_ TELECOM ESCUELA TÉCNICA VLC SUPERIOR DE UPV INGENIEROS DE TELECOMUNICACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Configuración de servicios VPN en entorno MPLS

José Cano Sáez

Tutor: Víctor Miguel Sempere Payá

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 2017-18

Valencia, 2 de julio de 2018

Resumen

El objetivo del presente trabajo es ofrecer un enfoque más practico al alumno sobre los conocimientos de redes MPLS aprendidos en la teoría de la asignatura de “Redes Públicas de Transporte”.

Para ello se han elaborado tres prácticas que profundizan en algunos de los conceptos más importantes de dichas redes, como son: el etiquetado de los paquetes, la Ingeniería de Tráfico o las VPN’s. En dichas prácticas se propone al alumno el montaje de una red real en el laboratorio con equipamiento Cisco, así como la configuración de la misma. Además, se realizarán capturas de paquetes en el entorno de la red con “*Wireshark”*, para poder identificar los diferentes campos de las cabeceras que conforman dichos paquetes.

Resum

L’objectiu del present treball és oferir un enfocament més practique a l'alumne sobre els coneixements de xarxes MPLS apresos en la teoria de l'assignatura de “Redes Públicas de Transporte”.

Per açò s'han elaborat tres pràctiques que aprofundeixen en alguns dels conceptes més importants d'aquestes xarxes, com són: l'etiquetatge dels paquets, l'Enginyeria de Tràfic o les VPN’s. En aquestes pràctiques es proposa a l'alumne el montatge d'una xarxa real en el laboratori amb equipament Cisco, així com la configuració de la mateixa. A més es realitzaran captures de paquets a l'entorn de la xarxa amb “Wireshark”, per a poder identificar els diferents camps de les capçaleres que conformen aquests paquets.

Abstract

The aim of the present work is to offer an approach more practise to the student on the knowledges of MPLS nets learnt in the theory of the subject of “Redes Públicas de Transporte”.

For this, three practices have been elaborated, these deepen in some of the most important concepts of these nets, as they are: the labeling of the packages, the Traffic Engineering or the VPN's. In these practices it is proposed to the student the setting of a real net in the laboratory with Cisco equipment, as well as the configuration of the same. Besides they will make captures of packages in the net environment with “Wireshark”, to be able to identify the different fields of the headers that conform these packages.

Índice

Capítulo 1. Objetivo

El objetivo del presente Trabajo Fin de Grado, en adelante TFG, es la realización de un dossier de prácticas para la asignatura de "*Redes Públicas de Transporte*", relacionadas con la parte del temario de MPLS de dicha asignatura.

Con dichas prácticas se pretende que los alumnos afiancen los conocimientos sobre MPLS obtenidos en las clases teóricas de la asignatura, mediante el montaje y configuración de una red MPLS con los equipos existentes en el laboratorio de Redes Telemáticas.

Dichas prácticas cubrirán los siguientes aspectos:

- diseño de la red teniendo en cuenta las prestaciones de los equipos presentes en el laboratorio.
- estudio de la arquitectura y el funcionamiento básico del protocolo MPLS, así como los comandos necesarios para su configuración, verificando en cada una de las etapas de diseño el funcionamiento de la red.
- estudio del protocolo LDP.
- se introducirán los conceptos de ingeniería de tráfico y balanceo de cargas.
- estudio del protocolo RSVP.
- establecimiento de VPN's de nivel 3.

Capítulo 2. Práctica 1 “Configuración básica de MPLS”

2.1 Introducción

El rápido crecimiento de Internet ha tenido un gran impacto sobre los tipos de servicios solicitados por los consumidores y el tipo de rendimiento que exigen a los productos que desean utilizar. En consecuencia, los proveedores de servicios se han visto en la obligación de desarrollar, gestionar y mejorar la infraestructura de sus redes **IP** en términos de rendimiento y control del tráfico a través de la Ingeniería de Tráfico.

La **Ingeniería de Tráfico** ofrece varios mecanismos para optimizar el rendimiento, modelado, medición, caracterización y control de tráfico en una red, para obtener objetivos específicos de rendimiento y ofrecer servicios competitivos de calidad a los clientes de esta.

Tradicionalmente el algoritmo de enrutamiento más utilizado por los ISP’s (Internet Services Provider) ha sido es el de “l*a ruta más corta*”, el problema de este tipo de algoritmos es que ciertos enlaces en la red se ven congestionados, mientras que existen otras rutas disponibles que no son utilizadas. Este tipo de enrutamiento provoca demoras impredecibles y pérdida de datos, sin embargo, no ha sido un problema para las aplicaciones tradicionales de Internet como son: web, correo electrónico, transferencia de archivos y similares. Pero la nueva generación de aplicaciones que incluyen audio y video streaming, exigen alto rendimiento, ancho de banda y baja latencia.

Para paliar esta situación, a mediados de la década de los 90 un grupo de investigación de la empresa CISCO, proponía un sistema de conmutación basado en etiquetas. El propósito principal era evitar tener routers actualizando y mirando continuamente las tablas de enrutamiento IP, lo que suponía una pérdida de tiempo fácilmente evitable. De aquí surge la tecnología MPLS.

MPLS (Multi-Protocol Label Switching), es una tecnología de conmutación de paquetes que se sitúa entre las capas 2 y 3, ver figura 1, que realiza enrutamiento de tráfico de manera rápida y efectiva, además de facilitar la Ingeniería de Tráfico, el despliegue de técnicas QoS o la utilización de VPN’s.

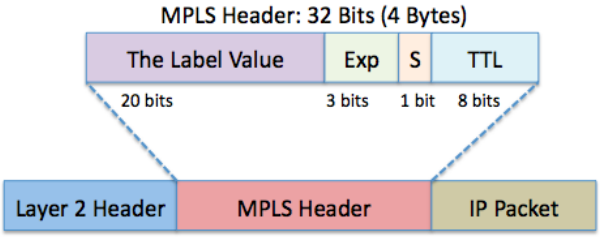

Figura 1. Cabecera MPLS

En las redes MPLS se utiliza la técnica de conmutación de etiquetas en lugar de los mecanismos clásicos de enrutamiento IP. La idea básica es tomar el software de control de un router IP, integrarlo con el rendimiento de reenvío con cambio de etiqueta de un switch ATM y crear un router extremadamente rápido y eficiente en cuanto a coste. En otras palabras, separar completamente el plano de control (enrutamiento) del plano de datos (reenvío de paquetes).

Los protocolos de enrutamiento de nivel 3 como OSPF o BGP se usan únicamente para funciones de control, ya que las decisiones de enrutamiento se toman en función de la etiqueta MPLS y no de la cabecera IP.

El etiquetado del tráfico se realiza a la entrada de la red, pero no en su salida. Es usado únicamente en los routers y es independiente del protocolo usado, lo que le permite ser utilizado sobre otros protocolos distintos a IP. Además, está diseñado para operar sobre cualquier tecnología de nivel de enlace de datos y red, ver figura 2, como podrían ser: Ethernet, ATM, Frame Relay, xDSL, SDH, lo que facilita su integración en redes ya existentes y el uso de una infraestructura de red unificada.

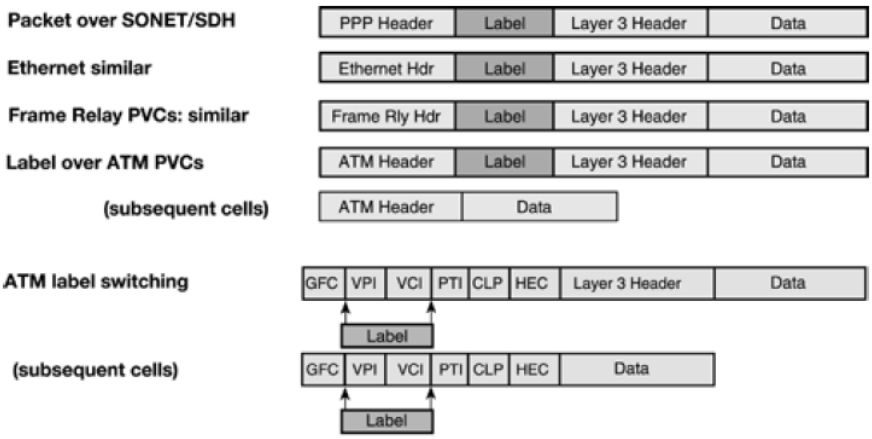

Figura 2. Técnicas de encapsulación MPLS

Así, un router asigna una etiqueta a cada una de las entradas de la tabla de enrutamiento y las distribuye a sus routers vecinos. Luego, cuando se pasan paquetes entre ellos, los routers solo tienen que leer la etiqueta MPLS para identificar el siguiente salto donde enviar el paquete. De esta forma los paquetes "fluyen" de un extremo a otro de la red y se consigue un enrutamiento a mayor velocidad a la vez que se disminuye el retardo y el jitter.

2.2 Componentes de una red MPLS

2.2.1 FEC (Forwarding Equivalence Class)

Es un conjunto de paquetes con similares características que son reenviados con la misma prioridad a través de un mismo LSP (Label Switched Path). Este grupo de paquetes están todos identificados por la misma etiqueta. Las FEC son una manera de distinguir un tipo de tráfico de otro.

2.2.2 LSP (Label Switch Path)

El LSP es el camino compuesto por uno o varios LSR (Label Switched Router) a través del cual se transmiten todos los paquetes pertenecientes a un determinado FEC. Estos caminos son unidireccionales, es decir, solo transmiten tráfico en un sentido.

MPLS soporta dos opciones para la creación de un LSP:

- LSP salto a salto: para el establecimiento de un LSP salto a salto, cada nodo elige de forma independiente el siguiente salto para encaminar un FEC.
- LSP explícito: en el caso de un LSP explicito, los LSR's no eligen de forma independiente, sino que un sólo LSR es el que define todos o la mayoría de los LSR's que conforman el LSP.

2.2.3 LSR (Label Switching Router)

Los LSR son todos aquellos routers que se encuentran dentro de una red MPLS. A diferencia de un router convencional, estos routers reenvían los paquetes en función de las etiquetas de los paquetes recibidos y no en función de la dirección IP de destino.

En una red MPLS podemos encontrar dos tipos de LSR:

- **LSR Core (LSR)**: no examinan la cabecera del paquete, solo miran la etiqueta MPLS, las intercambian y reenvían paquetes en base a estas.
- **LSR Edge (LER):** los LER son los routers frontera que operan en los bordes de una red MPLS. Estos routers son los encargados de convertir los paquetes IP en paquetes MPLS, o viceversa. Dependiendo de esta función, podemos diferenciar entre los routers de entrada (ingress) y los routers de salida (egress). Los primeros se sitúan en la entrada de la red y se encargan de asignar un FEC a los paquetes que reciben y de etiquetarlos para

que lleguen a su destino. Los routers de salida son los encargados de hacer la acción contraria, eliminar la etiqueta y entregar el paquete a su destinatario. En la figura 3 se pueden apreciar los diferentes tipos de routers existentes en una red MPLS.

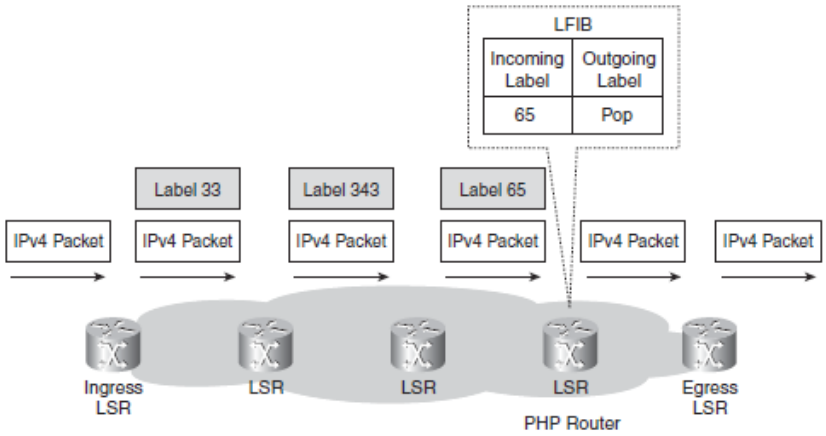

Figura 3. Topología red MPLS

Las funciones de un LSR son:

- mantenimiento de la tabla de encaminamiento **RIB** (Routing Information Base), proporcionada por el protocolo IGP.
- asignar e intercambiar etiquetas, en base a la tabla **LIB** (Label Information Base), o "*MPLS IP Routing Control"* como la denomina CISCO, obtenida mediante el uso de LDP.
- recibir paquete etiquetados, cambiar la etiqueta y reenviar al siguiente LSR, en función de la información contenida en la tabla **LFIB** (Label Forwarding Instance Base), o también llamada "*Label Forwarding Table*", que aúna la información de las anteriores tablas.

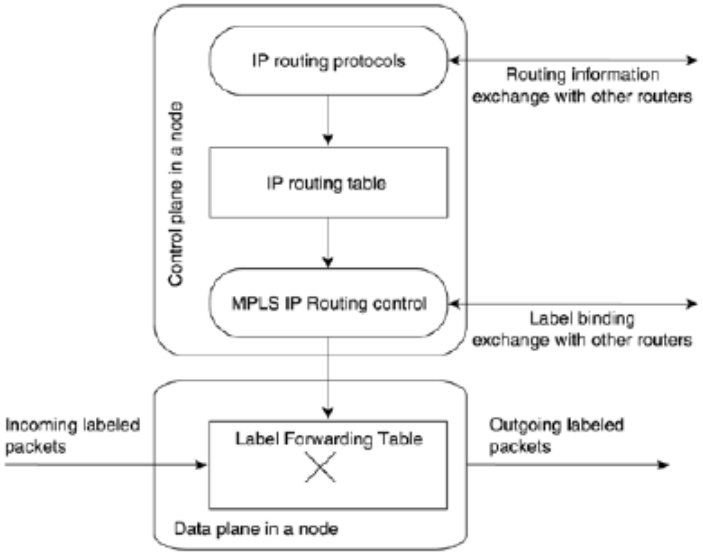

Figura 4. Funciones de un LSR

Un LER realiza todas las funciones de un LSR y adicionalmente:

- recibe paquetes IP, los etiqueta y envía.
- elimina etiquetas de los paquetes MPLS para reenviarlos como IP.

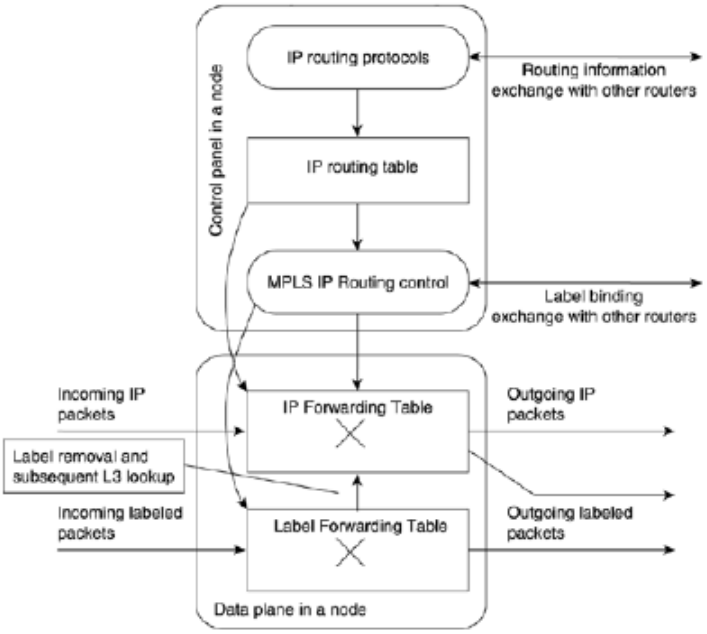

Figura 5. Funciones de un LER

En la figura 6 se pueden observar los comandos existentes en CISCO para mostrar las diferentes tablas utilizadas por MPLS y la forma en que estas se encuentran relacionadas.

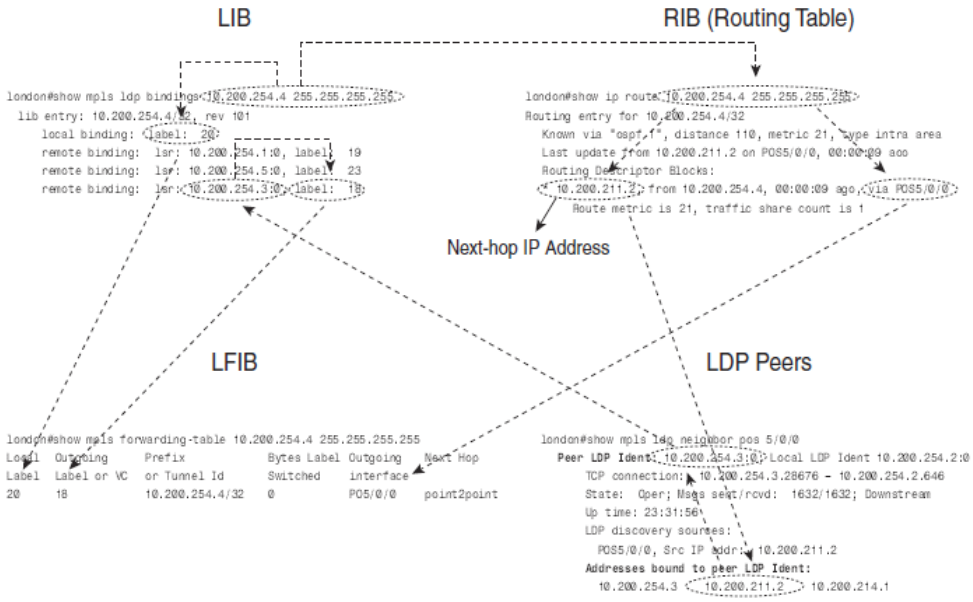

Figura 6. Tablas utilizadas en MPLS

2.2.4 Protocolos de distribución de etiquetas

Para mapear etiquetas en un LSP es necesario un protocolo de distribución de etiquetas. Existen diversas propuestas de protocolos para realizar dicha función como son:

- protocolo de distribución de etiquetas LDP (Label Distribution Protocol).
- protocolo de reserva de recursos con extensiones de Ingeniería de Tráfico RSVP-TE.
- protocolo de enrutamiento basado en restricciones LDP (CR-LDP).
- Multi-protocolo BGP.

Vamos a profundizar un poco en el protocolo **LDP**, que es el que utilizaremos en esta práctica.

El LDP es un protocolo que se usa para establecer y mantener asociaciones de etiquetas para un LSP asociado a un FEC.

El primer paso para del protocolo consiste en realizar el descubrimiento de vecinos, para ello cada router de la red envía un mensaje "*Hello*", utilizando el protocolo UDP y el puerto 646, es un mensaje de broadcast y se envía a la dirección IP multicast 224.0.0.2.

Una vez dos LSR se han descubierto a través de los mensajes LDP "*Hello*", intentarán establecer una sesión LDP entre ellos.

El router con dirección IP más alta intentará establecer una conexión TCP a través del puerto 646. Una vez negociados los parámetros de la conexión, la sesión LDP es establecida.

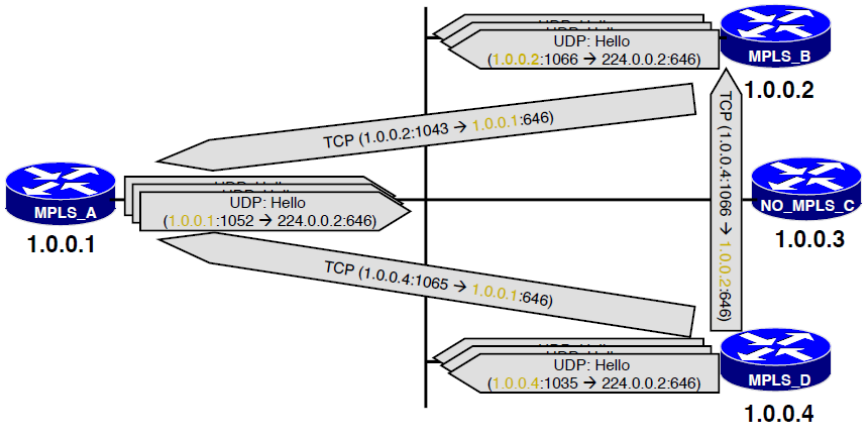

Figura 7. Establecimiento sesión LDP

En los routers CISCO existe un comando con el que se puede comprobar fácilmente que sesiones LDP tiene un router establecidas con diferentes peers, el comando es: **show mpls ldp neighbor**. Podemos ver el resultado de la ejecución de este en la figura 8.

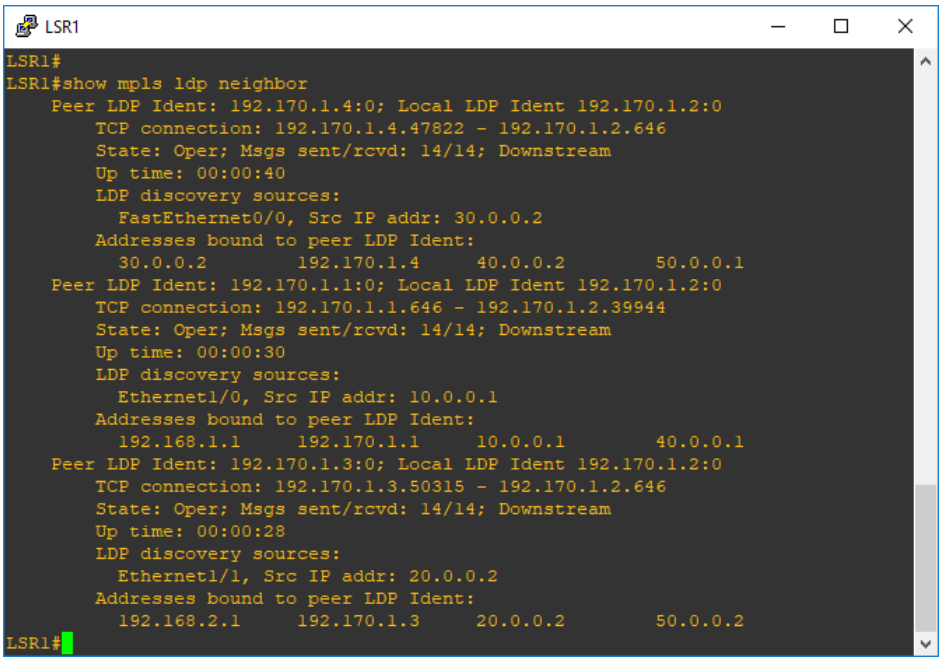

Figura 8. Conexiones LDP establecidas

Cuando se crea un enlace y se establece una sesión LDP entre dos LSR's se identifica como "*Hello Adjacency*". Cada LSR mantiene un timer para cada "*Hello Adjacency*" que se restaura cada vez que recibe un nuevo mensaje "*Hello*", si expira, se borra y se termina la sesión LDP.

Además del citado temporizador, cada router mantiene otro timer denominado "*KeepAlive*" que se resetea con cada LDP UDP recibida desde un peer, sirve para mantener la conexión activa. Si un LSR no tiene nada que enviar, envía un "*KeepAlive*" evitando así que expire la sesión.

Una vez establecida la sesión LDP se puede comenzar con el intercambio de etiquetas y la creación de caminos (LSP). El camino óptimo será escogido por el protocolo de routing IP, en nuestro caso utilizaremos OSPF. Se puede ver en la figura 9 de manera gráfica todo el proceso de establecimiento de la sesión LDP.

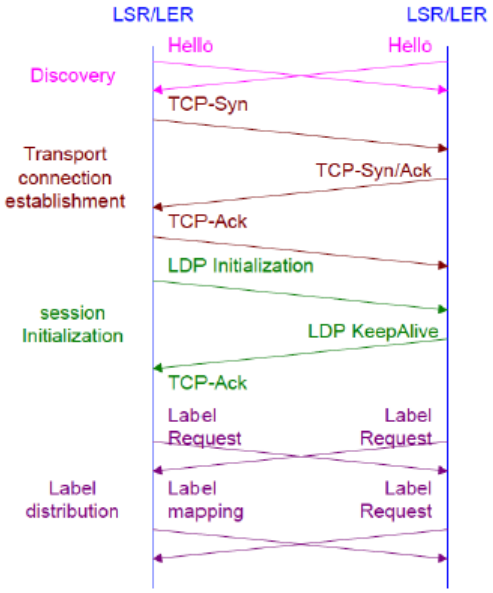

Figura 9. Establecimiento de la sesión LDP e intercambio de etiquetas

En la gestión del protocolo LDP, los peers utilizan los siguientes tipos de mensajes:

- **Notification**: informa a un LDP peer de un error fatal o de información de estado.
- **Hello**: se intercambian como parte del mecanismo de descubrimiento LDP.
- **Inicialization**: se solicita el establecimiento de una sesión LDP.
- **KeepAlive**: se monitoriza la integridad de una sesión LDP.
- **Address**: antes del envío de un "*label mapping*" o "*label request*" un LSR publicita sus direcciones de interface utilizando este mensaje.
- **Address withdraw**: se retira una dirección publicitada previamente.
- **Label mapping**: un LSR publicita un mapeado de una etiqueta a un FEC a sus LDP peers. (En LDP un FEC podría ser un prefijo o una IP de un LSR destino)
- **Label request**: un LSR lo envía a un LDP peer para solicitar un mapeado a un FEC concreto.
- **Label withdraw**: un LSR comunica a otro en el LDP peer que no puede utilizar un mapeado a un FEC que estaba utilizando.
- **Label release**: un LSR comunica a otro en el LDP peer que no necesita un mapeado específico a un FEC que previamente pidió.

2.3 Objetivos de la práctica

El objetivo de la presente práctica es familiarizarse con la tecnología y los conceptos básicos de MPLS (Multi-Protocol Label Switching), el protocolo LDP, así como su configuración en una red implementada con routers Cisco Systems.

Para ello, se deberán realizar las siguientes actividades:

- configurar el protocolo de routing IP, en nuestro caso se utilizará OSPF.
- introducir en los routers los comandos necesarios para la configuración de la red MPLS.
- verificar el comportamiento de la red MPLS, así como comprobar y visualizar las diferentes tablas utilizadas por MPLS en su funcionamiento.
- visualizar los diferentes paquetes que circulan por la red e identificar los campos pertenecientes a la configuración MPLS y LDP.

2.4 Materiales a utilizar

Para la realización de la presente práctica se utilizarán los routers CISCO 1841 disponibles en el laboratorio. En la figura 10 podemos ver una imagen de la trasera del mencionado router y en la tabla 1 la descripción de cada uno de los elementos presentes en la misma.

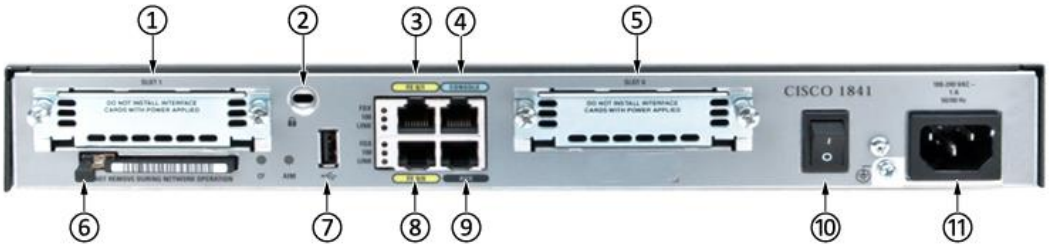

Figura 10. Trasera router CISCO 1841

ID	DESCRIPCIÓN
1	Slot de expansión 1
2	Accesorio para bloqueo
3	Puerto Fast Ethernet 0/0
4	Puerto de consola
5	Slot de expansión 0
6	Unidad compact flash
7	Puerto USB
8	Puerto Fast Ethernet 0/1
9	Puerto auxiliar
10	Interruptor de encendido
11	Entrada de alimentación

Tabla 1. Identificación elementos trasera CISCO 1841

Además de los citados routers, será necesaria una tarjeta expansora de 4 puertos Ethernet por cada router, la HWIC-4ESW, ver figura 11, también disponible en el laboratorio. La tarjeta deberá de ser instalada en el Slot 0 antes de conectar el router a la corriente.

El modelo 1841 sólo dispone de dos puertos Fast Ethernet, en la figura 12 se puede observar que dicho número de puertos es insuficiente para poder realizar el montaje de red propuesto, de ahí la necesidad de la tarjeta expansora, la cual nos proporciona 4 puertos adicionales de capa 2. Al ser puertos de capa 2, no se les podrá asignar directamente una dirección IP, por lo que será necesaria la utilización de una VLAN para poder asignar una dirección a cada uno de estos interfaces. Más adelante en el presente documento se explicará que comandos se deben de utilizar para la configuración de las diferentes VLAN's presentes en la topología de red propuesta.

La numeración de los interfaces de la HWIC-4ESW, como podemos apreciar en la figura 11, se inicia desde la derecha, por lo que el primer interfaz de la derecha será el FE 0/0/0 y el último el FE 0/0/3.

Figura 11. Tarjeta expansora Ethernet HWIC-4ESW

Aparte de los routers, cables de alimentación y latiguillos Fast Ethernet cruzados, será necesaria la utilización de tres PC's del laboratorio, en uno de ellos deberá estar instalado el analizador de redes "*Wireshark*" para realizar capturas de tráfico circulante por la red.

2.5 Diagrama de la topología de red

En la siguiente figura se observa la topología de la red a montar.

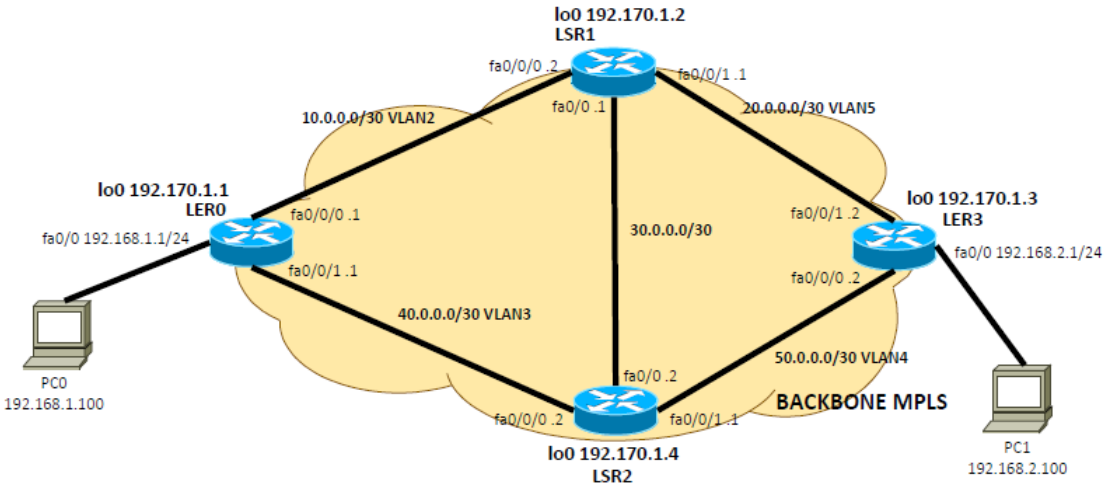

Figura 12. Diagrama de la red MPLS a configurar

En la siguiente tabla se especifican las diferentes direcciones de cada uno de los interfaces de cada router y PC's conectados a la red propuesta.

DISPOSITIVO	INTERFAZ	DIRECCIÓN IP	MÁSCARA DE RED	GATEWAY	VLAN
LER0	Fa0/0/0	10.0.0.1	255.255.255.252	-	VLAN 2
	Fa0/0/1	40.0.0.1	255.255.255.252	-	VLAN 3
	Fa0/0	192.168.1.1	255.255.255.0	-	-
	Lo0	192.170.1.1	255.255.255.255	-	-
LSR1	Fa0/0/0	10.0.0.2	255.255.255.252	-	VLAN 2
	Fa0/0/1	20.0.0.1	255.255.255.252	-	VLAN 5
	Fa0/0	30.0.0.1	255.255.255.252	-	-
	Lo0	192.170.1.2	255.255.255.255	-	-
LSR2	Fa0/0/0	40.0.0.2	255.255.255.252	-	VLAN 3
	Fa0/0/1	50.0.0.1	255.255.255.252	-	VLAN 4
	Fa0/0	30.0.0.2	255.255.255.252	-	-
	Lo0	192.170.1.4	255.255.255.255	-	-
LER3	Fa0/0/0	50.0.0.2	255.255.255.252	-	VLAN 4
	Fa0/0/1	20.0.0.2	255.255.255.252	-	VLAN 5
	Fa0/0	192.168.2.1	255.255.255.0	-	-
	Lo0	192.170.1.3	255.255.255.255	-	-
PC0	NIC	192.168.1.100	255.255.255.0	192.168.1.1	-
PC1	NIC	192.168.2.100	255.255.255.0	192.168.2.1	-

Tabla 2. Tabla de direccionamiento

Como se puede observar en la tabla 2, en todos los routers se utiliza una interfaz de loopback. La interfaz loopback es una interfaz virtual de red que identifica al propio dispositivo ante cualquier protocolo que lo requiera, como OSPF o LDP. Al no estar vinculada a una interfaz física, está siempre operativa.

Si no existiera esta interfaz los protocolos como OSPF o LDP utilizarían para identificar al router su dirección IP más alta, en tal caso, si ésta cayera el router debería utilizar otra dirección IP, lo que nos provocaría problemas de convergencia en la red e incluso si no se detectara ninguna interfaz activa perderíamos las sesiones OSPF, quedando el router descartado de la misma.

2.6 Configuración de la red

Antes de comenzar con la configuración, revisar el Anexo 0, donde se explica como cambiar la version del Sistema IOS del router. Durante el desarrollo de esta práctica y las siguientes trabajaremos con la release de IOS advipservicesk9-mz.150-1.M10, la cual soporta MPLS.

Empecemos ahora sí a configurar los diferentes equipos que conforman la red.

En primer lugar como paso previo, se deberá de borrar cualquier configuración existente en el router, para asegurarnos que no haya ninguna interferencia con la configuración que vamos a realizar. Para ello debemos de utilizar el comando **erase**, para borrar el archivo de configuración que se carga al inicializar el router, en modo priviligiado (utilizar el comando **enable** para acceder

al modo privilegiado). Introducir los comandos que aparecen a continuación, el resultado debe de ser similar al mostrado seguidamente:

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

A continuación, borraremos el archivo de configuración de las VLAN's en le caso que este exista, para ello utilizaremos el comando **delete**, la sintaxis es la siguiente:

```
Router#delete vlan.dat
Delete filename [vlan.dat]?
Delete flash:/vlan.dat? [confirm]
Router#
```

Al volver el indicador, ejecutar el comando **reload**, cuando se le solicite, confirmar el objetivo.

Después de que el router finalice el proceso de inicio, debemos elegir "**no**" utilizar el diálogo de configuración, como se muestra a continuación:

```
System Bootstrap, Version 12.3(8r)T8, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 2004 by cisco Systems, Inc.
PLD version 0x10
GIO ASIC version 0x127
c1841 processor with 131072 Kbytes of main memory
Main memory is configured to 64 bit mode with parity disabled


Readonly ROMMON initialized
program load complete, entry point: 0x8000f000, size: 0xc100

Initializing ATA monitor library.....
program load complete, entry point: 0x8000f000, size: 0xc100

Initializing ATA monitor library.....

program load complete, entry point: 0x8000f000, size: 0x27ce8b4
Self decompressing the image : ########################################################
################################################################################
#####
################################################################################
#####
############################## [OK]

Smart Init is enabled
smart init is sizing iomem
  ID          MEMORY_REQ        TYPE
             0X003AA110 public buffer pools
             0X00211000 public particle pools
             0X00020000 Crypto module pools
```

```
0X0056          0X00035600 Card in slot 0
                0X000021B8 Onboard USB

If any of the above Memory Requirements are
"UNKNOWN", you may be using an unsupported
configuration or there is a software problem and
system operation may be compromised.

Allocating additional 1448112 bytes to IO Memory.
PMem allocated: 123731968 bytes; IOMem allocated: 10485760 bytes

           Restricted Rights Legend

Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

         cisco Systems, Inc.
         170 West Tasman Drive
         San Jose, California 95134-1706



Cisco IOS Software, 1841 Software (C1841-ADVIPSERVICESK9-M), Version 15.0(1)M10,
 RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2013 by Cisco Systems, Inc.
Compiled Tue 26-Feb-13 12:36 by prod_rel_team
Image text-base: 0x600189D0, data-base: 0x63589100


This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

Installed image archive
Cisco 1841 (revision 5.0) with 120832K/10240K bytes of memory.
Processor board ID FCZ0935106L
6 FastEthernet interfaces
1 Virtual Private Network (VPN) Module
DRAM configuration is 64 bits wide with parity disabled.
```

```
191K bytes of NVRAM.
62720K bytes of ATA CompactFlash (Read/Write)


        --- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: no


Press RETURN to get started!

Cisco IOS Software, 1841 Software (C1841-ADVIPSERVICESK9-M), Version 15.0(1)M10,
RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2013 by Cisco Systems, Inc.
Compiled Tue 26-Feb-13 12:36 by prod_rel_team

 Router>
```

Antes de empezar a configurar nada, vamos a evitar que cuando se estén introduciendo comandos en la consola o revisando el resultado de un comando **show**, nos molesten las interrupciones que provocan los mensajes de logging de algunos sucesos estándar (p.e. el estado de las interfaces) o de un debug, bastará con introducir los siguientes comandos:

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#line console 0
Router(config-line)#logging synchronous
Router(config-line)#exit
```

Seguidamente desactivaremos la busqueda de DNS, de esta forma, cualquier error en la escritura de un comando simplemente hará que aparezca un mensaje indicando que el comando es desconocido o que no ha podido localizar el nombre de host, y evitará que el dispositivos nos deje colgados unos cuantos segundos hasta que aparezca el mensaje "*Unknown command or computer name, or unable to find computer address*". Los comandos a introducir son los siguientes:

```
Router#configure terminal
Router(config)#no ip domain lookup
Router(config)#exit
```

Antes de introducir cualquier configuración en el router, vamos a asignar a cada router el mismo nombre que aparece en la figura 12, para ello se utiliza el comando **hostname**, por ejemplo en el caso del LER0, la forma de cambiar el nombre sería:

```
Router(config)#hostname LER0
LER0(config)#
```

Repetir de manera análoga el anterior paso en cada uno de los restantes routers.

Empecemos realmente con la configuración propiamente dicha, en primer lugar, asignaremos las direcciones a cada uno de los diferentes interfaces de los routers, los comandos a introducir en el caso del LER0 serán los siguientes:

```
LER0(config)#interface Loopback0
LER0(config-if)#ip address 192.170.1.1 255.255.255.255
LER0(config-if)#exit
LER0(config)#interface FastEthernet0/0
LER0(config-if)#ip address 192.168.1.1 255.255.255.0
LER0(config-if)#no shutdown
LER0(config-if)#exit
LER0(config)#exit
```

Como se ha mencionado anteriormente, las interfaces de la tarjeta de expansión son de capa 2, por lo que para poder darles una dirección IP, es necesario previamente que sean asignadas a una VLAN. Para ello deberemos de declarar las VLAN's que utilizaremos, en el caso de LSR2, la declaración se haría de la siguiente manera.

```
LSR2#vlan database
LSR2(vlan)#vlan 3
VLAN 3 added:
        Name: VLAN0003
LSR2(vlan)#vlan 4
VLAN 4 added:
        Name: VLAN0004
LSR2(vlan)#exit
APPLY completed.
Exiting….
LSR2#
```

Crear todas las VLAN's necesarias en el resto de routers de la red.

Una vez creadas las VLAN, sus interfaces deben de ser declarados como como ***trunk***, con lo que el comando a introducir debería de ser el siguiente: **switchport mode *trunk***.

Un puerto trunk puede transportar tráfico de múltiples VLAN, por lo que, podemos tener múltiples VLAN en los routers y solo un enlace para transportar todo el tráfico. Los próximos comandos indican la forma de realizar la asignación de interfaces a la VLAN correspondiente y la adjudicación de dirección IP a la misma.

```
LER0(config)#interface FastEthernet0/0/0
LER0(config-if)#switchport access vlan 2
LER0(config-if)#switchport mode trunk
LER0(config-if)#exit
LER0(config)#interface FastEthernet0/0/1
LER0(config-if)#switchport access vlan 3
LER0(config-if)#switchport mode trunk
LER0(config-if)#exit
LER0(config)#interface Vlan 2
LER0(config-if)#ip address 10.0.0.1 255.255.255.252
LER0(config-if)#exit
LER0(config)#interface Vlan 3
LER0(config-if)#ip address 40.0.0.1 255.255.255.252
LER0(config)#exit
```

Repetir los anteriores pasos hasta configurar todos los interfaces utilizados en los diferentes routers que componen la red.

Una vez asignadas las direcciones a todos los interfaces, es necesario configurar el protocolo de routing, en este caso utilizaremos OSPF, por ser uno de los más extendidos.

Para habilitar OSPF, utilizaremos el comando **router ospf *id_proceso*** en el modo de configuración global (***id_proceso*** puede ser cualquier número entero desde 1 a 65.535).

Una vez en el modo de configuración del router, añadiremos todas las redes IP conectadas al router al proceso de enrutamiento OSPF mediante el comando **network**.

El formato de este comando es: **network *dir_IP máscara_wildcard* area *número_area***. La ***máscara_wildcard*** podemos definirla como la inversa de la máscara de red. El ***número_area*** hace referencia al área OSPF.

Un área OSPF no es más que un grupo de routers que comparten información de estado de enlace. Cuando todos los routers están dentro de la misma área de OSPF debe configurarse entonces los comandos **network** con la misma ***área-id*** en todos los routers. Podemos usar cualquier ***número_area***, pero es recomendable utilizar ***número_area*** 0 con OSPF de área única.

En este caso vamos a realizar el ejemplo de configuración para el LSR1, los comandos necesarios para configurar correctamente el protocolo OSPF en dicho router serían:

```
LSR1(config)#router ospf 1
LSR1(config-router)# network 10.0.0.0 0.0.0.3 area 0
LSR1(config-router)# network 20.0.0.0 0.0.0.3 area 0
LSR1(config-router)# network 30.0.0.0 0.0.0.3 area 0
LSR1(config-router)# network 192.170.1.2 0.0.0.0 area 0
LSR1(config-router)#exit
LSR1(config)#
```

Para comprobar que se ha configurado correctamente el OSPF ponemos utilizar el comando **show ip route**, el resultado debe de ser similar al siguiente:

```
LSR1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, 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
O       50.0.0.0 [110/2] via 30.0.0.2, 00:08:56, FastEthernet0/0
                 [110/2] via 20.0.0.2, 00:16:09, Vlan5
     20.0.0.0/30 is subnetted, 1 subnets
C       20.0.0.0 is directly connected, Vlan5
     40.0.0.0/30 is subnetted, 1 subnets
O       40.0.0.0 [110/2] via 30.0.0.2, 00:08:56, FastEthernet0/0
                 [110/2] via 10.0.0.1, 00:23:16, Vlan2
     10.0.0.0/30 is subnetted, 1 subnets
C       10.0.0.0 is directly connected, Vlan2
     192.170.1.0/32 is subnetted, 4 subnets
O       192.170.1.3 [110/2] via 20.0.0.2, 00:16:09, Vlan5
C       192.170.1.2 is directly connected, Loopback0
O       192.170.1.1 [sh110/2] via 10.0.0.1, 00:27:56, Vlan2
```

```
O      192.170.1.4 [110/2] via 30.0.0.2, 00:09:05, FastEthernet0/0
O    192.168.1.0/24 [110/2] via 10.0.0.1, 00:11:53, Vlan2
O    192.168.2.0/24 [110/2] via 20.0.0.2, 00:03:01, Vlan5
     30.0.0.0/30 is subnetted, 1 subnets
C      30.0.0.0 is directly connected, FastEthernet0/0
```

En la consola del router nos aparecen las direcciones de todas las redes que el router ha aprendido a direccionar, la letra "C" delante indica que la red está directamente conectada al mismo, mientras que una "O" significa que dicha red ha sido aprendida mediante OSPF.

Como podemos ver, en el caso de LSR1, la ruta hacia las redes 40.0.0.0/30, 50.0.0.0/30 se han aprendido mediante OSPF, ya que no están directamente conectadas al mismo. De modo similar ocurre para los dos PC's conectados en la red y los interfaces de loopback.

Una vez configurado el protocolo OSPF en todos los equipos de la red, deberíamos de ser capaces de tener acceso a todos los dispositivos de la red desde cualquier equipo que esté conectado a la misma.

Hacer ping desde diferentes equipos y comprobar que se puede acceder a cualquier interfaz o elemento de la red (en el caso de los PCs, asegurarse previamente que el firewall de Windows se encuentre deshabilitado, de no ser así filtrará los paquetes con origen en un PC y destino el otro).

También es posible averiguar la ruta que seguirá un paquete mediante el uso de comando **traceroute**, cuya sintaxis es la siguiente: **traceroute *ip_destino***.

Llegados a este punto, comentar que el software de los routers CISCO, el IOS, incorpora como funcionalidad estándar el balanceo de carga. Es inherente al proceso de reenvío en el router y se activa automáticamente si la tabla de ruteo tiene varias trayectorias a un destino. Se basa en los protocolos de ruteo estándar, como Routing Information Protocol (RIP), RIPv2, Enhanced Interior Gateway Routing Protocol (EIGRP) u Open Shortest Path First (OSPF).

A partir de este punto comenzaremos propiamente con la configuración de la red MPLS, para ello en el caso de CISCO es necesario activar la función CEF (Cisco Express Forwarding). Aunque normalmente esta viene habilitada por defecto.

Cuando se activa CEF el router construye, a partir de la información de la tabla de enrutamiento IP, otra tabla llamada FIB (Forwading Information Base), que especifica para cada posible red de destino la dirección del siguiente router y por lo tanto la interfaz que se debe utilizar. Así pues, la tabla FIB, es una versión simplificada de la tabla de rutas que acelera el proceso de enrutamiento de los paquetes, evitando las búsquedas recursivas.

Realmente CEF permite asociar una etiqueta, en el caso de IP la etiqueta es una dirección IP, con una interfaz de salida y con información de capa 2 del siguiente salto para el reenvío. De ahí que CISCO utilice esta tabla FIB para la implementación de MPLS cuando la etiqueta que se utiliza es la etiqueta de MPLS.

Para habilitar CEF el comando a utilizar es el siguiente:

```
LSR1(config)#ip cef
LSR1(config)#
```

Es necesario que se habilite CEF en cada uno de los routers pertenecientes a la red MPLS.

Ahora que ya tenemos configurados los routers de forma básica debemos de proceder a habilitar MPLS. Adicionalmente y para que posteriormente sea más fácil identificar la procedencia de los paquetes, limitaremos en cada uno de los routers el rango de etiquetas a asignar según la siguiente tabla:

ROUTER	RANGO ETIQUETAS
LER0	16-99 (min 16)
LSR1	100-199
LSR2	200-299
LER3	300-399

Tabla 3. Asignación de rangos de etiquetas por router

En primer lugar, tendremos que habilitar el procesamiento de MPLS, seguidamente será necesario indicar el protocolo que queremos utilizar para la distribución de etiquetas, en nuestro caso elegiremos LDP, por lo que el comando a utilizar será: **mpls label protocol *ldp***.

Por último, para limitar el rango de etiquetas local en cada router se utiliza el comando **mpls label range** [rango].

Siguiendo con el ejemplo del LSR1, los comandos necesarios para habilitar MPLS en el router son:

```
LSR1#configure terminal
LSR1(config)#mpls ip
LSR1(config)#mpls label protocol ldp
LSR1(config)#mpls label range 100 199
LSR1(config)#
```

Una vez configurado MPLS en el router, es necesario indicar que interfaces participan en la red, para ello se utiliza el comando **mpls ip** (no se utiliza en los interfaces de loopbak). En el caso que estamos utilizando como ejemplo, los comandos a introducir serían:

```
LSR1(config)#interface FastEthernet0/0
LSR1(config-if)#mpls ip
LSR1(config-if)#exit
LSR1(config)#interface FastEthernet0/0/0
LSR1(config-if)#mpls ip
LSR1(config-if)#exit
LSR1(config)#interface FastEthernet0/0/1
LSR1(config-if)#mpls ip
LSR1(config-if)#exit
LSR1(config)#interface vlan 2
LSR1(config-if)#mpls ip
LSR1(config-if)#exit
LSR1(config)#interface vlan 5
LSR1(config-if)#mpls ip
LSR1(config-if)#exit
LSR1(config)#exit
```

Con esto deberíamos de tener el router MPLS plenamente operativo, para comprobar que interfaces están funcionando con MPLS y el protocolo de distribución de etiquetas que utilizan, podemos utilizar el comando **show mpls interfaces**, cuyo resultado debería ser similar a:

```
LSR1#show mpls interfaces
Interface              IP         Tunnel   BGP     Static  Operational
FastEthernet0/0        Yes (ldp)  No       No      No      Yes
FastEthernet0/0/0      Yes (ldp)  No       No      No      Yes
FastEthernet0/0/1      Yes (ldp)  No       No      No      Yes
Vlan2                  Yes (ldp)  No       No      No      Yes
Vlan5                  Yes (ldp)  No       No      No      Yes
```

Repetir la configuración MPLS en el resto de routers de la red, para que ésta quede totalmente operativa.

Podemos comprobar de forma sencilla si la red está bien configurada con el comando **show mpls discovery**. Este comando muestra la información de descubrimiento de vecinos, empleando mensajes LDP Discovery HELLO, al mismo tiempo nos servirá para verificar que la interfaz está activa.

```
LSR2#show mpls ldp discovery
 Local LDP Identifier:
   192.170.1.4:0
   Discovery Sources:
   Interfaces:
      FastEthernet0/0/0 (ldp): xmit
      FastEthernet0/0/1 (ldp): xmit
      Vlan3 (ldp): xmit/recv
         LDP Id: 192.170.1.1:0
      Vlan4 (ldp): xmit/recv
         LDP Id: 192.170.1.3:0
```

El resultado del citado comando correspondiente al router LSR2, es el que vemos sobre estas líneas. Éste descubre a través de la interfaz Fa0/0/0 (VLAN 3) al router LER0, identificado por su loopback 192.170.1.1, el cual tiene activo el protocolo LDP. De forma análoga descubre a LER3 a través del interfaz Fa0/0/1 (VLAN 4).

Al lado de "ldp" podemos obervar el texto "xmit/recv", un resultado distinto a "xmit/recv" entre los routers directamente conectados indicaría un problema entre vecinos.

De igual manera comprobar las adyacencias con algún otro router y observar que los datos tengan coherencia con la red configurada.

Una vez configurada la red, es posible visualizar las diferentes tablas con las que trabajan los routers, por ejemplo, para visualizar la tabla LIB se debe utilizar el comando **show mpls ldp bindings**, al ejecutarlo en cualquiera de los routers de la red se obtendrá un resultado similar al siguiente:

```
 LSR1#show mpls ldp bindings
   tib entry: 10.0.0.0/30, rev 4
        local binding:  tag: imp-null
        remote binding: tsr: 192.170.1.1:0, tag: imp-null
        remote binding: tsr: 192.170.1.3:0, tag: 303
   tib entry: 20.0.0.0/30, rev 6
        local binding:  tag: imp-null
```

```
        remote binding: tsr: 192.170.1.1:0, tag: 17
        remote binding: tsr: 192.170.1.3:0, tag: imp-null
    tib entry: 30.0.0.0/30, rev 22
        local binding:  tag: imp-null
        remote binding: tsr: 192.170.1.3:0, tag: 305
        remote binding: tsr: 192.170.1.1:0, tag: 22
    tib entry: 40.0.0.0/30, rev 10
        local binding:  tag: 101
        remote binding: tsr: 192.170.1.1:0, tag: imp-null
        remote binding: tsr: 192.170.1.3:0, tag: 302
    tib entry: 50.0.0.0/30, rev 14
        local binding:  tag: 103
        remote binding: tsr: 192.170.1.3:0, tag: imp-null
        remote binding: tsr: 192.170.1.1:0, tag: 19
    tib entry: 192.168.1.0/24, rev 24
        local binding:  tag: 106
        remote binding: tsr: 192.170.1.1:0, tag: imp-null
        remote binding: tsr: 192.170.1.3:0, tag: 306
    tib entry: 192.168.2.0/24, rev 18
        local binding:  tag: 105
        remote binding: tsr: 192.170.1.3:0, tag: imp-null
        remote binding: tsr: 192.170.1.1:0, tag: 21
    tib entry: 192.170.1.1/32, rev 8
        local binding:  tag: 100
        remote binding: tsr: 192.170.1.1:0, tag: imp-null
        remote binding: tsr: 192.170.1.3:0, tag: 301
    tib entry: 192.170.1.2/32, rev 2
        local binding:  tag: imp-null
        remote binding: tsr: 192.170.1.1:0, tag: 16
        remote binding: tsr: 192.170.1.3:0, tag: 300
    tib entry: 192.170.1.3/32, rev 12
        local binding:  tag: 102
        remote binding: tsr: 192.170.1.3:0, tag: imp-null
        remote binding: tsr: 192.170.1.1:0, tag: 18
    tib entry: 192.170.1.4/32, rev 16
        local binding:  tag: 104
        remote binding: tsr: 192.170.1.1:0, tag: 20
        remote binding: tsr: 192.170.1.3:0, tag: 304
```

Una vez construida la tabla LIB y distribuidas las etiquetas a través del protocolo LDP, se construye la tabla LFIB, que se utilizará para realizar la conmutación de paquetes a través de la red. La tabla LFIB almacena la etiqueta asignada por el LSR vecino, "*Outgoing tag"*, la interfaz por donde enviar la trama MPLS, *"Outgoing interface"* y el siguiente salto, "*Next hop"*.

El comando para mostrar la tabla LFIB es el siguiente, **show mpls forwarding-table**, la salida de este, una vez ejecutado en el LSR2 y LER3, debe de ser similar a:

```
LSR2#show mpls forwarding-table
Local      Outgoing    Prefix           Bytes Label   Outgoing    Next Hop
Label      Label       or Tunnel Id     switched      interface
200        No Label    192.170.1.2/32   0             Fa0/0       30.0.0.1
201        Pop Label   192.170.1.1/32   0             Vl3         40.0.0.1
202        Pop Label   192.168.1.0/24   0             Vl3         40.0.0.1
203        Pop Label   20.0.0.0/30      0             Vl4         50.0.0.2
           No Label    20.0.0.0/30      0             Fa0/0       30.0.0.1
204        Pop Label   10.0.0.0/30      0             Vl3         40.0.0.1
           No Label    10.0.0.0/30      0             Fa0/0       30.0.0.1
205        Pop Label   192.170.1.3/32   0             Vl4         50.0.0.2
206        Pop Label   192.168.2.0/24   0             Vl4         50.0.0.2
```

```
LER3#show mpls forwarding-table
Local      Outgoing    Prefix           Bytes tag     Outgoing    Next Hop
Label      Label       or Tunnel Id     switched      interface
300        Pop tag     192.170.1.2/32   0             Vl5         20.0.0.1
301        202         192.170.1.1/32   0             Vl4         50.0.0.1
           100         192.170.1.1/32   0             Vl5         20.0.0.1
302        Pop Label   40.0.0.0/30      0             Vl4         50.0.0.1
303        Pop Label   10.0.0.0/30      0             Vl5         20.0.0.1
304        Pop Label   192.170.1.4/32   0             Vl4         50.0.0.1
305        Pop Label   30.0.0.0/30      0             Vl4         50.0.0.1
           Pop Label   30.0.0.0/30      0             Vl5         20.0.0.1
306        206         192.168.1.0      0             Vl4         50.0.0.1
           106         192.168.1.0      0             Vl5         20.0.0.1
```

Una vez finalizada la configuración de la práctica vamos a grabar cada uno de los archivos de configuración que están ejecutando los routers, para ello seguir las siguientes instrucciones.

Para copiar el archivo "running-config del router en cuestión a la compact flash de este, ejecutar:

```
LER0#copy running-config flash:startup-config
LER0#
```

Si lo que hubiéramos querido es copiar el archivo de configuración desde la compact flash al router, tendríamos que haber ejecutado:

```
LER0#copy flash:startup-config startup-config
LER0#
```

En la próxima práctica podríamos cargar este fichero en cada uno de los routers para no tener que volver a introducir de nuevo todos los comandos de configuración.

2.7 Ejercicios propuestos

1. **Utilizando el comando traceroute, determinar que ruta seguirán los paquetes desde el PC0 hasta el PC1. ¿Es esta ruta la qué siempre seguirán los paquetes?**

```
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 206 Exp 0] 0 msec
     10.0.0.2 [MPLS: Label 106 Exp 0] 4 msec
  2  20.0.0.2 44 msec
     50.0.0.2 48 msec
  3  192.168.2.100 4 msec 0 msec 4 msec
LER0#
```

Utilizando el comando traceroute podemos determinar la ruta que seguirán los paquetes desde el PC0 al PC1, el primer salto podrá ser LSR2 a través de interfaz fa0/0/1, usando la etiqueta 206. O bien se puede ir por LSR1 a través del fa0/0/0, etiquetándose el paquete con la etiqueta 106.

Desde LSR2, el siguiente salto será 50.0.0.2, eliminándose la etiqueta por ser LSR2 el PHP. Si el paquete llega desde LSR1, el siguiente salto será 20.0.0.2.

Como se puede leer en el guion de la práctica el camino que seguirán los paquetes podría variar debido al balanceo de carga, por este motivo se pueden apreciar las dos opciones encontradas por el protocolo IGP al ejecutar el traceroute.

2. **¿Por qué en alguna ocasión aparecen entradas duplicadas para el mismo destino?**

Cuando el router ha detectado varias rutas a un destino específico a través del protocolo de routing, selecciona la ruta con la mínima distancia administrativa, en este caso todos los caminos tienen el mismo coste, por lo que existe más de una ruta por la que el paquete puede ser enviado. Los routers Cisco tienen habilitado el balanceo de carga de forma predeterminada, por lo que usará ambas rutas para repartir la carga en la red.

3. **Ejecuta en alguno de los router de la red el comando "show mpls ldp bindings", ¿por qué en algún destino aparece la palabra "imp-null"?. ¿A qué es debido que no aparezca ninguna etiqueta asociada?**

Cuando la etiqueta o tag es *imp*-null, indica que el prefijo del paquete será reenviado con prefijo de red IP y no con la etiqueta MPLS, según el modo de funcionamiento PHP (Penultimate Hop Popping), o bien, por tener el router la red directamente conectada. De esta forma se evita una consulta innecesaria en la tabla LFIB en el LSR destino, cuando ya se conoce que el destino está conectado directamente a dicho LSR.

4. **A partir de la topología de red y de las tablas "forwarding table" y LIB del router LER0, construir las tablas RIB, FIB, LIB y LFIB de forma similar a los ejercicios realizados en la teoría de la asignatura.**

No tendremos en cuenta el etiquetado a las direcciones de loopback.

```
LER0#show mpls ldp bindings
    tib entry: 10.0.0.0/30, rev 4
        local binding:  tag: imp-null
        remote binding: tsr: 192.170.1.2:0, tag: imp-null
        remote binding: tsr: 192.170.1.4:0, tag: 203
    tib entry: 20.0.0.0/30, rev 14
        local binding:  tag: 19
        remote binding: tsr: 192.170.1.2:0, tag: imp-null
        remote binding: tsr: 192.170.1.4:0, tag: 204
    tib entry: 30.0.0.0/30, rev 22
        local binding:  tag: 22
        remote binding: tsr: 192.170.1.4:0, tag: imp-null
        remote binding: tsr: 192.170.1.2:0, tag: imp-null
    tib entry: 40.0.0.0/30, rev 6
        local binding:  tag: imp-null
        remote binding: tsr: 192.170.1.2:0, tag: 103
        remote binding: tsr: 192.170.1.4:0, tag: imp-null
    tib entry: 50.0.0.0/30, rev 10
        local binding:  tag: 17
        remote binding: tsr: 192.170.1.2:0, tag: 102
        remote binding: tsr: 192.170.1.4:0, tag: imp-null
    tib entry: 192.168.1.0/24, rev 18
        local binding:  tag: imp-null
        remote binding: tsr: 192.170.1.4:0, tag: 205
        remote binding: tsr: 192.170.1.2:0, tag: 105
    tib entry: 192.168.2.0/24, rev 20
        local binding:  tag: 21
        remote binding: tsr: 192.170.1.4:0, tag: 206
        remote binding: tsr: 192.170.1.2:0, tag: 106
    tib entry: 192.170.1.1/32, rev 2
        local binding:  tag: imp-null
        remote binding: tsr: 192.170.1.2:0, tag: 101
        remote binding: tsr: 192.170.1.4:0, tag: 202
    tib entry: 192.170.1.2/32, rev 12
        local binding:  tag: 18
        remote binding: tsr: 192.170.1.2:0, tag: imp-null
        remote binding: tsr: 192.170.1.4:0, tag: 201
    tib entry: 192.170.1.3/32, rev 16
        local binding:  tag: 20
        remote binding: tsr: 192.170.1.2:0, tag: 104
        remote binding: tsr: 192.170.1.4:0, tag: 200
    tib entry: 192.170.1.4/32, rev 8
        local binding:  tag: 16
        remote binding: tsr: 192.170.1.2:0, tag: 100
        remote binding: tsr: 192.170.1.4:0, tag: imp-null
```

LER0#show mpls forwarding-table

Local Label	Outgoing Label	Prefix or Tunnel Id	Bytes Label switched	Outgoing interface	Next Hop
16	Pop Label	192.170.1.4/32	0	Vl3	40.0.0.2
17	Pop Label	50.0.0.0/30	0	Vl3	40.0.0.2
18	Pop Label	192.170.1.2/32	174	Vl2	10.0.0.2
19	Pop Label	20.0.0.0/30	0	Vl2	10.0.0.2
20	200	192.170.1.3/32	0	Vl3	40.0.0.2
	104	192.170.1.3/32	0	Vl2	10.0.0.2
21	206	192.168.2.0/24	0	Vl3	40.0.0.2
	106	192.168.2.0/24	0	Vl2	10.0.0.2
22	Pop Label	30.0.0.0/30	0	Vl3	40.0.0.2
	Pop Label	30.0.0.0/30	0	Vl2	10.0.0.2

LER0

RIB

RED	SALTO
20.0.0.0/30	LSR1
30.0.0.0/30	LSR1
30.0.0.0/30	LSR2
50.0.0.0/30	LSR2
192.168.2.0/24	LSR1
192.168.2.0/24	LSR2

Tabla 4. Tabla RIB

FIB

RED	SALTO	LABEL
20.0.0.0/30	LSR1	-
30.0.0.0/30	LSR1	-
30.0.0.0/30	LSR2	-
50.0.0.0/30	LSR1	-
192.168.2.0/24	LSR1	106
192.168.2.0/24	LSR2	202

Tabla 5. Tabla FIB

LFIB

LABEL	ACTION	SALTO
17	POP	LSR2
19	POP	LSR1
20	200	LSR2
20	104	LSR1
21	206	LSR2
21	202	LSR2
22	POP	LSR2
22	POP	LSR1

Tabla 6. Tabla LFIB

En la última parte de la práctica pasaremos a utilizar el analizador de redes, para ello previamente nos valdremos de una de las funcionalidades de la tarjeta HWIC-4ESW, el "*port monitoring*" o SPAN. Mediante esta función podremos enviar una copia del tráfico de unos de los puertos de la tarjeta hacia otro y así poder conectar en este último un PC con "*Wireshark*" instalado y monitorizar el citado tráfico.

Nos vamos a centrar en el LSR1, capturando el tráfico de su interfaz Fa0/0/0 y reenviándolo hacia el interfaz Fa0/0/2, el cual tenemos disponible. Para ello previamente tendremos que configurar la sesión de SPAN, los comandos a utilizar son:

```
LSR1#configure terminal
LSR1(config)#monitor session 1 source interface FastEthernet0/0/0
LSR1(config)#monitor session 1 destination interface FastEthernet0/0/2
LSR1(config)#end
LSR1#
```

Ahora podemos conectar el "*Wireshark*" al puerto Fa0/0/2 y capturar el mismo tráfico que circula por el puerto Fa0/0/0.

5. **Ejecutar un ping desde PC0 a PC1. ¿Cómo aparecen encapsulados estos paquetes?. ¿Puedes reconocer los campos MPLS vistos en teoría?. (Si no se ve ningún paquete ICMP proveniente de 192.168.1.100, es debido a que el protocolo de IGP ha seleccionado la ruta a través de LSR2, habilitar el SPAN en el mismo y comprobarlo).**

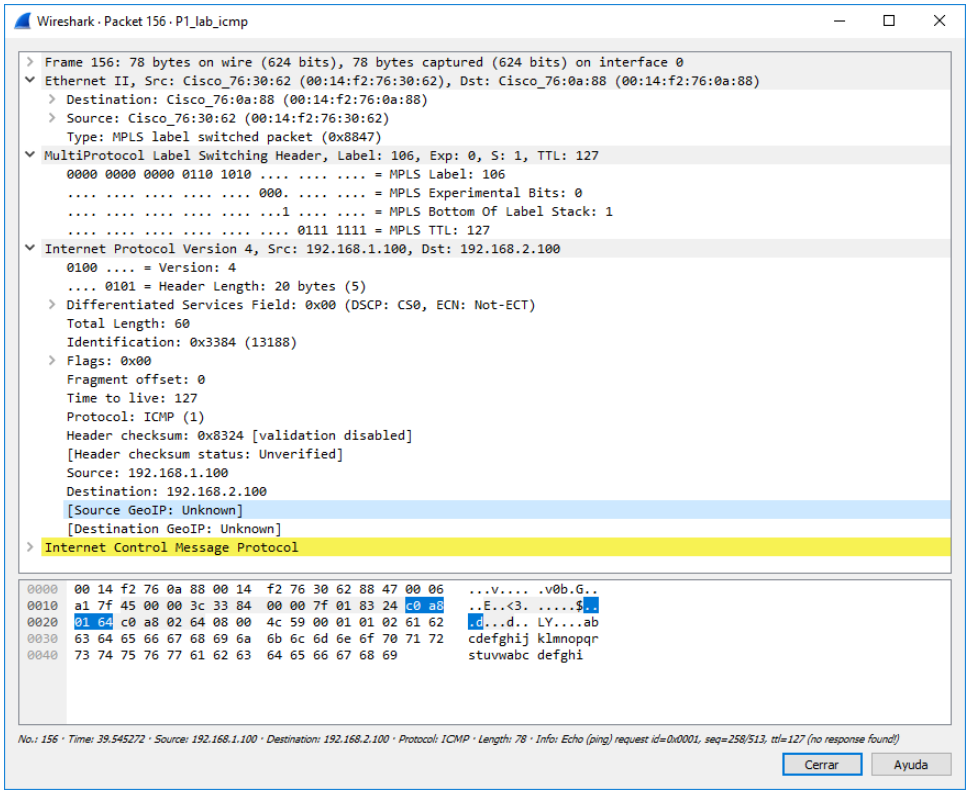

Figura 13. Captura de paquete MPLS

El comando ping se encapsula en ICMP, que a su vez viene dentro de un paquete IP. En nuestro caso, al ser una red MPLS, al paquete IP se le añade la cabecera MPLS.

Como se puede apreciar en la cabecera Ethernet, el EtherType es 0x8847 que como vimos en teoría se corresponde con: Ethernet+MPLS Unicast IP.

En la cabecera MPLS podemos observar los 4 campos que componen la misma: Label, Exp, S y TTL:

- Label: es el valor de la etiqueta, en este caso 106.
- Exp: llamados bits experimentales, se utilizan para identificar la clase de servicio. El valor es 0, no se están utilizando.
- S, cuando S=0 indica que hay etiquetas apiladas. No estamos trabajando con túneles ni nada similar, por lo que S=1.

Cuando hemos configurado la red hemos seleccionado LDP como protocolo de distribución de etiquetas, vamos a ver cómo trabaja para distribuir las mismas. Para ello utlizaremos el comando **clear mpls ldp neighbor *** en el LSR1, lo que eliminará todas las sesiones ldp establecidas con sus vecinos. Con el "*Wireshark*" conectado al puerto Fa0/0/2 del LSR1 captura el tráfico que se produce tras introducir el citado comando mientras que el LSR1 reestablece las sesiones LDP con sus vecinos.

6. **Identifica los paquetes de negociación de la sesión LDP y describe como se establece ésta.**

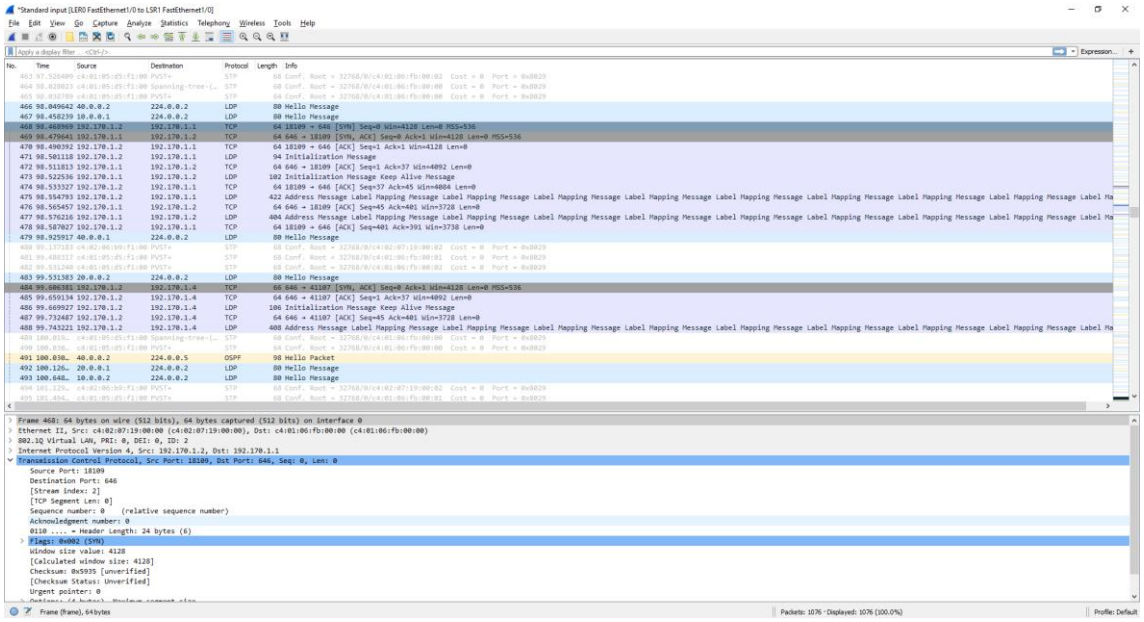

Figura 14. Establecimiento de la sesión LDP

En primer lugar, ver figura 14, y tras los mensajes "*Hello*", tal y como hemos visto en la teoría, se establece la conexión TCP, entre el router con la dirección IP más alta, en este caso LSR1 y el LER0, mientras que LER3 y LSR2 la establecen con el LSR1 por tener este una dirección IP más baja (lo0).

La conexión con el LER0 se establece a través del puerto 646, como ya sabíamos de teoría. Podemos identificar los 3 mensajes del handshake entre ambos routers: TCP-Syn, TCP-Syn/ACK y TCP-ACK.

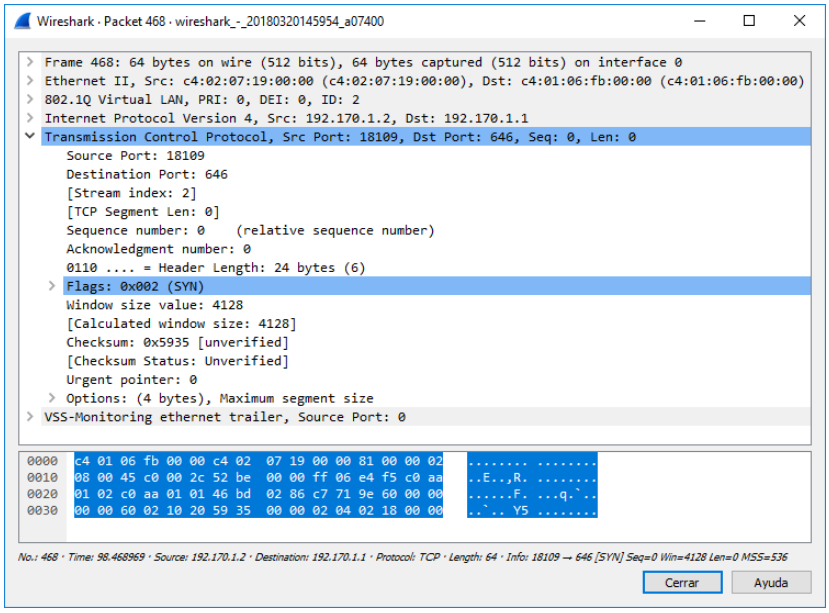

Figura 15. Mensaje TCP-Syn

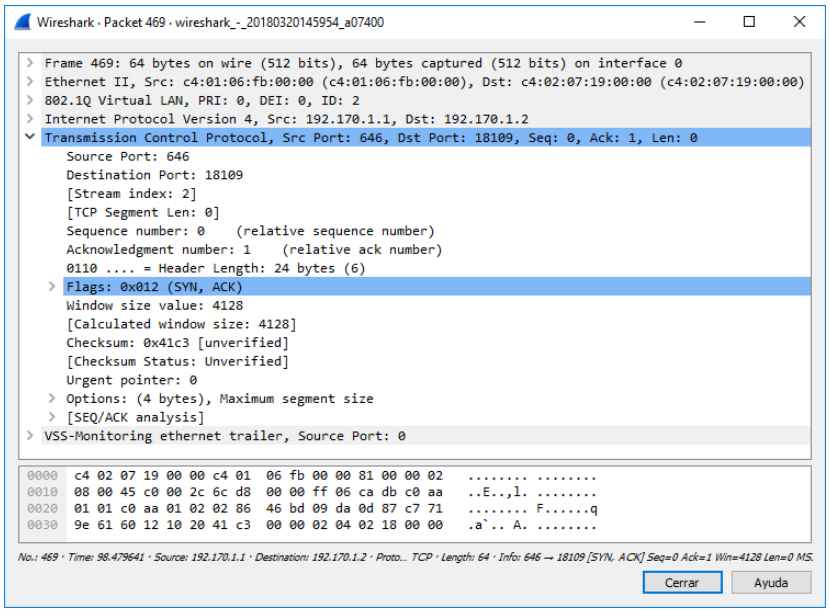

Figura 16. Mensaje TCP-Syn/ACK

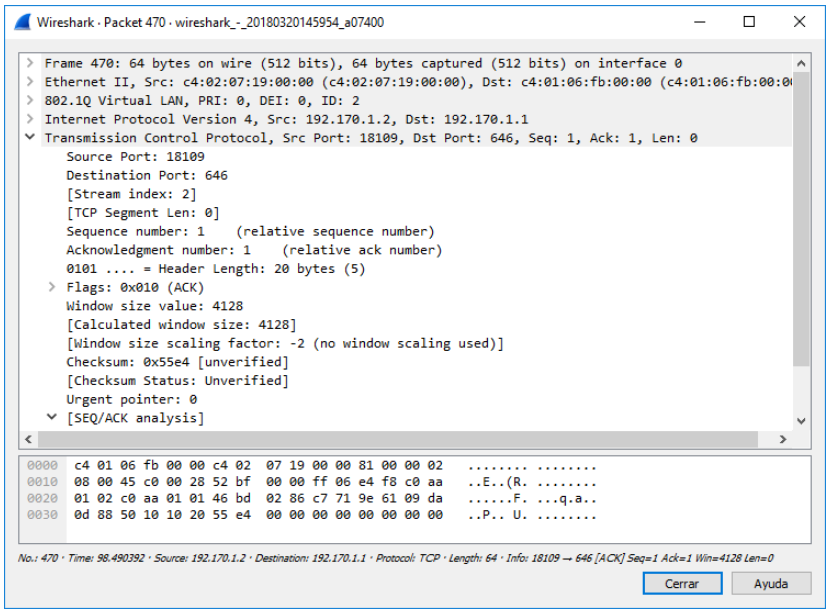

Figura 17. Mensaje TCP-ACK

Una vez establecida la conexión TCP entre ambos, LSR1 envía al LER0 el mensaje LDP de inicialización, para solicitar el establecimiento de la sesión LDP.

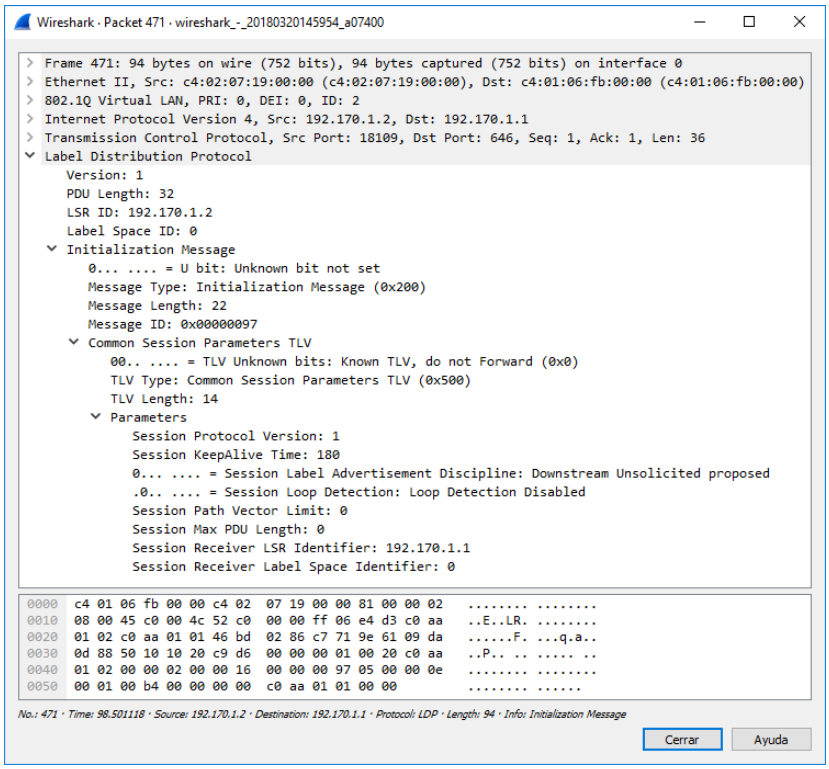

Figura 18. LDP Inizialization

Como podemos ver en la figura 18, la distribución de etiquetas es no solicitada.

Tras recibir el TCP-ACK y el mensaje "*Keep Alive*" comienza el proceso de distribución de etiquetas con los mensajes "*Address*" y "*Label Mapping*". En el mensaje "*Address*" cada router publicita las direcciones de sus interfaces, mientras que en el "*Label Mapping*" el router publicita el mapeado de un FEC a una etiqueta a sus vecinos. En este caso como podemos ver en la figura 19, ambos mensajes se encuentran en el mismo paquete.

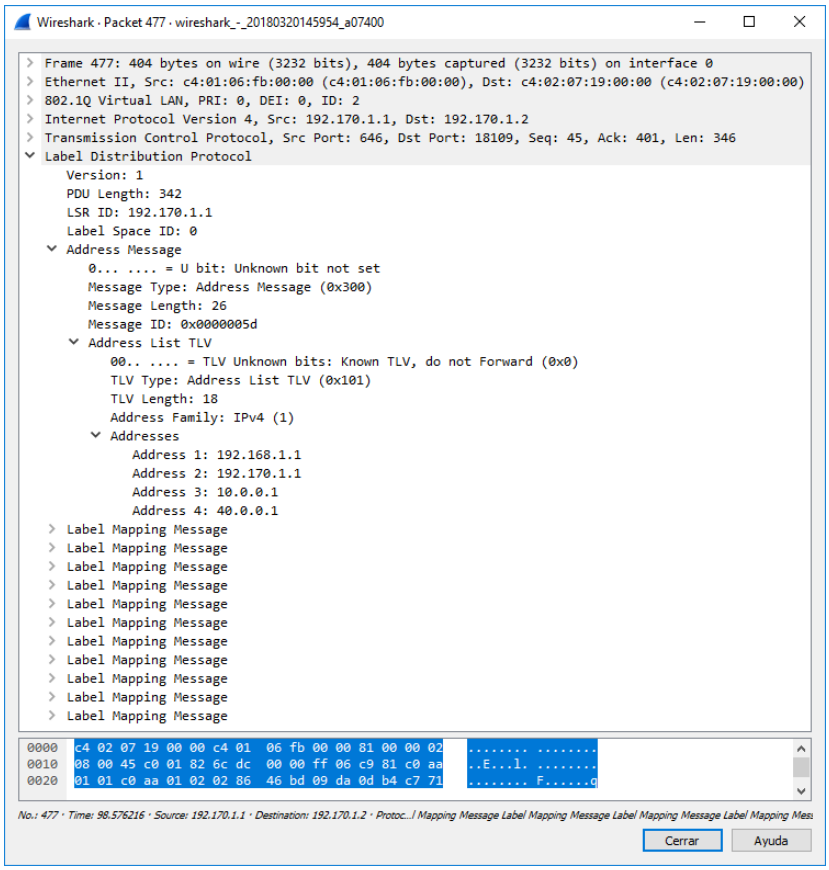

Figura 19. Mensaje "Address" y "Label Mapping" de LER0 a LSR1

En la figura 19, podemos observar que el LER0 le comunica al LSR1 la direcciones las 4 direcciones de sus interfaces, anteriormente LSR1 ya había realizado el mismo proceso.

Si desplegamos los diferentes mensajes "*Label Mapping*", podremos ver la asignación de etiquetas que hace LER0 a los diferentes FEC.

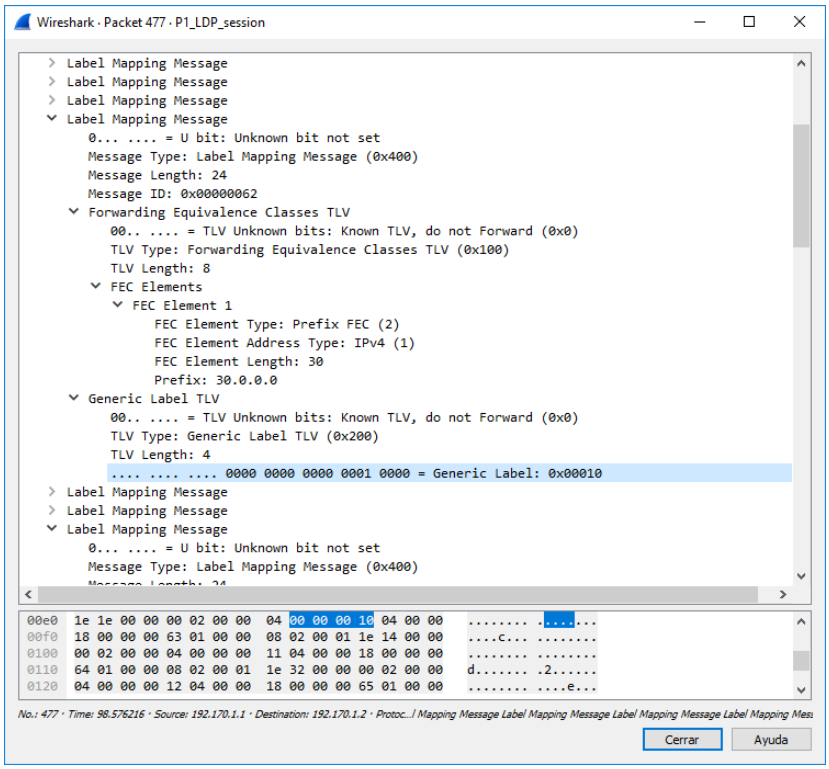

Figura 20. "*Label Mapping*" LER0

Como vemos en la figura 20, el LER0 asigna la etiqueta 16 (10#HEX) para alcanzar la red 30.0.0.0/30, si hacemos un **show mpls forwarding-table** comprobamos que efectivamente existe esa correspondencia, exactamente igual se realiza la asignación para el resto de destinos.

LER0#show mpls forwarding-table

Local Label	Outgoing Label	Prefix or Tunnel Id	Bytes Label switched	Outgoing interface	Next Hop
16	Pop Label	30.0.0.0/30	0	Vl3	40.0.0.2
	Pop Label	30.0.0.0/30	0	Vl2	10.0.0.2
17	Pop Label	20.0.0.0/30	0	Vl2	10.0.0.2
18	Pop Label	50.0.0.0/30	0	Vl3	40.0.0.2
19	Pop Label	192.170.1.2/32	1517	Vl2	10.0.0.2
20	Pop Label	192.170.1.4/32	2763	Vl3	40.0.0.2
21	205	192.168.2.0/24	0	Vl3	40.0.0.2
	105	192.168.2.0/24	0	Vl2	10.0.0.2
22	206	192.170.1.3/32	860	Vl3	40.0.0.2
	106	192.170.1.3/32	0	Vl2	10.0.0.2

En la tabla LFIB de LER0 observamos que LSR1 le ha pedido al LER0 que etiquete el destino 192.168.2.0/24 con 105, en la figura 21 podemos observar que efectivamente en el "*Label Mapping*" la etiqueta a utilizar coincide (69#HEX).

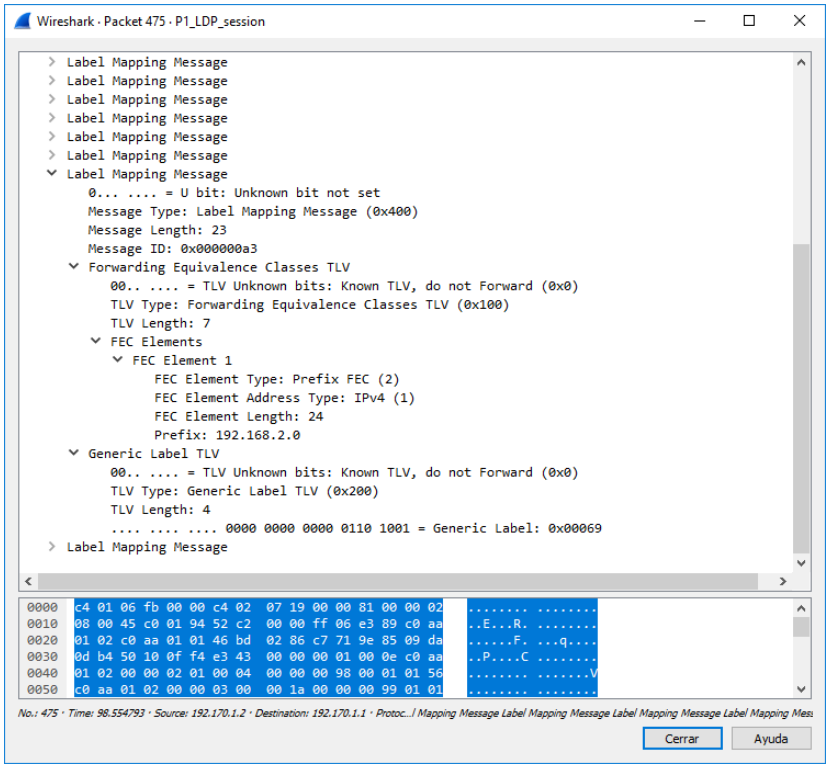

Figura 21. "*Label Mapping*" LSR1

En el resto de pares LDP se producirá un intercambio similar de mensajes LDP para realizar la asignación de etiquetas a los distintos FEC's.

2.8 ANEXO 0 "Actualización de versión de IOS desde la compact flash"

Antes de comenzar con el proceso de actualización, será necesario copiar en la compact flash que introduciremos en el router el fichero bin con la versión de IOS que queremos utilizar, para ello se pueden utilizar los lectores de tarjetas que incorporan los PC's del laboratorio.

Es primer lugar, tendremos que acceder al modo privilegiado mediante el comando **enable**, es recomendable borrar el fichero de inicio con el comando **erase startup-config**.

El fichero de IOS que utilizaremos en la presente práctica será c1841-advipservicesk9-mz.150-1.M10.bin, con lo que la secuencia de comandos a introducir en el router para arrancar con dicha ROM será la siguiente:

```
Router#configure terminal
Router(config)#no boot system
Router(config)#boot system flash: c1841-advipservicesk9-mz.150-1.M10.bin
Router(config)#exit
Router#reload
```

En primer lugar, con el comando **no boot system**, indicamos al router que elimine la imagen con la que arranca el sistema. Seguidamente con **boot system** especificamos al router la ruta de la imagen con la que debe de iniciar. Una vez indicada, es necesario realizar un reinicio del router mediante el comando **reload**. Introducir "*yes*" cuando pregunte si se desean guardar los cambios realizados en el sistema.

2.9 ANEXO 1 "Ficheros running-config de los routers"

LER0

```
!
version 15.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname LER0
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
!
!
dot11 syslog
ip source-route
!
!
!
!
ip cef
no ip domain lookup
no ipv6 cef
!
```

```
multilink bundle-name authenticated
!
mpls label range 16 99
mpls label protocol ldp
!
!
!
!
license udi pid CISCO1841 sn FCZ0935105S
!
redundancy
!
!
!
!
!
!
!
!
!
interface Loopback0
 ip address 192.170.1.1 255.255.255.255
 !
!
interface FastEthernet0/0
 ip address 192.168.1.1 255.255.255.0
 duplex auto
 speed auto
 !
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
 !
!
interface FastEthernet0/0/0
 switchport access vlan 2
 switchport mode trunk
 mpls ip
 !
!
interface FastEthernet0/0/1
 switchport access vlan 3
 switchport mode trunk
 mpls ip
 !
!
interface FastEthernet0/0/2
 !
!
interface FastEthernet0/0/3
 !
!
```

```
interface Vlan1
 no ip address
 shutdown
 !
!
interface Vlan2
 ip address 10.0.0.1 255.255.255.252
 mpls ip
 !
!
interface Vlan3
 ip address 40.0.0.1 255.255.255.252
 mpls ip
 !
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.0.0.3 area 0
 network 40.0.0.0 0.0.0.3 area 0
 network 192.168.1.0 0.0.0.255 area 0
 network 192.170.1.1 0.0.0.0 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
!
!
!
!
!
!
!
!
control-plane
 !
!
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
end
```

LSR1

```
!
version 15.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname LSR1
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
monitor session 1 source interface Fa0/0/0
monitor session 1 destination interface Fa0/0/2
!
!
!
dot11 syslog
ip source-route
!
!
!
!
ip cef
no ip domain lookup
no ipv6 cef
!
multilink bundle-name authenticated
!
mpls label range 100 199
mpls label protocol ldp
!
!
!
!
license udi pid CISCO1841 sn FCZ0935106L
!
redundancy
!
!
!
!
!
!
!
!
!
interface Loopback0
 ip address 192.170.1.2 255.255.255.255
 !
!
```

```
interface FastEthernet0/0
 ip address 30.0.0.1 255.255.255.252
 duplex auto
 speed auto
 mpls ip
 !
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
 !
!
interface FastEthernet0/0/0
 switchport access vlan 2
 switchport mode trunk
 mpls ip
 !
!
interface FastEthernet0/0/1
 switchport access vlan 5
 switchport mode trunk
 mpls ip
 !
!
interface FastEthernet0/0/2
 mpls ip
 !
!
interface FastEthernet0/0/3
 !
!
interface Vlan1
 no ip address
 shutdown
 !
!
interface Vlan2
 ip address 10.0.0.2 255.255.255.252
 mpls ip
 !
!
interface Vlan5
 ip address 20.0.0.1 255.255.255.252
 mpls ip
 !
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.0.0.3 area 0
 network 20.0.0.0 0.0.0.3 area 0
 network 30.0.0.0 0.0.0.3 area 0
 network 192.170.1.2 0.0.0.0 area 0
!
```

```
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
!
!
!
!
!
!
!
control-plane
 !
!
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
end
```

LSR2

```
!
version 15.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname LSR2
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
!
!
dot11 syslog
ip source-route
!
!
!
!
ip cef
no ip domain lookup
no ipv6 cef
!
multilink bundle-name authenticated
!
mpls label range 200 299
mpls label protocol ldp
!
!
!
!
license udi pid CISCO1841 sn FCZ1040113S
!
redundancy
!
!
!
!
!
!
!
!
!
interface Loopback0
 ip address 192.170.1.4 255.255.255.255
 !
!
interface FastEthernet0/0
 ip address 30.0.0.2 255.255.255.252
 duplex auto
```

```
 speed auto
 !
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
 !
!
interface FastEthernet0/0/0
 switchport access vlan 3
 switchport mode trunk
 mpls ip
 !
!
interface FastEthernet0/0/1
 switchport access vlan 4
 switchport mode trunk
 mpls ip
 !
!
interface FastEthernet0/0/2
 !
!
interface FastEthernet0/0/3
 !
!
interface Vlan1
 no ip address
 shutdown
 !
!
interface Vlan2
 no ip address
 !
!
interface Vlan3
 ip address 40.0.0.2 255.255.255.252
 mpls ip
 !
!
interface Vlan4
 ip address 50.0.0.1 255.255.255.252
 mpls ip
 !
!
router ospf 1
 log-adjacency-changes
 network 30.0.0.0 0.0.0.3 area 0
 network 40.0.0.0 0.0.0.3 area 0
 network 50.0.0.0 0.0.0.3 area 0
 network 192.170.1.4 0.0.0.0 area 0
!
ip forward-protocol nd
```

```
no ip http server
no ip http secure-server
!
!
!
!
!
!
!
!
!
control-plane
 !
!
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
end
```

LER3

```
!
version 15.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname LER3
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
!
!
dot11 syslog
ip source-route
!
!
!
!
ip cef
no ip domain lookup
no ipv6 cef
!
multilink bundle-name authenticated
!
mpls label range 300 399
mpls label protocol ldp
!
!
!
!
license udi pid CISCO1841 sn FHK104118Q6
!
redundancy
!
!
!
!
!
!
!
!
!
interface Loopback0
 ip address 192.170.1.3 255.255.255.255
 !
!
interface FastEthernet0/0
 ip address 192.168.2.1 255.255.255.0
 duplex auto
```

```
 speed auto
 !
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
 !
!
interface FastEthernet0/0/0
 switchport access vlan 4
 switchport mode trunk
 mpls ip
 !
!
interface FastEthernet0/0/1
 switchport access vlan 5
 switchport mode trunk
 mpls ip
 !
!
interface FastEthernet0/0/2
 !
!
interface FastEthernet0/0/3
 !
!
interface Vlan1
 no ip address
 shutdown
 !
!
interface Vlan4
 ip address 50.0.0.2 255.255.255.252
 mpls ip
 !
!
interface Vlan5
 ip address 20.0.0.2 255.255.255.252
 mpls ip
 !
!
router ospf 1
 log-adjacency-changes
 network 20.0.0.0 0.0.0.3 area 0
 network 50.0.0.0 0.0.0.3 area 0
 network 192.168.2.0 0.0.0.255 area 0
 network 192.170.1.3 0.0.0.0 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
```

```
!
!
!
!
!
!
!
control-plane
 !
!
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
end
```

Capítulo 3. Práctica 2 “Introducción a la ingeniería de tráfico en MPLS”

3.1 Introducción

La **Ingeniería de Tráfico** (TE) es una disciplina que procura la optimización del rendimiento de las redes operativas mediante la aplicación de diferentes funciones en la medición, caracterización, modelado y control del tráfico que circula por las mismas. Las mejoras del rendimiento mediante un eficiente manejo del tráfico y modo de utilización de recursos son los principales objetivos de esta.

Una ventaja práctica de la aplicación sistemática de los conceptos de la TE a las redes operacionales, es que ayuda a identificar y estructurar las metas y prioridades en términos de mejora de la calidad de servicio dado a los usuarios finales de los servicios de la red.

La TE se subdivide en dos ramas principalmente diferenciadas por sus objetivos:

- **orientada a tráfico**: su prioridad es la mejora de los indicadores relativos al transporte de datos, como por ejemplo: minimizar la pérdida de paquetes, minimizar el retardo, maximizar el rendimiento, obtener distintos niveles de acuerdo para brindar calidad de servicio, etc.
- **orientada a recursos**: esta rama se plantea como objetivo, la optimización de la utilización de los recursos de la red, de manera que, no se saturen partes de esta, mientras otras permanecen infrautilizadas, tomando principalmente el ancho de banda como recurso a optimizar.

Ambas ramas convergen en un objetivo global, que es minimizar la congestión. Un reto fundamental en la operación de una red, especialmente en redes IP públicas a gran escala, es incrementar la eficiencia de la utilización de recursos mientras se minimiza la posibilidad de congestión.

En casos de congestión de algunos enlaces, el problema se resolvía a base de añadir más capacidad a los enlaces. Mediante el uso de la TE el objetivo, es adaptar los flujos de tráfico a los recursos físicos de la red, equilibrando de forma óptima la utilización de esos recursos, de manera que no haya algunos que estén sobre utilizados, creando cuellos de botella, mientras otros puedan estar infrautilizados.

En resumen, TE provee de capacidades para alcanzar los siguientes objetivos:

- lograr un uso más eficiente del ancho de banda disponible, asegurando que ciertos recursos de la red no se encuentren sobre utilizados, mientras otros recursos son infrautilizados a lo largo de potenciales caminos alternativos.
- maximizar la eficiencia operacional.
- mejorar las características de la red orientadas al tráfico, minimizando la pérdida de paquetes, minimizando períodos prolongados de congestión y maximizando el rendimiento.
- mejorar las características estadísticas de la red (como pueden ser la tasa de pérdidas, variación del retardo y retardo de transferencia).
- proveer de un control preciso sobre cómo el tráfico es re-enrutado cuando el camino primario pierde la conectividad.

3.2 TE en MPLS

En el enrutamiento IP convencional, cada router toma decisiones de enrutamiento independientes basándose únicamente en la dirección IP destino que se encuentra en el encabezado de los paquetes IP. El principal problema con este tipo de enrutamiento, es que no considera los requerimientos de capacidad y tráfico que requieren los flujos de datos. El resultado es que algunos segmentos de la red pueden llegar a congestionarse, mientras existen rutas alternativas que son infrautilizadas. Incluso en situaciones de congestionamiento de la red, los protocolos de enrutamiento tradicionales continúan reenviando tráfico por el camino original o “ruta más corta”

hasta que se produce pérdida de paquetes, retardos y jitter que afectan especialmente a las aplicaciones sensibles al retardo como puede ser la voz sobre IP.

Para poder enrutar flujos de datos de aplicaciones interactivas que requieren bajo retardo y perdidas, es necesario utilizar los recursos de la red de forma más eficiente y el proceso mediante el cual se logra este objetivo se denomina como hemos mencionado antes, TE.

A través de los atributos de los protocolos IGP (Interior Gateway Protocol) se puede lograr establecer otros caminos y mitigar en cierta medida este problema. Sin embargo, para entornos de redes grandes, esta solución es difícil de implementar al tener que modificar en cada router de la red los atributos del protocolo IGP.

Un enfoque popular para eludir las insuficiencias de enrutamiento de los protocoles IGP es el uso de un modelo **overlay**, como por ejemplo es IP sobre ATM, ver figura 22.

El modelo overlay amplía las opciones de diseño, permitiendo implementar una topología de red virtual sobre una topología de red física. Dicha topología virtual se construye a partir de circuitos virtuales que son considerados como enlaces físicos para el IGP, lo que permite aprovechar de una forma más eficiente todos los recursos de la red.

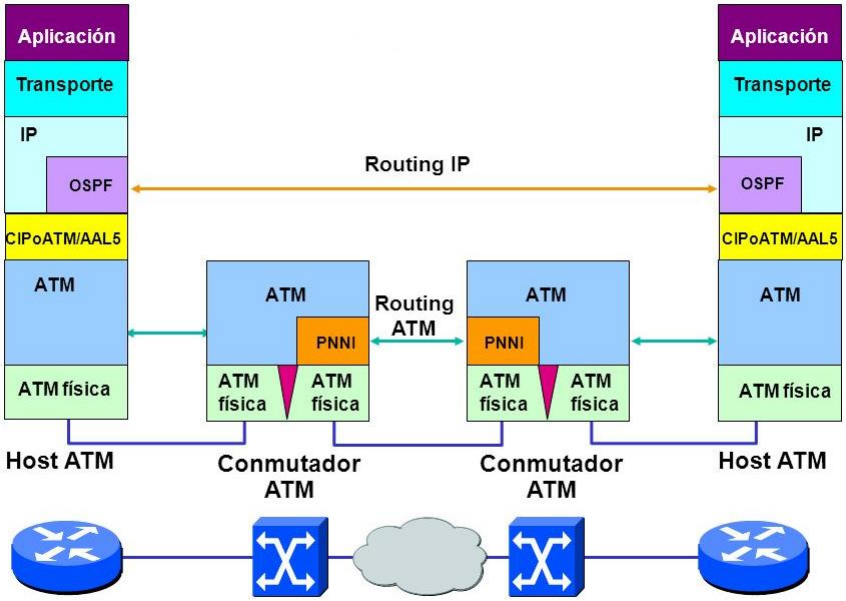

Figura 22. Modelo overlay IP sobre ATM

En MPLS la TE tiene una gran importancia, ya que puede proporcionar la mayoría de las funcionalidades disponibles en el modelo overlay de una manera integrada y sin ningún costo adicional. Algunas de las ventajas que ofrece MPLS en comparación con el modelo overlay incluyen:

- menos elementos de red.
- menos costes de operación.
- mayor fiabilidad ya que existen menos elementos de red en una determinada ruta.
- potencialmente menos latencia.
- arquitecturas de red simplificadas.

La RFC 2702 de la IETF describe un conjunto de capacidades que permiten a MPLS convertirse en un medio efectivo para implementar varias políticas de TE en redes IP, como son:

- MPLS permite crear fácilmente LSP's sin el paradigma del routing basado en la dirección IP destino, a través de acciones administrativas manuales o mediante la acción automática de los protocolos subyacentes.
- los LSP's pueden ser mantenidos de manera eficiente.

- se pueden crear "*traffic trunks*" (TT's) en los que se pueden mapear los FEC's. Un TT es una agregación de flujos de tráfico que pertenecen a la misma clase y que se envían a través de un camino común.
- un conjunto de atributos puede ser asociados a los TT's para modelar su comportamiento.
- se pueden asociar un conjunto de atributos a los recursos a fin de restringir el establecimiento de FEC's y TT's a través de ellos.
- MPLS permite tanto la agregación de tráfico y desagregación, mientras que el clásico enrutamiento basado en la IP destino permite solamente agregación.
- es relativamente fácil la integración de un marco de enrutamiento basado en restricciones con MPLS.
- una buena implementación de MPLS puede ofrecer un overhead significativamente inferior que otras alternativas para la TE.

Las tareas a realizar por MPLS dentro de la TE son 3, las cuales deben de realizarse de manera secuencial. Estas tareas son:

- mapearlos paquetes en FEC's, es necesario clasificar los paquetes en diferentes flujos o FEC's, ver figura 23.

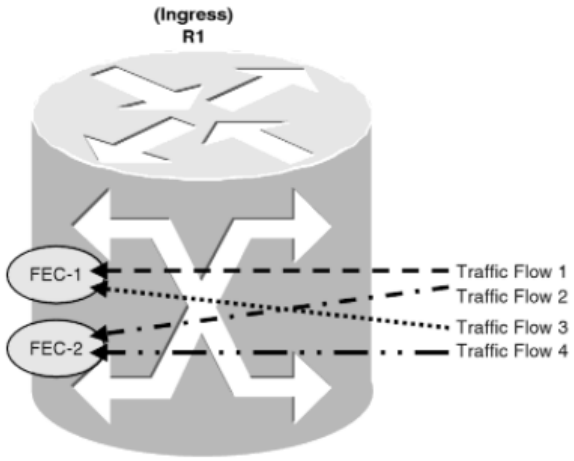

Figura 23. Definición de los flujos de tráfico

- mapear los FEC's Sobre TT's, ver figura 24. Una vez identificados los FEC's, los mismos tiene que ser mapeados sobre los TT's con sus necesidades de recursos correspondientes.

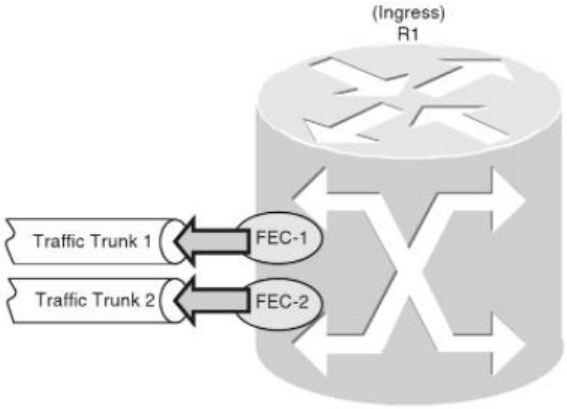

Figura 24. Mapeado de FEC's a TT's

- por último, hay que mapear los TT's sobre una topología física de red, ver figura 25. Esta es la tarea más importante de MPLS-TE. Los TT's necesitan un camino apropiado sobre

la red física que cumpla con sus requerimientos. Los protocolos de routing basados en restricciones son los encargados de realizar dichas tareas.

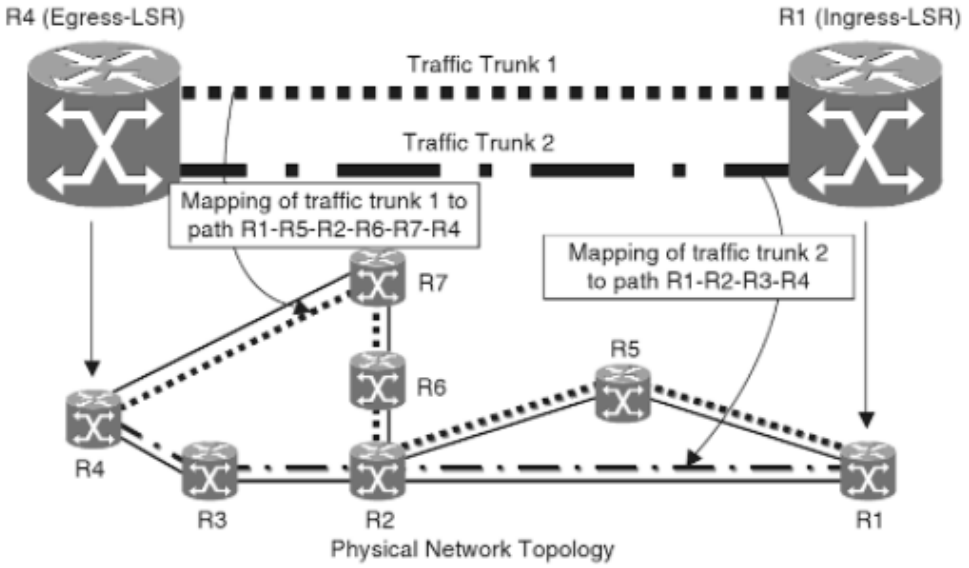

Figura 25. Mapeado de TT's a una topología física de red

3.3 Routing basado en restricciones

El routing basado en restricciones (**CBR**, Constraint Based Routing) es una de las capacidades funcionales más importantes para MPLS-TE. CBR selecciona la mejor ruta que cumple las restricciones establecidas, de manera que sea óptima respecto a alguna métrica escalar (por ejemplo, minimizar la cantidad de saltos o una métrica administrativa). Dichas restricciones son impuestas, por un lado, por políticas de enrutamiento que administran, gestionan y controlan el acceso a los recursos de la red y, por otro lado, por requisitos de calidad de servicio dados por el uso del ancho de banda, retardos, jitter y pérdidas de paquetes.

Para lograr estos objetivos, se utiliza el algoritmo **CSPF** (Constrained Shortest Path First), que es una extensión del algoritmo SPF (Shortest Path First). El algoritmo CSPF requiere que el LSR que realiza el cálculo del camino tenga información sobre todos los enlaces en la red. Esto impone una restricción en el tipo de protocolo de enrutamiento que se puede usar, es decir, se deben usar protocolos de estado de enlace como IS-IS u OSPF.

Para implementar la TE, MPLS-TE realizará las siguientes actividades:

- distribución de información.
- selección de caminos.
- activación de camino.
- envío de tráfico a través del camino.

Para desarrollar las actividades mencionadas, en MPLS-TE será necesario el trabajo conjunto de dos protocolos, CBR y RSVP-TE.

El protocolo CBR es un protocolo que se dedica exclusivamente a analizar todos los recursos de la red disponibles, para establecer el mejor LSP por donde circulará el túnel desde el origen (headend) hasta el destino (tailend).

Una vez establecido el LSP el protocolo RSVP se encargará de realizar la reserva de ancho de banda correspondiente en cada una de las interfaces.

RSVP es un protocolo de señalización, encargado de establecer y mantener el LSP del túnel, haciendo uso principalmente de dos tipos de mensajes:

- **Path Message**: se encarga de solicitar el ancho de banda requerido a lo largo del LSP del túnel.
- **Resv Message**: confirma la solicitud de reserva de ancho de banda que había realizado con el mensaje PATH y asigna una etiqueta al túnel.

En la figura 22, se puede observar de manera gráfica como se realiza dicho proceso.

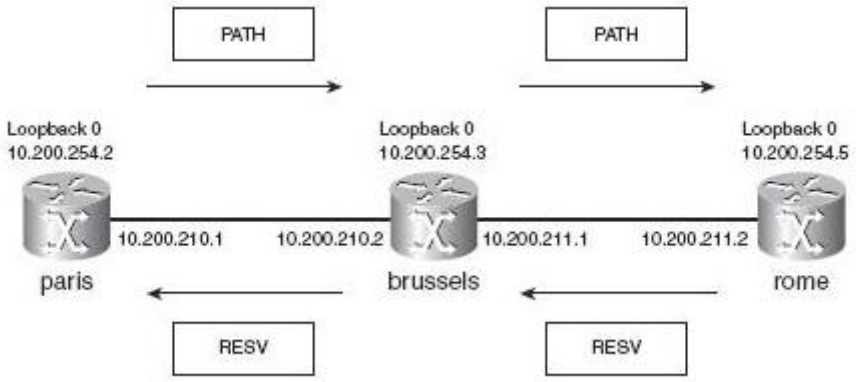

Figura 26. Establecimiento del LSP mediante RSVP-TE

3.4 Objetivos de la práctica

El objetivo de la presente práctica es familiarizarse con la tecnología y los conceptos de Ingeniería de Tráfico en MPLS, así como su configuración en una red implementada con routers Cisco Systems.

Para ello, se deberán realizar las siguientes actividades:

- introducir en los routers los comandos necesarios para utilizar MPLS-TE.
- definir túneles en la red y verificar el correcto funcionamiento de estos.
- visualizar los diferentes paquetes que circulan por la red e identificar los campos pertenecientes a los diferentes protocolos utilizados en MPLS-TE.

3.5 Materiales a utilizar

Para la realización de la presente práctica se utilizarán los routers CISCO 1841 disponibles en el laboratorio. En la figura 27 podemos ver una imagen de la trasera del mencionado router y en la tabla 7 la descripción de cada uno de los elementos presentes en la misma.

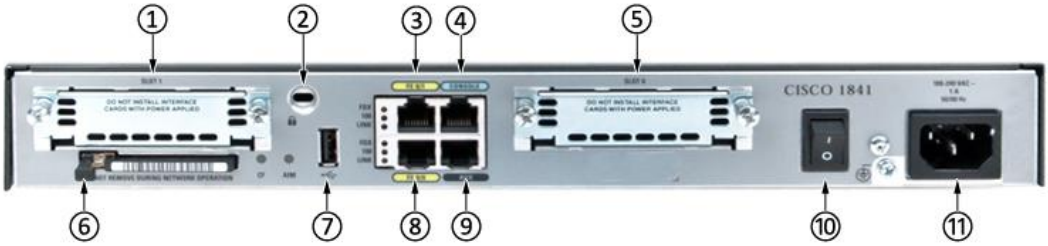

Figura 27. Trasera router CISCO 1841

ID	DESCRIPCIÓN
1	Slot de expansión 1
2	Accesorio para bloqueo
3	Puerto Fast Ethernet 0/0
4	Puerto de consola
5	Slot de expansión 0
6	Unidad compact flash
7	Puerto USB
8	Puerto Fast Ethernet 0/1
9	Puerto auxiliar
10	Interruptor de encendido
11	Entrada de alimentación

Tabla 7. Identificación elementos trasera CISCO 1841

Además de los citados routers, será necesaria una tarjeta expansora de 4 puertos Ethernet por cada router, la HWIC-4ESW, también disponible en el laboratorio. La tarjeta deberá ser instalada en el Slot 0 antes de conectar el router a la corriente.

La numeración de los interfaces de la HWIC-4ESW, como podemos apreciar en la figura 28, se inicia desde la derecha, por lo que el primer interfaz de la derecha será el FE 0/0/0 y el último el FE 0/0/3.

Figura 28. Tarjeta expansora Ethernet HWIC-4ESW

Aparte de los routers, cables de alimentación y latiguillos Fast Ethernet cruzados, será necesaria la utilización de cuatro PC's del laboratorio, uno de ellos deberá tener instalado el analizador de redes "*Wireshark*" para realizar capturas de tráfico circulante por la red.

3.6 Diagrama de la topología de red

En la siguiente figura se observa la topología de la red a montar.

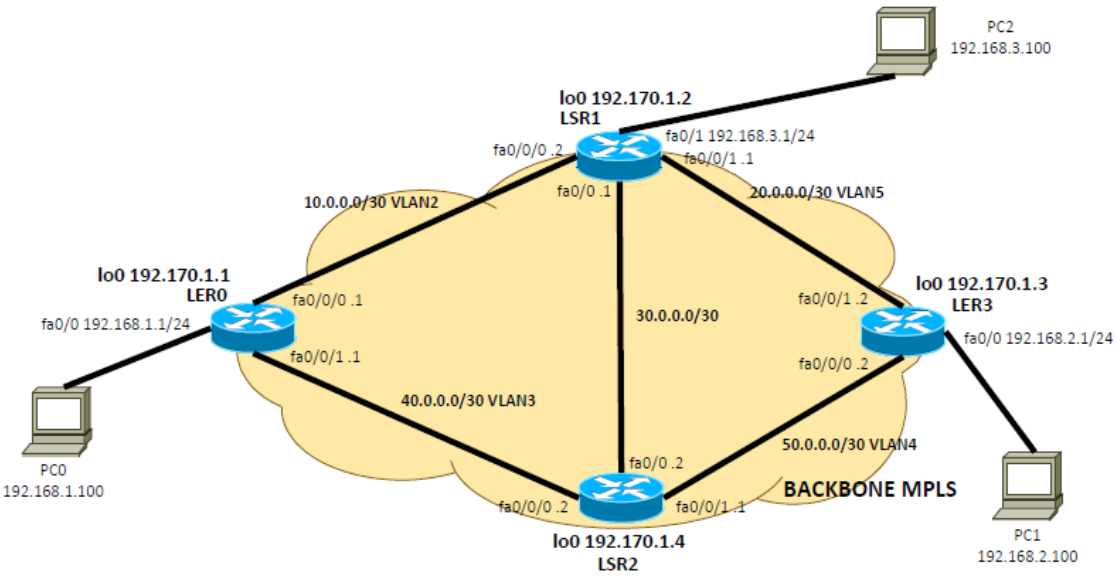

Figura 29. Diagrama de la red MPLS a configurar

En la siguiente tabla se especifican las diferentes direcciones de cada uno de los interfaces de cada router y PC's conectados a la red propuesta.

DISPOSITIVO	INTERFAZ	DIRECCIÓN IP	MÁSCARA DE RED	GATEWAY	VLAN
LER0	Fa0/0/0	10.0.0.1	255.255.255.252	-	VLAN 2
	Fa0/0/1	40.0.0.1	255.255.255.252	-	VLAN 3
	Fa0/0	192.168.1.1	255.255.255.0	-	-
	Lo0	192.170.1.1	255.255.255.255	-	-
LSR1	Fa0/0/0	10.0.0.2	255.255.255.252	-	VLAN 2
	Fa0/0/1	20.0.0.1	255.255.255.252	-	VLAN 5
	Fa0/0	30.0.0.1	255.255.255.252	-	-
	Fa0/1	192.168.31	255.255.255.0		-
	Lo0	192.170.1.2	255.255.255.255	-	-
LSR2	Fa0/0/0	40.0.0.2	255.255.255.252	-	VLAN 3
	Fa0/0/1	50.0.0.1	255.255.255.252	-	VLAN 4
	Fa0/0	30.0.0.2	255.255.255.252	-	-
	Lo0	192.170.1.4	255.255.255.255	-	-
LER3	Fa0/0/0	50.0.0.2	255.255.255.252	-	VLAN 4
	Fa0/0/1	20.0.0.2	255.255.255.252	-	VLAN 5
	Fa0/0	192.168.2.1	255.255.255.0	-	-
	Lo0	192.170.1.3	255.255.255.255	-	-
PC0	NIC	192.168.1.100	255.255.255.0	192.168.1.1	-
PC1	NIC	192.168.2.100	255.255.255.0	192.168.2.1	-
PC2	NIC	192.168.3.100	255.255.255.0	192.168.3.1	-

Tabla 8. Tabla de direccionamiento

3.7 Configuración de la red

Antes de comenzar con la TE, es necesario configurar la red para adecuarla a la nueva topología.

Partiendo de la práctica anterior será necesario modificar la configuración del router LSR1 arreglo a la nueva topología de red, es decir, configurar la dirección ip del nuevo interfaz y realizar los cambios necesarios en el protocolo OSPF. Para ello se deberemos introducir los siguientes comandos:

```
LSR1#configure terminal
LSR1(config)#router ospf 1
LSR1(config-router)# network 192.168.3.0 0.0.0.255 area 0
LSR1(config-router)#interface FastEthernet0/1
LSR1(config-if)#ip address 192.168.3.1 255.255.255.0
LSR1(config-if)#no shutdown
LSR1(config-if)#exit
```

Seguidamente pasaremos a configurar la red para dotarla de capacidades MPLS-TE, para ello vamos a establecer un par de túneles en la misma.

Para establecer un túnel, se ha de definir el Headend (inicio del túnel) y el Tailend (final del túnel). En nuestra red estableceremos un túnel entre LER0 y LER3, permitiendo la comunicación entre la red 192.168.1.100/24 conectada directamente al LER0 hacia la red 192.168.2.100/24 conectada directamente al LER3 a través del citado túnel. Puesto que los túneles son unidireccionales, procederemos a definir un túnel en el sentido contrario también, esto es desde LER3 hasta LER0.

Los túneles pueden crearse de forma dinámica o explicita. En el primer caso es el protocolo CBR el encargado de decidir cuál es el mejor camino hacia el Tailend, mientras que en la creación del túnel de forma explícita debe definirse el camino salto a salto.

Vamos a configurar los túneles utilizando ambos métodos, el túnel que va desde LER0 hasta el LER 3 lo haremos de manera dinámica, mientras que para el sentido contrario lo estableceremos de forma explícita, haciendo que atraviese LSR1 y LSR2.

Otros parámetros que deben tenerse en cuenta son:

- ancho de banda: debe definirse un ancho de banda del túnel para realizar la reserva en cada una de las interfaces por donde discurre el mismo. En el caso que alguna de las interfaces del LSP por donde transcurre el túnel no dispusiese de los recursos necesarios para establecer el mismo, éste no se levantaría.
- prioridad: a través de este atributo indicamos qué túnel es más importante que otro, de tal manera que, si dos túneles ocupasen el mismo LSP y no se dispusieran de los recursos necesarios para ambos, únicamente se establecería el túnel de prioridad más elevada.

Antes de establecer los túneles y para no tener que generar demasiado tráfico a la hora de desarrollar la presente práctica, ya que una de las cosas que pretendemos observar es cómo se comporta la red cuando se encuentra congestionada, vamos a limitar el ancho de banda de las interfaces, para ello utilizaremos el comando **rate limit**.

Vamos a establecer el ancho de banda en cada una de las interfaces de la red MPLS en 160 kbps, para ello los comandos a utilizar en el caso del LER0 serán los siguientes:

```
LER0#configure terminal
LER0(config)#interface Vlan 2
LER0(config-if)#rate-limit input 160000 160000 160000 conform-action transmit exceed-
action drop
LER0(config-if)#rate-limit output 160000 160000 160000 conform-action transmit exceed-
action drop
LER0(config-if)#interface Vlan 3
```

```
LER0(config-if)#rate-limit input 160000 160000 160000 conform-action transmit exceed-
action drop
LER0(config-if)#rate-limit output 160000 160000 160000 conform-action transmit exceed-
action drop
LER0(config-if)#exit
LER0(config)#exit
```

La sintaxis del comando es la siguiente: **rate-limit {input | output} [dscp** *dscp-value***] [access-group [rate-limit]** *acl-index***]** *bps burst-normal burst-max* **conform-action** *conform-action* **exceed-action** *exceed-action*.

Tal y como se puede apreciar se pueden limitar los flujos tanto en la entrada como en la salida del interfaz y permitir ráfagas, en nuestro caso hemos limitado tanto la entrada como la salida a 160 kbps, no permitiendo ninguna ráfaga.

Repetir el paso anterior en los diferentes interfaces de los routers que forman nuestra red.

Ahora comenzaremos propiamente con la configuración de TE, en primer lugar es necesario habilitar la ingeniería de tráfico de modo general y posteriormente en cada uno de los interfaces, para ello utilizaremos el comando **mpls traffic-eng tunnels**. En el caso del LER0 deberemos emplear los siguientes comandos:

```
LER0#configure terminal
LER0(config)#mpls traffic-eng tunnels
LER0(config)#interface FastEthernet0/0/0
LER0(config-if)#mpls traffic-eng tunnels
LER0(config-if)#ip rsvp bandwith 128 64
LER0(config-if)#interface FastEthernet0/0/1
LER0(config-if)#mpls traffic-eng tunnels
LER0(config-if)#ip rsvp bandwith 128 64
LER0(config-if)#interface Vlan 2
LER0(config-if)#mpls traffic-eng tunnels
LER0(config-if)#ip rsvp bandwith 128 64
LER0(config-if)#interface Vlan 3
LER0(config-if)#mpls traffic-eng tunnels
LER0(config-if)#ip rsvp bandwith 128 64
LER0(config-if)#exit
```

Asimismo, con el comando **ip rsvp bandwith**, habilitamos rsvp en la interfaz y reservamos un ancho de banda máximo para establecer el túnel en la misma, en este caso hemos hecho una reserva de 128 kbps, 64 kbps para cada tunel.

Podemos comprobar que hemos realizado correctamente la reserva de ancho de banda a través del comando **show ip rsvp interface**, el resultado debería ser similar al siguiente:

```
LSR1#show ip rsvp interface
interface      allocated   i/f max   flow max   sub max   VRF
Fa0/0          0           128K      64K        0
Fa0/0/0        0           128K      64K        0
Fa0/0/1        0           128K      64K        0
Vl2            64K         128K      64K        0
Vl5            0           128K      64K        0
```

El campo *allocated* nos indica la cantidad de ancho de banda que ha sido reservada en la interfaz.

Aparte de habilitar la TE en los interfaces, también es necesario activarla en el proceso OSPF para conocer el estado real de los enlaces de la red y poder aprovechar los menos utilizados. La forma de habilitarla es la siguiente:

```
LER0(config)#router ospf 1
LER0(config-router)#mpls traffic-eng router-id Loopback 0
LER0(config-router)#mpls traffic-eng area 0
LER0(config-router)#exit
LER0(config)#exit
```

Seguidamente repite los anteriores pasos en todos los routers que constituyen la red.

Una vez habilitada la TE en toda la red, procederemos a establecer los túneles, empezaremos por el túnel dinámico de LER0 a LER3.

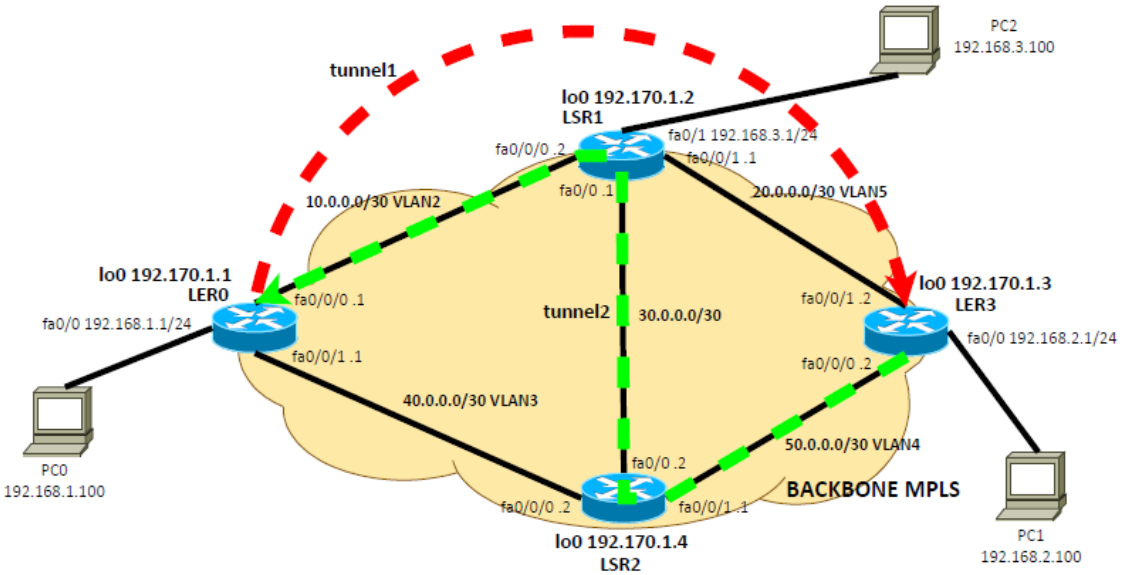

Figura 30. Túneles a establecer en la red

Un túnel se trata igual que una interfaz en CISCO, el primer paso es declarara la interfaz. Al primer túnel lo identificaremos como *tunnel 1*. El túnel ha de establecerse desde su inicio (Headend), en este caso desde LER0. Introducir los siguientes comandos para establecer el túnel dinámico:

```
LER0(config)#interface tunnel 1
LER0(config-if)#ip unnumbered loopback 0
LER0(config-if)#tunnel mode mpls traffic-eng
```

```
LER0(config-if)#tunnel destination 192.170.1.3
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 bandwith 64
LER0(config-if)#tunnel mpls traffic-eng priority 7 7
LER0(config-if)#exit
LER0(config)#
```

Como se puede apreciar existen diferentes comandos que definen el comportamiento del túnel, pasaremos a explicar el significado de cada uno:

- **ip unnumbered *loopback 0***, asignamos la ip de la interfaz de loopback al túnel.
- **tunnel mode mpls traffic-eng**, habilita el modo MPLS-TE en el túnel.
- **tunnel destination *191.170.1.3***, especifica el final del túnel.
- **tunnel mpls traffic-eng autoroute announce**, anuncia el túnel a través de OSPF, de esta manera todo el tráfico dirigido hacia el Tailend circulará a través del túnel.
- **tunnel mpls traffic-eng path-option 2 dynamic**, con *path option* indicamos el orden con el que se intenta establecer el túnel, un túnel con path-option 1 es prioritario frente a uno con 2. Si la interfaz no dispusiera de recursos suficientes para los dos túneles, únicamente establecería el primero.
  Mientras que con *dynamic* indicamos que el protocolo CBR se encargue de calcular el LSP del túnel.
- **tunnel mpls traffic-eng bandwith 64**, establece el ancho de banda reservado del túnel, en este caso 64 kbps.
- **tunnel mpls traffic-eng priority 7 7**, indica la prioridad del túnel, un valor menor indica mayor prioridad.

En este punto ya tenemos un túnel dinámico establecido entre LER0 y LER3. Seguidamente configuraremos el túnel explícito entre LER3 y LER0. Para ello utilizaremos los siguientes comandos:

```
LER3(config)#interface tunnel 2
LER3(config-if)#ip unnumbered loopback 0
LER3(config-if)#tunnel mode mpls traffic-eng
LER3(config-if)#tunnel destination 192.170.1.1
LER3(config-if)#tunnel mpls traffic-eng autoroute announce
LER3(config-if)#tunnel mpls traffic-eng path-option 1 explicit name tunel2
LER3(config-if)#tunnel mpls traffic-eng bandwith 64
LER3(config-if)#tunnel mpls traffic-eng priority 6 6
LER3(config-if)#exit
LER3(config)# ip explicit-path name tunel2
LER3(cfg-ip-expl-path)#next-address 192.170.1.4
LER3(cfg-ip-expl-path)#next-address 192.170.1.2
LER3(cfg-ip-expl-path)#next-address 192.170.1.1
LER3(cfg-ip-expl-path)#exit
LER3(config)#
```

Como podemos ver la única diferencia frente a un túnel dinámico, es que se especifican los saltos por los que discurre el túnel de manera manual.

Para comprobar que los túneles han sido correctamente establecidos, utilizaremos el comando **show mpls traffic-eng tunnels brief**, al ejecutarlo en alguno de los router de nuestra red, el resultado debería ser similar al siguiente:

```
LSR1#show mpls traffic-eng tunnels brief
```

```
Signalling Summary:
    LSP Tunnels Process:            running
    Passive LSP Listener:           running
    RSVP Process:                   running
    Forwarding:                     enabled
    Periodic reoptimization:        every 3600 seconds, next in 2045 seconds
    Periodic FRR Promotion:         Not Running
    Periodic auto-bw collection:    disabled
TUNNEL NAME             DESTINATION     UP IF    DOWN IF   STATE/PROT
LER3_t2                 192.170.1.1     Fa0/0    Vl2       up/up
Displayed 0 (of 0) heads, 1 (of 1) midpoints, 0 (of 0) tails
```

En este caso en LSR1 sólo visualizamos el túnel 2, pero es posible en determinadas ocasiones, como ya mencionamos en la primera práctica, que ambos túneles discurran a través de él.

Como podemos observar de la ejecución del anterior comando, en el caso del LSR1 observamos que está establecido uno de los dos túneles que habíamos definido, el otro túnel debe de haberse establecido a través de LSR2. Comprueba con el mismo comando en el LSR2 que el túnel se haya establecido correctamente. Existe también la posibilidad de que ambos túneles se establezcan atravesando el LSR1.

Otra manera de comprobar que el túnel se ha establecido correctamente es ejecutando el comando **show mpls traffic-eng tunnels *tunnel interface***, en alguno de los LER.

```
LER0#show mpls traffic-eng tunnels tunnel 1

Name: LER0_t1                           (Tunnel1) Destination: 192.170.1.3
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected

    path option 2, type dynamic (Basis for Setup, path weight 2)

  Config Parameters:
    Bandwidth: 64       kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 64       bw-based
    auto-bw: disabled
  Active Path Option Parameters:
    State: dynamic path option 2 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled

  InLabel  :  -
  OutLabel : Vlan3, 203
  RSVP Signalling Info:
       Src 192.170.1.1, Dst 192.170.1.3, Tun_Id 1, Tun_Instance 13
    RSVP Path Info:
      My Address: 40.0.0.1
      Explicit Route: 40.0.0.2 50.0.0.1 50.0.0.2 192.170.1.3
      Record Route:  NONE
      Tspec: ave rate=64 kbits, burst=1000 bytes, peak rate=64 kbits
    RSVP Resv Info:
      Record Route:  NONE
      Fspec: ave rate=64 kbits, burst=1000 bytes, peak rate=64 kbits
  Shortest Unconstrained Path Info:
    Path Weight: 2 (TE)
```

```
  Explicit Route: 40.0.0.1 40.0.0.2 50.0.0.1 50.0.0.2
                  192.170.1.3
History:
  Tunnel:
    Time since created: 31 minutes, 34 seconds
    Time since path change: 28 minutes, 56 seconds
  Current LSP:
    Uptime: 28 minutes, 56 seconds
```

Como podemos ver de la ejecución del último comando, el túnel está activo y señalizado. Entre otra información observamos la información RSVP, con la ruta explícita, el ancho de banda reservado, etc.

Por otra parte, si volvemos a ejecutar el comando **show mpls forwarding-table *destination-prefix* detail** en alguno de los dos LER, observaremos que el túnel para alcanzar su destino utiliza diferente etiquetado, al ser este asignado por el protocolo RSVP, por ejemplo, en el LER3 vemos seguidamente que utilizará la etiqueta 207, mientras que anterior práctica para alcanzar dicho destino se utilizaba la 205.

```
LER3#sh mpls forwarding-table 192.168.1.100 detail
Local      Outgoing    Prefix           Bytes Label   Outgoing    Next Hop
Label      Label       or Tunnel Id     switched      interface
303        No Label    192.168.1.0/24   0             Tu2         point2point
        MAC/Encaps=14/18, MRU=1500, Tag Stack{207}, via Vl4
        C40307320000C404074100008847 000D0000
        No output feature configured
```

Por último, vamos a utilizar la función SPAN de las HWIC como ya hicimos en la práctica 1. Haremos un mirroring del puerto FastEthernet0/0/0 del LSR1 sobre el puerto FastEthernet0/0/2, donde conectaremos el PC con el analizador de protocolos. Para ellos introducir los siguientes comandos en el LSR1:

```
LSR1#configure terminal
LSR1(config)#monitor session 1 source interface FastEthernet0/0/0
LSR1(config)#monitor session 1 destination interface FastEthernet0/0/2
LSR1(config)#end
LSR1#
```

3.8 Ejercicios propuestos

1. ¿Qué camino seguirán los paquetes de PC0 a PC1 en ambos sentidos?. ¿Coinciden con la ruta calculada por el protocolo IGP?. De no ser así, ¿a qué es debido?.

Para determinar la ruta que siguen los paquetes nos bastará con utilizar el comando traceroute desde los routers LER0 y LER3, ya que como se indica en el guion, los túneles no son bidireccionales.

En el primer caso obtenemos:

```
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 203 Exp 0] 4 msec 4 msec 0 msec
  2 50.0.0.2 4 msec 0 msec 4 msec
  3 192.168.2.100 0 msec 4 msec 0 msec
```

Como se puede observar coincide con la ruta calculada por el IGP, ya que el túnel discurre por el mismo camino.

En el caso del sentido contrario.

```
LER3#traceroute 192.168.1.100

Type escape sequence to abort.
Tracing the route to 192.168.1.100

  1 50.0.0.1 [MPLS: Label 207 Exp 0] 0 msec 4 msec 0 msec
  2 30.0.0.1 [MPLS: Label 100 Exp 0] 4 msec 0 msec 4 msec
  3 10.0.0.1 0 msec 4 msec 0 msec
  4 192.168.1.100 4 msec 0 msec 0 msec
```

Como podemos ver en este caso, el camino no coincide con el calculado por el IGP, ya que no es el camino más corto, pero al ser un túnel estático, dicho camino ha quedado establecido en la definición del túnel.

Seguidamente utilizaremos el analizador de paquetes para identificar los mensajes RSVP utilizados en la reserva de recursos del túnel.

2. Observar la captura de wireshark que estamos obteniendo e identificar los mensajes y campos explicados en la teoría de la asignatura.

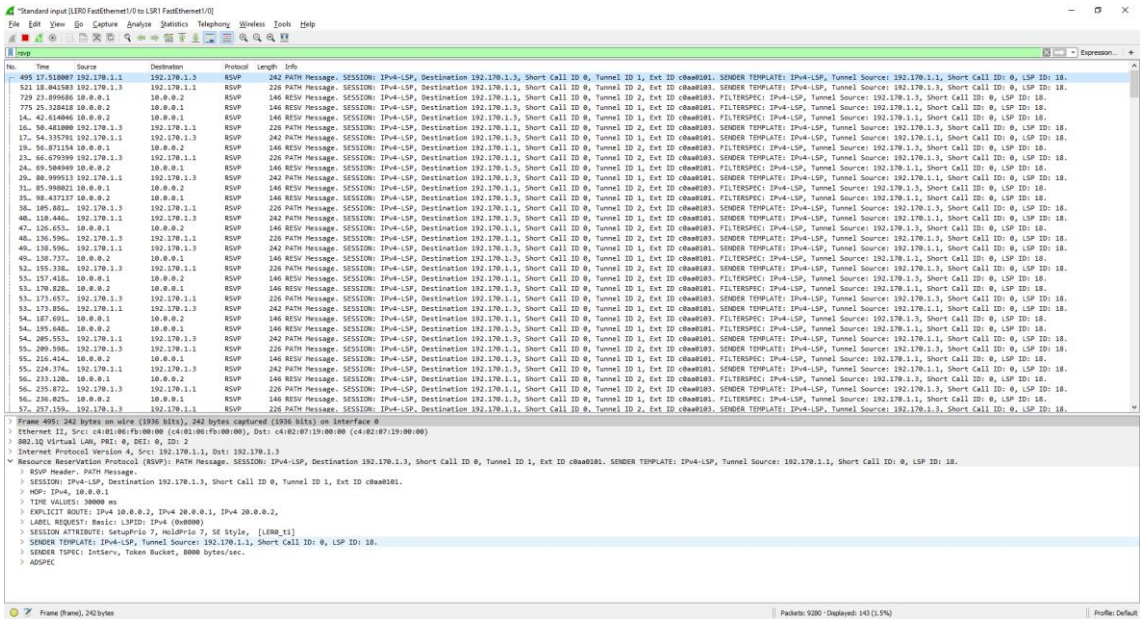

Figura 31. Mensajes RSVP para el establecimiento de los túneles

Como podemos ver en la captura de la figura 31 y según vimos en la teoría de la asignatura en la reserva de recursos del túnel intervienen dos tipos de mensajes RSVP, el PATH y el RSV.

Dentro de cada uno de los paquetes podemos identificar los diferentes objetos que lo componen, en el caso del PATH (ver figura 32), estos son:

- Session: identifica el túnel LSP siendo establecido con la IP del egress LSR, un identificador del túnel y la IP del ingress LSR. Como se puede apreciar en las capturas la IP del egress es la 192.170.1.1, el ID es 2 la IP del ingress 192.170.1.3. Tal y como habíamos definido en la constitución del túnel.
- RSVP Hop: dir IP e interface del LSR que envía el mensaje, en este caso el mensaje nos llega a través de 10.0.0.2.
- Sender Tspec: reserva de ancho de banda solicitada junto con min y max tamaño de paquete soportado por el túnel LSP. En este caso son 8000 bytes/s, es decir, los 64000 kbps que establecimos.
- Sender Template: contiene la dirección IP del emisor y el identificador del LSP, el emisor fue como hemos mencionado antes 192.170.1.3, mientras que el LSP ID es 18.

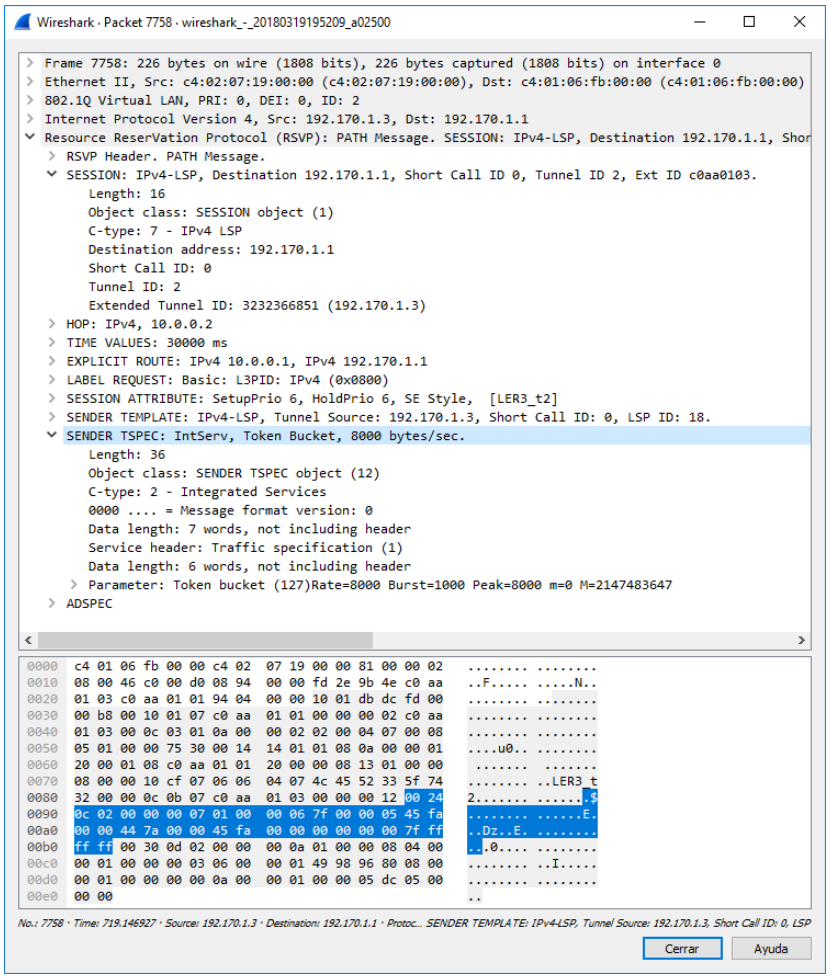

Figura 32. Campos del mensaje PATH

En cuanto al mensaje RESV (ver figura 33), los dos primeros campos son idénticos, analizaremos el resto:

- Style: sirve para definir el estilo de reserva FF (fixed filter) o SE (shared explicit), en nuestro caso Shared-Explicit, una sola reserva sobre el enlace común compartido.
- Flowspec: se genera como respuesta al Sender Tspec del mensaje Path y sirve para definir la QoS deseada, lo cual incluye una indicación de que servicio de control de la QoS está siendo solicitado, y los parámetros requeridos para ese servicio (RFC 2210).
- Filter Spec: en conjunción con el objeto Session define el conjunto de paquetes de datos o traffic trunk que recibe la QoS deseada definida por Flowspec.
- Label: etiqueta asignada al túnel por RSVP, es distinta a la determinada por LSP, y será la utilizada para encaminar los paquetes a través del túnel.

```
> SESSION: IPv4-LSP, Destination 192.170.1.3, Short Call ID 0, Tunnel ID 1, Ext ID c0aa0101.
> HOP: IPv4, 10.0.0.2
> TIME VALUES: 30000 ms
v STYLE: Shared-Explicit (18)
      Length: 8
      Object class: STYLE object (8)
      C-type: 1
      Flags: 0x00
      Style: Shared-Explicit (0x000012)
v FLOWSPEC: Controlled Load: Token Bucket, 8000 bytes/sec.
      Length: 36
      Object class: FLOWSPEC object (9)
      C-type: 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=8000 Burst=1000 Peak=8000 m=0 M=0
v FILTERSPEC: IPv4-LSP, Tunnel Source: 192.170.1.1, Short Call ID: 0, LSP ID: 14.
      Length: 12
      Object class: FILTER SPEC object (10)
      C-type: 7 - IPv4 LSP
      Sender IPv4 address: 192.170.1.1
      Sender LSP ID: 14
> LABEL: 106
```

Figura 33. Campos del mensaje RESV

Veamos ahora el comportamiento que tienen los túneles frente al tráfico que discurre a través de ellos. Para ello, vamos a utilizar el comando **ping** para generar tráfico en nuestra red. Abriremos un par de consolas en el PC0, desde el menú "Inicio→Ejecutar", teclear **cmd** y presionar "Enter".

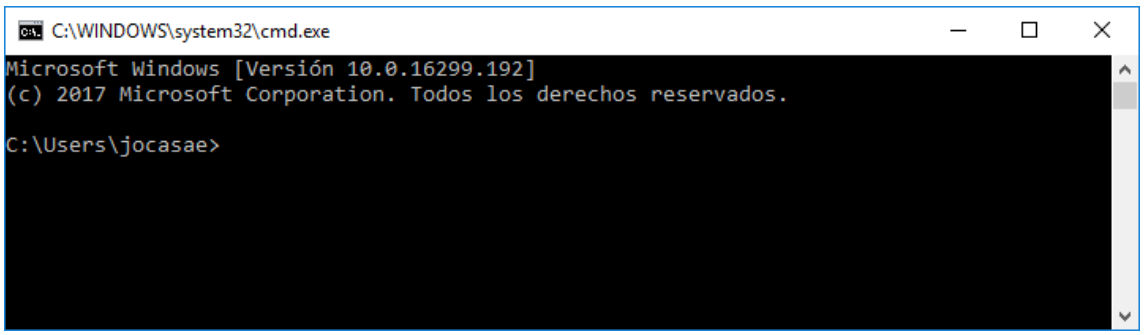

Figura 34. Consola de comandos, windows

La sintaxis del comando **ping** es la siguiente: ping [/t] [/a] [/n <Count>] [/l <Size>] [/f] [/I <TTL>] [/v <TOS>] [/r <Count>] [/s <Count>] [{/j <HostList> | /k <HostList>}] [/w <Timeout>] [/R] [/S <SrcAddr>] [/4] [/6] <TargetName>.

En nuestra consola teclearemos: **ping /t /l 19000 192.168.3.100**. Con esto generaremos un tráfico de aproximadamente 160 kbps de forma ininterrumpida hacia el PC2. El campo *l*, especifica el tamaño en bytes del campo de datos que se envía en el ping.

Transcurridos unos segundos, abrir otra consola en el mismo PC0 y teclear: **ping /t /l 8000 192.168.2.100**, de esta manera generamos un tráfico de aproximadamente 60 kbps hacia PC1.

Veamos ahora que comportamiento tienen dichos tráficos al circular por nuestra red, para ello nos valdremos de la utilidad de creación de gráficos de "*Wireshark*", para ello ir al menú "*Statistics→I/O Graph*".

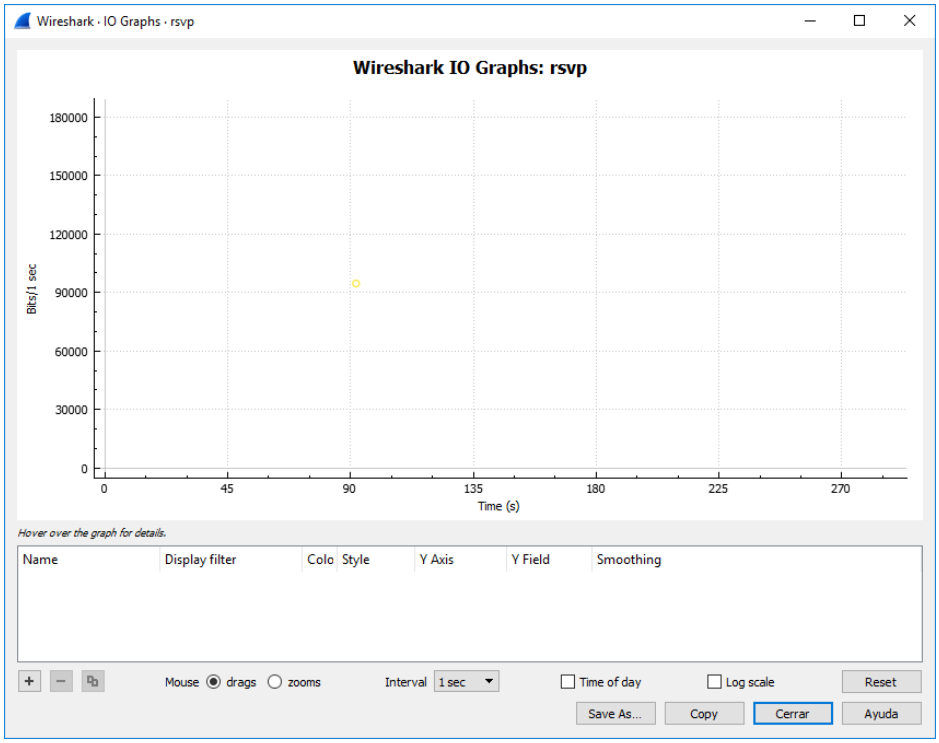

Figura 35. Wireshark, Statistics-I/O Graph

Presionar en el icono "+", "Add a new graph", dos veces para añadir dos líneas de gráfico. Seguidamente aplicaremos un filtro distinto a cada una de ellas. Para ello hacer click con el ratón sobre el campo "Display Filter", y en el primer caso añadir el siguiente filtro**: ip.dst == 192.168.2.100**. En el campo "Y Axis" seleccionar "bits" y por último en "Smoothing" seleccionar "10 interval SMA" para filtrar un poco los picos de la línea. Repetir el mismo proceso en la otra línea, que representará los paquetes con destino 192.168.3.100.

Si no aparecen en el gráfico paquetes con destino 192.168.2.100, será debido a que el túnel ha sido establecido a través de LSR2, en ese caso, desconectar momentáneamente el enlace entre LER0 y LSR2 para que el túnel se levante por LSR1 y volver a capturar. Para saber de antemano por donde discurre el túnel, nos bastaría con ejecutar el comando **show mpls traffic-eng tunnels tunnel 1** en el LER0 y observar el path del mismo.

3. Analiza el gráfico obtenido y explica el porqué de los valores representados.

El gráfico obtenido deberá ser similar al siguiente:

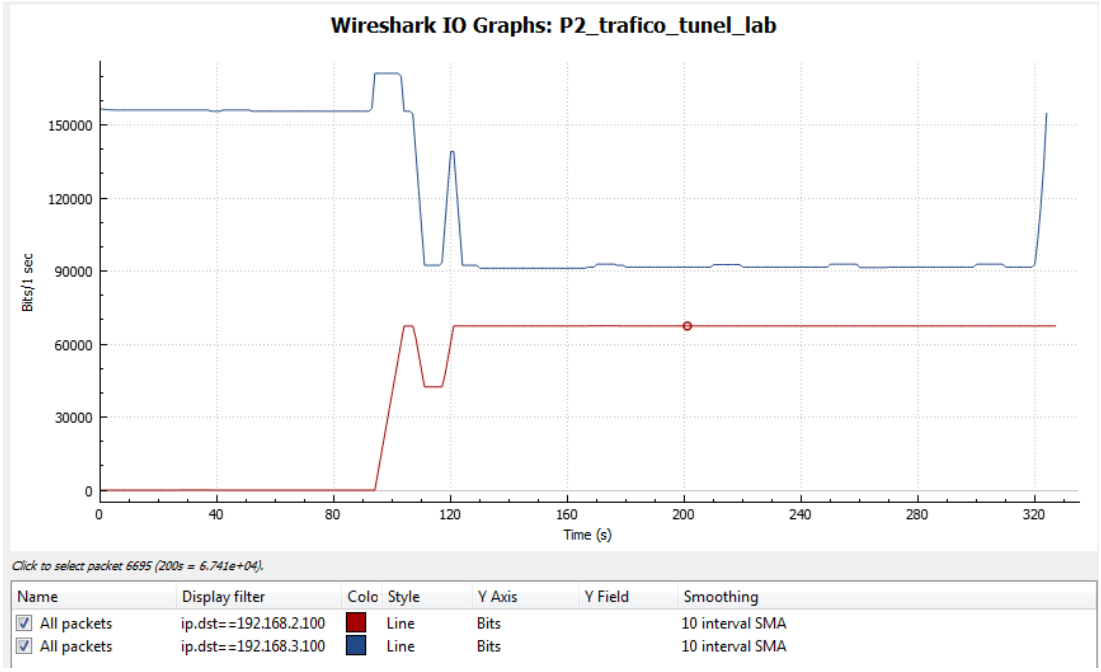

Figura 36. Paquetes de tráfico generados en la red

Como podemos apreciar en la figura 36 el total de tráfico que circula por la red es del orden de 160 kbps, evidentemente no puede ser superior, ya que hemos limitado el tráfico en todos los interfaces de la red a ese valor.

En un primer instante todo el tráfico que generamos hacia PC2, unos 160 kbps, discurre sin problemas por la red, pero al empezar a generar tráfico hacia PC1 este disminuye. Esto es debido a que el tráfico hacia PC1 discurre por el túnel, el cual tiene un ancho de banda reservado de 64 kbps, por lo que el tráfico hacia PC1 tiene prioridad frente al que se dirige hacia PC2. Puesto que el ancho de banda es limitado, el caudal hacia PC2 disminuye en la misma magnitud en la que aumenta el tráfico que circula hacia PC1.

Por último, vamos a analizar el diferente comportamiento que tienen ante fallos de comunicación en la red los diferentes tipos de túneles que hemos definido en la práctica. Para ello simularemos un fallo en la red desconectando el latiguillo conectado a la interfaz FastEthernet 0/0/0 del LER0.

4. ¿Qué efecto tiene la caída del enlace en cado uno de los dos túneles definidos en la red?

Si ejecutamos el comando **sh mpls traffic-eng tunnels *tunnel 1*** en el LER0 antes de que caiga el enlace podemos observar el estado del tunel actualmente, así como su ruta, que vemos pasa por el LSR1.

```
LER0#sh mpls traffic-eng tunnels tunnel 1

Name: LER0_t1                           (Tunnel1) Destination: 192.170.1.3
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected

    path option 2, type dynamic (Basis for Setup, path weight 2)

  Config Parameters:
    Bandwidth: 64       kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
```

```
  Metric Type: TE (default)
  AutoRoute:  enabled   LockDown: disabled  Loadshare: 64       bw-based
  auto-bw: disabled

 InLabel  :  -
 OutLabel : Vlan2, 103
 RSVP Signalling Info:
     Src 192.170.1.1, Dst 192.170.1.3, Tun_Id 1, Tun_Instance 28
  RSVP Path Info:
    My Address: 10.0.0.1
    Explicit Route: 10.0.0.2 20.0.0.1 20.0.0.2 192.170.1.3
    Record Route:  NONE
    Tspec: ave rate=64 kbits, burst=1000 bytes, peak rate=64 kbits
  RSVP Resv Info:
    Record Route:  NONE
    Fspec: ave rate=64 kbits, burst=1000 bytes, peak rate=64 kbits
 Shortest Unconstrained Path Info:
  Path Weight: 2 (TE)
  Explicit Route: 10.0.0.1 10.0.0.2 20.0.0.1 20.0.0.2
                  192.170.1.3
 History:
  Tunnel:
    Time since created: 7 minutes, 12 seconds
    Time since path change: 46 seconds
  Current LSP:
    Uptime: 48 seconds
```

Al ejecutar el mismo comando instantes después de que caiga el enlace obtenemos:

```
LER0#sh mpls traffic-eng tunnels tunnel 1

Name: LER0_t1                          (Tunnel1) Destination: 192.170.1.3
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected

    path option 2, type dynamic (Basis for Setup, path weight 2)

  Config Parameters:
    Bandwidth: 64       kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 64       bw-based
    auto-bw: disabled

  InLabel  :  -
  OutLabel : Vlan3, 208
  RSVP Signalling Info:
      Src 192.170.1.1, Dst 192.170.1.3, Tun_Id 1, Tun_Instance 29
    RSVP Path Info:
      My Address: 40.0.0.1
      Explicit Route: 40.0.0.2 50.0.0.1 50.0.0.2 192.170.1.3
      Record Route:  NONE
      Tspec: ave rate=64 kbits, burst=1000 bytes, peak rate=64 kbits
    RSVP Resv Info:
      Record Route:  NONE
```

```
    Fspec: ave rate=64 kbits, burst=1000 bytes, peak rate=64 kbits
  Shortest Unconstrained Path Info:
   Path Weight: 2 (TE)
   Explicit Route: 40.0.0.1 40.0.0.2 50.0.0.1 50.0.0.2
                   192.170.1.3
  History:
   Tunnel:
     Time since created: 11 minutes, 12 seconds
     Time since path change: 26 seconds
   Current LSP:
     Uptime: 28 seconds
     Selection: reoptimation
   Prior LSP:
     ID: path option 2 [28]
     Removal Trigger: path verification failed
```

Como se puede apreciar el túnel se ha reencaminado a través de LSR2 de forma automática. En el caso del túnel 2, como se puede observar seguidamente el túnel se encuentra caído.

```
LER3#sh mpls traffic-eng tunnels tunnel 2

Name: LER3_t2                          (Tunnel2) Destination: 192.170.1.1
  Status:
    Admin: up         Oper: down    Path: not valid    Signalling: Down
    path option 1, type explicit tunel2

  Config Parameters:
    Bandwidth: 64       kbps (Global)  Priority: 6  6    Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 64       bw-based
    auto-bw: disabled

  Shortest Unconstrained Path Info:
    Path Weight: 2 (TE)
    Explicit Route: 50.0.0.2 50.0.0.1 40.0.0.2 40.0.0.1
                    192.170.1.1
  History:
    Tunnel:
      Time since created: 13 minutes, 42 seconds
      Time since path change: 2 minutes, 55 seconds
    Prior LSP:
      ID: path option 1 [26]
      Removal Trigger: path verification failed
      Last Error: PCALC:: No addresses to connect 192.170.1.2 to 0.0.0.0
```

Al caer el enlace, el protocolo RSVP se encarga de establecer un nuevo camino para el túnel definido de manera dinámica, siempre que los enlaces dispongan de los recursos necesarios, mientras que el definido de forma explícita necesitará de la intervención del administrador de la red que deberá de establecer otra ruta alternativa.

3.9 ANEXO 1 “Ficheros running-config de los routers”

LER0

```
!
version 15.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname LER0
!
boot-start-marker
boot system flash:c1841-advipservicesk9-mz.150-1.M10.bin
boot-end-marker
!
!
no aaa new-model
!
!
!
dot11 syslog
ip source-route
!
!
!
!
ip cef
no ip domain lookup
no ipv6 cef
!
multilink bundle-name authenticated
!
mpls traffic-eng tunnels
mpls label range 16 99
mpls label protocol ldp
!
!
!
!
license udi pid CISCO1841 sn FCZ0935105S
!
redundancy
!
!
!
!
!
!
!
!
!
interface Loopback0
 ip address 192.170.1.1 255.255.255.255
 !
```

```
!
interface Tunnel1
 ip unnumbered Loopback0
 tunnel mode mpls traffic-eng
 tunnel destination 192.170.1.3
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth 64
 tunnel mpls traffic-eng path-option 2 dynamic
 !
 no routing dynamic
!
interface FastEthernet0/0
 ip address 192.168.1.1 255.255.255.0
 duplex auto
 speed auto
 !
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
 !
!
interface FastEthernet0/0/0
 switchport access vlan 2
 switchport mode trunk
 mpls traffic-eng tunnels
 mpls ip
 !
 ip rsvp bandwidth 128 64
!
interface FastEthernet0/0/1
 switchport access vlan 3
 switchport mode trunk
 mpls traffic-eng tunnels
 mpls ip
 !
 ip rsvp bandwidth 128 64
!
interface FastEthernet0/0/2
 shutdown
 !
!
interface FastEthernet0/0/3
 shutdown
 !
!
interface Vlan1
 no ip address
 !
!
interface Vlan2
 ip address 10.0.0.1 255.255.255.252
```

```
 rate-limit input 160000 160000 160000 conform-action transmit exceed-action drop
 rate-limit output 160000 160000 160000 conform-action transmit exceed-action drop
 mpls traffic-eng tunnels
 mpls ip
 !
 ip rsvp bandwidth 128 64
!
interface Vlan3
 ip address 40.0.0.1 255.255.255.252
 rate-limit input 160000 160000 160000 conform-action transmit exceed-action dro p
 rate-limit output 160000 160000 160000 conform-action transmit exceed-action drop
 mpls traffic-eng tunnels
 mpls ip
 !
 ip rsvp bandwidth 128 64
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 log-adjacency-changes
 network 10.0.0.0 0.0.0.3 area 0
 network 40.0.0.0 0.0.0.3 area 0
 network 192.168.1.0 0.0.0.255 area 0
 network 192.170.1.1 0.0.0.0 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
!
!
!
!
!
!
!
control-plane
 !
!
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
end
```

LSR1

```
!
version 15.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname LSR1
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
monitor session 1 source interface Fa0/0/0
monitor session 1 destination interface Fa0/0/2
!
!
!
dot11 syslog
ip source-route
!
!
!
!
ip cef
no ip domain lookup
no ipv6 cef
!
multilink bundle-name authenticated
!
mpls traffic-eng tunnels
mpls label range 100 199
mpls label protocol ldp
!
!
!
!
license udi pid CISCO1841 sn FCZ0935106L
!
redundancy
!
!
!
!
!
!
!
!
!
interface Loopback0
 ip address 192.170.1.2 255.255.255.255
 !
```

```
!
interface FastEthernet0/0
 ip address 30.0.0.1 255.255.255.252
 rate-limit input 160000 160000 160000 conform-action transmit exceed-action drop
 rate-limit output 160000 160000 160000 conform-action transmit exceed-action drop
 duplex auto
 speed auto
 mpls traffic-eng tunnels
 mpls ip
 !
 ip rsvp bandwidth 128 64
!
interface FastEthernet0/1
 ip address 192.168.3.1 255.255.255.0
 duplex auto
 speed auto
 !
!
interface FastEthernet0/0/0
 switchport access vlan 2
 switchport mode trunk
 mpls traffic-eng tunnels
 mpls ip
 !
 ip rsvp bandwidth 128 64
!
interface FastEthernet0/0/1
 switchport access vlan 5
 switchport mode trunk
 mpls traffic-eng tunnels
 mpls ip
 !
 ip rsvp bandwidth 128 64
!
interface FastEthernet0/0/2
 !
!
interface FastEthernet0/0/3
 shutdown
 !
!
interface Vlan1
 no ip address
 !
!
interface Vlan2
 ip address 10.0.0.2 255.255.255.252
 rate-limit input 160000 160000 160000 conform-action transmit exceed-action drop
 rate-limit output 160000 160000 160000 conform-action transmit exceed-action drop
 mpls traffic-eng tunnels
 mpls ip
 !
 ip rsvp bandwidth 128 64
!
interface Vlan5
```

```
 ip address 20.0.0.1 255.255.255.252
 rate-limit input 160000 160000 160000 conform-action transmit exceed-action drop
 rate-limit output 160000 160000 160000 conform-action transmit exceed-action drop
 mpls traffic-eng tunnels
 mpls ip
 !
 ip rsvp bandwidth 128 64
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 log-adjacency-changes
 network 10.0.0.0 0.0.0.3 area 0
 network 20.0.0.0 0.0.0.3 area 0
 network 30.0.0.0 0.0.0.3 area 0
 network 192.168.3.0 0.0.0.255 area 0
 network 192.170.1.2 0.0.0.0 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
!
!
!
!
!
!
!
control-plane
 !
!
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
end
```

LSR2

```
!
version 15.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname LSR2
!
boot-start-marker
boot system flash:c1841-advipservicesk9-mz.150-1.M10.bin
boot-end-marker
!
!
no aaa new-model
!
!
!
dot11 syslog
ip source-route
!
!
!
!
ip cef
no ip domain lookup
no ipv6 cef
!
multilink bundle-name authenticated
!
mpls traffic-eng tunnels
mpls label range 200 299
mpls label protocol ldp
!
!
!
!
license udi pid CISCO1841 sn FCZ1040113S
!
redundancy
!
!
!
!
!
!
!
!
!
!
interface Loopback0
 ip address 192.170.1.4 255.255.255.255
 !
!
interface FastEthernet0/0
```

```
 ip address 30.0.0.2 255.255.255.252
 rate-limit input 160000 160000 160000 conform-action transmit exceed-action drop
 rate-limit output 160000 160000 160000 conform-action transmit exceed-action drop
 duplex auto
 speed auto
 mpls traffic-eng tunnels
 mpls ip
 !
 ip rsvp bandwidth 128 64
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
 !
!
interface FastEthernet0/0/0
 switchport access vlan 3
 switchport mode trunk
 mpls traffic-eng tunnels
 mpls ip
 !
 ip rsvp bandwidth 128 64
!
interface FastEthernet0/0/1
 switchport access vlan 4
 switchport mode trunk
 mpls traffic-eng tunnels
 mpls ip
 !
 ip rsvp bandwidth 128 64
!
interface FastEthernet0/0/2
 shutdown
 !
!
interface FastEthernet0/0/3
 shutdown
 !
!
interface Vlan1
 no ip address
 !
!
interface Vlan3
 ip address 40.0.0.2 255.255.255.252
 rate-limit input 160000 160000 160000 conform-action transmit exceed-action drop
 rate-limit output 160000 160000 160000 conform-action transmit exceed-action drop
 mpls traffic-eng tunnels
 mpls ip
 !
 ip rsvp bandwidth 128 64
!
interface Vlan4
```

```
 ip address 50.0.0.1 255.255.255.252
 rate-limit input 160000 160000 160000 conform-action transmit exceed-action drop
 rate-limit output 160000 160000 160000 conform-action transmit exceed-action drop
 mpls traffic-eng tunnels
 mpls ip
 !
 ip rsvp bandwidth 128 64
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 log-adjacency-changes
 network 30.0.0.0 0.0.0.3 area 0
 network 40.0.0.0 0.0.0.3 area 0
 network 50.0.0.0 0.0.0.3 area 0
 network 192.170.1.4 0.0.0.0 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
!
!
!
!
!
!
!
control-plane
 !
!
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
end
```

LER3

```
!
version 15.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname LER3
!
boot-start-marker
boot system flash:c1841-advipservicesk9-mz.150-1.M10.bin
boot-end-marker
!
!
no aaa new-model
!
!
!
dot11 syslog
ip source-route
!
!
!
!
ip cef
no ip domain lookup
no ipv6 cef
!
multilink bundle-name authenticated
!
mpls traffic-eng tunnels
mpls label range 300 399
mpls label protocol ldp
!
!
!
!
license udi pid CISCO1841 sn FHK104118Q6
!
redundancy
!
!
!
!
!
!
!
!
!
!
interface Loopback0
 ip address 192.170.1.3 255.255.255.255
 !
!
interface Tunnel2
```

```
 ip unnumbered Loopback0
 tunnel mode mpls traffic-eng
 tunnel destination 192.170.1.1
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 6 6
 tunnel mpls traffic-eng bandwidth 64
 tunnel mpls traffic-eng path-option 1 explicit name tunel2
 !
 no routing dynamic
!
interface FastEthernet0/0
 ip address 192.168.2.1 255.255.255.0
 duplex auto
 speed auto
 !
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
 !
!
interface FastEthernet0/0/0
 switchport access vlan 4
 switchport mode trunk
 mpls traffic-eng tunnels
 mpls ip
 !
 ip rsvp bandwidth 128 64
!
interface FastEthernet0/0/1
 switchport access vlan 5
 switchport mode trunk
 mpls traffic-eng tunnels
 mpls ip
 !
 ip rsvp bandwidth 128 64
!
interface FastEthernet0/0/2
 shutdown
 !
!
interface FastEthernet0/0/3
 shutdown
 !
!
interface Vlan1
 no ip address
 !
!
interface Vlan4
 ip address 50.0.0.2 255.255.255.252
 rate-limit input 160000 160000 160000 conform-action transmit exceed-action drop
 rate-limit output 160000 160000 160000 conform-action transmit exceed-action drop
```

```
 mpls traffic-eng tunnels
 mpls ip
 !
 ip rsvp bandwidth 128 64
!
interface Vlan5
 ip address 20.0.0.2 255.255.255.252
 rate-limit input 160000 160000 160000 conform-action transmit exceed-action drop
 rate-limit output 160000 160000 160000 conform-action transmit exceed-action drop
 mpls traffic-eng tunnels
 mpls ip
 !
 ip rsvp bandwidth 128 64
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 log-adjacency-changes
 network 20.0.0.0 0.0.0.3 area 0
 network 50.0.0.0 0.0.0.3 area 0
 network 192.168.2.0 0.0.0.255 area 0
 network 192.170.1.3 0.0.0.0 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
!
ip explicit-path name tunel2 enable
 next-address 192.170.1.4
 next-address 192.170.1.2
 next-address 192.170.1.1
!
!
!
!
!
!
!
control-plane
 !
!
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
end
```

Capítulo 4. VPN MPLS

4.1 Introducción

VPN (Virtual Private Network) MPLS o redes privadas virtuales MPLS, es la más popular y usada implementación de la tecnología MPLS, su popularidad ha crecido exponencialmente en los últimos años. Aunque muchos proveedores de servicios las han implementado como sustitutas de sus antiguas redes ATM o Frame Relay, muchas grandes compañías las están desarrollando dada su escalabilidad y la capacidad de dividir redes en redes más pequeñas, lo cual es muchas veces útil en empresas de gran tamaño, donde con la misma infraestructura tienes que dar servicio a departamentos individuales.

Una VPN es una red que emula redes privadas sobre una infraestructura común. La característica fundamental de una VPN es que todas las ubicaciones conectadas a la misma deben poder utilizar infraestructura común con otras ubicaciones de otra VPN y tener el tráfico completamente separado. Si hablamos de VPN's de nivel IP se amplían mucho las posibilidades, como puede ser ofrecer conectividad entre VPN's distintas en incluso conectividad a internet entre ellas.

Las VPN MPLS son posibles porque el proveedor de servicios dispone de una red MPLS por debajo, que desvincula el plano de control del plano de tráfico lo cual es imposible con una red IP tradicional.

En MPLS existen dos tipos de VPN's L2 y L3 VPN que se corresponden con los niveles 2 y 3 de la capa OSI.

4.2 L3 VPN

Las VPN's de nivel 3 se conocen como 2547 VPN's ó BGP/MPLS IP VPN's. Tienen 3 componentes básicos, tal y como se puede apreciar en la figura 37:

- CE (Customer Edge), router de la red del cliente que sirve de interface con el PE.
- PE (Provider Edge), router de la red del proveedor que hace de ingress LSR a la red del proveedor.
- P (Provider), router de la red del proveedor que transporta tráfico en el backbone pero que no tiene conocimiento de VPN's.

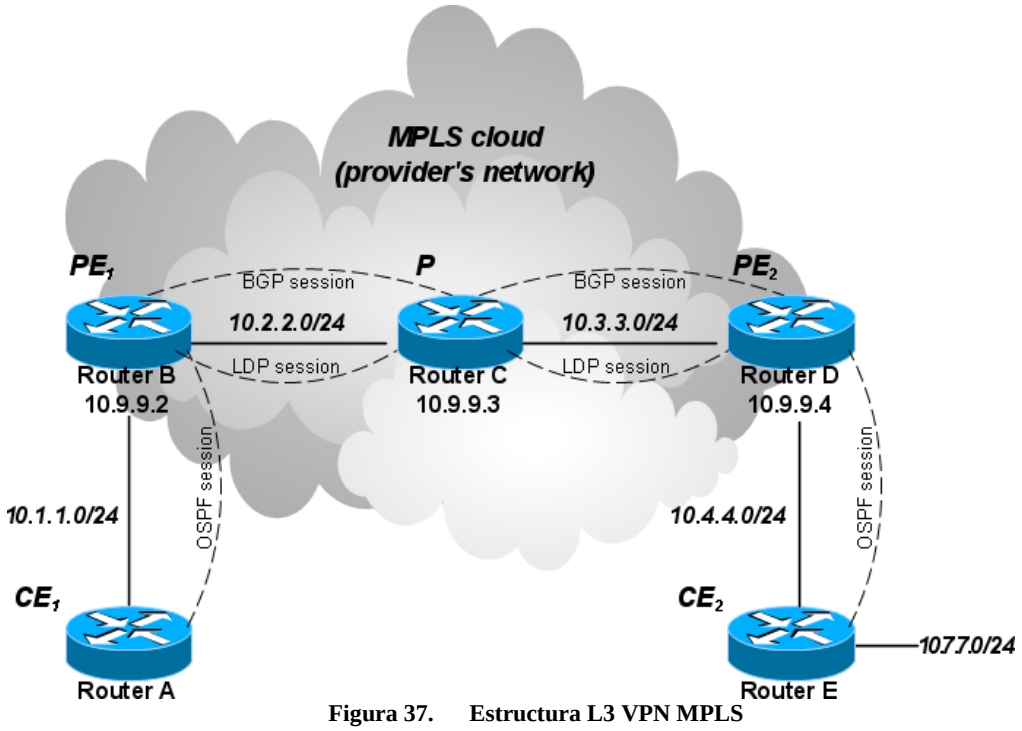

Figura 37. Estructura L3 VPN MPLS

Debido a que tanto los routers CE como los PE interactúan a nivel 3, es necesario que hablen entre ellos un protocolo de routing dinámico (o rutas estáticas). El CE solo tiene un equipo conectado fuera de su ubicación, el PE. El CE no tiene conectividad física directa con ningún otro CE. El nombre de este modelo se llama peer-to-peer ya que el CE y el PE tienen una conexión de nivel 3.

Una VPN debe ser privada, por ello los clientes pueden tener su propio plan de direccionamiento, puede usar, tanto direccionamiento público como privado e incluso se puede repetir direccionamiento entre clientes. Si los paquetes fuesen reenviados como paquetes IP en los nodos P habría un problema de routing, y si no se les permitiese a los clientes tener su propio direccionamiento, este debería ser asignado por el proveedor de servicios.

Suponiendo esto, los paquetes podrían ser reenviados atendiendo a su dirección IP destino en cada router de la red del proveedor. Esto significa que tanto los nodos P como los nodos PE deberían tener una tabla de rutas completa con el direccionamiento de cada cliente y esa tabla podría ser muy grande. El único protocolo de routing capaz de manejar semejante tabla es BGP, por lo que tanto nodos P como nodos PE deberían hablar BGP entre ellos. Llegados a este caso no sería un esquema válido debido a que no es un entorno privado para cada cliente.

Otra solución sería que tanto LSR's P como PE manejasen tablas de rutas distintas para cada cliente. Debería haber tantos procesos de routing como VPN's de cliente hubiera configuradas en la red.

Esta no es una solución muy escalable ya que cada vez que un nuevo cliente se diese de alta en la red habría que configurar en cada nodo (tanto P, como PE) un proceso de routing. Además, al entrar un paquete a la red a través de un PE, ¿Cómo se podría identificar a que VPN pertenece?. La solución pasaría por modificar el paquete IP añadiéndole un campo de identificación de VPN. Entonces los nodos P deberían mirar además del campo IP destino el campo de VPN para reenviar adecuadamente el paquete.

Una solución escalable es que los routers P no tuviesen consciencia de VPN's lo que les liberaría de la carga de tener información de las rutas para cada VPN. Precisamente esto es la solución que ofrece MPLS. Los paquetes IP de cada cliente son etiquetados en la red MPLS para conseguir una VPN privada para cada cliente. Además, los routers P no necesitan conocer la tabla de rutas gracias a la utilización de dos etiquetas MPLS. Por lo tanto, BGP no es necesario en los routers P. Las rutas para cada VPN sólo se manejan en los nodos PE al igual que solo hay concepto de VPN en los PE's lo que hace que las VPN MPLS sean una solución escalable.

4.3 Arquitectura L3 VPN

4.3.1 Virtual Routing Forwarding (VRF)

Una VRF es una instancia de enrutamiento y reenvío en la VPN. Es el nombre que recibe la combinación de la tabla de routing de la VPN, la CEF de la VRF y los protocolos de routing IP asociados en el router PE. Un nodo PE tiene una instancia de VRF para cada VPN asociada.

En la figura 38, podemos ver como un nodo PE tiene su tabla de rutas global IP y también una tabla de routing VRF por cada VPN conectada al PE.

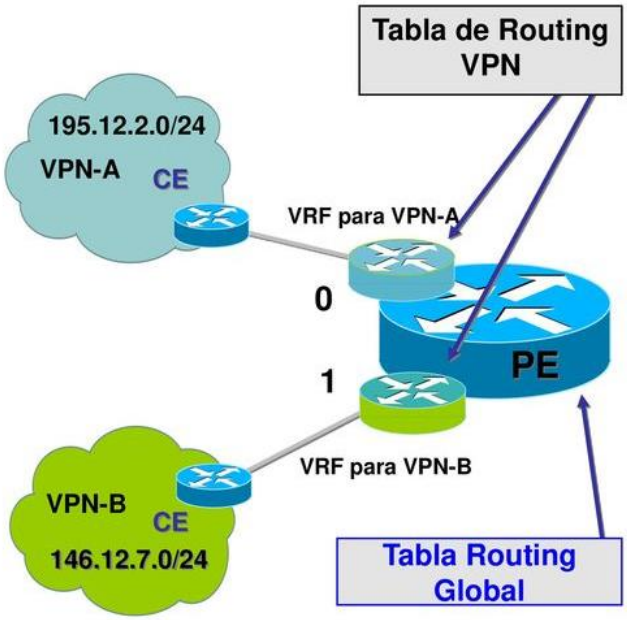

Figura 38. VRFs en nodo PE

Como la tabla de rutas debe estar separada y ser privada para cada cliente dentro de un nodo PE, cada VPN debe tener su propia tabla de rutas. Esta tabla de rutas privada se llama tabla de rutas VRF. El interfaz del PE que conecta con el CE puede pertenecer solo a una VRF por lo que todos los paquetes recibidos en la interfaz de esa VRF se identifican inequívocamente como pertenecientes a esa VRF.

4.3.2 Route Distingisher (RD)

Permite la superposición de direcciones entre clientes.

Los prefijos de la VPN se propagan a través de la VPN sobre MPLS mediante Multiprotocol-BGP. El problema es que cuando BGP transporte estos prefijos sobre la red deben ser únicos y si los clientes tienen direccionamiento IP solapado el routing podría ser erróneo.

Para solucionar este problema se crea el concepto de RD que convierte los prefijos IP en únicos. Cada prefijo de cada cliente recibe un identificador único RD para distinguir el mismo prefijo de distintos clientes. El prefijo deriva de la combinación del prefijo IP y del RD y se llama prefijo VPNv4. El MP-BGP transporta los prefijos VPNv4 entre los routers PE.

El RD es un campo de 64 bits pero no indica a que VRF pertenece el prefijo. La función del RD no es ser un identificador de VPN ya que algunos escenarios de VPN's más complejos pueden requerir más de un RD por VPN. Cada instancia de VRF en un nodo PE debe tener un RD asignado.

El valor del campo del RD puede tener dos formatos: ***ASN:nn*** o ***DirecciónIP:nn*** donde *nn* representa un número. El formato más usado comúnmente es *ASN:nn* donde *ASN* es el número de sistema autónomo asignado por IANA al proveedor de servicio y nn es el número que el proveedor de servicio asigna unívocamente a la VRF. El RD no impone semántica y se usa solamente para identificar de manera única las rutas de la VPN. La combinación del RD y el prefijo IP proporciona un prefijo VPNv4 de 96 bits de longitud como se aprecia en la figura 39.

Figura 39. VPNv4

4.3.3 Route Target (RT)

Si sólo se usara el RD para identificar la VPN, la comunicación entre sedes de distintas VPN's sería problemática y a veces esto es necesario, p.e cuando dos clientes necesitan acceder a un mismo recurso (DMZ, servidor, segmento de red, etc.…).

Una sede de un cliente A no podría comunicarse con una sede de un cliente B porque los RD's no coincidirían. El concepto de sedes de distintos clientes con comunicación entre si se llaman extranet VPN. El caso más sencillo de comunicación entre sedes de un mismo cliente (de la misma VPN) se conoce como intranet VPN. La comunicación entre sedes se controla mediante otra funcionalidad de la VPN MPLS llamada Route Target (RT).

Un RT es una comunidad extendida de BGP que indica que rutas deben ser importadas de MP-BGP a la VRF. Exportar un RT significa que a cada ruta VPNv4 exportada se le añade una comunidad BGP extendida (esto es el RT), cuando esta ruta se redistribuye de la tabla de rutas VRF al MP-BGP. Importar un RT significa que para cada ruta VPNv4 recibida de MP-BGP se comprueba si su comunidad extendida (RT) coincide con alguna de las asociadas a alguna VRF. Si coincide el prefijo se incluye en la tabla de rutas VRF como una ruta IP. Si no coincide el prefijo es rechazado.

La figura 40 muestra como los RT's controlan que rutas se importan en cada VRF desde los PE's remotos y con que RT's se exportan los prefijos VPNv4 hacia los PE's remotos.

Más de un RT puede ser asociado a un prefijo VPNv4. Para que la importación hacia la VRF se permita, sólo es necesario que un RT del prefijo VPNv4 coincida con alguno de los RT's importados en esa VRF.

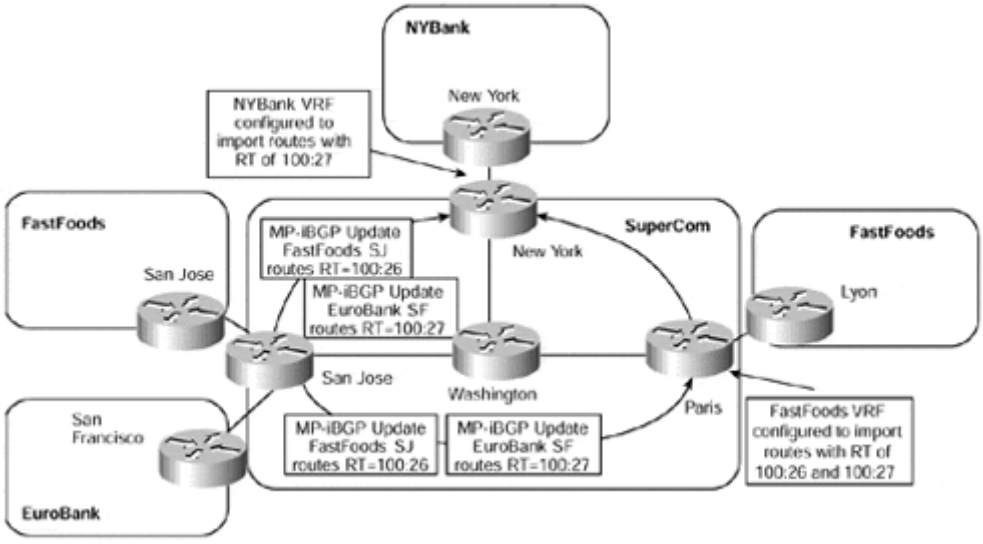

Figura 40. Funcionamiento de los RT's

En la figura 40 por ejemplo, el router del core "SuperCom" situado en San José exporta las rutas para la VPN de FastFood con el RT 100:26 y para la VPN de EuroBank con el RT 100:27. La VRF de NYBank ubicada en el router del core de New York importa el RT 100:27, esto implica que sólo contendrá rutas hasta la sede de EuroBank en San Francisco.

La VRF de FastFoods en el router del core ubicado en París importa los RT's 100:26 y 100:27, por lo que tendrá acceso a la sede de FastFood de San José y la sede de San Francisco del EuroBank.

4.3.4 Propagación de rutas VPNv4

Las VRF's separan las rutas de cliente en los nodos PE, pero absolutamente todos los prefijos son transportados a través de la red MPLS. Potencialmente pueden ser cientos de miles de rutas ya que pueden ser numerosas las VPN's de cliente configuradas. Para este transporte de rutas, BGP es el protocolo ideal ya que está muy testado y es sumamente estable para el manejo de grandes tablas de rutas, por eso es el protocolo estandarizado para internet. Gracias a la transformación de prefijos IP en prefijos VPNv4 (RD + prefijo IP), todas las rutas se pueden transportar de manera segura a través de la red.

El nodo PE recibe rutas IP desde el CE mediante un IGP o mediante eBGP (external BGP). Estas rutas IP de una VPN determinada se insertan en una tabla de rutas VRF. Esta VRF depende de la que esté configurada sobre el interfaz del PE que conecta con el CE que envía las rutas. Estas rutas IP se convierten en rutas VPNv4 una vez que los prefijos IP se añaden al RD correspondiente, es entonces cuando entran en el proceso de MP-BGP. BGP se encarga de distribuir estas rutas VPNv4 hacia todos los PE's en esa VPN.

El que la ruta VPNv4, después de separarse del RD, sea añadida a la tabla VRF como ruta IP o no depende de si los RT's permiten la importación a esta VRF. Esas rutas IP son entonces anunciadas al router CE mediante eBGP u otro protocolo de routing.

Para comprender todos estos procesos, en la figura 41 se pueden observar los pasos que se establecen para que se produzca la comunicación IP entre dos CE's a través de una VPN MPLS.

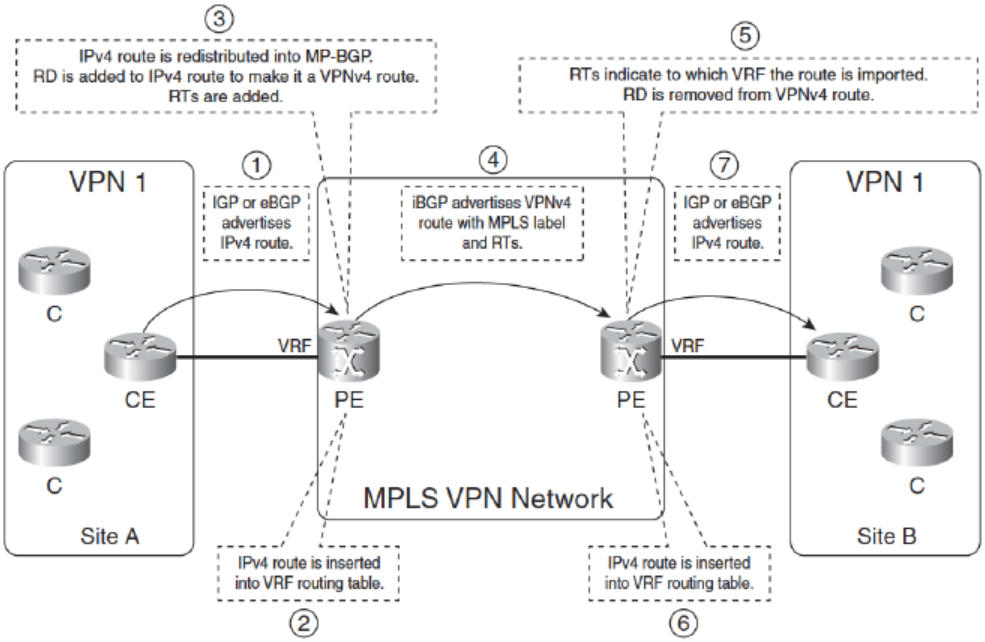

Figura 41. Propagación de rutas en una VPN MPLS

4.3.5 Envío de paquetes en VPN MPLS

Los paquetes no pueden ser reenviados como paquetes puros IP entre dos sedes. Los routers P no pueden reenviarlos porque no tienen información alguna de VRF's. MPLS soluciona este problema etiquetando los paquetes. La manera más habitual es hacerlo con LDP entre todos los routers P y PE así todo el tráfico IP es reenviado basado en etiquetas.

También se puede usar RSVP con extensiones para Ingeniería de Tráfico, pero LDP es lo más común.

Los paquetes IP son reenviados basándose en etiquetas desde el Ingress PE hasta el Egress PE. Un nodo P nunca tiene que tiene que consultar la cabecera IP. Esta es la manera en que los paquetes se conmutan entre el Ingress PE y el Egress PE. Esta etiqueta se llama etiqueta *IGP*, ya que es la etiqueta que se asocia a un prefijo IP en la tabla de routing global de los routers P y PE y es anunciada por el IGP.

El tráfico de VPN a VPN tiene dos etiquetas en una red VPN MPLS. La etiqueta externa es la etiqueta IGP y es distribuida mediante LDP o RSVP entre todos los routers P y PE. La etiqueta interna es la etiqueta *VPN* que es anunciada por MP-BGP de PE a PE. Los routers P consultan la etiqueta IGP para reenviar los paquetes hacia el nodo PE correcto. Los Egress PE usan la etiqueta de VPN para reenviar el paquete al CE correcto.

En la figura 42 podemos ver como es el reenvío de paquetes en una red VPN MPLS. Los paquetes entran en el router PE en la VRF asociada al interfaz como un paquete IP. Es reenviado a través de la red VPN MPLS con dos etiquetas. Los routers P reenvían el paquete mirando la etiqueta externa. Esta etiqueta externa es intercambiada en cada nodo P. Las etiquetas son eliminadas en el Egress PE y el paquete es enviado como un paquete IP sobre el interfaz que corresponda a la VRF en el CE.

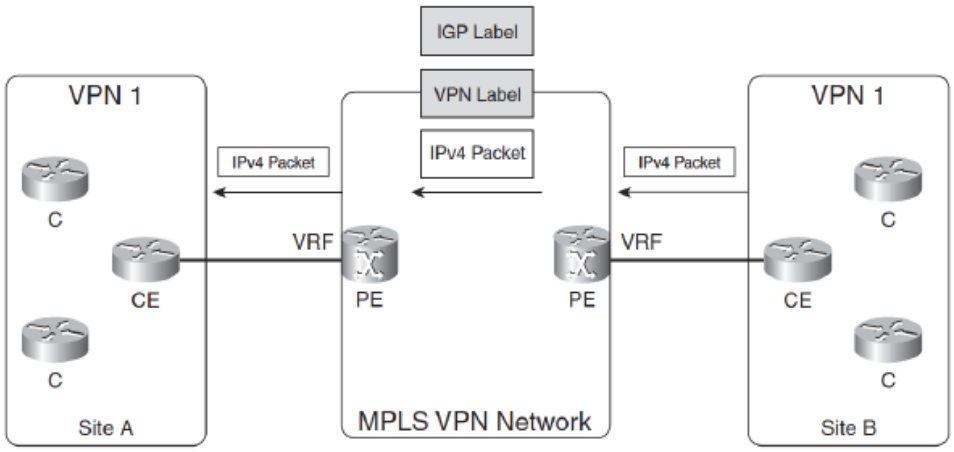

Figura 42. Envío de paquetes en una red VPN MPLS

4.4 Arquitectura L2 VPN

En los últimos años otra aplicación basada en la tecnología MPLS ha emergido, teniendo una gran acogida por los clientes y los ISP's, las MPLS/VPN de Capa 2.

Estas redes tienen la naturaleza de ser multiprotocolo, es decir, pueden transportar tanto tráfico IP como tráfico no IP, gran parte de las especificaciones del IETF sobre como transportar el tráfico de Capa 2 (Ethernet, Frame Relay, ATM, HDLC, PPP) a través de una red MPLS, están ya descritas. Esto da la oportunidad a los ISP's de con la misma infraestructura MPLS, transportar los tráficos asociados a servicios tradicionales, en especial el Frame Relay en nuestro país.

Relacionado con las VPLS (Virtual Private LAN Services), hay dos borradores destacados en los Grupos de Trabajo del IETF, "draft Kompella" y "draft Lasserre-VKompella". A pesar de que las especificaciones mencionadas todavía no están estandarizadas, varios fabricantes de equipos ya soportan el borrador Martini del IETF (IETF Martini drafts). Los borradores Martini definen los mecanismos de encapsular y distribuir etiquetas para transportar Frame Relay, ATM, Ethernet, HDLC, PPP, sobre una red MPLS.

Cisco como líder en el mercado de los fabricantes de infraestructura de redes, para transportar tramas de Capa 2 sobre un backbone IP/MPLS propone una solución que denomina AToM (Any Transport over MPLS). Esta solución de Cisco habilita a los ISP's, a proveer conectividad de Capa 2 entre los sitios de los clientes utilizando una misma infraestructura de red basada en paquetes IP/MPLS. Los ISP's pueden realizar las conexiones tradicionales Frame Relay, ATM y

las conexiones Ethernet sobre un backbone IP/MPLS. AToM soporta los siguientes tipos de transporte: ATM AAL5 sobre MPLS, ATM Cell Relay sobre MPLS, Ethernet sobre MPLS, Frame Relay sobre MPLS, PPP sobre MPLS y HDLC sobre MPLS.

4.5 Objetivos de la práctica

El objetivo de la presente práctica es familiarizarse con la tecnología y los conceptos de VPN's sobre MPLS, así como su configuración en una red implementada con routers Cisco Systems.

Para ello, se deberán realizar las siguientes actividades:

- introducir en los routers los comandos necesarios para configurar una L3 VPN.
- verificar el correcto funcionamiento de la VPN establecida en la red.
- visualizar los diferentes paquetes que circulan por la red e identificar los campos pertenecientes a los diferentes protocolos utilizados en VPN MPLS.

4.6 Materiales a utilizar

Para la realización de la presente práctica se utilizarán los routers CISCO 1841 disponibles en el laboratorio. En la figura 43 podemos ver una imagen de la trasera del mencionado router y en la tabla 9 la descripción de cada uno de los elementos presentes en la misma.

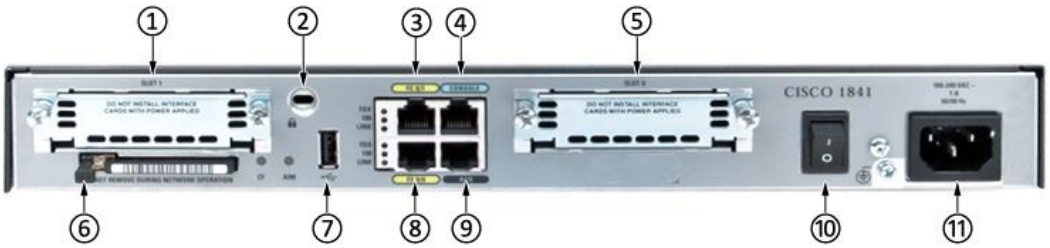

Figura 43. Trasera router CISCO 1841

ID	DESCRIPCIÓN
1	Slot de expansión 1
2	Accesorio para bloqueo
3	Puerto Fast Ethernet 0/0
4	Puerto de consola
5	Slot de expansión 0
6	Unidad compact flash
7	Puerto USB
8	Puerto Fast Ethernet 0/1
9	Puerto auxiliar
10	Interruptor de encendido
11	Entrada de alimentación

Tabla 9. Identificación elementos trasera CISCO 1841

Además de los citados routers, será necesaria una tarjeta expansora de 4 puertos Ethernet por cada router, la HWIC-4ESW, también disponible en el laboratorio. La tarjeta se instalará en el Slot 0 antes de conectar el router a la corriente.

La numeración de los interfaces de la HWIC-4ESW se inicia desde la derecha, por lo que el primer interfaz de la derecha será el FE 0/0/0 y el último el FE 0/0/3, tal y como podemos ver en la figura 44.

Figura 44. Tarjeta expansora Ethernet HWIC-4ESW

Aparte de los routers, cables de alimentación y latiguillos Fast Ethernet cruzados, será necesaria la utilización de tres PC's del laboratorio, en uno de ellos deberá estar instalado el analizador de redes "*Wireshark*" para realizar capturas de tráfico circulante por la red.

4.7 Diagrama de la topología de red

En la figura 45 se observa la topología de la red a montar.

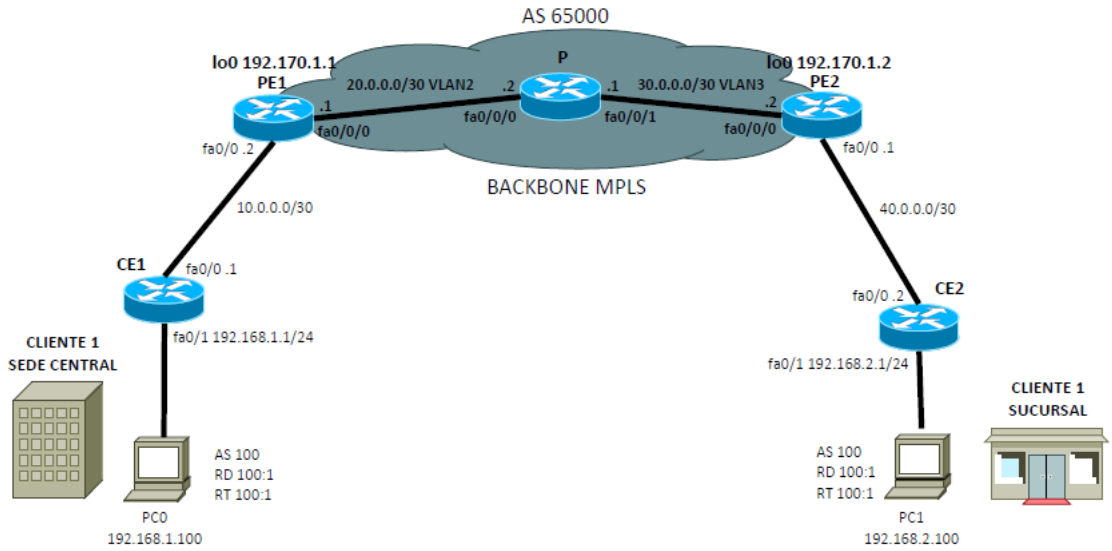

Figura 45. Diagrama de la red a configurar

DISPOSITIVO	INTERFAZ	DIRECCIÓN IP	MÁSCARA DE RED	GATEWAY	VLAN
CE1	Fa0/0	10.0.0.1	255.255.255.252	-	-
	Fa0/1	192.168.1.1	255.255.255.0	-	-
PE1	Fa0/0	10.0.0.2	255.255.255.252	-	-
	Lo0	192.170.1.1	255.255.255.255	-	-
	Fa0/0/0	20.0.0.1	255.255.255.252	-	VLAN 2
P	Fa0/0/0	20.0.0.2	255.255.255.252	-	VLAN 2
	Fa0/0/1	30.0.0.1	255.255.255.252	-	VLAN 3
PE2	Fa0/0	40.0.0.1	255.255.255.252	-	-
	Lo0	192.170.1.2	255.255.255.255	-	-
	Fa0/0/0	30.0.0.2	255.255.255.252	-	VLAN 3
CE2	Fa0/0	40.0.0.1	255.255.255.252	-	-
	Fa0/1	192.168.2.1	255.255.255.0	-	-
PC0	NIC	192.168.1.100	255.255.255.0	192.168.1.1	-
PC1	NIC	192.168.2.100	255.255.255.0	192.168.2.1	-

Tabla 10. Tabla de direccionamiento

Como podemos apreciar en la anterior figura, vamos a establecer una VPN en una empresa que dispone de unas oficinas centrales y una sucursal ubicada en otra ciudad. En la central se dispone de un router que hace de Gateway, el CE1. De manera análoga tenemos a CE2 en el espacio de la sucursal.

Para establecer la VPN y enlazar los PE's con los CE's, se utilizará el protocolo eBGP, que nos permitirá intercambiar la información de routing entre los diferentes sistemas autónomos, en la frontera formada por los interfaces Fa0/0 de ambos routers PE, donde se crean las tablas VRF, que actúan como un router lógico, permitiendo definir caminos virtuales entre la sede central y la sucursal remota. Así pues, se crearán dos VRF's, a las que llamaremos Cliente1.

En primer lugar, comenzaremos asignando las direcciones a los interfaces del backbone, en el caso del router PE1, los comandos a introducir serían:

```
Router#configure terminal
Router(config)#hostname PE1
PE1(config)#line console 0
PE1(config-line)#logging synchronous
PE1(config-line)#interface Loopback0
PE1(config-if)#ip address 192.170.1.1 255.255.255.255
PE1(config-if)#interface FastEthernet0/0
PE1(config-if)#ip address 10.0.0.2 255.255.255.252
PE1(config-if)#no shutdown
PE1(config-if)#interface FastEthernet0/0/0
PE1(config-if)#switchport access vlan 2
PE1(config-if)#switchport mode trunk
PE1(config-if)#interface Vlan 2
PE1(config-if)#ip address 20.0.0.1 255.255.255.252
PE1(config-if)#exit
```

```
PE1(config)#exit
```

Completar el proceso con el resto de routers del backbone MPLS y continuar hasta configurar todos los routers que conforman nuestra red.

Continuemos configurando el protocolo IGP dentro del bakbone MPLS, en este caso y como venimos haciendo en las anteriores prácticas, utilizaremos OSPF, para el caso del router P los comandos a introducir serán:

```
P(config)#router ospf 1
P(config-router)#network 20.0.0.0 0.0.0.3 area 0
P(config-router)#network 30.0.0.0 0.0.0.3 area 0
P(config-router)#exit
P(config)#
```

Siguiendo con la configuración de OSPF, en el caso del PE2, éste se configuraría de la siguiente forma:

```
PE2(config)#router ospf 1
PE2(config-router)#network 30.0.0.0 0.0.0.3 area 0
PE2(config-router)#network 192.170.1.2 0.0.0.0 area 0
PE2(config-router)#exit
PE2(config)#
```

Repetir estos mismos pasos para el caso de PE1.

Como ya hicimos en anteriores prácticas, es necesario habilitar ip cef, mpls ip, de forma general y en cada interfaz perteneciente al backbone MPLS, es decir, en los routers PE1, P y PE2, además del protocolo de distribución de etiquetas. Utilizaremos los siguientes comandos:

```
PE1#configure terminal
PE1(config)#ip cef
PE1(config)#mpls ip
PE1(config)#mpls label protocol ldp
PE1(config)#interface Fast Ethernet 0/0/0
PE1(config-if)#mpls ip
PE1(config-if)#interface Vlan 2
PE1(config-if)#mpls ip
PE1(config-if)#exit
PE1(config)#
```

Repetir los anteriores pasos en las interfaces pertenecientes a la red MPLS en los routers P y PE2.

Seguiremos con la VPN propiamente dicha, vamos a habilitar el routing y el forwarding en la misma, para ello crearemos una tabla VRF a la que denominaremos Cliente1, será necesario ejecutar los siguientes comandos en ambos routers PE:

```
PE1(config)#ip vrf Cliente1
PE1(config-vrf)#rd 100:1
PE1(config-vrf)#route-target import 100:1
PE1(config-vrf)#route-target export 100:1
PE1(config-vrf)#exit
PE1(config)#exit
```

```
PE2(config)#ip vrf Cliente1
PE2(config-vrf)#rd 100:1
PE2(config-vrf)#route-target import 100:1
PE2(config-vrf)#route-target export 100:1
PE2(config-vrf)#exit
PE2(config)#exit
```

En primer lugar se define el RD, como ya sabemos de la teoría de la asignatura, el RD identifica las rutas VPN y se representa como ASN:nn (ASN, Autonomous System Number), en este caso hemos establecido que el sistema autónomo sea el 100 y lo identificamos como 1, el RD debe de ser único y diferente para cada VPN de cliente. De esta forma podemos diferenciar entre distintos clientes, aunque los mismos utilicen rangos de direcciones IP superpuestos.

Seguidamente definimos los RT, los RT nos indican las rutas de la VRF que se distribuirán hacia el otro PE a través de la red MPLS. Dichas rutas se intercambiarán mediante MP-BGP. En este caso sólo importartemos y exportaremos la ruta hacia el único sistema autónomo que hemos definido, el 100.

Una vez definidas las VRF, es necesario asignarlas a una interfaz, en el caso de nuestra red, estos interfaces serán los fa0/0 de ambos routers PE, la asignación se realiza de la siguiente manera:

```
PE1(config)#interface FastEthernet0/0
PE1(config-if)#ip vrf forwarding Cliente1
PE1(config-if)#ip address 10.0.0.2 255.255.255.252
PE1(config-if)#exit
PE1(config)#
```

Observar que al asignar la vrf a la interfaz, la dirección ip del mismo es eliminada, por lo que es necesario volver a asignar dicha IP. Repetir el mismo proceso para el router PE2.

Con el comando **show ip route vrf Cliente1**, podemos comprobar que dicha interfaz se ha añadido a la tabla de enrutamiento de la vrf, como podemos se puede observar a continuación, desapareciendo de la tabla de routing global.

```
PE1#sh ip route vrf Cliente1

Routing Table: Cliente1
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/30 is subnetted, 1 subnets
C       10.0.0.0 is directly connected, FastEthernet0/0
PE1#
```

Podemos comprobar que la vrf se ha creado correctamente con el siguiente comando:

```
PE2#sh ip vrf

  Name                Default RD        Interfaces
  Cliente1            100:1             Fa0/0
PE1#
```

Como podemos comprobar en el router PE2 existe una vrf de nombre Cliente1, asignada al interfaz Fa0/0 con RD 100:1, tal y como hemos configurado previamente.

Una vez definidas las VRF, estableceremos la sesión BGP entre los routers PE:

```
PE1(config)#router bgp 65000
PE1(config-router)#neighbor 192.170.1.2 remote-as 65000
PE1(config-router)#neighbor 192.170.1.2 update-source loopback 0
PE1(config-router)#exit
PE1(config)#
```

En primer lugar, con el comando **router bgp *sistema-autónomo***, se habilita bgp en el router, en el caso que nos ocupa hemos utilizado el AS 65000 para el backbone MPLS.

Un **Sistema Autónomo**, AS, es un grupo de redes IP que poseen una política de rutas propia e independiente, es decir, realiza su propia gestión del tráfico que fluye entre él y los restantes Sistemas Autónomos que la red.

Una vez habilitado BGP en el router, es necesario habilitar su peer con el que intercambiará información, esto se realiza con el comando **neighbor**, en el caso de PE1, su vecino es PE2, el cual se identifica mediante su dirección de loopback. Además, es necesario indicar el AS al que pertenece. Por último, se indica que se tome como origen para las actualizaciones de BGP el interfaz loopback0.

Los siguientes comandos ilustran el proceso de habilitación de BGP en el PE2.

```
PE2(config)#router bgp 65000
PE2(config-router)#neighbor 192.170.1.1 remote-as 65000
PE2(config-router)#neighbor 192.170.1.1 update-source loopback 0
PE2(config-router)#exit
PE2(config)#
```

Seguidamente para habilitar MP-BGP tendremos que activar la familia vpnv4 dentro del proceso bgp, para ello deberemos introducir los siguientes comandos:

```
PE2(config)#router bgp 65000
PE2(config-router)#address-family vpnv4
PE2(config-router-af)#neighbor 192.170.1.1 activate
PE2(config-router-af)#neighbor 192.170.1.1 send-community extended
PE2(config-router-af)# exit-address-family
PE2(config-router)#
```

Repetir los mismos pasos en el router PE1.

Para comprobar que el proceso se ha realizado de forma correcta y tenemos comunicación con nuestro peer vpnv4, podemos utlizar el comando **show bgp vpnv4 all summary**, el resultado obtenido debería de ser similar al siguiente:

```
PE1#sh ip bgp vpnv4 all summary
BGP router identifier 192.170.1.1, local AS number 65000
BGP table version is 4, main routing table version 4
1 network entries using 274 bytes of memory
1 path entries using 136 bytes of memory
2/1 BGP path/bestpath attribute entries using 496 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
1 BGP extended community entries using 24 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 954 total bytes of memory
BGP activity 1/0 prefixes, 1/0 paths, scan interval 15 secs

Neighbor        V  AS      MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down   State/PfxRcd
192.170.1.2     4  65000   66      66       4       0    0     01:01:23  1
```

El siguiente paso es definir la familia ipv4, en dicha familia se declaran los vecinos con los que se van a intercambiar rutas ipv4, en nuestro caso estos serán los router CE1 y CE2, dependiendo de que router PE vayamos a configurar.

Vamos a definir la familia ipv4 en el router PE2, de forma análoga repetir el proceso en el router PE1. BGP para prevenir bucles, por defecto no aprende rutas que provengan del mismo AS, por lo que en un principio no aprendería las rutas provenientes de la central al tener el mismo AS, con el comando **as-override** podemos evitar esto.

Este comando sustituye el número de AS de las rutas provenientes de 40.0.0.2 por el suyo propio, en este caso sustituiría el 100 por 65000, por lo que PE1 podrá aprender las rutas provenientes de CE2.

```
PE2(config)#router bgp 65000
PE2(config-router)#address-family ipv4 vrf Cliente1
PE2(config-router-af)#neighbor 40.0.0.2 remote-as 100
PE2(config-router-af)#neighbor 40.0.0.2 activate
PE2(config-router-af)#neighbor 40.0.0.2 as override
PE2(config-router-af)#exit-address-family
PE2(config-router)#
```

Sigamos ahora con la configuración de los routers de proveedor, donde tendremos que configurar el protocolo eBGP (external BGP) para poder establecer la comunicación con estos. Aprovecharemos para comunicar la dirección de la red del cliente dentro del mismo proceso BGP, evitándonos así configurar OSPF en el cliente y posteriormente redistribuir esta ruta hacia BGP.

```
PE1(config)#router bgp 100
PE1(config-router)#neighbor 10.0.0.2 remote-as 65000
PE1(config-router)#network 192.168.1.0 mask 255.255.255.0
PE1(config-router)#exit
PE1(config)#
```

Repetir los anteriores comandos para configurar el router PE2.

Para comprobar que la anterior ruta ha sido aprendida por el router mediante bgp, utilizar el comando **show ip bgp vpnv4 vrf Cliente1**, si se ha configurado correctamente, el resultado obtenido debería ser similar al siguiente:

```
PE1#show ip bgp vpnv4 vrf Cliente1
BGP table version is 4, local router ID is 192.170.1.1
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: 100:1 (default for vrf Cliente1)
*> 192.168.1.0      10.0.0.1            0               0       100 i
*>i192.168.2.0      192.170.1.2         0       100     0       100 i
PE1#
```

La i que podemos ver delante de la ruta hacia la sucursal, significa que esta ha sido aprendida mediante iBGP (internal BGP).

En estos instantes hemos finalizado la configuración de la VPN, con lo que los equipos de la Central deberían de poder comunicarse con los de la sucursal remota y viceversa a través de la misma.

Por último, vamos a utilizar la función SPAN de las HWIC como ya hicimos en las prácticas 1 y 2. Haremos un mirroring del puerto FastEthernet0/0/0 del P sobre el puerto FastEthernet0/0/2, donde conectaremos el PC con el analizador de protocolos. Para ellos introducir los siguientes comandos:

```
P#configure terminal
P(config)#monitor session 1 source interface FastEthernet0/0/0
P(config)#monitor session 1 destination interface FastEthernet0/0/2
P(config)#end
P(config)#
```

4.8 Ejercicios propuestos

Conecta el equipo con el analizador de protocolos "*Wireshark*" al interfaz FastEthernet0/0/2 del router P, para capturar el tráfico circulante por la red.

Para generar tráfico en la red vamos a utilizar el comando **ping**. Abriremos una consola en el PC0, desde el menú "*Inicio→Ejecutar*", teclear **cmd** y presionar "*Enter*".

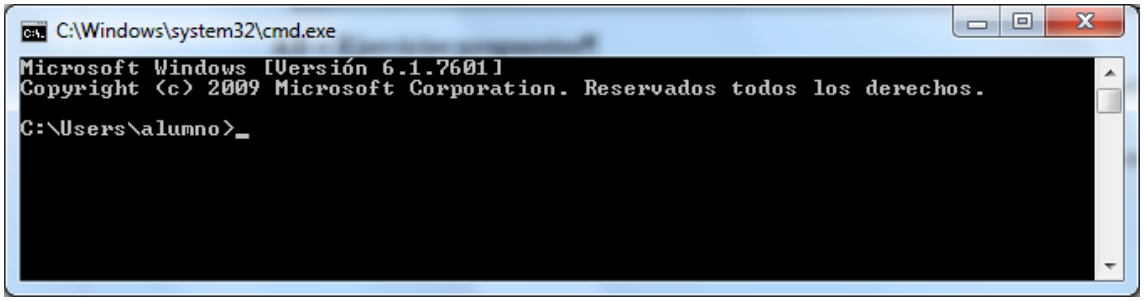

Figura 46. Consola de comandos, windows

1. **Haz un ping desde PC0 hasta PC1, observar la captura que obtenemos en "Wireshark". Analiza los campos de los paquetes capturados, ¿cuál es la principal diferencia respecto de los paquetes que capturamos en la primera práctica?.**

Como podemos ver en la figura 47 el paquete tiene los mismos campos que un paquete MPLS normal, la única diferencia es que esta vez tiene dos etiquetas MPLS, una para la ruta vpnv4 y otra para la ruta OSPF.

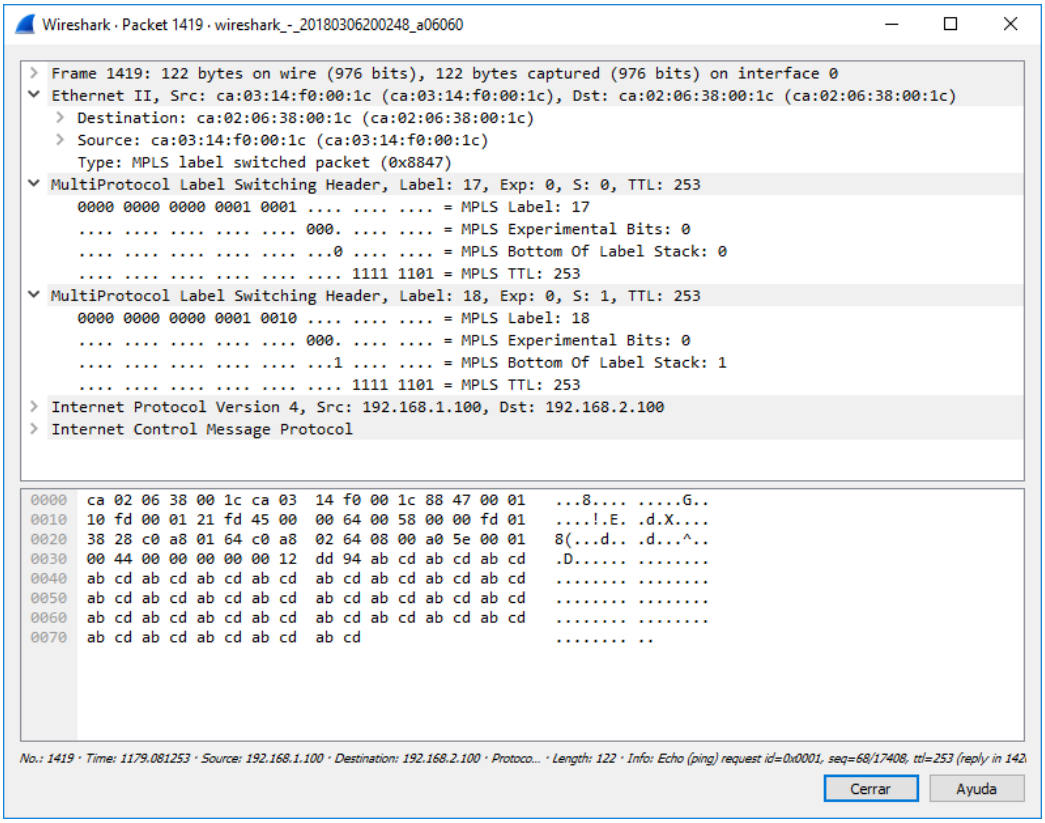

Figura 47. Captura de paquete MPLS en red VPN

2. **¿Puedes identificar para que se utilizan las etiquetas que llevan los paquetes?**

La cabecera MPLS que se corresponde con la etiqueta 18, tiene el bit S de la cabecera MPLS a 1, lo que implica que es la última etiqueta, la utilizada por el router P para encaminar los paquetes según OSPF.

La etiqueta 17 por tanto, es la utilizada en la ruta VPN.

Seguidamente vamos a analizar cómo se envían los atributos de la VPN mediante vpnv4. Hemos visto que para esto se utiliza MP-BGP, dichos paquetes son enviados al inicio de la sesión BGP entre ambos routers PE. Vamos a provocar un reinicio de la sesión BGP, lo que provocará que se pierda la conexión TCP que establece BGP entre ambos PE, para ello realizaremos un **reload** del router P, antes copiar la configuración existente en el router a la ram, mediante el comando **wr**.

```
P#wr
Building configuration...
[OK]
P#reload
Proceed with reload? [confirm]
```

3. ¿Qué paquetes tipo de paquetes BGP puedes identificar aparte de los KeepAlive?. ¿Puedes relacionar alguno con los conceptos visto en la teoría de la asignatura?.

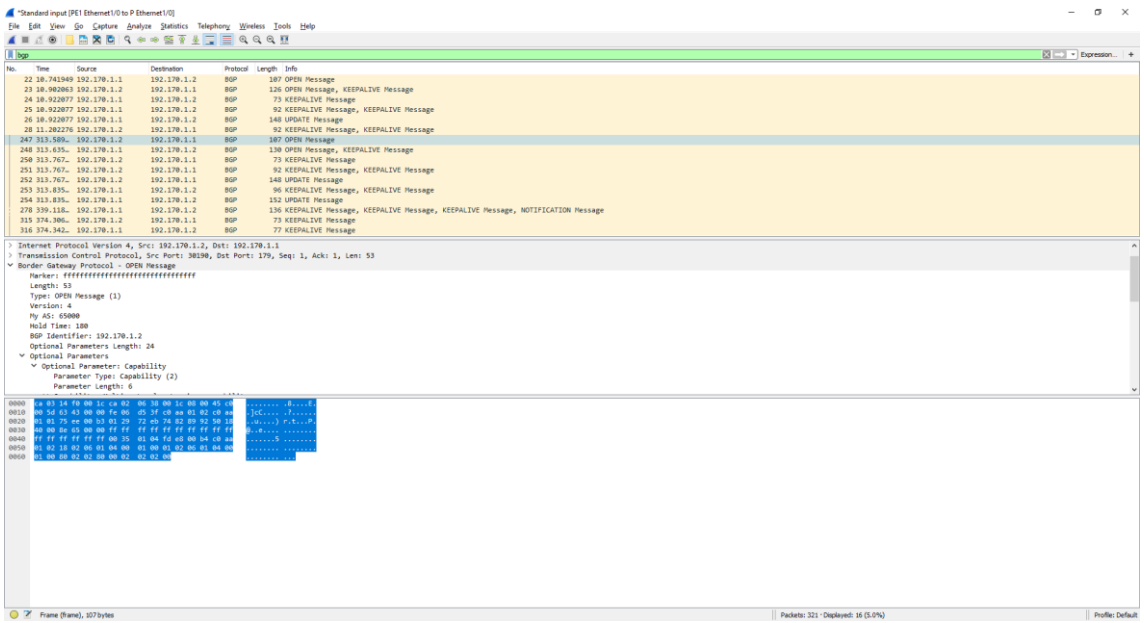

Figura 48. Paquetes BGP capturados al inición de la sesión

Identificamos dos tipos de paquetes, los OPEN Messagge y los UPDATE Message. En los primeros, ver figura 49, se transmiten los parámetros generales de la conexión, como son: la versión de BGP utilizada, el número del AS local, el identificador BGP, etc.

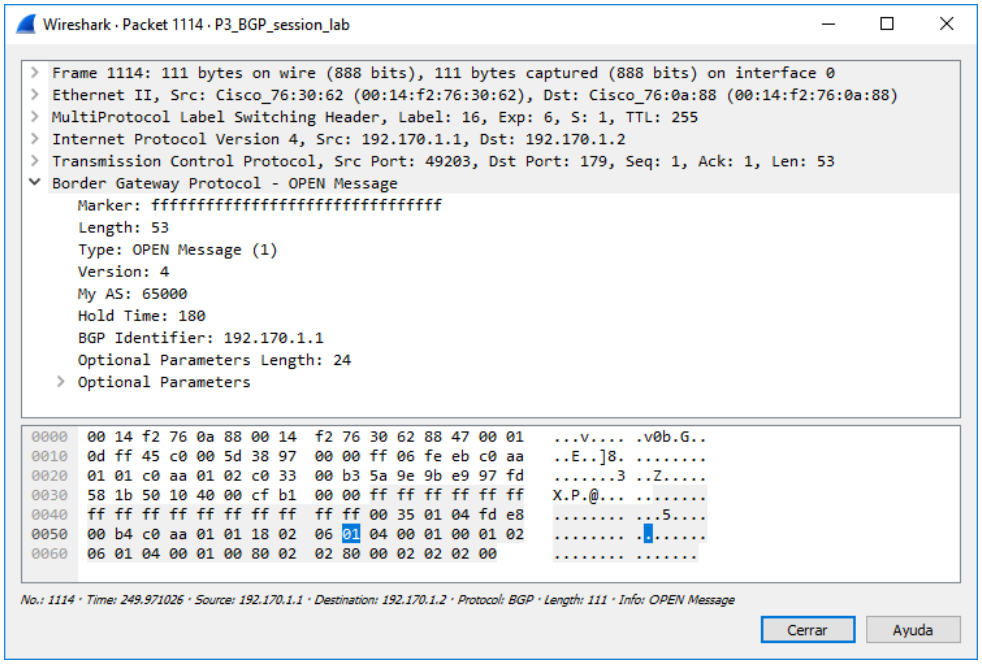

Figura 49. Captura paquete OPEN Message BGP

En los UPDATE Message se envía la información de la red, rutas y sus atributos, entre los que se encuentran los de la VPN.

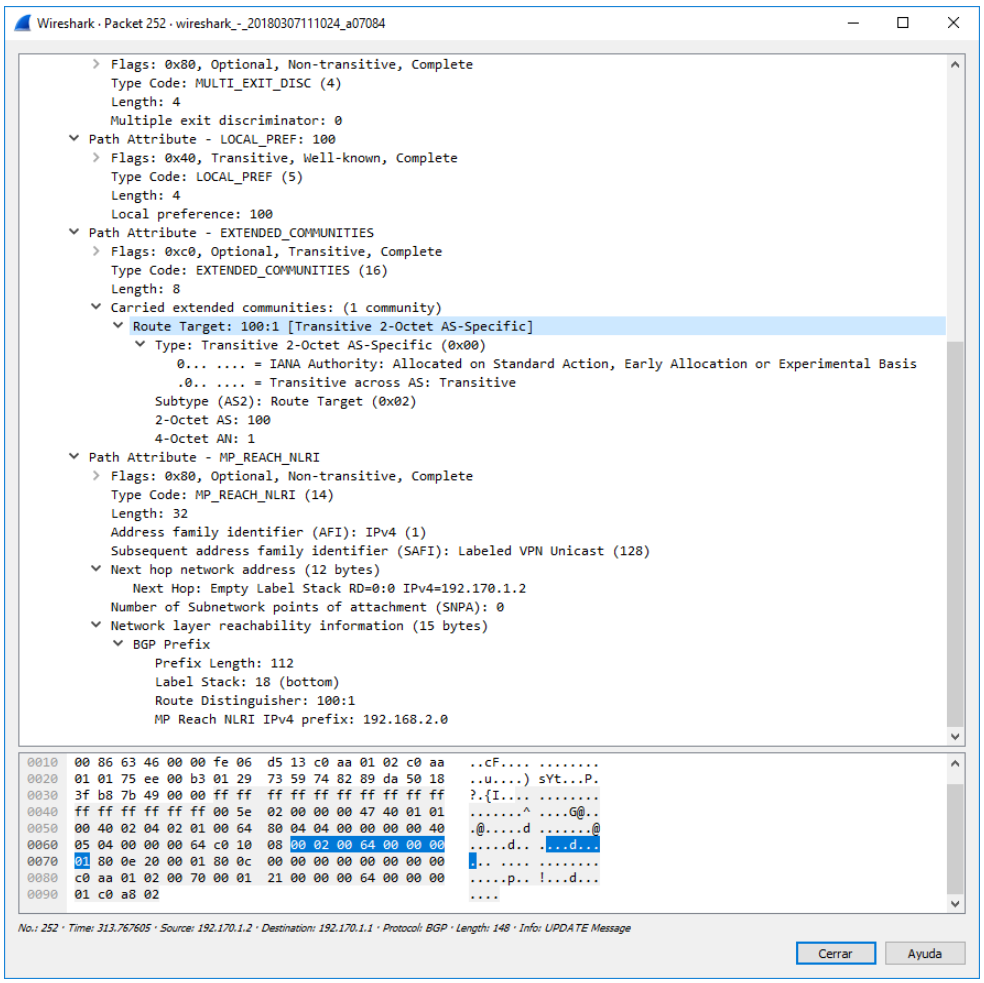

Figura 50. Captura paquete UPDATE Message BGP

Como se puede apreciar en la figura 50, en estos paquetes es donde se envía al peer PE la información de los RT, RD y AS de los clientes de la VPN, así como otros muchos parámetros, como el espacio de direccionamiento de la red del cliente, etc.

4. ¿Qué cambios serían necesarios realizar a nivel de configuración y topología para poder crear una VPN y dar servicio a otro cliente adicional?.

A nivel de topología deberíamos de conectar los routers de cliente en ambos router PE.

Seguidamente, deberíamos de crear otra vrf y definir los RD's y los RT's a importar y exportar, estos deben de ser diferentes a los del otro cliente, ya que el espacio de direccionamiento utilizado podría ser el mismo.

A continuación, deberíamos de asignar a las interfaces del router PE conectadas a los routers de cliente la vrf creada y modificar la familia ipv4 del proceso bgp de ambos routers PE para incluir al nuevo cliente.

Será necesario añadir al router PE la nueva red que conecta con el router del cliente, en nuestro caso al proceso OSPF.

Por último, tendremos que comunicar la nueva red al router PE, desde el router de cliente, mediante el protocolo de routing que elijamos, en el desarrollo de la práctica hemos utilizado BGP.

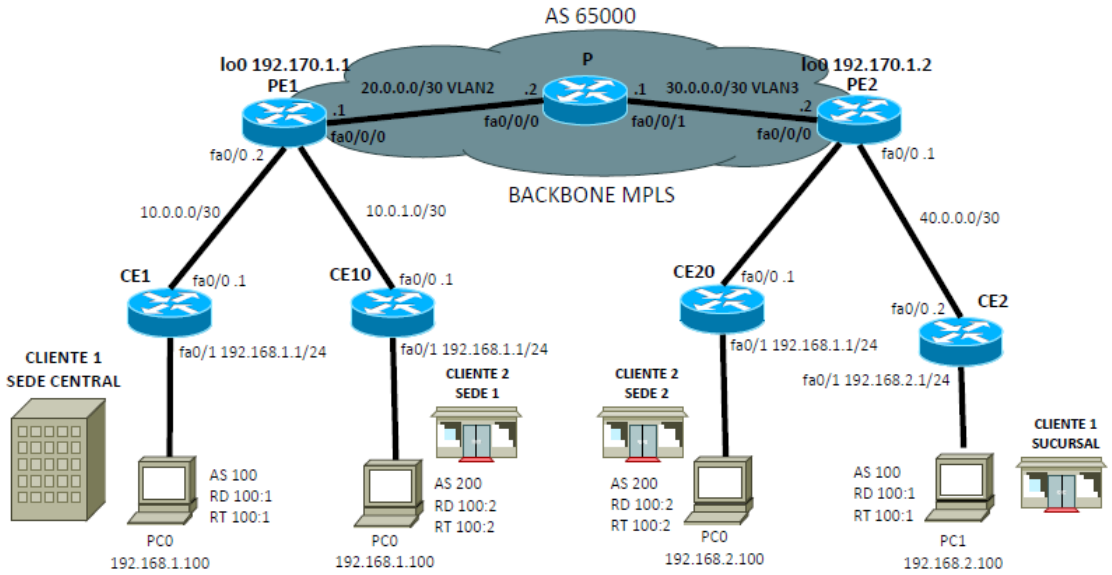

Figura 51. Configuración VPN con dos clientes

5. ¿Podría la sede central del cliente 1 comunicarse con la sede 1 del cliente 2?

Teóricamente sí, se debería de crear un RT diferente que ambos importarían y exportarían, si bien tendrían un problema de direccionamiento, ya que comparte e mismo espacio de direcciones.

4.9 ANEXO 1 “Ficheros running-config de los routers”

CE1

```
!
version 15.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname CE1
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
!
!
dot11 syslog
ip source-route
!
!
!
!
ip cef
no ip domain lookup
no ipv6 cef
!
multilink bundle-name authenticated
!
!
!
!
!
license udi pid CISCO1841 sn FCZ0935105N
!
redundancy
!
!
!
!
!
!
!
!
!
interface FastEthernet0/0
 ip address 10.0.0.1 255.255.255.252
 duplex auto
 speed auto
 !
!
interface FastEthernet0/1
```

```
 ip address 192.168.1.1 255.255.255.0
 duplex auto
 speed auto
 !
!
!
interface Vlan1
 no ip address
 !
!
router bgp 100
 no synchronization
 bgp log-neighbor-changes
 network 192.168.1.0
 neighbor 10.0.0.2 remote-as 65000
 no auto-summary
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
!
!
!
!
!
!
!
!
control-plane
 !
!
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
end
```

CE2

```
!
version 15.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname CE2
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
!
!
dot11 syslog
ip source-route
!
!
!
!
ip cef
no ip domain lookup
no ipv6 cef
!
multilink bundle-name authenticated
!
!
!
!
!
license udi pid CISCO1841 sn FHK104118Q6
!
redundancy
!
!
!
!
!
!
!
!
!
interface FastEthernet0/0
 ip address 40.0.0.2 255.255.255.252
 duplex auto
 speed auto
 !
!
interface FastEthernet0/1
 ip address 192.168.2.1 255.255.255.0
 duplex auto
```

```
 speed auto
 !
!
interface FastEthernet0/0/0
 !
!
!
interface Vlan1
 no ip address
 !
!
router bgp 100
 no synchronization
 bgp log-neighbor-changes
 network 192.168.2.0
 neighbor 40.0.0.1 remote-as 65000
 no auto-summary
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
!
!
!
!
!
!
!
control-plane
 !
!
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
end#
```

PE1

```
!
version 15.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE1
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
!
!
dot11 syslog
ip source-route
!
!
!
!
ip vrf Cliente1
 description Sede Central Cliente 1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
ip cef
no ip domain lookup
no ipv6 cef
!
multilink bundle-name authenticated
!
mpls label protocol ldp
!
!
!
!
license udi pid CISCO1841 sn FCZ0935105S
!
redundancy
!
!
!
!
!
!
!
!
!
!
interface Loopback0
 ip address 192.170.1.1 255.255.255.255
```

```
 !
!
interface FastEthernet0/0
 ip vrf forwarding Cliente1
 ip address 10.0.0.2 255.255.255.252
 duplex auto
 speed auto
 !
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
 !
!
interface FastEthernet0/0/0
 switchport access vlan 2
 switchport mode trunk
 mpls ip
 !
!
interface FastEthernet0/0/1
 shutdown
 !
!
interface FastEthernet0/0/2
 shutdown
 !
!
interface FastEthernet0/0/3
 shutdown
 !
!
interface Vlan1
 no ip address
 !
!
interface Vlan2
 ip address 20.0.0.1 255.255.255.252
 mpls ip
 !
!
router ospf 1
 log-adjacency-changes
 network 20.0.0.0 0.0.0.3 area 0
 network 192.170.1.1 0.0.0.0 area 0
!
router bgp 65000
 no synchronization
 bgp log-neighbor-changes
 neighbor 192.170.1.2 remote-as 65000
 neighbor 192.170.1.2 update-source Loopback0
 no auto-summary
 !
```

```
 address-family vpnv4
  neighbor 192.170.1.2 activate
  neighbor 192.170.1.2 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf Cliente1
  no synchronization
  neighbor 10.0.0.1 remote-as 100
  neighbor 10.0.0.1 activate
  neighbor 10.0.0.1 as-override
 exit-address-family
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
!
!
!
!
!
!
!
control-plane
 !
!
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
end
```

PE2

```
!
version 15.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE2
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
!
!
dot11 syslog
ip source-route
!
!
!
!
ip vrf Cliente1
 description Sucursal Cliente 1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
ip cef
no ip domain lookup
no ipv6 cef
!
multilink bundle-name authenticated
!
mpls label protocol ldp
!
!
!
!
license udi pid CISCO1841 sn FCZ1040113S
!
redundancy
!
!
!
!
!
!
!
!
!
!
interface Loopback0
 ip address 192.170.1.2 255.255.255.255
```

```
 !
!
interface FastEthernet0/0
 ip vrf forwarding Cliente1
 ip address 40.0.0.1 255.255.255.252
 duplex auto
 speed auto
 !
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
 !
!
interface FastEthernet0/0/0
 switchport access vlan 3
 switchport mode trunk
 mpls ip
 !
!
interface FastEthernet0/0/1
 shutdown
 !
!
interface FastEthernet0/0/2
 shutdown
 !
!
interface FastEthernet0/0/3
 shutdown
 !
!
interface Vlan1
 no ip address
 !
!
interface Vlan3
 ip address 30.0.0.2 255.255.255.252
 mpls ip
 !
!
router ospf 1
 log-adjacency-changes
 network 30.0.0.0 0.0.0.3 area 0
 network 192.170.1.2 0.0.0.0 area 0
!
router bgp 65000
 bgp log-neighbor-changes
 neighbor 192.170.1.1 remote-as 65000
 neighbor 192.170.1.1 update-source Loopback0
 !
 address-family ipv4
  no synchronization
```

```
 neighbor 192.170.1.1 activate
 no auto-summary
exit-address-family
!
address-family vpnv4
 neighbor 192.170.1.1 activate
 neighbor 192.170.1.1 send-community extended
exit-address-family
!
address-family ipv4 vrf Cliente1
 no synchronization
 neighbor 40.0.0.2 remote-as 100
 neighbor 40.0.0.2 activate
 neighbor 40.0.0.2 as-override
exit-address-family
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
!
!
!
!
!
!
!
control-plane
 !
!
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
end
```

P

```
!
version 15.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
monitor session 1 source interface Fa0/0/0
monitor session 1 destination interface Fa0/0/2
!
!
!
dot11 syslog
ip source-route
!
!
!
!
ip cef
no ip domain lookup
no ipv6 cef
!
multilink bundle-name authenticated
!
mpls label protocol ldp
!
!
!
!
license udi pid CISCO1841 sn FCZ0935106L
!
redundancy
!
!
!
!
!
!
!
!
!
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
```

```
 !
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
 !
!
interface FastEthernet0/0/0
 switchport access vlan 2
 switchport mode trunk
 mpls ip
 !
!
interface FastEthernet0/0/1
 switchport access vlan 3
 switchport mode trunk
 mpls ip
 !
!
interface FastEthernet0/0/2
 !
!
interface FastEthernet0/0/3
 shutdown
 !
!
interface Vlan1
 no ip address
 !
!
interface Vlan2
 ip address 20.0.0.2 255.255.255.252
 mpls ip
 !
!
interface Vlan3
 ip address 30.0.0.1 255.255.255.252
 mpls ip
 !
!
router ospf 1
 log-adjacency-changes
 network 20.0.0.2 0.0.0.0 area 0
 network 30.0.0.1 0.0.0.0 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
!
!
!
```

```
!
!
!
!
control-plane
 !
!
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
end
```

Capítulo 5. Bibliografía

[1] Nam-Kee Tan ,“MPLS for Metropolitan Area Networks”, Auerback, 2004.

[2] “*Cisco IOS Multiprotocol Label Switching Configuration Guide*”, Release 12.2 SR, Cisco Systems Inc., 2010.

[3] Pepelnjak, I.; Guichard, J., “M*PLS and VPN Architectures”,* Cisco Press, 2002.

[4] “*Cisco 1800 Series Software Configuration Guide*”, Cisco Systems Inc., 2004.