremos el tráfico que circula por nuestra red. En este caso en el enlace que une LER0 con LSR3. Haga click derecho sobre el enlace y seleccione *Start Capture*. Aparecerá un icono de una lupa sobre este enlace y se abrirá Wireshark automáticamente. (Si no aparece, vuelva a hacer click derecho en el enlace y seleccione *Start Wireshark*)

**Ejecute un ping desde PC-1 a PC-2. ¿Qué paquetes ICMP salen encapsulados en MPLS? Observe los distintos campos MPLS.**

No.	Time	Source	Destination	Protocol	Length	Info
4	2.967522	192.168.1.100	192.168.2.100	ICMP	102	Echo (ping) request id=0x766a, seq=1/256, ttl=63 (reply in 5)
5	3.057444	192.168.2.100	192.168.1.100	ICMP	102	Echo (ping) reply id=0x766a, seq=1/256, ttl=63 (request in 4)
7	4.109747	192.168.1.100	192.168.2.100	ICMP	102	Echo (ping) request id=0x786a, seq=2/512, ttl=63 (reply in 8)
8	4.163109	192.168.2.100	192.168.1.100	ICMP	102	Echo (ping) reply id=0x786a, seq=2/512, ttl=63 (request in 7)
11	5.217333	192.168.1.100	192.168.2.100	ICMP	102	Echo (ping) request id=0x796a, seq=3/768, ttl=63 (reply in 11)
12	5.286244	192.168.2.100	192.168.1.100	ICMP	102	Echo (ping) reply id=0x796a, seq=3/768, ttl=63 (request in 12)
13	6.335131	192.168.1.100	192.168.2.100	ICMP	102	Echo (ping) request id=0x7a6a, seq=4/1024, ttl=63 (reply in 14)
14	6.382579	192.168.2.100	192.168.1.100	ICMP	102	Echo (ping) reply id=0x7a6a, seq=4/1024, ttl=63 (request in 13)
17	7.425395	192.168.1.100	192.168.2.100	ICMP	102	Echo (ping) request id=0x7b6a, seq=5/1280, ttl=63 (reply in 18)
18	7.516254	192.168.2.100	192.168.1.100	ICMP	102	Echo (ping) reply id=0x7b6a, seq=5/1280, ttl=63 (request in 17)

Figura 34. Captura paquetes ICMP

```

> Frame 95: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on int
▼ Ethernet II, Src: ca:01:13:1c:00:06 (ca:01:13:1c:00:06), Dst: ca:04:20:1c:00:06
  > Destination: ca:04:20:1c:00:1c (ca:04:20:1c:00:1c)
  > Source: ca:01:13:1c:00:06 (ca:01:13:1c:00:06)
  Type: MPLS label switched packet (0x8847)
▼ MultiProtocol Label Switching Header, Label: 309, Exp: 0, S: 1, TTL: 255
  0000 0000 0001 0011 0101 .... = MPLS Label: 309
  .... = MPLS Experimental Bits: 0
  .... = MPLS Bottom Of Label Stack: 1
  .... 1111 1111 = MPLS TTL: 255
▼ Internet Protocol Version 4, Src: 40.0.0.1, Dst: 192.168.2.100
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 100
  Identification: 0x0210 (528)
  > Flags: 0x0000
  Time to live: 255
  Protocol: ICMP (1)
  Header checksum: 0xce7b [validation disabled]
  [Header checksum status: Unverified]
  Source: 40.0.0.1
  Destination: 192.168.2.100
  > Internet Control Message Protocol
  
```

Figura 35. Paquete MPLS enviado por LER0

Los paquetes ICMP Echo Request van encapsulados en IP, que a su vez van dentro de MPLS.

El EtherType, en la cabecera Ethernet, tiene un valor de 0x8847, que corresponde con la trama Ethernet Unicast que lleva MPLS.

Dentro de la cabecera MPLS, de 32 bits, se encuentran 4 campos:

- MPLS Label: valor local de la etiqueta que el router ha asignado al paquete. Al ser un LER de entrada ha añadido una etiqueta al paquete IP. En este caso su valor es 409.
- Exp o bits experimentales: 3 bits que identifican la clase de servicio (CoS). Es 0, por lo que no se están usando y el paquete no tiene ninguna prioridad.
- S o Stack: indica si hay etiquetas apiladas. Como no hay túneles, S = 1, siendo la última etiqueta en la pila.
- TTL: tiempo de vida de un paquete antes de ser descartado de la red. El valor es de 255 porque no ha habido ningún salto anterior.

#### 4. Monitorice esta vez el enlace LSR3-LSR4. ¿Qué diferencias observa en los paquetes enviados por el LSR3 respecto a los enviados por el LER0?

```
> Frame 17: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interface 0
  > Ethernet II, Src: ca:04:20:1c:00:06 (ca:04:20:1c:00:06), Dst: ca:05:1f:cc:00:06 (ca:05:1f:cc:00:06)
    > Destination: ca:05:1f:cc:00:06 (ca:05:1f:cc:00:06)
    > Source: ca:04:20:1c:00:06 (ca:04:20:1c:00:06)
    Type: MPLS Label switched packet (0x8847)
  > MultiProtocol Label Switching Header, Label: 409, Exp: 0, S: 1, TTL: 62
    0000 0000 0001 1001 1001 .... = MPLS Label: 409
    .... = MPLS Experimental Bits: 0
    .... = MPLS Bottom Of Label Stack: 1
    .... 0011 1110 = MPLS TTL: 62
  > Internet Protocol Version 4, Src: 192.168.1.100, Dst: 192.168.2.100
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 84
    Identification: 0x6a7a (27258)
    > Flags: 0x0000
    Time to live: 63
    Protocol: ICMP (1)
    Header checksum: 0x8c16 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.100
    Destination: 192.168.2.100
  > Internet Control Message Protocol
```

Figura 36. Paquete MPLS enviado por LSR3

El LER añade una etiqueta al paquete IP, mientras que el LSR ha intercambiado esa etiqueta por otra (Swap 409).

El valor del TTL no coincide con el TTL del paquete IP. Al haber un salto desde el inicio del LSP, el TTL ha disminuido en 1 (62).

5. Para ver cómo trabaja el protocolo de distribución de etiquetas, eliminaremos primero las sesiones establecidas anteriormente con el comando “*Clear mpls ldp neighbor \**” en el LSR3. Tras introducir el comando, el LSR3 reestablecerá las sesiones LDP con sus vecinos.

Mientras, capture el tráfico con el Wireshark en el enlace LER0-LSR3 e identifique los mensajes de la negociación.

No.	Time	Source	Destination	Protocol	Length	Info
18	13.233217	40.0.0.2	224.0.0.2	LDP	76	Hello Message
20	14.107000	40.0.0.1	224.0.0.2	LDP	76	Hello Message
21	14.120741	192.170.1.4	192.170.1.1	TCP	60	42206 → 646 [SYN] Seq=0 Win=4128 Len=0 MSS=536
22	14.128136	192.170.1.1	192.170.1.4	TCP	60	646 → 42206 [SYN, ACK] Seq=0 Ack=1 Win=4128 Len=0 MSS=536
23	14.131307	192.170.1.4	192.170.1.1	TCP	60	42206 → 646 [ACK] Seq=1 Ack=1 Win=4128 Len=0
24	14.173549	192.170.1.4	192.170.1.1	LDP	90	Initialization Message
25	14.183063	192.170.1.1	192.170.1.4	TCP	60	646 → 42206 [ACK] Seq=1 Ack=37 Win=4092 Len=0
26	14.193633	192.170.1.1	192.170.1.4	LDP	98	Initialization Message Keep Alive Message
27	14.205258	192.170.1.4	192.170.1.1	TCP	60	42206 → 646 [ACK] Seq=37 Ack=45 Win=4084 Len=0
28	14.247544	192.170.1.4	192.170.1.1	LDP	530	Address Message Label Mapping Message Label Mapping Message Label Mapping Message Label Mapping Message Label Mapping M...
29	14.257016	192.170.1.1	192.170.1.4	TCP	60	646 → 42206 [ACK] Seq=45 Ack=513 Win=3616 Len=0
30	14.267586	192.170.1.1	192.170.1.4	LDP	512	Address Message Label Mapping Message Label Mapping Message Label Mapping Message Label Mapping Message Label Mapping M...
31	14.279216	192.170.1.4	192.170.1.1	TCP	60	42206 → 646 [ACK] Seq=513 Ack=503 Win=3626 Len=0
34	17.682114	40.0.0.2	224.0.0.2	LDP	76	Hello Message
35	18.107924	40.0.0.1	224.0.0.2	LDP	76	Hello Message
36	21.885164	40.0.0.2	224.0.0.2	LDP	76	Hello Message

Figura 37. Captura paquetes establecimiento LDP

Primero se envían mensajes Hello periódicamente para buscar LSR peers.

Tras encontrar 2 LSR vecinos. Se produce el establecimiento de la conexión TCP por el puerto 646. Podemos identificar 3 tipos de mensajes:

```
Wireshark · Packet 21 · -
> Frame 21: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
> Ethernet II, Src: ca:04:20:1c:00:1c (ca:04:20:1c:00:1c), Dst: ca:01:13:1c:00:06 (ca:01:13:1c:00:06)
> Internet Protocol Version 4, Src: 192.170.1.4, Dst: 192.170.1.1
▼ Transmission Control Protocol, Src Port: 42206, Dst Port: 646, Seq: 0, Len: 0
  Source Port: 42206
  Destination Port: 646
  [Stream index: 1]
  [TCP Segment Len: 0]
  Sequence number: 0 (relative sequence number)
  [Next sequence number: 0 (relative sequence number)]
  Acknowledgment number: 0
  0110 .... = Header Length: 24 bytes (6)
  > Flags: 0x002 (SYN)
    Window size value: 4128
    [Calculated window size: 4128]
    Checksum: 0x0e8e [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  > Options: (4 bytes), Maximum segment size
  > [Timestamps]
```

Figura 38. TCP-SYN Message

```
Wireshark · Packet 22 · -
> Frame 22: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
> Ethernet II, Src: ca:01:13:1c:00:06 (ca:01:13:1c:00:06), Dst: ca:04:20:1c:00:1c (ca:04:20:1c:00:1c)
> Internet Protocol Version 4, Src: 192.170.1.1, Dst: 192.170.1.4
▼ Transmission Control Protocol, Src Port: 646, Dst Port: 42206, Seq: 0, Ack: 1, Len: 0
  Source Port: 646
  Destination Port: 42206
  [Stream index: 1]
  [TCP Segment Len: 0]
  Sequence number: 0 (relative sequence number)
  [Next sequence number: 0 (relative sequence number)]
  Acknowledgment number: 1 (relative ack number)
  0110 .... = Header Length: 24 bytes (6)
  > Flags: 0x012 (SYN, ACK)
    Window size value: 4128
    [Calculated window size: 4128]
    Checksum: 0xc4a0 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  > Options: (4 bytes), Maximum segment size
  > [SEQ/ACK analysis]
  > [Timestamps]
```

Figura 39. TCP-SYN/ACK Message



```
Wireshark · Packet 23 · -
> Frame 23: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
> Ethernet II, Src: ca:04:20:1c:00:1c (ca:04:20:1c:00:1c), Dst: ca:01:13:1c:00:06 (ca:01:13:1c:00:06)
> Internet Protocol Version 4, Src: 192.170.1.4, Dst: 192.170.1.1
▼ Transmission Control Protocol, Src Port: 42206, Dst Port: 646, Seq: 1, Ack: 1, Len: 0
  Source Port: 42206
  Destination Port: 646
  [Stream index: 1]
  [TCP Segment Len: 0]
  Sequence number: 1 (relative sequence number)
  [Next sequence number: 1 (relative sequence number)]
  Acknowledgment number: 1 (relative ack number)
  0101 ... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
  Window size value: 4128
  [Calculated window size: 4128]
  [Window size scaling factor: -2 (no window scaling used)]
  Checksum: 0xd8c1 [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  > [SEQ/ACK analysis]
  > [Timestamps]
```

Figura 40. TCP-ACK Message

Una vez se haya establecido la conexión TCP, los routers negocian los distintos parámetros de la sesión LDP usando mensajes de inicialización. Algunos parámetros son: el identificador del LSR, la versión del protocolo o la distribución de etiquetas. En este caso la distribución es no solicitada.

```
> Frame 366: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface 0
> Ethernet II, Src: ca:04:20:1c:00:1c (ca:04:20:1c:00:1c), Dst: ca:01:13:1c:00:06 (ca:01:13:1c:00:06)
> Internet Protocol Version 4, Src: 192.170.0.4, Dst: 192.170.0.1
> Transmission Control Protocol, Src Port: 52841, Dst Port: 646, Seq: 1, Ack: 1, Len: 36
▼ Label Distribution Protocol
  Version: 1
  PDU Length: 32
  LSR ID: 192.170.0.4
  Label Space ID: 0
  ▼ Initialization Message
    0... .... = U bit: Unknown bit not set
    Message Type: Initialization Message (0x200)
    Message Length: 22
    Message ID: 0x0000012b
  ▼ Common Session Parameters
    00.. .... = TLV Unknown bits: Known TLV, do not Forward (0x0)
    TLV Type: Common Session Parameters (0x500)
    TLV Length: 14
  ▼ Parameters
    Session Protocol Version: 1
    Session KeepAlive Time: 180
    0... .... = Session Label Advertisement Discipline: Downstream Unsolicited proposed
    .0.. .... = Session Loop Detection: Loop Detection Disabled
    Session Path Vector Limit: 0
    Session Max PDU Length: 0
    Session Receiver LSR Identifier: 192.170.0.1
    Session Receiver Label Space Identifier: 0
```

Figura 41. Mensaje de Inicialización LDP

Cuando una nueva sesión LDP es establecida, los LSR anuncian las direcciones de sus interfaces usando los mensajes Address.

```

> Frame 369: 530 bytes on wire (4240 bits), 530 bytes captured (4240 bits) on interface
> Ethernet II, Src: ca:04:20:1c:00:1c (ca:04:20:1c:00:1c), Dst: ca:01:13:1c:00:06 (ca:0
> Internet Protocol Version 4, Src: 192.170.0.4, Dst: 192.170.0.1
> Transmission Control Protocol, Src Port: 52841, Dst Port: 646, Seq: 37, Ack: 45, Len:
> Label Distribution Protocol
▼ Label Distribution Protocol
  Version: 1
  PDU Length: 454
  LSR ID: 192.170.0.4
  Label Space ID: 0
  ▼ Address Message
    0... .... = U bit: Unknown bit not set
    Message Type: Address Message (0x300)
    Message Length: 26
    Message ID: 0x0000012d
    ▼ Address List
      00.. .... = TLV Unknown bits: Known TLV, do not Forward (0x0)
      TLV Type: Address List (0x101)
      TLV Length: 18
      Address Family: IPv4 (1)
      ▼ Addresses
        Address 1: 30.0.0.2
        Address 2: 192.170.0.4
        Address 3: 50.0.0.1
        Address 4: 40.0.0.2
    > Label Mapping Message
    > Label Mapping Message
    > Label Mapping Message
    > Label Mapping Message
    > Label Mapping Message
  
```

Figura 42. Address Message

Dentro del mensaje Address, aparecen las secciones *Label Mapping*, en las que se muestran las asignaciones de etiquetas a los distintos FECs.

```

▼ Label Mapping Message
  0... .... = U bit: Unknown bit not set
  Message Type: Label Mapping Message (0x400)
  Message Length: 24
  Message ID: 0x00000149
  ▼ FEC
    00.. .... = TLV Unknown bits: Known TLV, do not Forward (0x0)
    TLV Type: FEC (0x100)
    TLV Length: 8
    ▼ FEC Elements
      ▼ FEC Element 1
        FEC Element Type: Prefix FEC (2)
        FEC Element Address Type: IPv4 (1)
        FEC Element Length: 30
        Prefix: 60.0.0.0
    ▼ Generic Label
      00.. .... = TLV Unknown bits: Known TLV, do not Forward (0x0)
      TLV Type: Generic Label (0x200)
      TLV Length: 4
      .... .... 0000 0000 0001 0011 0001 = Generic Label: 0x00131
  > Label Mapping Message
  > Label Mapping Message
  > Label Mapping Message
  
```

Figura 43. Label Mapping 60.0.0.0

La etiqueta asignada por el LSR3 para alcanzar la red 60.0.0.0/30 es 305 (131 en hexadecimal).  
Si observamos la tabla LFIB comprobaremos que coincide:

```
LSR3#show mpls forwarding-table
Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC  or Tunnel Id    switched   interface
300    Pop tag    20.0.0.0/30     0          Fa0/0     30.0.0.1
301    Pop tag    10.0.0.0/30     0          Fa1/0     40.0.0.1
        Pop tag    10.0.0.0/30     0          Fa0/0     30.0.0.1
302    Pop tag    192.168.1.0/24  0          Fa1/0     40.0.0.1
303    Pop tag    192.170.0.1/32  0          Fa1/0     40.0.0.1
304    Pop tag    192.170.0.2/32  0          Fa0/0     30.0.0.1
305    Pop tag    60.0.0.0/30     0          Fa0/1     50.0.0.2
306    407       192.170.0.3/32  0          Fa0/1     50.0.0.2
        105       192.170.0.3/32  0          Fa0/0     30.0.0.1
307    Pop tag    70.0.0.0/30     0          Fa0/1     50.0.0.2
308    Pop tag    192.170.0.5/32  0          Fa0/1     50.0.0.2
309    409       192.168.2.0/24  0          Fa0/1     50.0.0.2
310    410       192.170.0.6/32  0          Fa0/1     50.0.0.2
```

Figura 44. Tabla LFIB del LSR3

En las asignaciones en las que aparecen como valor de etiqueta un 3 (valor incluido en el rango reservado), indica que es una etiqueta Implicit Null y que la red destino está conectada directamente.

```

v Label Mapping Message
  0... .... = U bit: Unknown bit not set
  Message Type: Label Mapping Message (0x400)
  Message Length: 24
  Message ID: 0x00000142
  v FEC
    00.. .... = TLV Unknown bits: Known TLV, do not Forward (0x0)
    TLV Type: FEC (0x100)
    TLV Length: 8
    v FEC Elements
      v FEC Element 1
        FEC Element Type: Prefix FEC (2)
        FEC Element Address Type: IPv4 (1)
        FEC Element Length: 30
        Prefix: 40.0.0.0
      v Generic Label
        00.. .... = TLV Unknown bits: Known TLV, do not Forward (0x0)
        TLV Type: Generic Label (0x200)
        TLV Length: 4
        .... .... 0000 0000 0000 0000 0011 = Generic Label: 0x00003
  > Label Mapping Message

```

Figura 45. Label Mapping 40.0.0.0

Se comprueba que el valor Implicit Null es el mismo que aparece en la tabla LIB del LER0 en la entrada de la red 40.0.0.0/30. Esta red está conectada directamente al LER0 y LSR3.

```
LER0#sh mpls ldp bindings 40.0.0.0 30
tib entry: 40.0.0.0/30, rev 4
local binding: tag: imp-null
remote binding: tsr: 192.170.0.2:0, tag: 100
remote binding: tsr: 192.170.0.4:0, tag: imp-null
```

Figura 46. Tabla LIB de LER0 entrada 40.0.0.0

## Capítulo 4. Práctica 2 “Simulación de una red MPLS-TE”

### 4.1 Introducción

Uno de los grandes problemas de las redes IP es la congestión, ocurre cuando, durante un periodo de tiempo, la tasa de llegada de los paquetes excede la capacidad de salida y produce un aumento de los retardos de tránsito, los jitters y las pérdidas de paquetes.

Este problema es causado principalmente por la escasez de recursos en la red y el uso ineficiente de los recursos debido a los protocolos de enrutamiento, ya que éstos utilizan el algoritmo de camino más corto, sin tomar en cuenta la disponibilidad de recursos en los enlaces, produciéndose una saturación de algunos enlaces por estar sobreutilizados (cuello de botella).

El primer caso se soluciona incorporando nuevos recursos y aplicando técnicas de control de congestión como añadir más capacidad a los enlaces.

La mala gestión de los recursos se puede resolver con el uso de la **Ingeniería de Tráfico** (TE).

Las técnicas de TE tratan de adaptar los flujos de tráfico a los recursos de red disponibles, para así minimizar la congestión e incrementar la eficiencia del uso de recursos.

Independientemente de la técnica utilizada, el fin de la ingeniería de tráfico es desarrollar las estructuras para optimizar el rendimiento de las redes y así, proporcionar calidad en el servicio de red a los usuarios. Por lo tanto, está orientada tanto a tráfico como a recursos.

- A tráfico: mejorar los aspectos relacionados al transporte de datos: minimizar retardos, pérdidas de paquetes, etc.
- A recursos: optimización del uso de recursos de red, principalmente el ancho de banda.

Los beneficios que otorga el uso de TE:

- Disminuye costos
- Retrasar o evitar el aumento de la capacidad de la red
- Maximizar el ancho de banda
- Mejorar el control en caso de fallos

### 4.2 MPLS-TE

La ingeniería de tráfico puede ser implementada en cualquier tipo de red, pero sólo MPLS ofrece ciertas ventajas:

- Lidiar con congestiones en los enlaces.
- Re-enrutamiento automático de paquetes ante fallos en la red.
- Balanceo de carga y elección de rutas sin cambiar las métricas del protocolo de enrutamiento.

[RFC 2702] MPLS se convierte en el medio perfecto para implementar TE en redes IP gracias a una serie de capacidades:

- Agregación y desagregación de tráfico para redireccionar flujos desde partes de la red sobreutilizadas a otras infrautilizadas.
- Creación sencilla de LSPs explícitos de manera manual o dinámica (automatizada por protocolos subyacentes) y su eficiente mantenimiento.
- Creación de “Traffic Trunks”, agregación unidireccional de flujos de tráfico perteneciente a un mismo FEC y enviados a través de un LSP común.
- Asociación de atributos a Traffic Trunks para modelar su comportamiento.
- Routing con restricciones y protección de trunks.
- Optimiza la función de routing para maniobrar tráfico por la red.

MPLS dentro de TE realiza varias tareas secuenciales:

1. Mapea paquetes a FECs según distintos criterios de clasificación.
2. Mapea FECs a Traffic Trunks
3. Mapea trunks a la topología física de la red. Esta tarea es realizada por los protocolos de routing basados en restricciones.

### 4.3 Routing basado en restricciones (CBR)

CBR (Constraint Based Routing) selecciona la mejor ruta según las restricciones establecidas en la red de manera que sea óptima respecto a alguna métrica.

Para ello usa el algoritmo de enrutamiento basado en restricciones **CSPF** (Constrain Shortest Path First). Este algoritmo requiere que el LSR de inicio calcule la ruta más corta cumpliendo una serie de restricciones como el ancho de banda, delay, etc.

Una vez el camino ha sido determinado, se convierte en Ruta Explícita (ER) con un listado de las direcciones IP de los routers del camino y es difundido mediante el protocolo RSVP.

### 4.4 RSVP-TE

MPLS utiliza un protocolo de señalización y reserva de recursos para crear el LSP del túnel. Mantiene una base de datos que asocia las etiquetas a los caminos y son actualizados por medio de un protocolo de routing.

Los túneles implementan un interfaz de salida asociado con un camino definido internamente a la red, no necesariamente el camino calculado por IGP. Son unidireccionales, y una vez configurados se les asigna una etiqueta. Un túnel consta de tres partes:

- Headend: router origen. Donde se configura el túnel.
- Tailend: router destino.
- Midpoint: todos los routers intermedios del camino.

Sus mensajes son:

- **PATH**: inicia el proceso. Lleva el ER y las características de ancho de banda requerido. Path Tear para eliminar el túnel. Si un enlace se cae o se desaloja un túnel de un enlace, los routers envían PATHerr a los extremos del túnel, los cuáles buscan un nuevo camino.
- **RESV**: asigna y transporta la etiqueta del túnel. Confirma la reserva de ancho de banda realizada por el mensaje PATH.

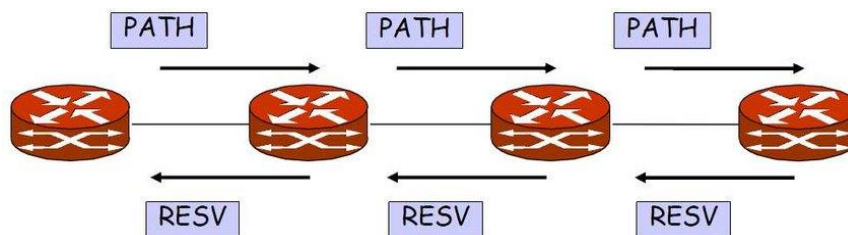


Figura 47. Mensajes RSVP

Cada uno de los mensajes contiene campos que lo caracterizan, algunos de los más importantes son:

- Session: identifica la dirección del flujo del tráfico junto al nodo de origen y destino.
- Sender\_Template y Sender\_Tspec: usados por los LSRs intermedios para calificar el nodo origen y especificar el tráfico dentro del mensaje PATH.
- Filter-spec, FlowSpec y Style: utilizados por el mensaje RESV para definir la reserva del tráfico, especificar los requerimientos de recursos y diseñar el estilo de reserva.

## 4.5 Fast Reroute

Como se ha explicado anteriormente una de las ventajas del uso de MPLS con TE es el re-encaminamiento automático de los paquetes ante cualquier fallo en la red.

En las redes IP, cuando se produce un fallo en un nodo o enlace, los protocolos de encaminamiento vuelven a calcular la ruta óptima para dirigir el tráfico. Durante ese proceso, se pueden producir retrasos y pérdidas de paquetes. MPLS-TE ofrece un mecanismo para restaurar caminos y minimizar pérdidas llamado conmutación de protección o Fast Reroute, definido en [RFC 4090] y [RFC 6894].

Para usar Fast Reroute es necesario que exista una ruta de recuperación o backup calculada y establecida previamente a que ocurra cualquier fallo en el LSP principal. Cuando se detecta un fallo, este mecanismo dirige el tráfico hacia la ruta de backup que, al estar ya configurada, no se pierde tiempo en recalcularla y se minimiza la pérdida de paquetes.

No podremos comprobar este proceso en nuestra red, ya que esta función no está disponible en las familias de routers Cisco permitidas en GNS3.

## 4.6 Objetivos

En esta práctica se va a realizar una configuración básica de una red MPLS con Ingeniería de Tráfico en el entorno simulado de GNS3. Entre los distintos objetivos se encuentran:

- Diseño y configuración de una red MPLS-TE.
- Entender el protocolo RSVP-TE y observar su funcionamiento mediante una serie de actividades.

## 4.7 Topología de red

En esta segunda práctica utilizaremos la red creada en la anterior práctica con algunos pequeños cambios en la topología.

Para realizar funciones de envío de tráfico de gran tamaño, los PCs virtuales en GNS3 suelen dar fallos y desconectarse del servidor, por lo que los cambiaremos por routers que hagan la función de PC.

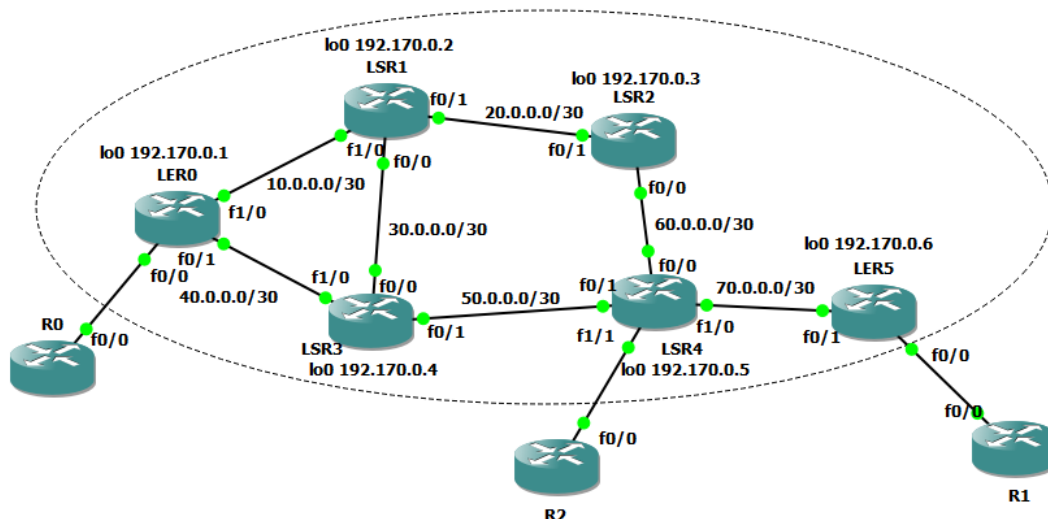


Figura 48. Diagrama de la topología de red

La tabla de direccionamiento de los routers añadidos es la siguiente:

Dispositivo	Interfaz	Dirección IP	Máscara de red
R0	Fa0/0	192.168.1.100	255.255.255.0
R1	Fa0/0	192.168.2.100	255.255.255.0
R2	Fa0/0	192.168.3.100	255.255.255.0

Figura 7. Tabla de direccionamiento

## 4.8 Configuración

### 4.8.1 Añadir direcciones y OSPF

Una vez cambiados los VPCS por routers en la red creada anteriormente, les asignaremos sus direcciones IP y máscara, y el protocolo OSPF.

```
R2#configure terminal
R2(config)#interface fa0/0
R2(config-if)#ip address 192.168.3.100 255.255.255.0
R2(config-if)#no shutdown
R2(config)#router ospf 1
R2(config-router)#network 192.168.3.0 0.0.0.255 area 0
```

En el router LSR4 tendremos que asignar una dirección al puerto FastEthernet1/1 y añadir una nueva red en el protocolo OSPF.

### 4.8.2 Configuración de TE

La función de Ingeniería de Tráfico en MPLS solo se puede habilitar tras haber activado MPLS en la red. Procederemos a habilitarla, primero a nivel global y después en cada uno de los interfaces con el comando “*mpls traffic-eng tunnels*”.

También habilitaremos el protocolo RSVP en cada interfaz, reservando un ancho de banda máximo para crear los túneles. En nuestro caso reservaremos 64 kbps (32 kbps por túnel).

```
LSR3(config)#mpls traffic-eng tunnels
LSR3(config)#int fa0/0
LSR3(config-if)#mpls traffic-eng tunnels
LSR3(config-if)#ip rsvp bandwidth 64 32
LSR3(config-if)#int fa0/1
LSR3(config-if)#mpls traffic-eng tunnels
LSR3(config-if)#ip rsvp bandwidth 64 32
LSR3(config-if)#int fa1/0
LSR3(config-if)#mpls traffic-eng tunnels
LSR3(config-if)#ip rsvp bandwidth 64 32
LSR3(config-if)#exit
```

Para comprobar que se ha realizado correctamente la reserva podemos usar el comando “*show ip rsvp interface*”:

```
LER0#show ip rsvp interface
interface    allocated  i/f max  flow max sub max
Fa0/1       32K       64K   32K   0
Fa1/0       0         64K   32K   0
```

Figura 49. Reserva de recursos

En el apartado *allocated* se indica que se ha realizado una reserva de ancho de banda de 32 kbps en el interfaz.

Tendremos que activar también TE en el proceso OSPF para conocer los estados de los interfaces y las rutas de la red.

```
LSR3#conf t
LSR3(config)#router ospf 1
LSR3(config-router)#mpls traffic-eng router-id Loopback 0
LSR3(config-router)#mpls traffic-eng area 0
```

Repetir todos los pasos anteriores en cada router de la red MPLS.

Para visualizar las sesiones RSVP establecidas utilizaríamos “*show ip rsvp sender*”.

```
LSR3#show ip rsvp sender
To          From          Pro DPort Sport Prev Hop      I/F      BPS
192.170.0.1 192.170.0.6   0  2    9   30.0.0.1    Fa0/0    32K
192.170.0.6 192.170.0.1   0  1   14  40.0.0.1    Fa1/0    32K
```

Figura 50. Sesiones RSVP

En LSR3 aparecen ambas sesiones RSVP establecidas, una por túnel, indicando el ancho de banda reservado.

### 4.8.3 Establecimiento túneles

Tras habilitar la Ingeniería de Tráfico en toda la red, pasaremos a la creación de los túneles.

El túnel se puede crear de forma dinámica o explícita según el comando “*tunnel mpls traffic-eng path-option número {dynamic/explicit {name nombre-tunel}}*”. Path-option indica el orden con el que se establece el túnel.

Como los túneles son unidireccionales, se creará un túnel en cada sentido. Uno de LER0 a LER5 y otro en sentido contrario.

Primero estableceremos el túnel dinámico entre los dos LER. Lo identificaremos como *tunnel 1* y se tendrá que establecer desde su router de inicio. Para el direccionamiento usaremos su dirección de loopback (“*ip unnumbered loopback 0*”) y usando el comando de ruta automática, indicaremos al IGP que utilice el túnel como próximo salto para alcanzar el destino final (“*tunnel mpls traffic-eng autoroute announce*”).

En la configuración de los túneles se determina las restricciones que usarán los equipos al ejecutar el algoritmo **CSPF** para calcular el LSP. Algunas de las restricciones son la prioridad, el ancho de banda o la métrica.

Asignaremos un ancho de banda (bandwidth) de 32kbps y una prioridad con respecto al otro túnel a través del comando “*tunnel mpls traffic-eng priority setup-priority hold-priority*”. Siendo 7 la más alta y 0 la más baja.

Los comandos se muestran a continuación:



```
LER0(config)#int tunnel 1
LER0(config-if)#ip unnumbered loopback 0
LER0(config-if)#tunnel destination 192.170.0.6
LER0(config-if)#tunnel mode mpls traffic-eng
LER0(config-if)#tunnel mpls traffic-eng autoroute announce
LER0(config-if)#tunnel mpls traffic-eng path-option 2 dynamic
LER0(config-if)#tunnel mpls traffic-eng bandwidth 32
LER0(config-if)#tunnel mpls traffic-eng priority 7 7
```

Una vez establecido el túnel dinámico, configuraremos el túnel explícito entre ambos LER. La única diferencia en la configuración es la especificación de los saltos por los que transita el túnel mediante “**ip explicit-path name nombre\_tunnel**” y “**next address dir\_IP**”.

```
LER5(config)#int tunnel 2
LER5(config-if)#ip unnumbered loopback 0
LER5(config-if)#tunnel destination 192.170.0.1
LER5(config-if)#tunnel mode mpls traffic-eng
LER5(config-if)#tunnel mpls traffic-eng autoroute announce
LER5(config-if)#tunnel mpls traffic-eng path-option 1 explicit name tunel2
LER5(config-if)#tunnel mpls traffic-eng bandwidth 32
LER5(config-if)#tunnel mpls traffic-eng priority 6 6

LER5(config)#ip explicit-path name tunel2
LER5(cfg-ip-expl-path)#next-address 192.170.0.5
LER5(cfg-ip-expl-path)#next-address 192.170.0.3
LER5(cfg-ip-expl-path)#next-address 192.170.0.2
LER5(cfg-ip-expl-path)#next-address 192.170.0.4
LER5(cfg-ip-expl-path)#next-address 192.170.0.1
```

Con el comando “**show mpls traffic-eng tunnels brief**” podemos comprobar si los túneles han sido correctamente creados.

```
LSR2#show mpls traffic-eng tunnels brief
Signalling Summary:
  LSP Tunnels Process:      running
  RSVP Process:            running
  Forwarding:              enabled
  Periodic reoptimization: every 3600 seconds, next in 2745 seconds
  Periodic auto-bw collection: disabled
TUNNEL NAME                DESTINATION      UP IF    DOWN IF  STATE/PROT
LER5_t2                    192.170.0.1     Fa0/0   Fa0/1   up/up
Displayed 0 (of 0) heads, 1 (of 1) midpoints, 0 (of 0) tails
LSR2#
```

Figura 51. Información túneles en LSR2

Por el LSR2, únicamente pasa el túnel 2 (LER5\_t2) que está correctamente establecido.

En el caso del LSR3, podemos observar que ambos túneles discurren por él como hemos definido anteriormente (LER0\_t1 y LER5\_t2).

```
LSR3#show mpls traffic-eng tunnels brief
Signalling Summary:
  LSP Tunnels Process:      running
  RSVP Process:            running
  Forwarding:              enabled
  Periodic reoptimization: every 3600 seconds, next in 2560 seconds
  Periodic auto-bw collection: disabled
TUNNEL NAME                DESTINATION      UP IF    DOWN IF    STATE/PROT
LER0_t1                    192.170.0.6     Fa1/0   Fa0/1     up/up
LER5_t2                    192.170.0.1     Fa0/0   Fa1/0     up/up
Displayed 0 (of 0) heads, 2 (of 2) midpoints, 0 (of 0) tails
```

Figura 52. Información túneles en LSR3

#### 4.8.4 Verificación final

En los LER también se puede comprobar el correcto establecimiento de los túneles con el comando “**show mpls traffic-eng tunnels tunnel *interface***”.

Se muestran parámetros como el ancho de banda establecido, la prioridad, la etiqueta asignada o las direcciones de los nodos por los que transcurre.

Para el LER0 veremos el Túnel 1 (figura 53) y para el LER5 el 2 (figura 54).

```
LER0#show mpls traffic-eng tunnels tunnel 1
Name: LER0_t1 (Tunnel1) Destination: 192.170.0.6
Status:
  Admin: up      Oper: up      Path: valid      Signalling: connected
  path option 2, type dynamic (Basis for Setup, path weight 3)
Config Parameters:
  Bandwidth: 32 kbps (Global) Priority: 7 7 Affinity: 0x0/0xFFFF
  Metric Type: TE (default)
  AutoRoute: enabled LockDown: disabled Loadshare: 32 bw-based
  auto-bw: disabled
InLabel : -
OutLabel : FastEthernet0/1, 311
RSVP Signalling Info:
  Src 192.170.0.1, Dst 192.170.0.6, Tun_Id 1, Tun_Instance 14
RSVP Path Info:
  My Address: 40.0.0.1
  Explicit Route: 40.0.0.2 50.0.0.1 50.0.0.2 70.0.0.2
                  70.0.0.1 192.170.0.6
  Record Route: NONE
  Tspec: ave rate=32 kbits, burst=1000 bytes, peak rate=32 kbits
RSVP Resv Info:
  Record Route: NONE
  Fspec: ave rate=32 kbits, burst=1000 bytes, peak rate=32 kbits
Shortest Unconstrained Path Info:
  Path Weight: 3 (TE)
  Explicit Route: 40.0.0.1 40.0.0.2 50.0.0.1 50.0.0.2
                  70.0.0.2 70.0.0.1 192.170.0.6
History:
Tunnel:
  Time since created: 19 minutes, 19 seconds
  Time since path change: 18 minutes, 24 seconds
Current LSP:
  Uptime: 18 minutes, 24 seconds
```

Figura 53. Parámetros túnel 1

```
LER5#show mpls traffic-eng tunnels tunnel 2

Name: LER5_t2 (Tunnel2) Destination: 192.170.0.1
Status:
  Admin: up      Oper: up      Path: valid      Signalling: connected

  path option 1, type explicit tunel2 (Basis for Setup, path weight 5)

Config Parameters:
  Bandwidth: 32 kbps (Global) Priority: 6 6 Affinity: 0x0/0xFFFF
  Metric Type: TE (default)
  AutoRoute: enabled LockDown: disabled Loadshare: 32 bw-based
  auto-bw: disabled

InLabel : -
OutLabel : FastEthernet0/1, 410
RSVP Signalling Info:
  Src 192.170.0.6, Dst 192.170.0.1, Tun_Id 2, Tun_Instance 9
RSVP Path Info:
  My Address: 70.0.0.1
  Explicit Route: 70.0.0.2 60.0.0.2 60.0.0.1 20.0.0.2
                  20.0.0.1 30.0.0.1 30.0.0.2 40.0.0.2
                  40.0.0.1 192.170.0.1
  Record Route: NONE
  Tspec: ave rate=32 kbits, burst=1000 bytes, peak rate=32 kbits
RSVP Resv Info:
  Record Route: NONE
  Fspec: ave rate=32 kbits, burst=1000 bytes, peak rate=32 kbits
Shortest Unconstrained Path Info:
  Path Weight: 3 (TE)
  Explicit Route: 70.0.0.1 70.0.0.2 50.0.0.2 50.0.0.1
                  40.0.0.2 40.0.0.1 192.170.0.1

History:
  Tunnel:
    Time since created: 21 minutes, 45 seconds
    Time since path change: 20 minutes, 55 seconds
  Current LSP:
    Uptime: 20 minutes, 55 seconds
```

Figura 54. Parámetros túnel 2

## 4.9 Actividades propuestas

1. Ejecute el comando “*show mpls forwarding-table 192.168.2.100 detail*” en LER0 para visualizar la tabla LFIB y observe la cabecera MPLS de un paquete. **¿Qué diferencias observa en el etiquetado respecto a la práctica anterior?**

\* Los valores de las etiquetas pueden variar con cada reinicio de sesión.

```
LER0#show mpls forwarding-table 192.168.2.100 detail
Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC  or Tunnel Id    switched   interface
24     Untagged  192.168.2.0/24  0          Tu1        point2point
      MAC/Encaps=14/18, MRU=1500, Tag Stack{311}, via Fa0/1
      CA042DB8001CCA01045800068847 00137000
      No output feature configured
      Per-packet load-sharing
```

Figura 55. Tabla LFIB destino 192.168.2.100

En la pila de etiquetas o tag stack se observa que el LER0 utiliza la 311 para alcanzar R1 (en la práctica anterior la etiqueta era 309). Esta etiqueta es la exterior en la pila y es usada para indicar qué túnel TE reenviará los paquetes, en este caso el túnel 1.

```
MultiProtocol Label Switching Header, Label: 311, Exp: 0, S: 1, TTL: 255
0000 0000 0001 0011 0111 .... .. = MPLS Label: 311
..... 000. .... .. = MPLS Experimental Bits: 0
..... ..1 .... .. = MPLS Bottom Of Label Stack: 1
..... 1111 1111 = MPLS TTL: 255
```

Figura 56. Cabecera MPLS

Si observamos la cabecera MPLS de un paquete enviado, el bit del campo S está puesto a 1 al tratarse de la etiqueta del túnel TE, como se ha explicado en las actividades de la práctica 1.

2. Si mandamos paquetes a R0 y R1 desde los LERs, ¿qué caminos seguirán?

```
LER0#traceroute 192.168.2.100
Type escape sequence to abort.
Tracing the route to 192.168.2.100

 1 40.0.0.2 [MPLS: Label 311 Exp 0] 704 msec 168 msec 176 msec
 2 50.0.0.2 [MPLS: Label 409 Exp 0] 316 msec 252 msec 184 msec
 3 70.0.0.1 360 msec 168 msec 188 msec
 4 192.168.2.100 508 msec 112 msec 68 msec
```

Figura 57. Traceroute a R1

Los paquetes dirigidos al router R1 circulan por el camino del túnel 1. Este camino es la ruta más corta calculada por el protocolo de routing IGP (OSPF), debido a que el túnel ha sido creado como dinámico.

```

LER5#traceroute 192.168.1.100
Type escape sequence to abort.
Tracing the route to 192.168.1.100

 0 10.0.0.1 [MPLS: Label 310 Exp 0] 100 msec 100 msec 100 msec
 1 70.0.0.2 [MPLS: Label 410 Exp 0] 188 msec 176 msec 92 msec
 2 60.0.0.1 [MPLS: Label 203 Exp 0] 120 msec 64 msec 64 msec
 3 20.0.0.1 [MPLS: Label 102 Exp 0] 104 msec 72 msec 20 msec
 4 30.0.0.2 [MPLS: Label 310 Exp 0] 84 msec 20 msec 100 msec
 5 40.0.0.1 52 msec 52 msec 56 msec
 6 192.168.1.100 172 msec 176 msec 104 msec

```

Figura 58. Traceroute a R0

Se puede apreciar en este caso, como la ruta sigue el camino del túnel 2, a pesar de no ser el camino más corto, ya que así ha sido establecido cuando hemos definido el túnel estático.

### 3. Usando el Wireshark capture los mensajes de RSVP para la reserva de recursos en cada túnel e identifique sus campos.

No.	Time	Source	Destination	Protocol	Length	Info
6	3.846860	192.170.0.1	192.170.0.6	RSVP	254	PATH Message. SESSION: IPv4-LSP, Destination 192.170.0.6, Short Call ID 0, Tunnel ID 1, Ext ID c0aa0001. SENDER TEMPLAT...
8	7.465181	40.0.0.2	40.0.0.1	RSVP	142	RESV Message. SESSION: IPv4-LSP, Destination 192.170.0.6, Short Call ID 0, Tunnel ID 1, Ext ID c0aa0001. FILTERSPEC: IP...
12	10.894917	40.0.0.1	40.0.0.2	RSVP	142	RESV Message. SESSION: IPv4-LSP, Destination 192.170.0.1, Short Call ID 0, Tunnel ID 2, Ext ID c0aa0006. FILTERSPEC: IP...
33	31.030320	40.0.0.2	40.0.0.1	RSVP	142	RESV Message. SESSION: IPv4-LSP, Destination 192.170.0.6, Short Call ID 0, Tunnel ID 1, Ext ID c0aa0001. FILTERSPEC: IP...
34	31.337606	192.170.0.1	192.170.0.6	RSVP	254	PATH Message. SESSION: IPv4-LSP, Destination 192.170.0.6, Short Call ID 0, Tunnel ID 1, Ext ID c0aa0001. SENDER TEMPLAT...
41	37.161788	192.170.0.6	192.170.0.1	RSVP	222	PATH Message. SESSION: IPv4-LSP, Destination 192.170.0.1, Short Call ID 0, Tunnel ID 2, Ext ID c0aa0006. SENDER TEMPLAT...
60	54.163609	40.0.0.1	40.0.0.2	RSVP	142	RESV Message. SESSION: IPv4-LSP, Destination 192.170.0.1, Short Call ID 0, Tunnel ID 2, Ext ID c0aa0006. FILTERSPEC: IP...
64	60.585814	192.170.0.6	192.170.0.1	RSVP	222	PATH Message. SESSION: IPv4-LSP, Destination 192.170.0.1, Short Call ID 0, Tunnel ID 2, Ext ID c0aa0006. SENDER TEMPLAT...
75	70.120666	192.170.0.1	192.170.0.6	RSVP	254	PATH Message. SESSION: IPv4-LSP, Destination 192.170.0.6, Short Call ID 0, Tunnel ID 1, Ext ID c0aa0001. SENDER TEMPLAT...
83	77.347430	40.0.0.2	40.0.0.1	RSVP	142	RESV Message. SESSION: IPv4-LSP, Destination 192.170.0.6, Short Call ID 0, Tunnel ID 1, Ext ID c0aa0001. FILTERSPEC: IP...
95	88.747482	192.170.0.6	192.170.0.1	RSVP	222	PATH Message. SESSION: IPv4-LSP, Destination 192.170.0.1, Short Call ID 0, Tunnel ID 2, Ext ID c0aa0006. SENDER TEMPLAT...
100	91.045122	40.0.0.1	40.0.0.2	RSVP	142	RESV Message. SESSION: IPv4-LSP, Destination 192.170.0.1, Short Call ID 0, Tunnel ID 2, Ext ID c0aa0006. FILTERSPEC: IP...
116	106.740253	192.170.0.1	192.170.0.6	RSVP	254	PATH Message. SESSION: IPv4-LSP, Destination 192.170.0.6, Short Call ID 0, Tunnel ID 1, Ext ID c0aa0001. SENDER TEMPLAT...
117	106.881897	40.0.0.1	40.0.0.2	RSVP	142	RESV Message. SESSION: IPv4-LSP, Destination 192.170.0.1, Short Call ID 0, Tunnel ID 2, Ext ID c0aa0006. FILTERSPEC: IP...
118	107.903900	192.170.0.6	192.170.0.1	RSVP	222	PATH Message. SESSION: IPv4-LSP, Destination 192.170.0.1, Short Call ID 0, Tunnel ID 2, Ext ID c0aa0006. SENDER TEMPLAT...
124	113.138214	40.0.0.2	40.0.0.1	RSVP	142	RESV Message. SESSION: IPv4-LSP, Destination 192.170.0.6, Short Call ID 0, Tunnel ID 1, Ext ID c0aa0001. FILTERSPEC: IP...
139	129.666235	192.170.0.1	192.170.0.6	RSVP	254	PATH Message. SESSION: IPv4-LSP, Destination 192.170.0.6, Short Call ID 0, Tunnel ID 1, Ext ID c0aa0001. SENDER TEMPLAT...
152	139.113342	192.170.0.6	192.170.0.1	RSVP	222	PATH Message. SESSION: IPv4-LSP, Destination 192.170.0.1, Short Call ID 0, Tunnel ID 2, Ext ID c0aa0006. SENDER TEMPLAT...

Figura 59. Captura de mensajes RSVP

En la reserva de recursos de los túneles aparecen dos tipos de mensajes: RESV y PATH.

En el mensaje RESV (Figura 60 y 61) podemos observar los siguientes campos:

- Session: dirección IP del LSR destino e ID asignado al túnel (1 para túnel 1 y 2 para el segundo túnel).
- Hop: dirección del siguiente salto.
- Style: estilo de la reserva. Puede ser “Fixed Filter” o “Shared Explicit”. En ambos túneles el estilo es Shared-Explicit, es decir, una reserva única en el enlace compartido.
- Flowspec: reserva de ancho de banda y sus parámetros (4000 bytes/sec).
- FilterSpec: conjunto de paquetes que reciben la reserva definida.
- Label: etiqueta asignada al túnel para encaminar los paquetes a través de él. Diferente a la etiqueta asignada por LDP. Como hemos visto antes, el LER asigna la etiqueta 311 para el túnel 1 y 105 para el túnel 2.

```

v Resource ReserVation Protocol (RSVP): RESV Message. SESSION: IPv4-LSP, Destination 192.170.0.6,
  > RSVP Header. RESV Message.
  v SESSION: IPv4-LSP, Destination 192.170.0.6, Short Call ID 0, Tunnel ID 1, Ext ID c0aa0001.
    Length: 16
    Object class: SESSION object (1)
    C-type: IPv4-LSP (7)
    Destination address: 192.170.0.6
    Short Call ID: 0
    Tunnel ID: 1
    Extended Tunnel ID: 3232366593 (192.170.0.1)
  > HOP: IPv4, 40.0.0.2
  > TIME VALUES: 30000 ms
  v STYLE: Shared-Explicit (18)
    Length: 8
    Object class: STYLE object (8)
    C-Type: Style (1)
    Flags: 0x00
    Style: Shared-Explicit (0x000012)
  v FLOWSPEC: Controlled Load: Token Bucket, 4000 bytes/sec.
    Length: 36
    Object class: FLOWSPEC object (9)
    C-Type: Integrated Services (2)
    0000 .... = Message format version: 0
    Data length: 7 words, not including header
    Service header: Controlled Load (5)
    Data length: 6 words, not including header
  > Parameter: Token bucket (127)Rate=4000 Burst=1000 Peak=4000 m=0 M=0
  v FILTERSPEC: IPv4-LSP, Tunnel Source: 192.170.0.1, Short Call ID: 0, LSP ID: 21.
    Length: 12
    Object class: FILTER SPEC object (10)
    C-Type: IPv4 LSP (7)
    Sender IPv4 address: 192.170.0.1
    LSP ID: 21
  > LABEL: 311
  
```

Figura 60. Mensaje RESV túnel 1

```

v Resource ReserVation Protocol (RSVP): RESV Message. SESSION: IPv4-LSP, Destination 192.170.0.1,
  > RSVP Header. RESV Message.
  v SESSION: IPv4-LSP, Destination 192.170.0.1, Short Call ID 0, Tunnel ID 2, Ext ID c0aa0006.
    Length: 16
    Object class: SESSION object (1)
    C-type: IPv4-LSP (7)
    Destination address: 192.170.0.1
    Short Call ID: 0
    Tunnel ID: 2
    Extended Tunnel ID: 3232366598 (192.170.0.6)
  > HOP: IPv4, 20.0.0.1
  > TIME VALUES: 30000 ms
  > STYLE: Shared-Explicit (18)
  > FLOWSPEC: Controlled Load: Token Bucket, 4000 bytes/sec.
  v FILTERSPEC: IPv4-LSP, Tunnel Source: 192.170.0.6, Short Call ID: 0, LSP ID: 7.
    Length: 12
    Object class: FILTER SPEC object (10)
    C-Type: IPv4 LSP (7)
    Sender IPv4 address: 192.170.0.6
    LSP ID: 7
  > LABEL: 105
  
```

Figura 61. Mensaje RESV túnel 2

En el mensaje PATH (Figura 62 y 63) los primeros campos son iguales que en el mensaje RESV. El resto de los campos son:

- Explicit Route: direcciones de los nodos por los que traspasa cada LSP.
- Sender Template: dirección del router emisor e ID del LSP (12 para el túnel 1 y 7 para el túnel 2).
- Sender Tspec: reserva de ancho de banda solicitada. 4000 bytes/s (los 32 kbps establecidos).

```
> Internet Protocol Version 4, Src: 192.170.0.1, Dst: 192.170.0.6
▼ Resource Reservation Protocol (RSVP): PATH Message. SESSION: IPv4-LSP, Destination 192.170.0.6, Short Call ID 0, Tunnel ID 1,
  > RSVP Header. PATH Message.
  > SESSION: IPv4-LSP, Destination 192.170.0.6, Short Call ID 0, Tunnel ID 1, Ext ID c0aa0001.
  > HOP: IPv4, 40.0.0.1
  > TIME VALUES: 30000 ms
  ▼ EXPLICIT ROUTE: IPv4 40.0.0.2, IPv4 50.0.0.1, IPv4 50.0.0.2, ...
    Length: 52
    Object class: EXPLICIT ROUTE object (20)
    C-Type: 1
    > IPv4 Subobject - 40.0.0.2, Strict
    > IPv4 Subobject - 50.0.0.1, Strict
    > IPv4 Subobject - 50.0.0.2, Strict
    > IPv4 Subobject - 70.0.0.2, Strict
    > IPv4 Subobject - 70.0.0.1, Strict
    > IPv4 Subobject - 192.170.0.6, Strict
  > LABEL REQUEST: Basic: L3PID: IPv4 (0x0800)
  > SESSION ATTRIBUTE: SetupPrio 7, HoldPrio 7, SE Style, [LER0_t1]
  ▼ SENDER TEMPLATE: IPv4-LSP, Tunnel Source: 192.170.0.1, Short Call ID: 0, LSP ID: 12.
    Length: 12
    Object class: SENDER TEMPLATE object (11)
    C-Type: IPv4 LSP (7)
    Sender IPv4 address: 192.170.0.1
    Short Call ID: 0
    LSP ID: 12
  ▼ SENDER TSPEC: IntServ, Token Bucket, 4000 bytes/sec.
    Length: 36
    Object class: SENDER TSPEC object (12)
    C-Type: Integrated Services (2)
    0000 ... = Message format version: 0
    Data length: 7 words, not including header
    Service header: Traffic specification (1)
    Data length: 6 words, not including header
    > Parameter: Token bucket (127)Rate=4000 Burst=1000 Peak=4000 m=0 M=2147483647
  > ADSPEC
```

Figura 62. Mensaje PATH túnel 1

```
▼ Resource Reservation Protocol (RSVP): PATH Message. SESSION: IPv4-LSP, Destination 192.170.0.1, Sho
  > RSVP Header. PATH Message.
  > SESSION: IPv4-LSP, Destination 192.170.0.1, Short Call ID 0, Tunnel ID 2, Ext ID c0aa0006.
  > HOP: IPv4, 20.0.0.2
  > TIME VALUES: 30000 ms
  ▼ EXPLICIT ROUTE: IPv4 20.0.0.1, IPv4 30.0.0.1, IPv4 30.0.0.2, ...
    Length: 52
    Object class: EXPLICIT ROUTE object (20)
    C-Type: 1
    > IPv4 Subobject - 20.0.0.1, Strict
    > IPv4 Subobject - 30.0.0.1, Strict
    > IPv4 Subobject - 30.0.0.2, Strict
    > IPv4 Subobject - 40.0.0.2, Strict
    > IPv4 Subobject - 40.0.0.1, Strict
    > IPv4 Subobject - 192.170.0.1, Strict
  > LABEL REQUEST: Basic: L3PID: IPv4 (0x0800)
  > SESSION ATTRIBUTE: SetupPrio 6, HoldPrio 6, SE Style, [LERS_t2]
  ▼ SENDER TEMPLATE: IPv4-LSP, Tunnel Source: 192.170.0.6, Short Call ID: 0, LSP ID: 7.
    Length: 12
    Object class: SENDER TEMPLATE object (11)
    C-Type: IPv4 LSP (7)
    Sender IPv4 address: 192.170.0.6
    Short Call ID: 0
    LSP ID: 7
  > SENDER TSPEC: IntServ, Token Bucket, 4000 bytes/sec.
  > ADSPEC
```

Figura 63. Mensaje PATH túnel 2

4. Genere tráfico en la red con destino router R1 y R2. Primero generaremos tráfico de gran tamaño desde LER0 hacia R2 con el comando **“ping 192.168.3.100 repeat 400 size 12000”**.

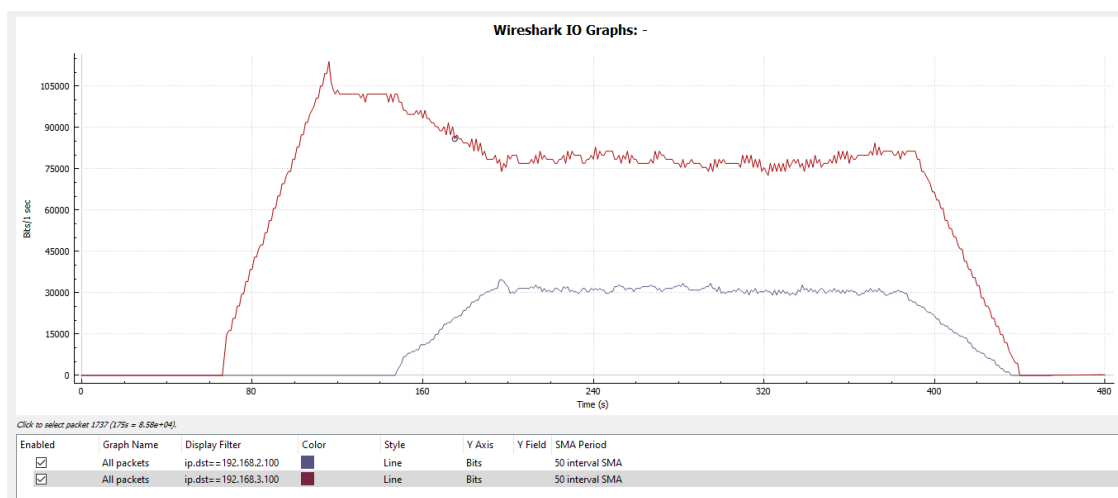
Como GNS3 no permite abrir dos terminales en el mismo nodo, desde el R0 enviaremos paquetes con un tamaño de 32 kbps al R1 con el comando **“ping 192.168.2.100 repeat 300 size 4000”**.

Utilizando la herramienta para crear gráficas que ofrece Wireshark (**Statistics-I/O Graph**) crearemos una para monitorizar los dos tráfico a la vez. En la ventana IO Graphs configuraremos los siguientes campos:

- El símbolo “+” para añadir líneas de gráfico.
- El campo “Display Filter” para filtrar el tráfico generado. Escribiremos los filtros: **“ip.dst==192.168.2.100”** y **“ip.dst==192.168.3.100”**, uno para cada línea respectivamente.
- “Y Axis” para ver el tráfico en bits por segundo.
- “SMA Period” para filtrar los picos de las líneas. Al ser bastante inestable, será necesario un Interval alto, en este caso se ha usado **“50 Interval SMA”**. Se puede ir cambiando hasta ver las líneas más estables, aunque dará lugar a un resultado menos real.

**Analice dicha gráfica y explique qué ha ocurrido con ambos tráfico.**

\*Cabe destacar que no es recomendable usar GNS3 para enviar tráfico de gran tamaño a velocidad limitada de esta manera. Al realizar esta prueba se ha observado como la memoria RAM usada aumentaba progresivamente, al igual que la utilización de CPU del equipo host. Lo que produce que el resultado sea inestable: latencia, pérdidas de paquetes, inestabilidad en el tamaño de paquetes, etc. De todas formas, el objetivo principal de esta actividad se puede apreciar.



**Figura 64. Tráfico generado en la red enlace LER0-LSR3**

En la gráfica resultante, se observa que los paquetes dirigidos a R2 recorren la red con el tamaño indicado hasta que se empieza a generar tráfico hacia R1. Este tráfico discurre por el túnel 1 con un ancho de banda de 32 kbps y una mayor prioridad.

El tráfico hacia R2 se ve disminuido en la misma cantidad en la que el otro tráfico se ve aumentado, debido a haber limitado el ancho de banda en dicho túnel (Balanceo de carga).



## Capítulo 5. Práctica 3 “Simulación Básica de una red MPLS VPN”

### 5.1 Introducción

Una VPN (Virtual Private Network) es una tecnología que permite crear redes privadas en la infraestructura de internet pública proporcionando confidencialidad y seguridad. Existen dos modelos según su implementación:

- Overlay VPN: incluye tecnologías como Frame Relay, ATM, IPsec, etc.
- Peer to peer VPN: con red de proveedores común e implementada con routers compartidos y ACLs, routers independientes para cada cliente o mediante MPLS (MPLS VPN).

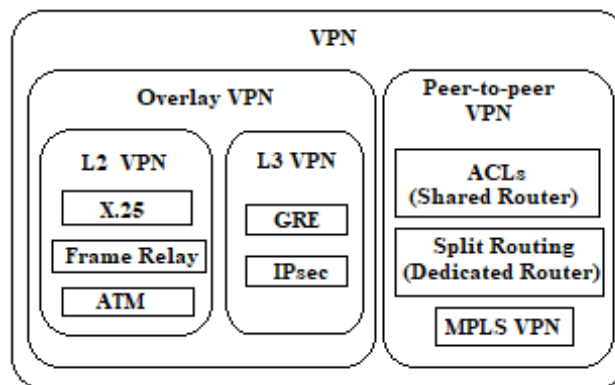


Figura 65. Modelos VPN

Cuando VPN se utiliza con MPLS, permite que varios clientes se interconecten de modo transparente a través de una red de proveedor de servicios (backbone MPLS), pudiéndose enviar paquetes IP entre ellos. La red proveedora puede ofrecer conectividad a varias VPN IP distintas, apareciendo cada una de ellas como una red privada, separada del resto de redes.

MPLS VPN puede implementarse tanto a nivel 2 como a nivel 3 de la capa OSI.

En las VPNs de capa 3 (L3VPN) la responsabilidad de crear y administrar túneles de tráfico privado entre los clientes recae en el proveedor usando MPLS.

### 5.2 Componentes y arquitectura L3VPN

#### 5.2.1 Customer Edge (CE)

Router perteneciente a la red del cliente conectado a los routers frontera de la red de proveedores a nivel 3. Intercambia rutas con los vecinos mediante cualquier protocolo de routing.

No forma parte del backbone MPLS por lo que no conoce su mecanismo, únicamente envía y recibe información de las rutas y la intercambia con el router PE.

#### 5.2.2 Provider Edge (PE)

Router frontera de la red del proveedor de servicios conectado al router CE. Contiene rutas VPN y establece diversos protocolos de enrutamiento para mantener rutas con clientes o routers de la red P. Contiene una tabla de enrutamiento (VRF) independiente para cada cliente.

Para realizar rutas entre los PE vecinos de la red se utiliza el protocolo BGP.

#### 5.2.3 Provider (P)

Router MPLS en el backbone de la red. Nunca está conectado a la red cliente.

No lleva rutas VPN, ya que solo posee información de la red del proveedor en sus tablas de routing.

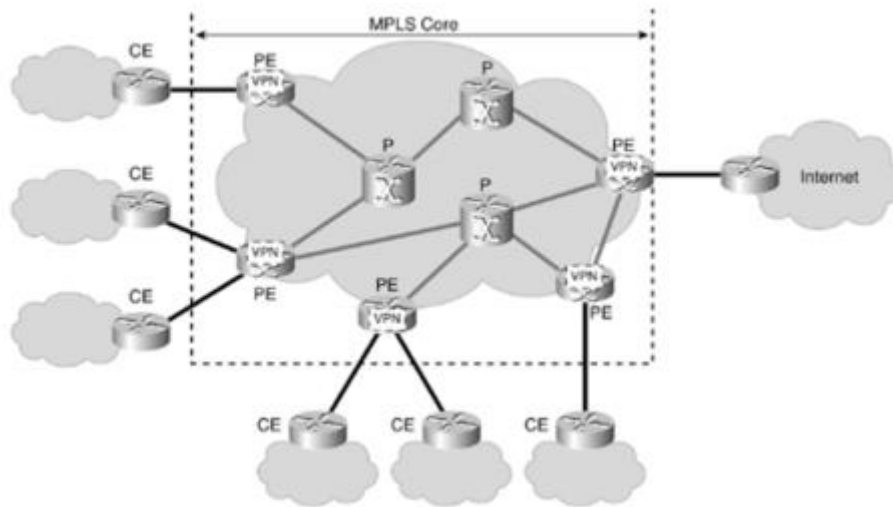


Figura 66. Topología red L3VPN MPLS

#### 5.2.4 Virtual Routing Forwarding (VRF)

Instancia de enrutamiento aislada y única dentro de un router. Consiste en una tabla de routing, una tabla CEF derivada y un grupo de los interfaces que usan dichas tablas. Pueden existir múltiples VRFs en los PE, una por cada VPN conectada al router.

Cuando un paquete enviado por el CE llega al PE, se utiliza la tabla de encaminamiento VRF asignada a ese emplazamiento para determinar la ruta a seguir por el paquete.

#### 5.2.5 Route Distinguishers (RD)

Identificador de rutas VPN que se antepone a la dirección de red para formar un prefijo único. Son 64 bits y se representa mediante ASN:nn (número de sistema autónomo y número asignado por proveedor). Para IPv4 se forma las direcciones VPNv4 (96 bits) intercambiadas únicamente entre los routers PE.

Los valores de RD no tienen un significado específico, están diseñados para generar rutas únicas cuando hay solapamiento. Son útiles cuando varios clientes comparten el mismo espacio de direccionamiento y se conectan al mismo PE.

#### 5.2.6 Route Targets (RT)

Valor numérico definido por cada PE que está asociado a las rutas que exporta a los puertos BGP.

Dos tipos de RT:

- Export RT: identifican los sitios remotos a donde se exportará una ruta.
- Import RT: utilizado por PE para seleccionar las rutas a importar en sus tablas VRF.

Para aceptar una nueva ruta el RT de importación y de exportación deben de coincidir. Son distribuidos por las actualizaciones BGP.

En casos de VPNs con solapamientos, estos valores son utilizados para identificar la asociación de la VPN.

### 5.3 MP-BGP

[RFC 2858] El Multiprotocolo BGP es una extensión del protocolo BGP utilizado para propagar direcciones y los atributos que las acompañan. Usado únicamente entre los PE.

Se puede clasificar de dos formas en función del AS:

- **BGP externo (eBGP):** la sesión BGP se establece entre routers de diferentes sistemas autónomos.
- **BGP interno (iBGP):** la sesión BGP se establece entre routers que forman parte del mismo sistema autónomo.

Los peers intercambian 4 tipos de mensajes una vez se haya establecido la sesión TCP:

- **Open:** abre sesión BGP entre vecinos. Se envían y negocian los parámetros del protocolo de routing del router.
- **KeepAlive:** enviados periódicamente para mantener la sesión abierta.
- **Update:** actualizan las tablas de rutas. Añaden, modifican o borran rutas.
- **Notification:** enviado cuando se produce algún error y cerrar la sesión BGP.

### 5.4 Propagación de rutas y envío de paquetes en MPLS VPN

BGP es utilizado para transportar rutas de manera segura por la red. EL proceso que se realiza para la propagación de las rutas es el siguiente:

1. Los routers PE reciben actualizaciones con direcciones IPv4 desde los routers CE mediante eBGP o un protocolo de encaminamiento configurado. Estas rutas IP se almacenan en la tabla VRF a la que pertenezcan.
2. Las rutas VPNv4 son propagadas a los PE. Para crearlas se añaden los RD delante de los prefijos IP. También se ponen los Export RT para especificar a qué VPN está asociada.

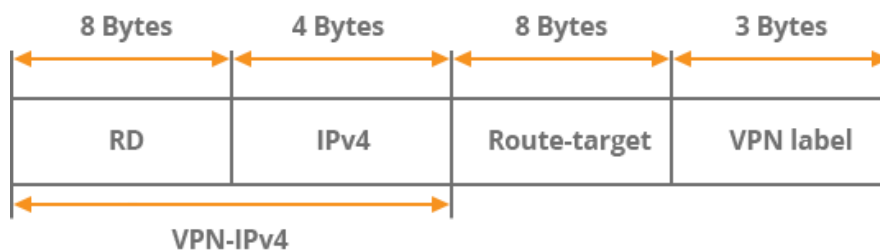


Figura 67. Mensaje de actualización MP-BGP

3. Los PE reciben actualizaciones de MP-BGP e importan rutas VPN de entrada en sus VRF correspondientes según los valores de Import RT asociados a esas rutas y tablas VRF.
4. Las rutas son añadidas en las VRF y redistribuidas mediante eBGP o el protocolo de routing que se está ejecutando entre los routers PE y CE para ser propagadas a la red del cliente.

Una vez las rutas IP y VPNv4 han sido propagadas, se habrá establecido comunicación IP entre CE y se procederá al envío de paquetes.

Los paquetes se reenvían basándose en etiquetas entre los routers PE peers. El tráfico entre VPNs tiene una pila de 2 etiquetas en la red del proveedor añadidas por el PE de ingreso y eliminadas por el PE de salida. La externa es la etiqueta IGP, asociada a un prefijo o dirección IP en la tabla de encaminamiento global de la red P y es distribuida mediante un protocolo de distribución de etiquetas (LDP o RSVP) entre los routers P y PE. Es utilizada por P para reenviar los paquetes al PE.

La segunda etiqueta es la perteneciente a la VPN, anunciada por MP-BGP entre ambos PE y es utilizada para reenviar los paquetes al CE correcto. Posee un valor de 1 en el bit S.

## 5.5 Objetivos

En esta práctica se va a realizar una configuración básica de una red VPN sobre MPLS en el simulador GNS3. Los distintos objetivos son:

- Diseño y configuración de una red VPN-MPLS.
- Entender el protocolo BGP y observar su funcionamiento mediante una serie de actividades.

## 5.6 Topología de red

Vamos a crear una red L3 VPN-MPLS formada por 5 routers c7200 con las mismas propiedades que en las prácticas anteriores. En ella estableceremos una VPN con dos routers cliente CE, los cuales queremos comunicar.

Utilizaremos los protocolos OSPF y BGP y la creación de tablas VRF para intercambiar información de direccionamiento entre proveedores y clientes.

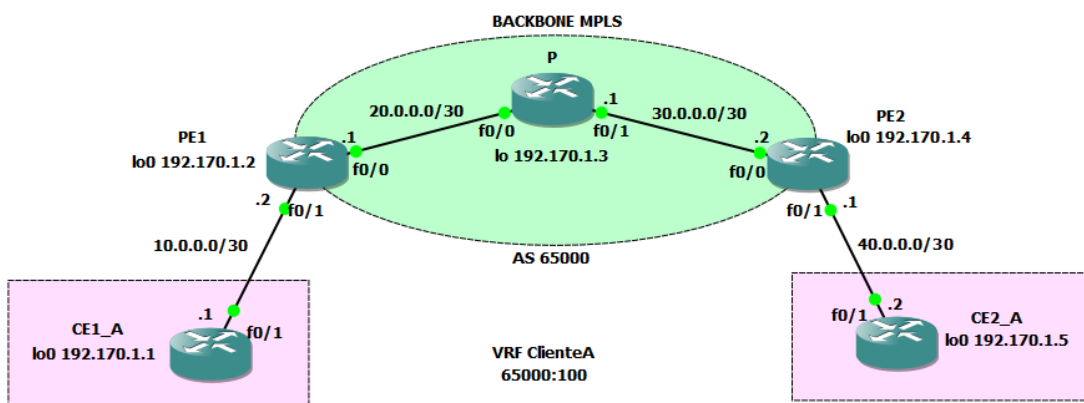


Figura 68. Diagrama de la red VPN MPLS

Dispositivo	Interfaz	Dirección IP	Máscara de red
CE1	Lo0	192.170.1.1	255.255.255.255
	Fa0/1	10.0.0.1	255.255.255.252
PE1	Lo0	192.170.1.2	255.255.255.255
	Fa0/0	20.0.0.1	255.255.255.252
	Fa0/1	10.0.0.2	255.255.255.252
P	Lo0	192.170.1.3	255.255.255.255
	Fa0/0	20.0.0.2	255.255.255.252
	Fa0/1	30.0.0.1	255.255.255.252
PE2	Lo0	192.170.1.4	255.255.255.255
	Fa0/0	30.0.0.2	255.255.255.252
	Fa0/1	40.0.0.1	255.255.255.252
CE2	Lo0	192.170.1.5	255.255.255.255
	Fa0/1	40.0.0.2	255.255.255.252

Tabla 8. Tabla de direccionamiento



## 5.7 Configuración

### 5.7.1 Asignación de direcciones

Al igual que en las anteriores prácticas, primero asignaremos cada una de las direcciones de los interfaces y loopbacks de cada router.

```
PE1#configure terminal
PE1(config)#line console 0
PE1(config-line)#logging synchronous
PE1(config-line)#int loopback0
PE1(config-if)#ip address 192.170.1.2 255.255.255.255
PE1(config-if)#int fa0/0
PE1(config-if)#ip add 20.0.0.1 255.255.255.252
PE1(config-if)#no shutdown
PE1(config-if)#int fa0/1
PE1(config-if)#ip add 10.0.0.2 255.255.255.252
PE1(config-if)#no shutdown
```

Usaremos OSPF como IGP entre todos los routers proveedores, únicamente en los interfaces pertenecientes al backbone MPLS (Router PE1, PE2 y P).

```
PE1(config)#router ospf 1
PE1(config-router)#network 20.0.0.0 0.0.0.3 area 0
PE1(config-router)#network 192.170.1.2 0.0.0.0 area 0
```

### 5.7.2 MPLS

Habilitaremos ip cef, mpls ip y el protocolo de distribución de etiquetas (LDP) a nivel global en los nodos P, PE1 y PE2.

También se asignará mpls ip únicamente en los interfaces pertenecientes al backbone MPLS.

```
PE2(config)#ip cef
PE2(config)#mpls ip
PE2(config)#mpls label protocol ldp
PE2(config)#int fa0/0
PE2(config-if)#mpls ip
```

Comprobaremos si el descubrimiento de vecinos se ha realizado de forma correcta con el comando “*show mpls ldp neighbor*”. Solo deben aparecer las direcciones de las redes y routers pertenecientes a la red del proveedor.

```
P#show mpls ldp neighbor
Peer LDP Ident: 192.170.1.2:0; Local LDP Ident 192.170.1.3:0
TCP connection: 192.170.1.2.646 - 192.170.1.3.38071
State: Oper; Msgs sent/rcvd: 10/11; Downstream
Up time: 00:02:36
LDP discovery sources:
  FastEthernet0/0, Src IP addr: 20.0.0.1
Addresses bound to peer LDP Ident:
  20.0.0.1      192.170.1.2
Peer LDP Ident: 192.170.1.4:0; Local LDP Ident 192.170.1.3:0
TCP connection: 192.170.1.4.56494 - 192.170.1.3.646
State: Oper; Msgs sent/rcvd: 10/11; Downstream
Up time: 00:02:24
LDP discovery sources:
  FastEthernet0/1, Src IP addr: 30.0.0.2
Addresses bound to peer LDP Ident:
  30.0.0.2      192.170.1.4
```

Figura 69. Descubrimiento de vecinos en backbone

### 5.7.3 MP-BGP

Crearemos la sesión BGP únicamente entre los routers PE. Usada para establecer las rutas de los clientes entre cada uno de los dos routers PE, asegurando que cada uno conozca el siguiente salto (la dirección de loopback del PE peer).

Con el comando “**router bgp AS**” habilitaremos BGP en el router. Un AS o Sistema Autónomo consiste en un conjunto de redes con su propia política de enrutamiento. En este caso usaremos el AS 65000, el cual es uno de los más utilizados y extendidos.

Como la sesión BGP se realiza entre peers, es necesario habilitar al peer de cada router con **neighbor** e indicar a qué AS pertenece. El PE vecino será identificado con su dirección de loopback.

También activaremos la familia de direcciones vpnv4. Bajo esta configuración habrá que habilitar las comunidades peer-to-peer con el comando “**neighbor dir\_IP send-community [estandar | extended | both]**”. Las comunidades son un atributo del protocolo BGP usadas para servicios VPN que proporcionan capacidad adicional en el etiquetado de rutas y en el enrutamiento.

Los comandos para PE1 son los siguientes:

```
PE1(config)#router bgp 65000
PE1(config-router)#neighbor 192.170.1.4 remote-as 65000
PE1(config-router)#neighbor 192.170.1.4 update-source Loopback0

PE1(config-router)#address-family vpnv4
PE1(config-router-af)#neighbor 192.170.1.4 activate
PE1(config-router-af)#neighbor 192.170.1.4 send-community extended
PE1(config-router-af)#exit-address-family
```

Realice el mismo proceso en el router PE2.

Para verificar la configuración, usaremos el comando “**show ip bgp vpnv4 all summary**”. Podemos observar que se ha establecido correctamente la conexión con el peer vpnv4.

```

PE2#show ip bgp vpv4 all summary
BGP router identifier 192.170.1.4, local AS number 65000
BGP table version is 16, main routing table version 16
3 network entries using 411 bytes of memory
3 path entries using 204 bytes of memory
3/2 BGP path/bestpath attribute entries using 372 bytes of memory
1 BGP extended community entries using 40 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1027 total bytes of memory
BGP activity 6/3 prefixes, 6/3 paths, scan interval 15 secs

Neighbor      V   AS MsgRcvd MsgSent  TblVer  InQ  OutQ Up/Down  State/PfxRcd
192.170.1.2   4 65000    46     46     16    0    0 00:38:42      3
  
```

Figura 70. Información BGP

#### 5.7.4 VRF

El cliente se situará en una tabla VRF configurada con un RD y RT de importación y exportación.

Primero se define el RD, el cual identifica la ruta VPN y es representado como ASN:nn (Número del Sistema Autónomo). Debe ser único y diferente para cada cliente, para poder identificarlos.

Después definimos los Route Targets, que indican qué rutas se distribuirán al peer PE según la VPN que identifique.

En este caso se ha elegido el mismo valor para RD y RT por simpleza, pero no es necesario.

```

PE2(config)#ip vrf ClienteA
PE2(config-vrf)#rd 65000:100
PE2(config-vrf)#route-target import 65000:100
PE2(config-vrf)#route-target export 65000:100
  
```

También se puede escribir el comando **“route-target both valor\_RT”** para definir los RTs de importación y exportación juntos.

A continuación, se asignará a la VRF creada los interfaces de cada PE con los que comunica, en este caso los fastEthernet0/1:

```

PE2(config-vrf)#int fa0/1
PE2(config-if)#ip vrf forwarding ClienteA
PE2(config-if)#ip add 40.0.0.1 255.255.255.252
  
```

La IP del interfaz se ha tenido que volver a introducir debido a que es eliminada al aplicar el forwarding de VRF.

Realice los pasos anteriores en el router PE1.

Comprobaremos que dicha tabla VRF ha sido correctamente creada con su correspondiente RD y asignada a los interfaces:

```

PE1#show ip vrf
Name           Default RD      Interfaces
ClienteA      65000:100      Fa0/1
  
```

Figura 71. Tabla VRF

### 5.7.5 Routing CE-PE

Para establecer la comunicación entre los routers PE y CE configuraremos un proceso adicional de OSPF. También se puede realizar mediante rutas estáticas o con el protocolo EBGp, configurando un número de AS diferente al del backbone MPLS.

En el caso de los routers CE, bastaría con permitir todas las direcciones IP.

```
CE1(config)#router ospf 1
CE1(config-router)#passive-interface Loopback0
CE1(config-router)#network 0.0.0.0 255.255.255.255 area 0
```

Realice los mismos pasos en el router CE2.

En los PE usaremos un id de proceso distinto a 1 para que no coincida con el proceso ya establecido en el interior del backbone. Si hubiera más VRFs cada id sería distinto.

```
PE1(config)#router ospf 100 vrf ClienteA
PE1(config-router)#network 10.0.0.2 0.0.0.0 area 0
```

```
PE2(config)#router ospf 100 vrf ClienteA
PE2(config-router)#network 40.0.0.1 0.0.0.0 area 0
```

### 5.7.6 Redistribución rutas

Redistribuiremos en ambos routers PE las rutas aprendidas mediante BGP a la VRF y viceversa. Este proceso se realiza bajo la familia de direcciones IPv4.

```
PE2(config)#router bgp 65000
PE2(config-router)#address-family ipv4 vrf ClienteA
PE2(config-router-af)#redistribute ospf 100 vrf ClienteA

PE2(config-router-af)#router ospf 100 vrf ClienteA
PE2(config-router)#redistribute bgp 65000 subnets
```

El comando “**show ip route vrf name\_vrf**” nos mostrará la tabla de routing de nuestra VRF. Las direcciones del peer cliente y servidor aparecen con una letra B indicando que han sido aprendidas mediante BG y no mediante otro protocolo de routing.



```
PE2#sh ip route vrf ClienteA

Routing Table: ClienteA
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

 40.0.0.0/30 is subnetted, 1 subnets
C    40.0.0.0 is directly connected, FastEthernet0/1
 10.0.0.0/30 is subnetted, 1 subnets
B    10.0.0.0 [200/0] via 192.170.1.2, 01:01:27
 192.170.1.0/32 is subnetted, 2 subnets
B    192.170.1.1 [200/2] via 192.170.1.2, 01:01:27
O    192.170.1.5 [110/2] via 40.0.0.2, 00:02:28, FastEthernet0/1
 192.168.1.0/32 is subnetted, 1 subnets
```

Figura 72. Tabla de ruta de la VRF ClienteA

También podemos comprobar la configuración con “show ip bgp vpnv4 vrf ClienteA”

```
PE1#sh ip bgp vpnv4 vrf ClienteA
BGP table version is 30, local router ID is 192.170.1.2
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
Route Distinguisher: 65000:100 (default for vrf ClienteA)
*> 10.0.0.0/30      0.0.0.0           0         32768 ?
*>i40.0.0.0/30     192.170.1.4       0         100    0 ?
*> 192.170.1.1/32  10.0.0.1          2         32768 ?
*>i192.170.1.5/32  192.170.1.4       2         100    0 ?
```

Figura 73. Información direcciones VPN de la tabla BGP

Muestra únicamente las direcciones de red y loopback de los routers cliente. La i que se encuentra delante de las rutas indica que han sido aprendidas mediante internal BGP (iBGP). Al haber usado OSPF como protocolo de routing entre CE y PE, y no external BGP, aparece el símbolo “?” en Path.

### 5.7.7 Verificación final

Para verificar la correcta conectividad de los routers del cliente realizaremos un ping entre CE1 y CE2:

```
CE1#ping 192.170.1.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.170.1.5, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/66/92 ms
```

Figura 74. Ping entre routers del cliente

Los paquetes llegan correctamente, atravesando la red del proveedor.

## 5.8 Actividades propuestas

1. Realice un ping desde el router CE1 a CE2 y capture el tráfico en el enlace PE1-P con el Wireshark. ¿Qué diferencias observa en los paquetes respecto a la práctica 1?

```
> Frame 64: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface 0
▼ Ethernet II, Src: ca:09:0c:3a:00:08 (ca:09:0c:3a:00:08), Dst: ca:08:0c:2c:00:08 (ca:08:0c:2c:00:08)
  > Destination: ca:08:0c:2c:00:08 (ca:08:0c:2c:00:08)
  > Source: ca:09:0c:3a:00:08 (ca:09:0c:3a:00:08)
  Type: MPLS label switched packet (0x8847)
▼ MultiProtocol Label Switching Header, Label: 17, Exp: 0, S: 0, TTL: 254
  0000 0000 0000 0001 0001 .... .. = MPLS Label: 17
  .... .. 000. .... = MPLS Experimental Bits: 0
  .... .. 0 .... = MPLS Bottom Of Label Stack: 0
  .... .. 1111 1110 = MPLS TTL: 254
▼ MultiProtocol Label Switching Header, Label: 20, Exp: 0, S: 1, TTL: 254
  0000 0000 0000 0001 0100 .... .. = MPLS Label: 20
  .... .. 000. .... = MPLS Experimental Bits: 0
  .... .. 1 .... = MPLS Bottom Of Label Stack: 1
  .... .. 1111 1110 = MPLS TTL: 254
> Internet Protocol Version 4, Src: 10.0.0.1, Dst: 192.170.1.5
> Internet Control Message Protocol
```

Figura 75. Paquete MPLS

Observamos que el paquete contiene los mismos campos que un paquete MPLS, pero en este caso aparecen dos etiquetas anidadas: 17 y 20.

La etiqueta 20, lleva el bit S o stack a 1, lo que indica que se trata de la última etiqueta, utilizada para la ruta encaminada mediante OSPF en el backbone MPLS.

La etiqueta 17 pertenece a la VPN y la identifica para que el PE sepa a dónde reenviar el paquete.

2. Observe el establecimiento de la sesión BGP reiniciando el router P con el botón *reload*. ¿Qué tipo de paquetes BGP aparecen?

No.	Time	Source	Destination	Protocol	Length	Info
57	75.461487	192.170.1.2	192.170.1.4	BGP	77	KEEPALIVE Message
80	91.719933	192.170.1.4	192.170.1.2	BGP	73	KEEPALIVE Message
134	190.153078	192.170.1.2	192.170.1.4	BGP	77	KEEPALIVE Message
144	285.539572	192.170.1.4	192.170.1.2	BGP	73	KEEPALIVE Message
195	318.001737	192.170.1.2	192.170.1.4	BGP	77	KEEPALIVE Message
371	778.800183	192.170.1.4	192.170.1.2	BGP	107	OPEN Message
372	779.231600	192.170.1.2	192.170.1.4	BGP	130	OPEN Message, KEEPALIVE Message
373	779.304415	192.170.1.4	192.170.1.2	BGP	73	KEEPALIVE Message
374	779.388834	192.170.1.4	192.170.1.2	BGP	92	KEEPALIVE Message, KEEPALIVE Message
375	779.418068	192.170.1.4	192.170.1.2	BGP	284	UPDATE Message, UPDATE Message
376	780.231664	192.170.1.2	192.170.1.4	BGP	96	KEEPALIVE Message, KEEPALIVE Message
377	780.231921	192.170.1.2	192.170.1.4	BGP	288	UPDATE Message, UPDATE Message
379	782.098044	192.170.1.4	192.170.1.2	BGP	113	KEEPALIVE Message, KEEPALIVE Message, NOTIFICATION Message
440	891.688189	192.170.1.4	192.170.1.2	BGP	73	KEEPALIVE Message
443	894.000021	192.170.1.2	192.170.1.4	BGP	77	KEEPALIVE Message

Figura 76. Captura de paquetes BGP

Se identifican cuatro tipos de paquetes: KeepAlive Message, Open Message, Update Message y Notification Message.

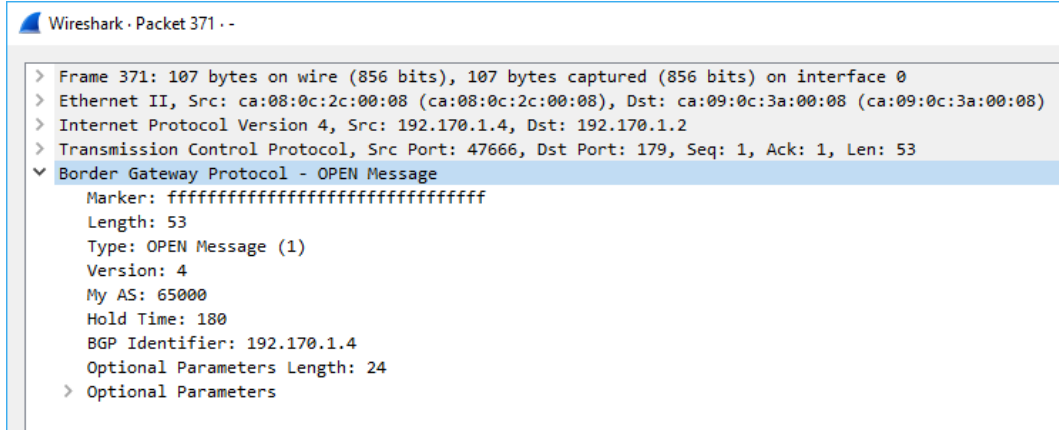
Los mensajes KeepAlive son enviados periódicamente para mantener la conexión y confirmar que ambos extremos siguen activos en la sesión BGP.

El mensaje de Notificación es enviado cuando se produce un error y se tiene que cerrar la sesión.

Los OPEN Message (figura 77) transmiten parámetros para establecer la sesión BGP. Algunos de los parámetros son:

- Número de versión de BGP usada. Es importante que las versiones de ambos peer coincidan.

- Identificador BGP el cual corresponde con la IP del vecino (192.170.1.4)
- Número del AS local, en este caso 65000.



```
Wireshark · Packet 371 · -
> Frame 371: 107 bytes on wire (856 bits), 107 bytes captured (856 bits) on interface 0
> Ethernet II, Src: ca:08:0c:2c:00:08 (ca:08:0c:2c:00:08), Dst: ca:09:0c:3a:00:08 (ca:09:0c:3a:00:08)
> Internet Protocol Version 4, Src: 192.170.1.4, Dst: 192.170.1.2
> Transmission Control Protocol, Src Port: 47666, Dst Port: 179, Seq: 1, Ack: 1, Len: 53
▼ Border Gateway Protocol - OPEN Message
  Marker: ffffffffffffffffffffffffffffffffff
  Length: 53
  Type: OPEN Message (1)
  Version: 4
  My AS: 65000
  Hold Time: 180
  BGP Identifier: 192.170.1.4
  Optional Parameters Length: 24
  > Optional Parameters
```

Figura 77. Paquete OPEN Message BGP

Los UPDATE Messages o mensajes de actualización (figura 78) son enviados al peer PE cada vez que hay una modificación en una ruta o es creada una nueva. Contienen información de las rutas y sus atributos, algunos de ellos son:

- Origin: Indica mediante qué proceso ha sido aprendida la ruta. En este caso es Incomplete al haber sido aprendida por redistribución. Una i indicaría IBGP y e EBGP.
- Community: Los destinos son agrupados en comunidades. Contiene la información de los RTs y AS. En este caso la hemos definido como extendida.
- Mp\_reach\_nlri: Información de la familia ipv4 y del RD.

```

> Path Attribute - ORIGIN: INCOMPLETE
> Path Attribute - AS_PATH: empty
> Path Attribute - MULTI_EXIT_DISC: 0
▼ Path Attribute - LOCAL_PREF: 100
  > Flags: 0x40, Transitive, Well-known, Complete
  Type Code: LOCAL_PREF (5)
  Length: 4
  Local preference: 100
▼ Path Attribute - EXTENDED_COMMUNITIES
  > Flags: 0xc0, Optional, Transitive, Complete
  Type Code: EXTENDED_COMMUNITIES (16)
  Length: 32
▼ Carried extended communities: (4 communities)
  ▼ Route Target: 65000:100 [Transitive 2-Octet AS-Specific]
    > Type: Transitive 2-Octet AS-Specific (0x00)
    Subtype (AS2): Route Target (0x02)
    2-Octet AS: 65000
    4-Octet AN: 100
    > OSPF Domain Identifier: 0:6554112 [Transitive 2-Octet AS-Specific]
    > OSPF Route Type: Area: 0.0.0.0, Type: Network [Transitive Experimental]
    > OSPF Router ID: 40.0.0.1 [Transitive Experimental]
▼ Path Attribute - MP_REACH_NLRI
  > Flags: 0x80, Optional, Non-transitive, Complete
  Type Code: MP_REACH_NLRI (14)
  Length: 33
  Address family identifier (AFI): IPv4 (1)
  Subsequent address family identifier (SAFI): Labeled VPN Unicast (128)
▼ Next hop network address (12 bytes)
  Next Hop: Empty Label Stack RD=0:0 IPv4=192.170.1.4
  Number of Subnetwork points of attachment (SNPA): 0
▼ Network layer reachability information (16 bytes)
  ▼ BGP Prefix
    Prefix Length: 118
    Label Stack: 19 (bottom)
    Route Distinguisher: 65000:100
    MP Reach NLRI IPv4 prefix: 40.0.0.0
  
```

Figura 78. Paquete UPDATE Message BGP

3. Una de las técnicas que permite el protocolo BGP es el uso de reflectores de rutas. Añada un nuevo router en la backbone MPLS conectado únicamente al router P (figura 79). Asigne las direcciones y protocolos necesarios para la comunicación y rehaga las sesiones BGP de los PEs con el nuevo router de la misma forma hecha anteriormente y añadiendo el comando “neighbor IP\_loopback route-reflector-client” en la familia vpnv4.

¿Qué ventajas presenta añadir un Reflector de Rutas en la red?

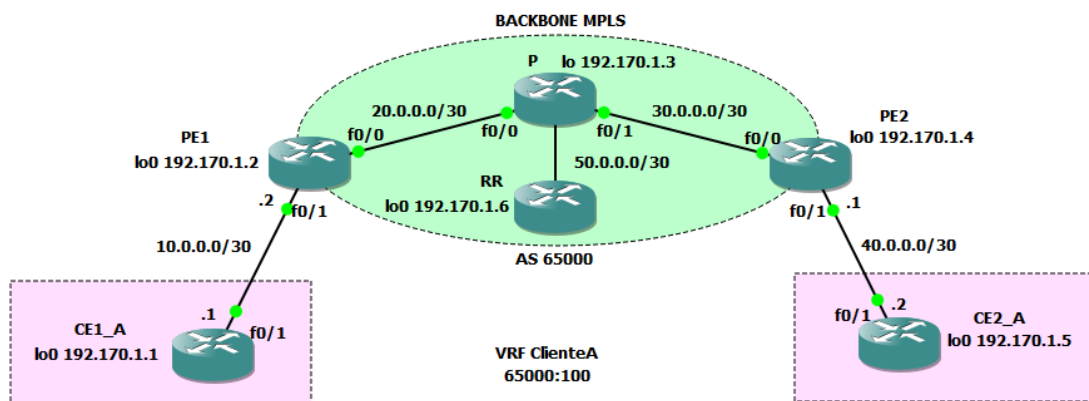


Figura 79. Diagrama de red con Reflector de Rutas

Los comandos que se introducirán en el Reflector de Rutas son los siguientes:

```
RR(config)#router bgp 65000
RR(config-router)#neighbor 192.170.1.2 remote-as 65000
RR(config-router)#neighbor 192.170.1.4 remote-as 65000
RR(config-router)#neighbor 192.170.1.4 update-source Loopback 0
RR (config-router)#neighbor 192.170.1.2 update-source Loopback 0

RR(config-router)#address-family vpvv4
RR (config-router-af)#neighbor 192.170.1.4 activate
RR(config-router-af)#neighbor 192.170.1.2 activate
RR(config-router-af)#neighbor 192.170.1.2 send-community both
RR(config-router-af)#neighbor 192.170.1.4 send-community both
RR(config-router-af)#neighbor 192.170.1.4 route-reflector-client
RR(config-router-af)#neighbor 192.170.1.2 route-reflector-client
```

Una vez establecidas todas las sesiones BGP observamos como tiene conexión con los routers PE.

```
RR#sh ip bgp sum
BGP router identifier 192.170.1.6, local AS number 65000
BGP table version is 1, main routing table version 1

Neighbor      V    AS MsgRcvd MsgSent  TblVer  InQ  OutQ Up/Down  State/PfxRcd
192.170.1.2   4 65000      6      9        1    0    0 00:01:06      0
192.170.1.4   4 65000      6      8        1    0    0 00:00:45      0
```

Figura 80. Conexiones BGP del router reflector

Los reflectores de rutas son vecinos iBGP utilizados para informar de rutas a otros routers iBGP. Cuando reciben una ruta de sus clientes la anuncia al resto y si la recibe de otro equipo no cliente suyo únicamente la anuncia a sus clientes.

Al igual que el router P, no tiene ninguna información de las VRFs configuradas ni del direccionamiento de los clientes.

Es muy útil para grandes topologías de redes en las que hay que conectar múltiples sitios remotos de uno o varios clientes. Solo se realizaría la sesión BGP entre el router PE del sitio y el reflector, evitando que se realicen sesiones con cada uno de los peer PE y reduciendo la cantidad de sesiones iBGP.

#### 4. Si se quisiera añadir un nuevo cliente a la red, ¿qué cambios habría que realizar en la configuración y topología de la red?

En la red deberíamos añadir dos nuevos routers CE conectados cada uno a su respectivo PE.

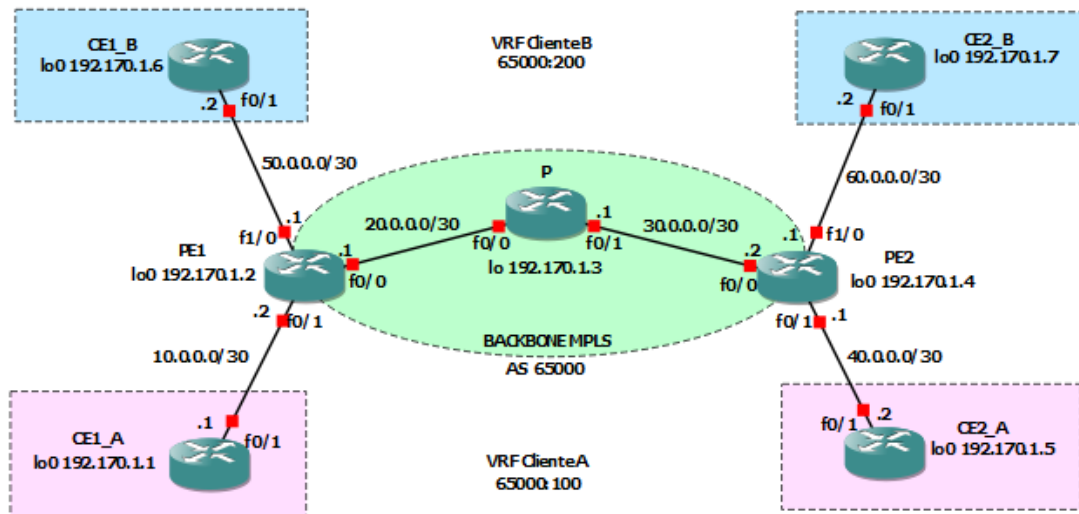


Figura 81. Diagrama de red con 2 clientes

Habría que crear una VRF nueva y definir su RD y RTs de importación y exportación, siendo diferentes a los del otro cliente para evitar solapamiento. Asignar la VRF creada a los interfaces de PE conectados a los routers cliente.

```
PE2#sh ip vrf
```

Name	Default RD	Interfaces
ClienteA	65000:100	Fa0/1
ClienteB	65000:200	Fa1/0

Figura 82. Tablas VRF creadas

Posteriormente, añadir mediante el protocolo de routing OSPF la nueva red que conecta el router CE al router PE, con un id de proceso diferente al del otro cliente.

Por último, modificar la familia ipv4 y redistribuir las nuevas rutas creadas. También se puede realizar mediante E-BGP con un identificador AS distinto.

```
PE1#sh ip route vrf ClienteB

Routing Table: ClienteB
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

 50.0.0.0/30 is subnetted, 1 subnets
 C    50.0.0.0 is directly connected, FastEthernet1/0
 192.170.1.0/32 is subnetted, 2 subnets
 B    192.170.1.7 [200/2] via 192.170.1.4, 00:01:04
 O    192.170.1.6 [110/2] via 50.0.0.2, 00:00:24, FastEthernet1/0
 60.0.0.0/30 is subnetted, 1 subnets
 B    60.0.0.0 [200/0] via 192.170.1.4, 00:01:04
```

Figura 83. Rutas tabla VRF ClienteB



## Capítulo 6. Bibliografía

- [1] Pepelnjak, I., and Guichard, J., “Arquitecturas MPLS y VPN”, Cisco Press, 2002.
- [2] Ernesto Ariganello, Enrique Barrientos Sevilla, “Redes CISCO. CCNP a fondo. Guía de estudio para profesionales”, Ra-Ma S.A Ed., 2010.
- [3] Documentation GNS3 <http://docs.gns3.com/> [Online]
- [4] Cisco “Configuración básica de MPLS usando OSPF” [https://www.cisco.com/c/es\\_mx/support/docs/multiprotocol-label-switching-mpls/mpls/13736-mplsospf.html](https://www.cisco.com/c/es_mx/support/docs/multiprotocol-label-switching-mpls/mpls/13736-mplsospf.html) [Online]
- [5] “Cisco IOS Multiprotocol Label Switching Configuration Guide”, Release 12.2 SR, Cisco Systems Inc., 2010