UNIVERSITAT POLITÈCNICA DE VALÈNCIA

_ TELECOM ESCUELA TÉCNICA VLC SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería de Telecomunicación

DESARROLLO Y CONFIGURACIÓN DE UN ENTORNO MPLS EN EQUIPOS MIKROTIK Y SIMULADO EN GNS3

Trabajo Fin de Grado

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

AUTOR/A: Carpio Ortiz, David

Tutor/a: Sempere Paya, Víctor Miguel

CURSO ACADÉMICO: 2023/2024

Resumen

En este trabajo se va a montar una red de equipos MikroTik, concretamente del modelo CRS326-24G-2S+RM. Su objetivo es enfocar de manera práctica los conocimientos de redes MPLS obtenidos de manera teórica en la asignatura Redes Públicas de Transporte.

Las prácticas que se han elaborado profundizan en los diferentes conceptos de MPLS, Etiquetado LDP de Paquetes, Redes Privadas Virtuales (VPN) y Servicio de Redes Privadas Virtuales (VPLS).

Esta red MPLS también se podrá implementar en el simulador GNS3 y se desarrollará de igual forma todo el proceso.

Resum

En aquest treball es muntarà una xarxa d'equips Mikrotik, concretament del model CRS326-24G-2S+RM. El seu objectiu és enfocar de manera pràctica els coneixements de xarxes MPLS obtinguts de manera teòrica a l'assignatura Xarxes Públiques de Transport.

Les pràctiques que s'han elaborat aprofundeixen els diferents conceptes de MPLS, Etiquetatge LDP de Paquets, Xarxes Privades Virtuals (VPN) i Servei de Xarxes Privades Virtuals (VPLS).

Aquesta xarxa MPLS també es podrà implementar al simulador GNS3 i es desenvoluparà de la mateixa manera tot el procés.

Abstract

In this proyect, a network of Mikrotik equipment, specifically the CRS326-24G-2S+RM model, is going to be set up. Its objective is to focus in a practical way the knowledge of MPLS networks obtained in a theoretical way in the subject "Redes Públicas de Transporte".

The practices that have been elaborated deepen in the different concepts of MPLS, Virtual Private Network (VPN) and Virtual Private Network Service (VPLS).

This MPLS network can also be implemented in the simulator GNS3 and the whole process will be developed in the same way.

Índice

Índice de figuras

Índice de tablas

1. Objetivos del trabajo

El objetivo principal de este trabajo es el desarrollo de redes MPLS tanto en un entorno físico en el laboratorio como en un entorno simulado en GNS3. El trabajo se dividirá en tres prácticas orientadas a la parte del protocolo MPLS de la asignatura "Redes Públicas de Transporte".

Estas prácticas serán realizadas con los routers MikroTik CRS326-24G-2S+RM disponibles en el laboratorio de "Redes Telemáticas". Con esto se pretende profundizar en los contenidos impartidos en las clases de teoría, centrándose así en las diferentes aplicaciones que tiene una red MPLS.

Estas mismas prácticas podrán realizarse en el simulador GNS3, para que los alumnos puedan desarrollar las redes de una forma más avanzada, detallada y visual, aprovechando las herramientas que proporciona dicho simulador.

Los objetivos específicos que se detallan en el trabajo son:

- Diseñar y configurar redes básicas MPLS con RouterOS de MikroTik.
- Estudiar la arquitectura y comprobar el funcionamiento de MPLS.
- Establecer y estudiar el protocolo LDP en Mikrotik con diversas pruebas.
- Establecer y estudiar el protocolo BGP.
- Diseñar y configurar redes MPLS VPN de nivel 3 en MikroTik.
- Diseñar y configurar redes MPLS VPLS de nivel 2 en MikroTik.
- Optimizar los recursos disponibles de WireShark para comprender los protocolos que intervienen en la red MPLS.
- Configurar y comprender el funcionamiento del simulador GNS3 y de la máquina virtual VMware Workstation Player.

2. Práctica 0: "Configuración en RouterOS de MikroTik"

2.1. Introducción

MikroTik es una empresa letona que desarrolla software y hardware para soluciones de redes. Su producto principal es el sistema operativo MikroTik RouterOS, que se utiliza en enrutadores y dispositivos de redes para ofrecer una amplia gama de funcionalidades y servicios.

- Enrutamiento
- Firewall
- VPN
- Hostpot
- Administración centralizada
- Balanceo de carga
- Wireless

MikroTik ofrece una amplia gama de productos de hardware, desde enrutadores de nivel empresarial hasta dispositivos más pequeños y económicos.

En estas prácticas se hará uso de los Routers MikroTik CRS326-24G-2S+RM como ya se ha ido comentando a lo largo del documento. La razón de ello son las múltiples ventajas que nos proporcionan estos equipos frente a los de CISCO.

- Equipos mucho más económicos.
- Funcionan todos sus puertos en capa 3 (L3 – IP).
- MikroTik RouterOS es altamente personalizable y ofrece una amplia gama de características y funcionalidades.
- Tiene una curva de aprendizaje más suave en comparación con Cisco. La interfaz gráfica de usuario (GUI), Winbox, es intuitiva y fácil de usar, lo que facilita la configuración y administración de dispositivos.

2.2. MikroTik RouterOS

MikroTik RouterOS es un sistema operativo de enrutador desarrollado por MikroTik en 1997 para la gestión y configuración de su propio Hardware conocido como RouterBOARD.

MikroTik RouterOS se basa en el núcleo Linux y está diseñado para brindar un alto rendimiento, estabilidad y seguridad en entornos de red. Ofrece una amplia gama de características y funcionalidades que permiten a los administradores de redes configurar, administrar y controlar sus redes de manera efectiva.

2.2.1. Winbox

Winbox es la interfaz gráfica de usuario (GUI) proporcionada por MikroTik para administrar y configurar los dispositivos de red que utilizaremos en las siguientes prácticas. Es una herramienta popular y ampliamente utilizada por su facilidad de uso.

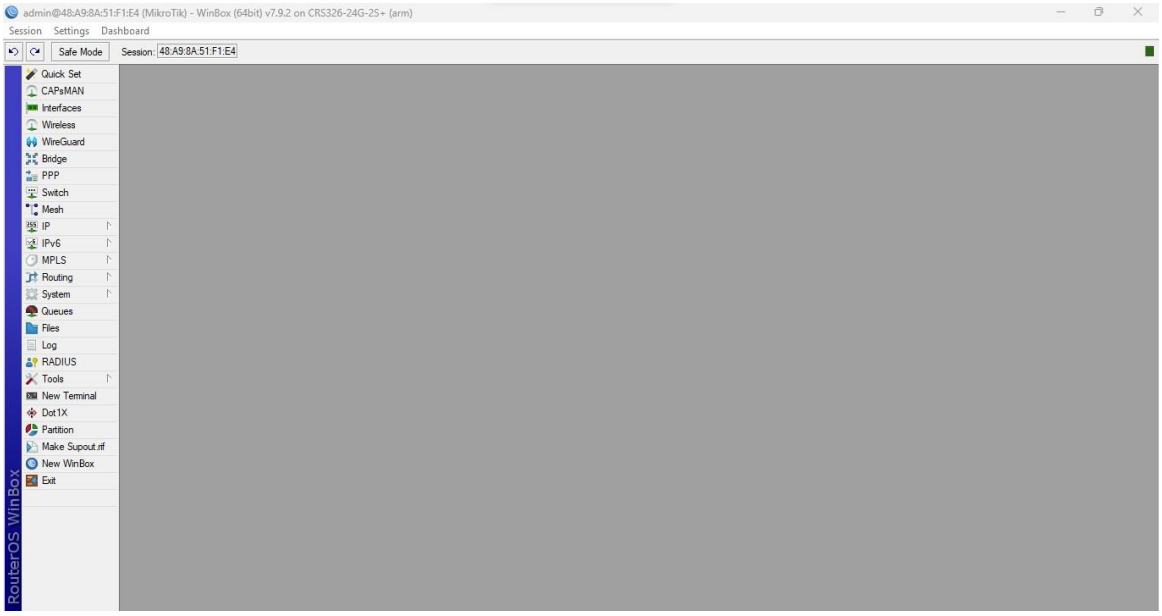


Fig. 1. Interfaz Winbox.

Como se puede observar, Winbox es una interfaz gráfica que permite configurar de manera rápida y sencilla un equipo MikroTik sin necesidad de hacer uso del *Terminal*.

Cuando trabajemos con los equipos físicos del laboratorio, nosotros vamos a utilizar únicamente dicho *Terminal* para poder realizar cualquier configuración similar en entornos que no disponen de un GUI.

Fig. 2. Interfaz Winbox. Menú Terminal.

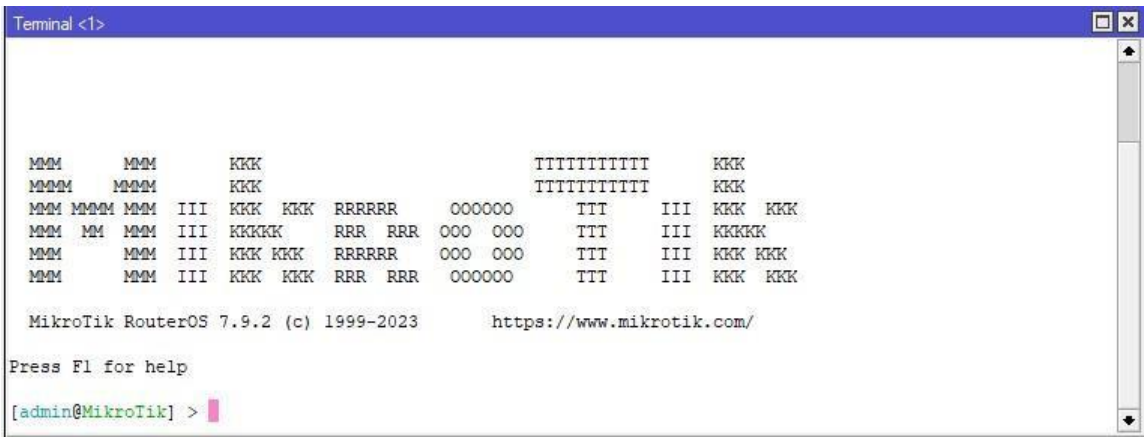


Fig. 3. Interfaz Winbox. Menú Terminal. Pestaña del Terminal.

En el entorno de simulación GNS3 utilizaremos la CLI (Comand Line Interface) incluida en el simulador. En este caso el cliente SSH original de Windows, PuTTY.

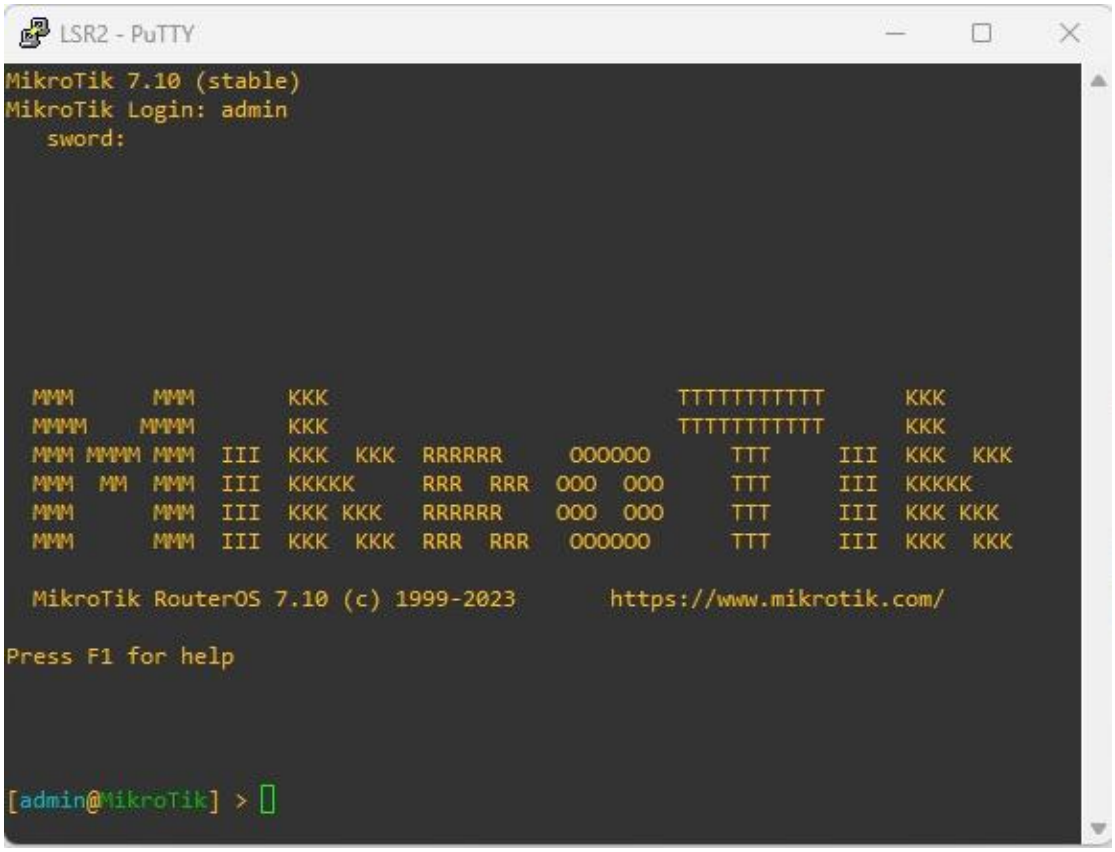


Fig. 4. Interfaz PuTTY.

Aunque GNS3 nos proporciona el terminal PuTTY de Windows, podemos utilizar la interfaz de Winbox en GNS3 mejorando la topología de red de forma que tengamos las interfaces de cada router a la interfaz de red de nuestro PC.

Para ello, necesitamos un elemento *Cloud*, que aunque solo tiene dos interfaces, podemos utilizar un *switch* para conectar nuestros 6 routers al *Cloud*.

En nuestro caso utilizaremos 2 *Switches* para mantener la topología más limpia visualmente. De forma que la topología quedará como se muestra a continuación.

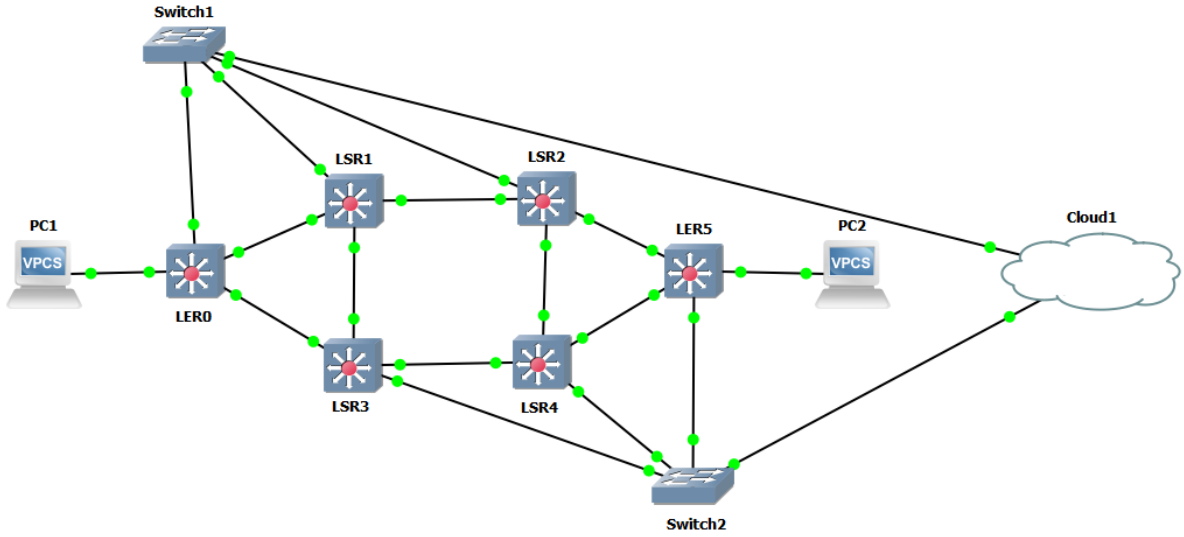


Fig. 5. Topología de red. Conexión Cloud para Winbox.

Para realizar la conexión, recomendamos utilizar las últimas interfaces *ethernet* de los routers, para evitar problemas ante posibles cambios de la topología. En este caso hemos utilizado la *ether24*.

Una vez se realiza la conexión y los routers están plenamente funcionado, podemos ejecutar Winbox para ver que los routers están accesibles. A continuación, se muestra lo comentado.

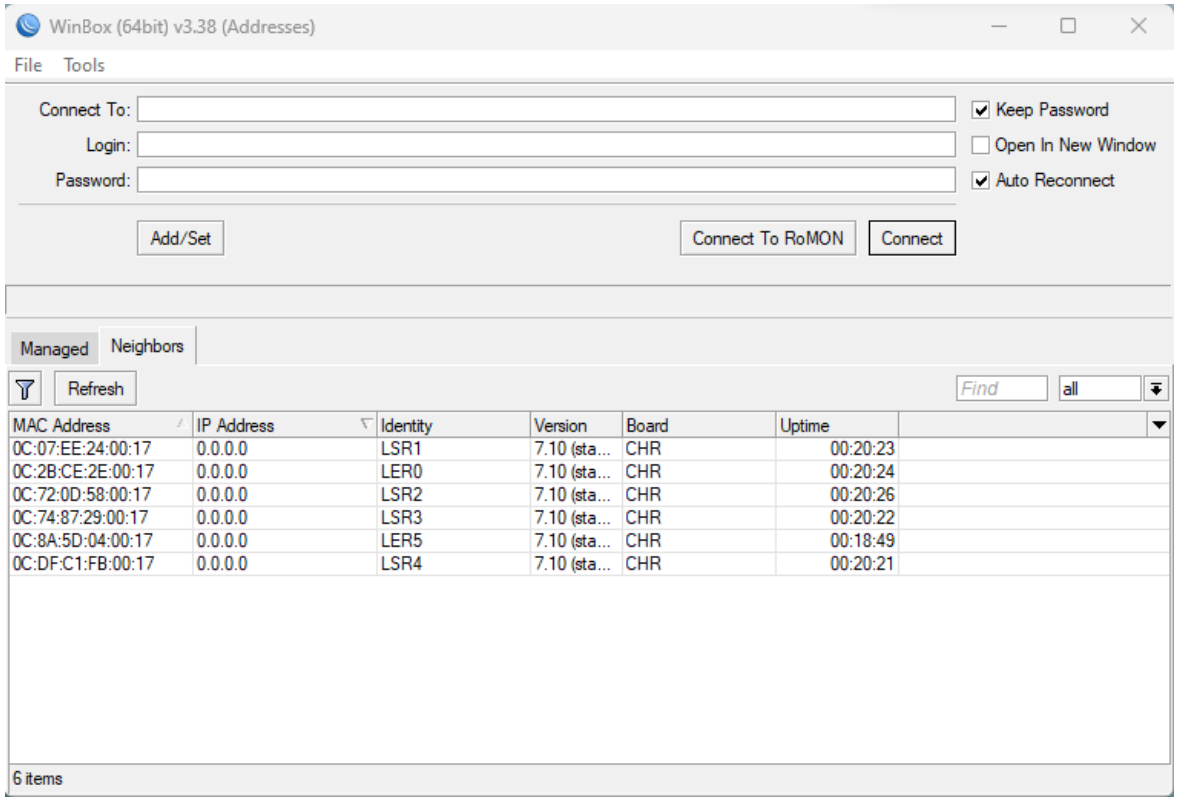


Fig. 6. Routers de GNS3 en Winbox.

2.2.2. Principales comandos

MikroTik RouterOS se basa en acceder a las diferentes pestañas o grupos que se pueden observar en Winbox y realizar en ellos las diferentes configuraciones de los parámetros.

Esta singularidad es heredada del entorno Linux, por lo tanto, si has trabajado con el terminal de este Sistema Operativo te resultará familiar la forma de proceder.

Algunos comandos y características se muestran a continuación:

- **/ :** la barra separadora seguida de los nombres de las pertinentes pestañas y subpestañas nos permite acceder a los grupos de configuración. Escribiendo únicamente la barra separadora podemos salir de todas las pestañas.
    - Ejemplo para acceder a configurar las direcciones ip's:

```
[admin@MikroTik] > /ip address
[admin@MikroTik] /ip/address >
```

- **.. :** si introducimos dos puntos seguidos podemos volver a la pestaña de un nivel superior e incluso acceder con la misma ejecución a una nueva subpestaña.
    - Ejemplo para salir de la pestalla ***address*** y acceder a la pestalla ***route*** también de ***ip***:

```
[admin@MikroTik] /ip/address > .. route
[admin@MikroTik] /ip/route >
```

- **add:** añadir nuevo valor en la pestaña en la que nos encontramos, por ejemplo, una dirección ip. Si solo introducimos ***add***, nos pedirán los valores necesarios, para una dirección ip nos pedirán la *dirección* y la *interfaz.*
- **set:** actualizar un valor de la entrada ya registrada, por ejemplo, podríamos actualizar la interfaz de una dirección ip. Al ejecutar el comando nos pedirán el número de la entrada sobre la que queremos aplicar el cambio.
- **remove:** eliminar una entrada ya registrada, por ejemplo, podríamos eliminar una dirección ip junto con su interfaz añadida por error.
- **\:** si a lo largo de las prácticas se muestra la contrabarra, esta indica que el comando que se está ejecutando no termina ahí y continúa en la siguiente línea.
- **export:** ejecutando este comando sobre el directorio raíz, se observará toda la configuración aplicada. Pero también se puede realizar sobre cualquier subgrupo de menús para observar únicamente la configuración aplicada sobre este.
- **print:** con este comando se observa la configuración aplicada sobre un submenú concreto, muy similar a lo que ocurre con *export.* Es más indicado para ver las diferentes entradas creadas antes de realizar un set o un *remove*. También permite mostrar ciertas tablas ejecutando el *print* una vez se ha accedido al directorio en cuestión que la contiene.
- **/system/identity:** desde este contexto es posible cambiar el nombre del router para que así se muestre en el terminal sobre el que ejecutaremos nuestros comandos.
    - **Ejemplo para LER0:**

```
[admin@MikroTik] > system/identity
[admin@MikroTik] /system/identity > set name=LER0
[admin@LER0] >
```

Al igual que ocurre con el sistema operativo de CISCO, podemos acortar los comandos hasta el punto de que el SO lo reconozca como una instrucción única. Además, haciendo uso del tabulador se completan los comandos o en su defecto se muestran las posibilidades de campos a completar.

Se muestra en la siguiente figura el caso para acceder a añadir una IP:

Fig. 7. Uso del tabulador como ayuda.

2.2.3. Control de errores

A partir de los comandos mencionados en el apartado anterior podemos llevar a cabo cualquier gestión de errores que se pueda llegar a producir a lo largo de las prácticas. Pero para una mejor comprensión se procede a continuación con un ejemplo simple.

Antes de centrarnos en los errores, se muestra el uso del comando *set* en el cambio de nombre del Router a través de *system/identity.*

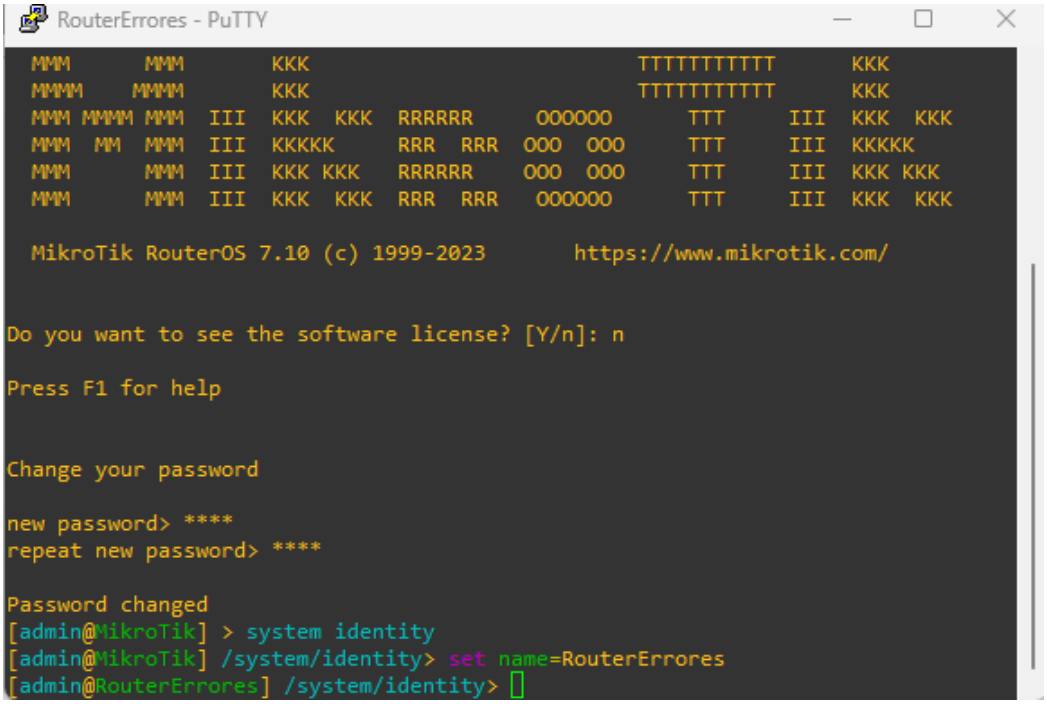


Fig. 8. Control de errores. Nombre del Router.

Ahora nos vamos a mover al directorio inicial con / y accederemos a *ip/address* para configurar una serie de IPs con sus respectivas interfaces físicas.

```
[admin@RouterErrores] /system/identity> /
[admin@RouterErrores] > ip
[admin@RouterErrores] /ip> address
[admin@RouterErrores] /ip/address> add
address: 10.0.0.1/30
interface: ether2
[admin@RouterErrores] /ip/address> add
address: 20.0.0.1/30
interface: ether3
[admin@RouterErrores] /ip/address> add
address: 30.0.0.1
interface: ether4
[admin@RouterErrores] /ip/address> add
address: 40.0.0.1/30
interface: ether2
[admin@RouterErrores] /ip/address>
```

Fig. 9. Control de errores. Configuración de IPs.

Con el comando *add* hemos introducido una serie de IPs con sus *interfaces*, pero como se puede observar, en una de ellas no hemos especificado la *máscara* deseada */30* y en otra de ellas hemos repetido la interfaz física.

Para solucionar estos errores, primero ejecutaremos un *print* para visualizar bien las diferentes entradas que tenemos y saber sobre cuáles de todas ellas debemos proceder con la edición o el borrado.

```
[admin@RouterErrores] /ip/address> print
Columns: ADDRESS, NETWORK, INTERFACE
# ADDRESS       NETWORK    INTERFACE
0 10.0.0.1/30   10.0.0.0   ether2
1 20.0.0.1/30   20.0.0.0   ether3
2 30.0.0.1/32   30.0.0.1   ether4
3 40.0.0.1/30   40.0.0.0   ether2
[admin@RouterErrores] /ip/address>
```

Fig. 10. Control de errores. print de direcciones IPs.

Vemos que las asignaciones se muestras en formato tabla siendo la primera columna un identificativo numérico seguido por la dirección IP, la subred a la que pertenece y la interfaz física.

Para solucionar el error con la máscara no es necesario borrar toda la asignación ya que es posible editar la dirección IP de la siguiente manera.

```
[admin@RouterErrores] /ip/address> print
Columns: ADDRESS, NETWORK, INTERFACE
# ADDRESS       NETWORK    INTERFACE
0 10.0.0.1/30   10.0.0.0   ether2
1 20.0.0.1/30   20.0.0.0   ether3
2 30.0.0.1/32   30.0.0.1   ether4
3 40.0.0.1/30   40.0.0.0   ether2
[admin@RouterErrores] /ip/address> edit
number: 2
value-name: address
```

Fig. 11. Control de errores. Edición de una IP (1).

Hemos ejecutado el comando *edit* que nos pide primero la entrada a editar y a continuación el nombre del valor a editar. Nosotros indicamos la entrada *2* y el valor *address*. Con esta información ejecutamos con *Enter* ya que para editar la dirección IP se nos abrirá sobre el mismo terminal una ventana para realizar dicha edición.

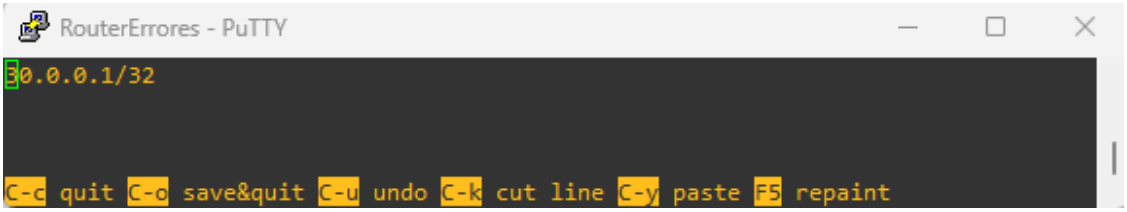


Fig. 12. Control de errores. Edición de una IP (2).

Se nos muestra la dirección IP sobre la que modificaremos la máscara. Nos desplazaremos con el cursor para la modificación y guardaremos haciendo uso del comando por teclado *Ctrl + o* como se indica en la línea inferior de ayuda.

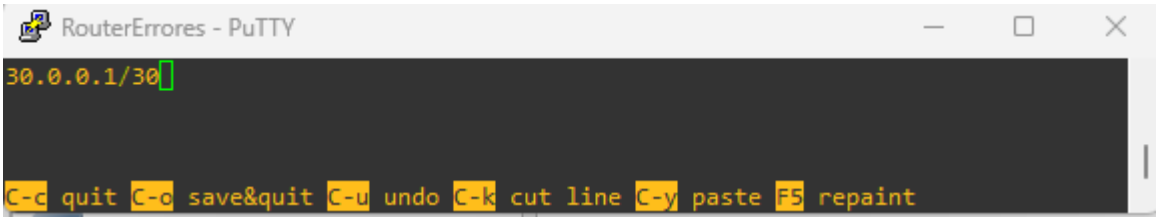


Fig. 13. Control de errores. Edición de una IP (3).

Una vez hemos guardado la modificación, se nos vuelve a mostrar el terminal como lo estábamos viendo con anterioridad. De tal forma que podemos ejecutar de nuevo *print* para ver que la modificación se ha llevado a cabo con éxito.

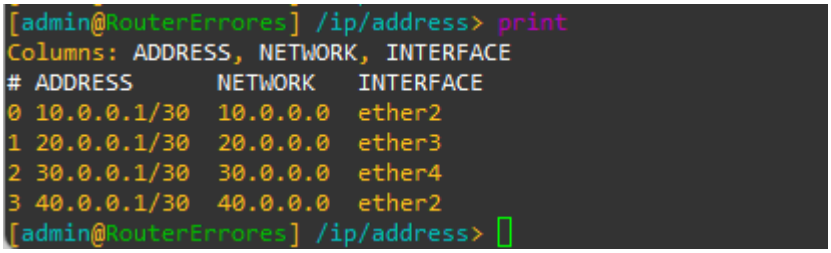


Fig. 14. Control de errores. Edición de una IP (4).

Una vez hemos visto como corregir un error a través del comando edit, vamos a ver como borrar una entrada de la tabla. Esto lo haremos en los casos que no queramos reutilizar ningún valor de la entrada.

En este caso como ya tenemos una dirección IP para la interfaz *ether2*, queremos borrar la entrada sin conservar la dirección IP. Se procede de la siguiente manera.

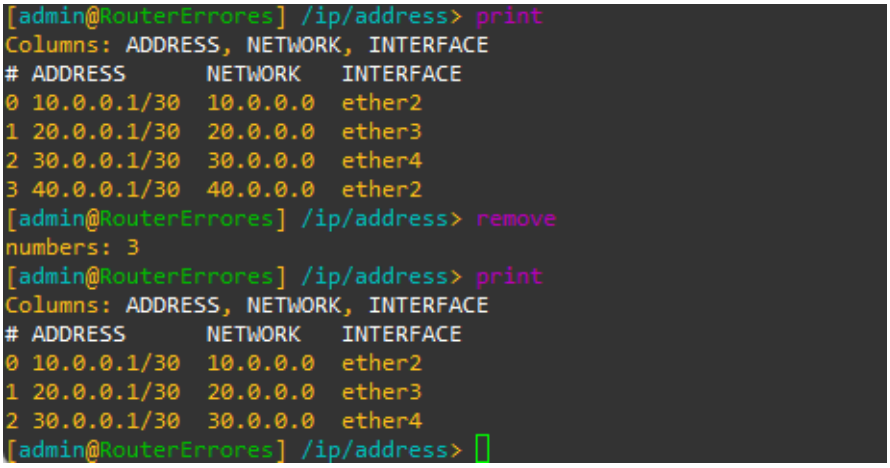


Fig. 15. Control de errores. Borrar entrada completa en ip/address.

Para borrar la entrada mencionada, ejecutamos el comando *remove* e indicamos el número en cuestión. Para comprobar el éxito de la ejecución, volvemos a ejecutar el comando *print.*

2.3. Actualizar RouterOS del MikroTik

Los equipos MikroTik con los que vamos a trabajar en el laboratorio están basados en un SO propio denominado RouterOS. Este Sistema Operativo se actualiza con poca frecuencia ya que se está implementando poco a poco las funcionalidades para la *versión 7.*

En este caso, para realizar la práctica, nosotros necesitamos al menos **RouterOS 7.10**.

Si los equipos no están actualizados, a continuación, se detalla el proceso para llevarlo a cabo:

Una vez abrimos *winbox* y vemos los equipos conectados, podemos ver la versión que utiliza como se puede comprobar en la captura siguiente:

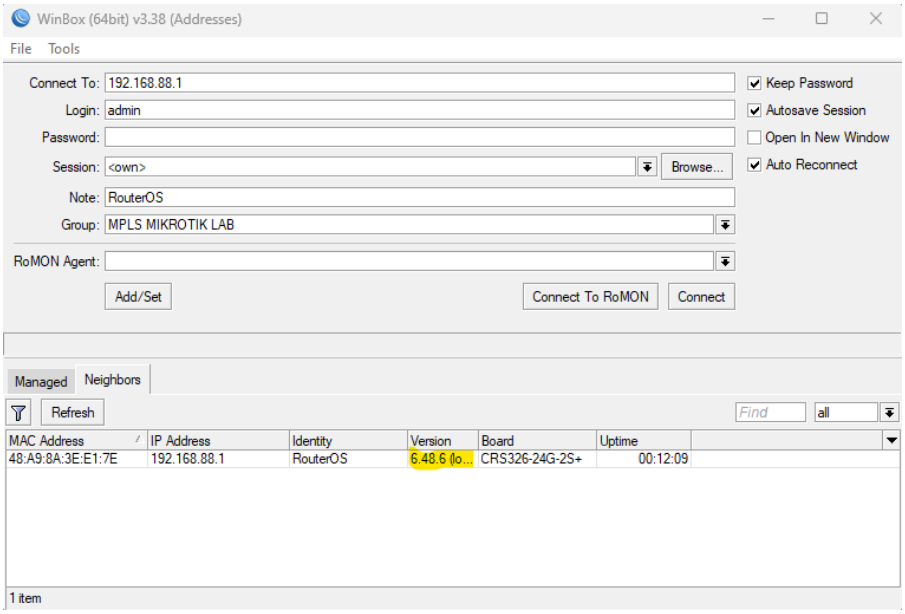


Fig. 16. Versión actual de RouterOS.

Ya dentro de la interfaz de *Winbox*, seleccionaremos la sección *Files*, donde arrastraremos el archivo con la versión actualizada que podemos encontrar en:

https://download.mikrotik.com/routeros/7.10/routeros-7.10-arm.npk

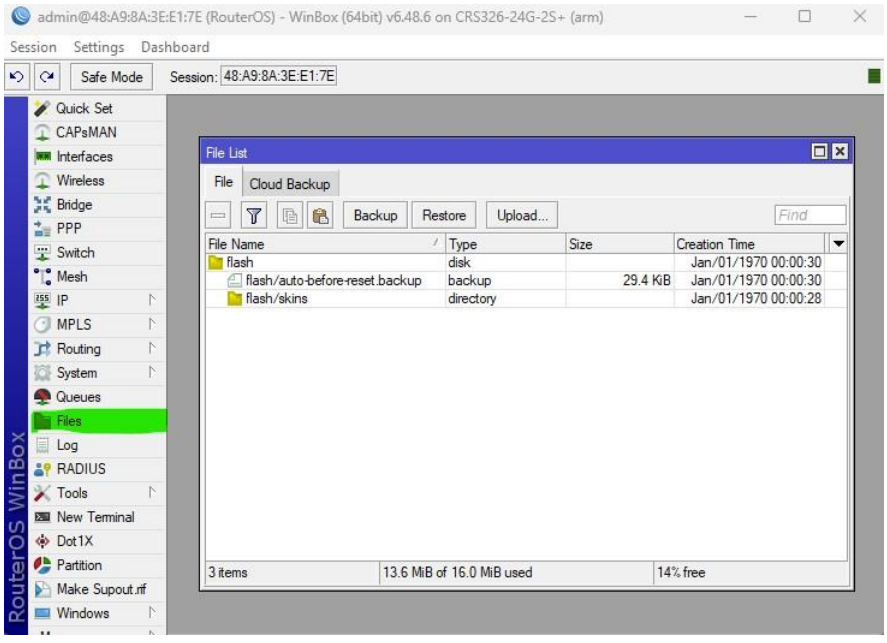


Fig. 17. Sección Files por defecto.

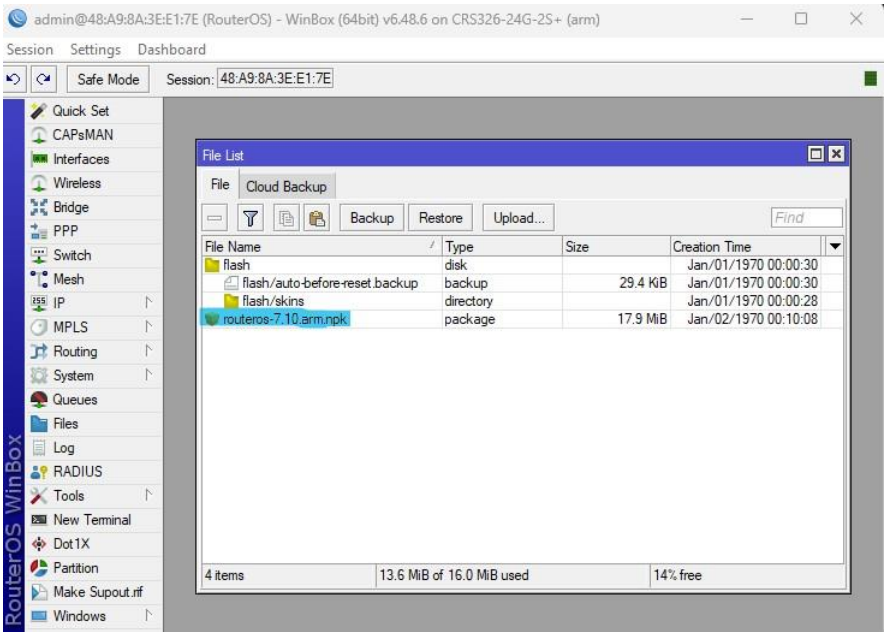


Fig. 18. Sección Files con archivo de la nueva versión.

Con el archivo cargado, abrimos el *Terminal* y ejecutamos *system reebot* para reiniciar el router para actualizar la versión.

Este archivo cargado es el equivalente al utilizado en GNS3 para la instalación de los equipos virtuales. Pero en ningún caso son el mismo tipo de archivo porque en GNS3 estamos utilizando un archivo *Raw Disk Image* que se instala en la máquina virtual para su utilización enGNS3.

2.4. Softwares necesarios

2.4.1. Introducción

Para realizar las prácticas sin usar los equipos físicos de MikroTik y por tanto utilizar el entorno de simulación GNS3, necesitamos una serie de requisitos a nivel de software. Como se menciona, necesitamos *GNS3* para montar la simulación, pero aparte necesitamos hacer uso de *VMware WorkStation Player* para una fácil instalación de los equipos virtuales y de *WireShark* para la captura de los paquetes generados en la red.

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

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

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

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

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

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

Algunas de sus características más importantes son:

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

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

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

2.4.2. Máquina Virtual

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

Primero instalaremos el software de VMware WorkStation Player en su versión gratuita para uso no comercial desde su página web oficial:

https://www.vmware.com/es/products/workstation-player/workstation-player-evaluation.html

Tendremos que seleccionar "DESCARGAR AHORA" como se muestra en la imagen. Esto iniciará la descarga de un ejecutable *.exe* que tendremos que abrir para la instalación.

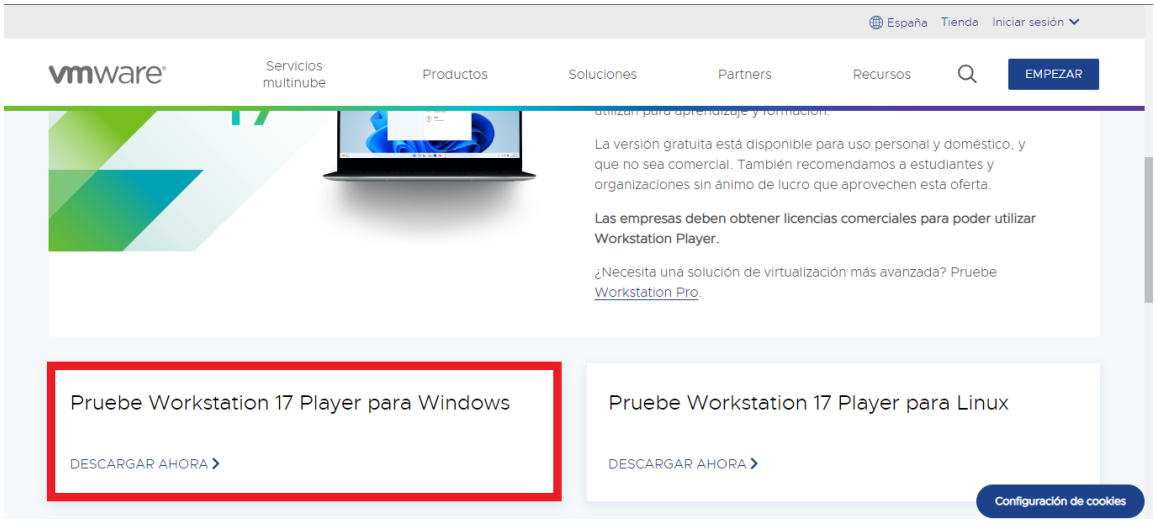


Fig. 19. Instalación VMware. Web de descarga.

A continuación, se muestra el flujo de instalación, aunque este no conlleva ningún paso complejo más allá de aceptar las condiciones, cambiar el directorio de instalación, etc.

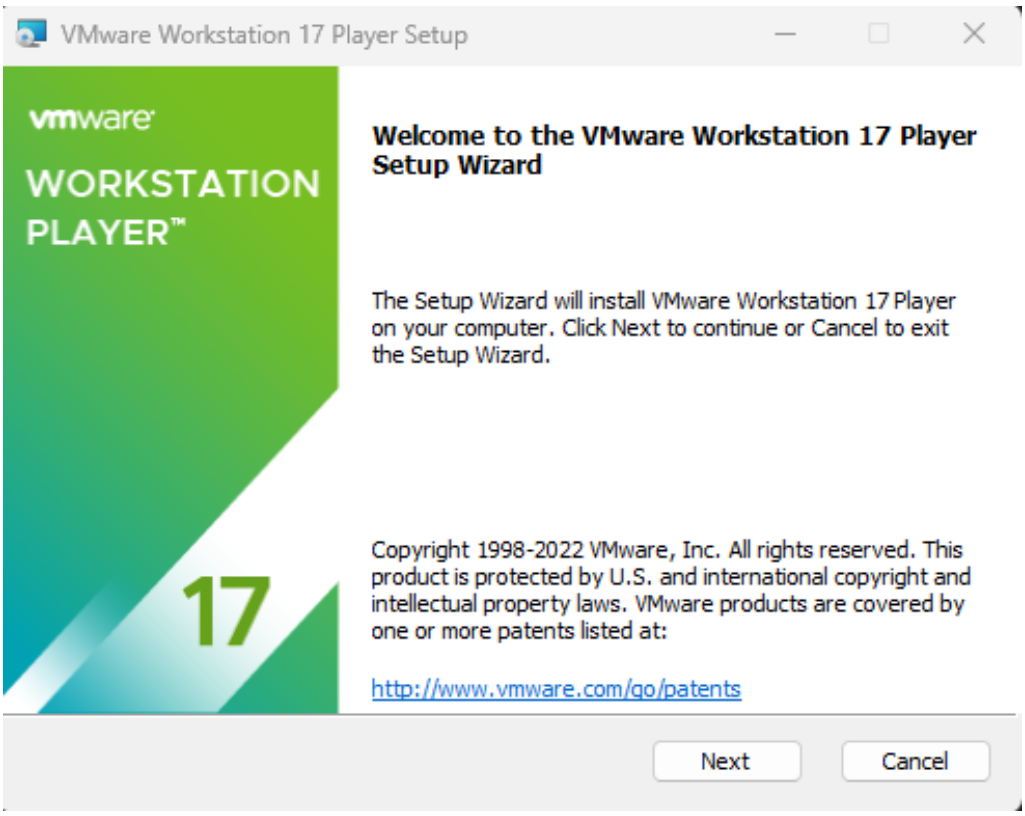


Fig. 20. Instalación VMware. Flujo de instalación (1).

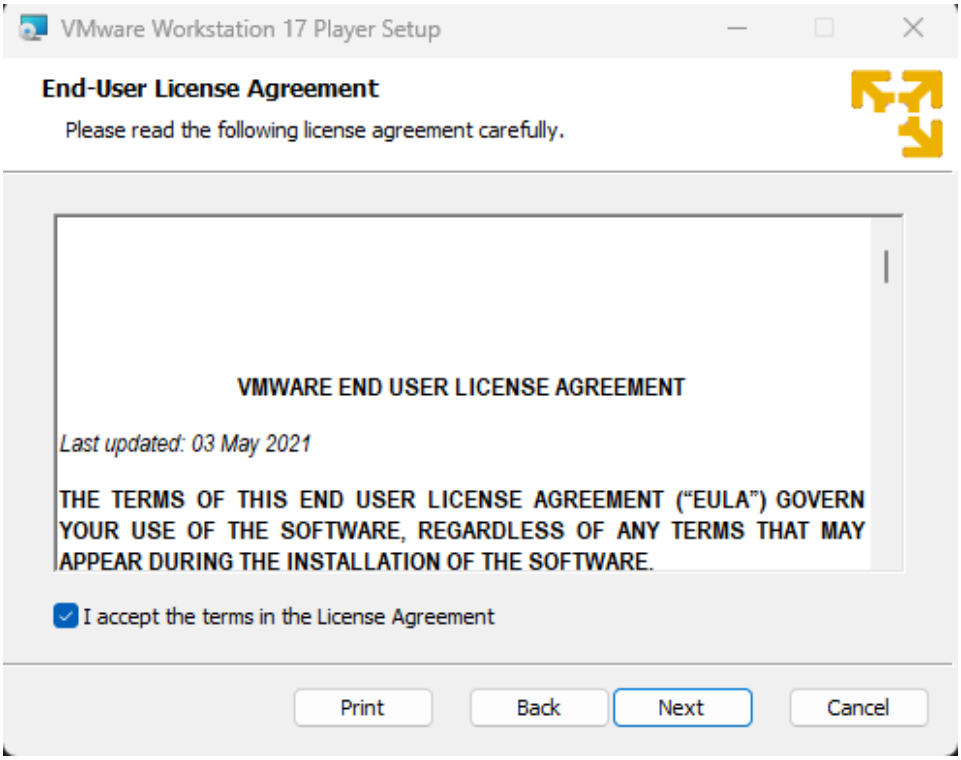


Fig. 21. Instalación VMware. Flujo de instalación (2).

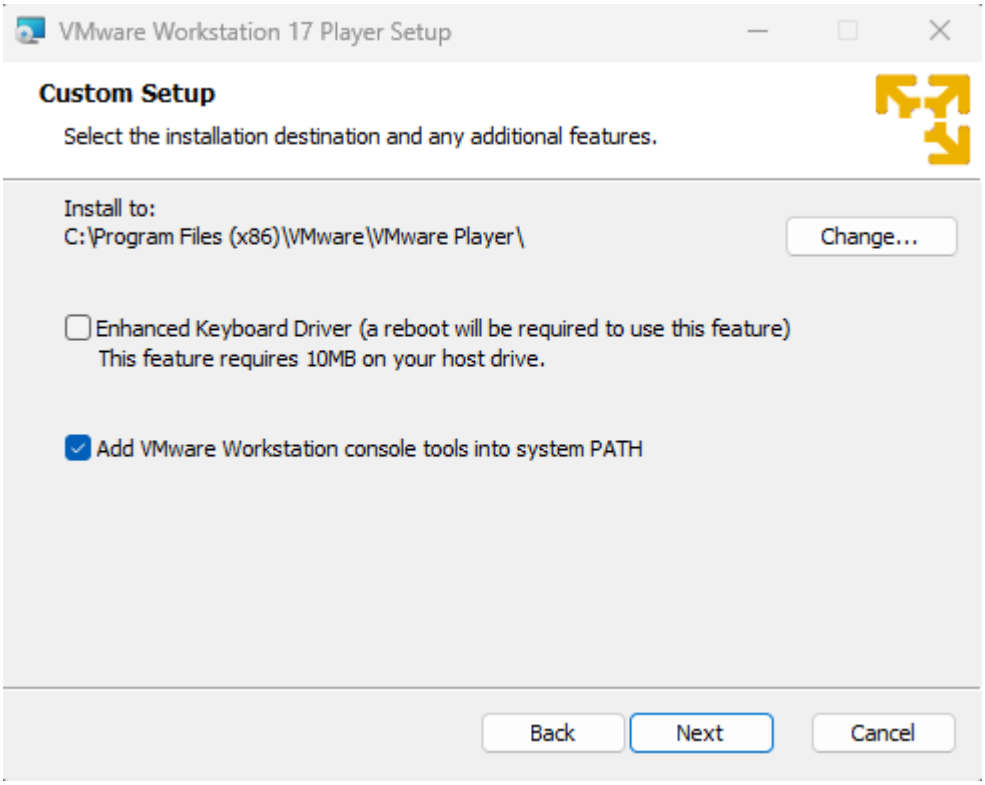


Fig. 22. Instalación VMware. Flujo de instalación (3).

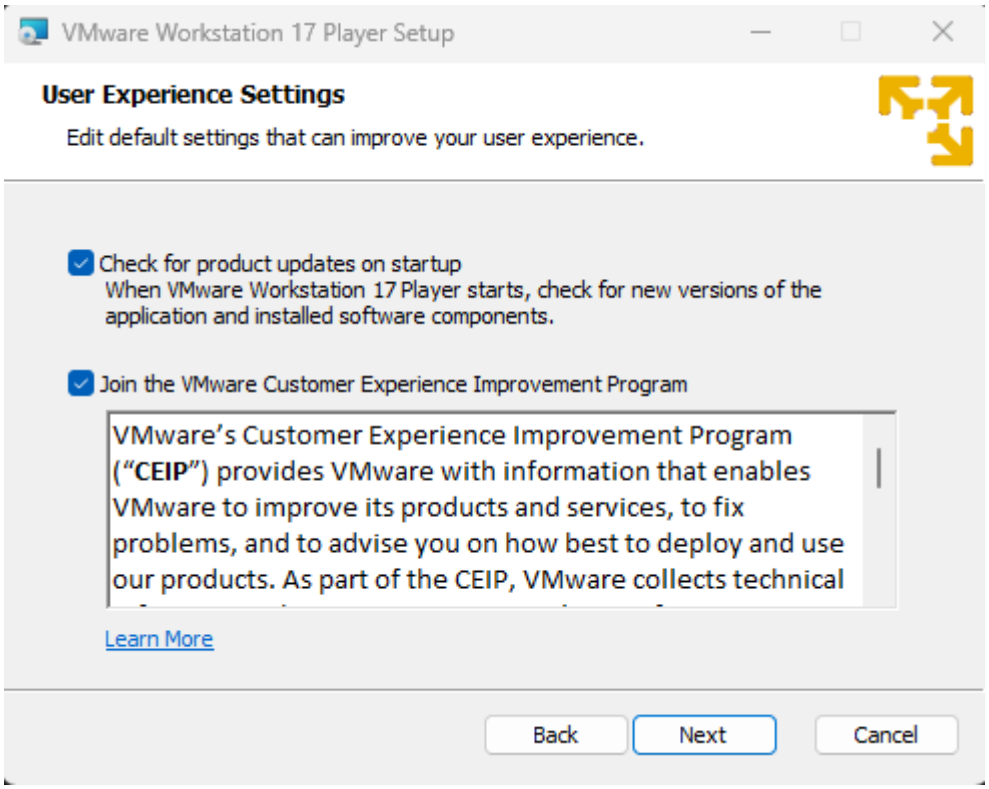


Fig. 23. Instalación VMware. Flujo de instalación (4).

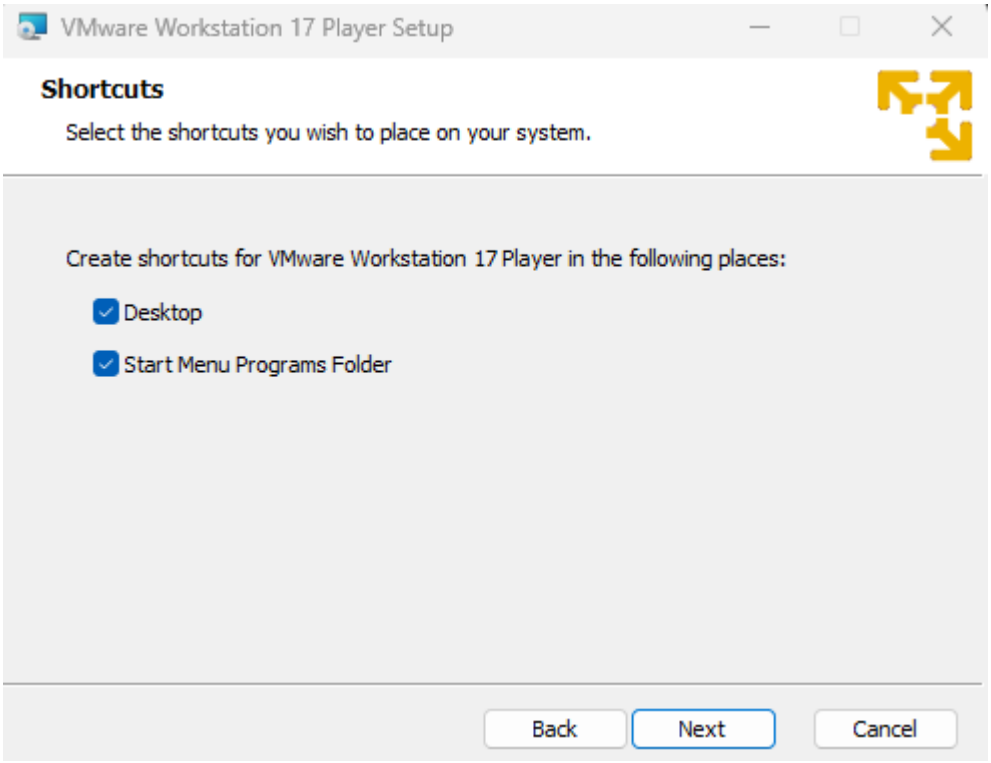


Fig. 24. Instalación VMware. Flujo de instalación (5).

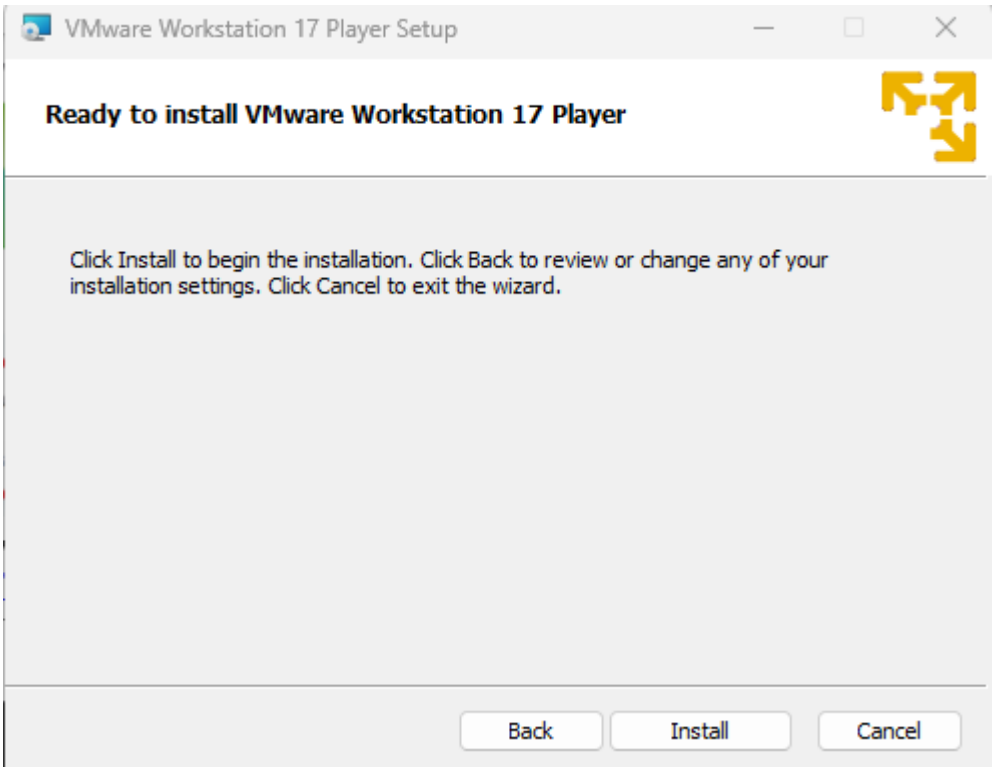


Fig. 25. Instalación VMware. Flujo de instalación (6).

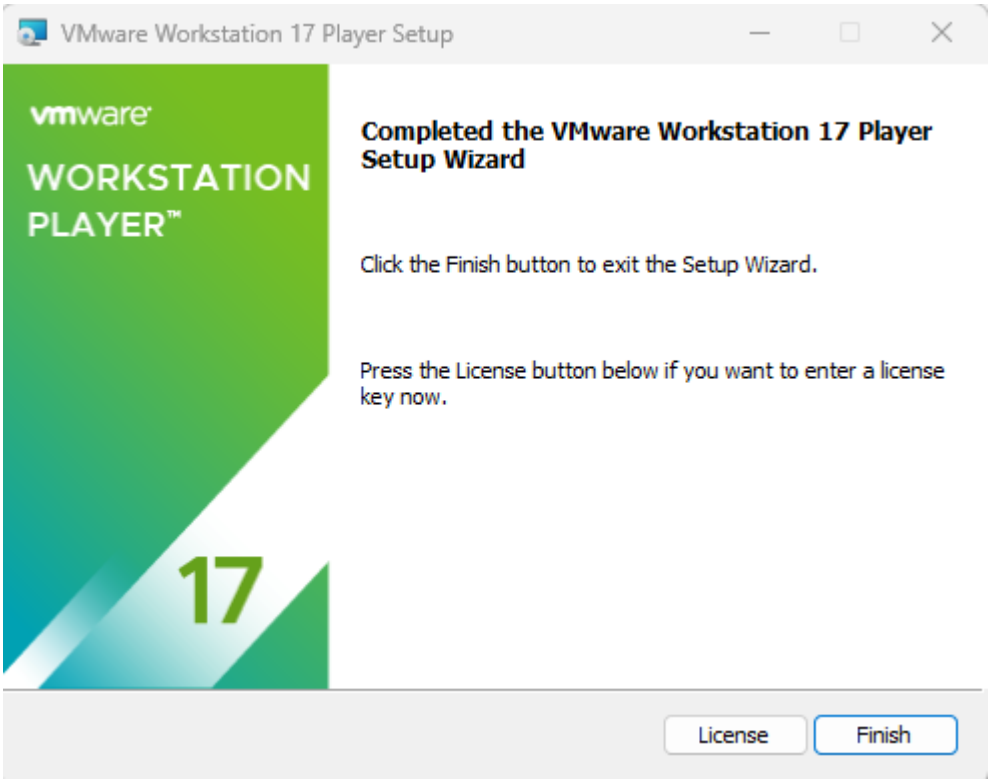


Fig. 26. Instalación VMware. Flujo de instalación (7).

Con VMware instalado accederemos a la página web oficial de GSN3 (https://www.gns3.com) para descargar la imagen de la máquina virtual que nos permitirá instalar nuestro router de Mikrotik.

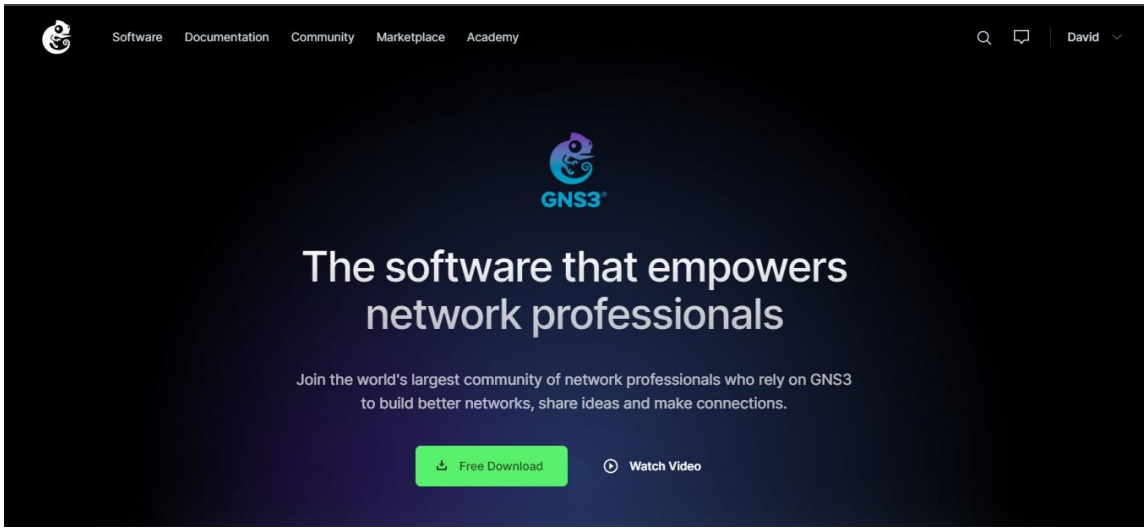


Fig. 27. Inicio de gns3.com

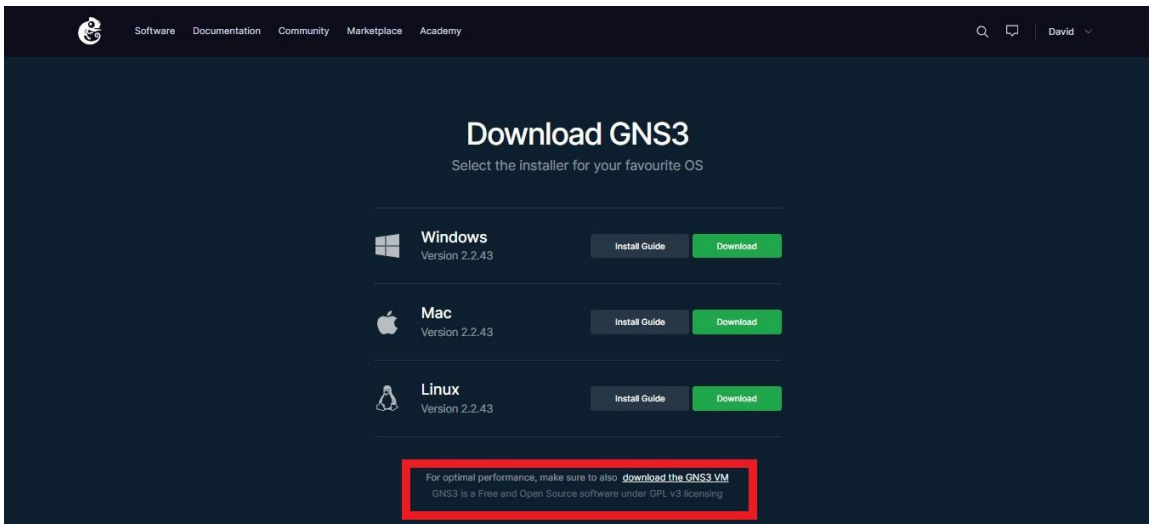


Fig. 28. Descarga de GNS3 y acceso a descarga de las VM.

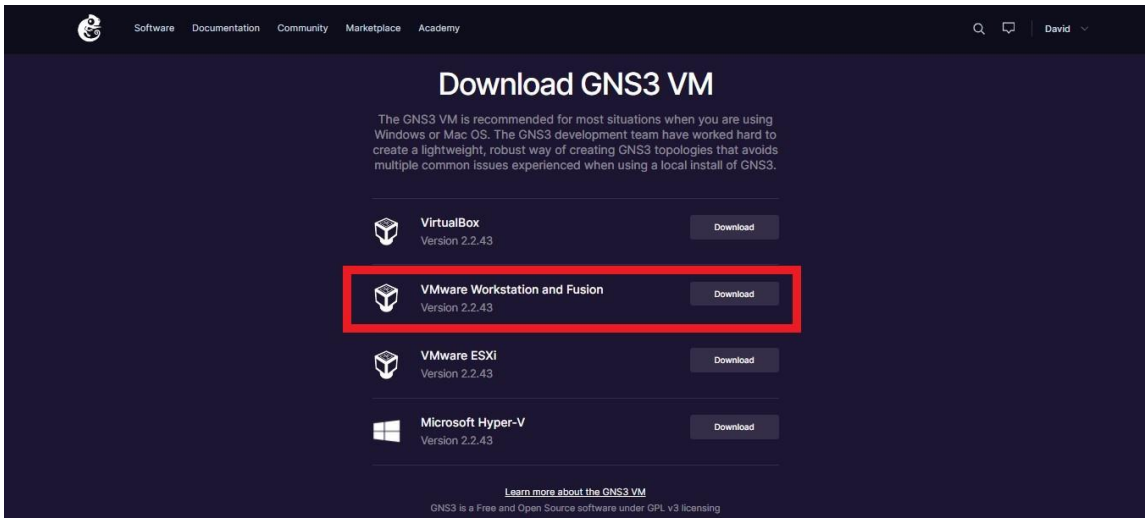


Fig. 29. Descarga de la VM para diferentes softwares de virtualización.

La versión debe coincidir con la versión del GNS3. Descomprímala y haga doble click sobre la misma para importarla a VMware WorkSatation Player.

Primero tendremos que elegir el nombre de la máquina virtual y la ubicación de esta. Una vez hecho seleccionaremos “Import”.

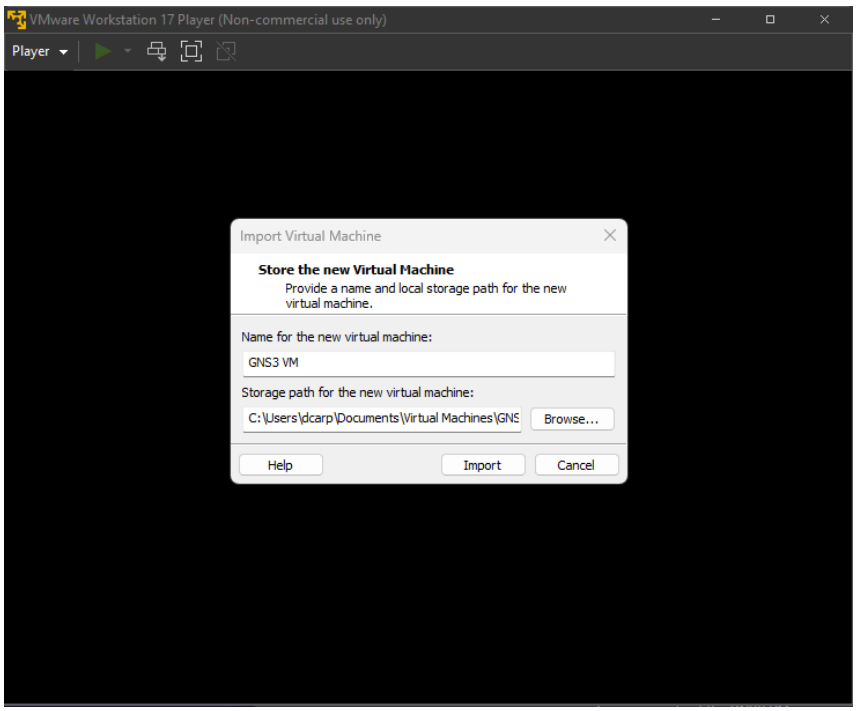


Fig. 30. VMware. Importar máquina virtual.

Tras la importación, se inicia automáticamente la VM pero nos salta un aviso de que el virtualizador *Intel VT-x/EPT* no es soportado. Seleccionemos lo que seleccionemos nos saltará un error que abortará la VM.

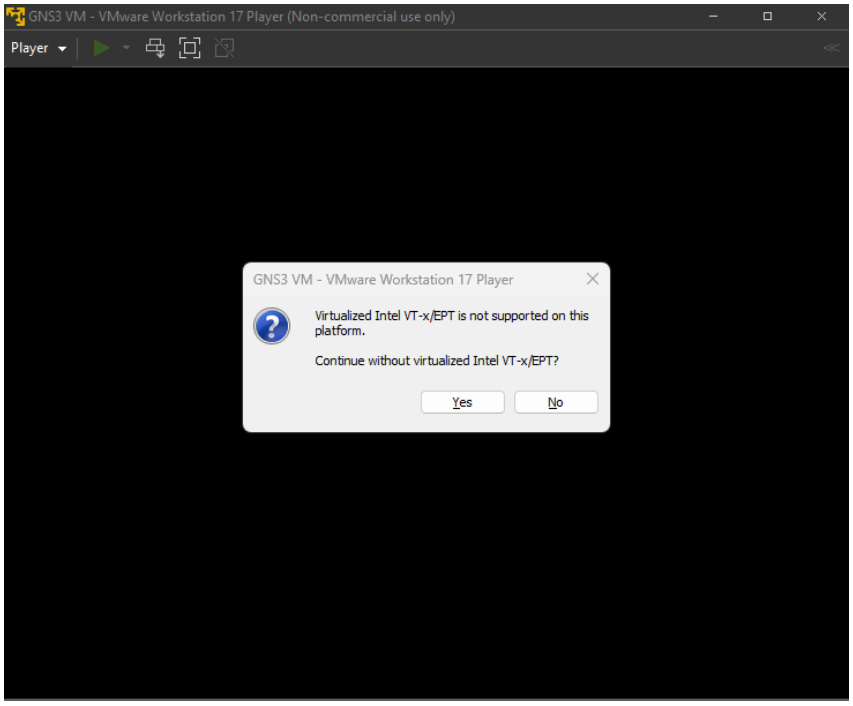


Fig. 31. VMware. Aviso de virtualizador Intel VT-x/EPT.

Para solucionar esta casuística y configurar la máquina virtual con ciertas especificaciones, accederemos a la configuración de la VM de la siguiente manera:

Primero, seleccionaremos nuestra máquina virtual y en la parte inferior seleccionaremos *"Edit virtual machine settings"*.

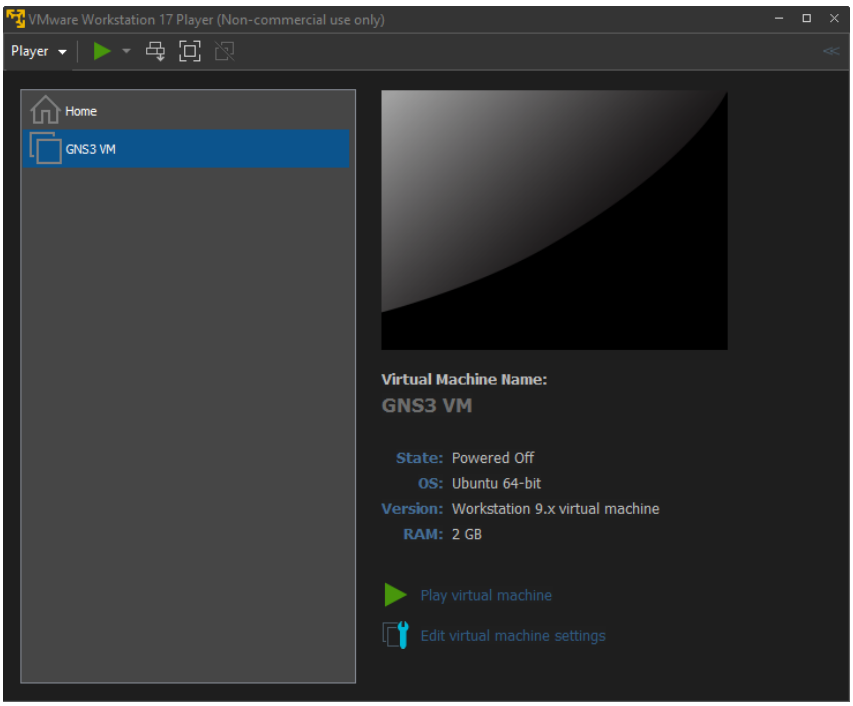


Fig. 32. VMware. Configurar VM.

Ahora, aumentaremos la memoria a 4096 MB.

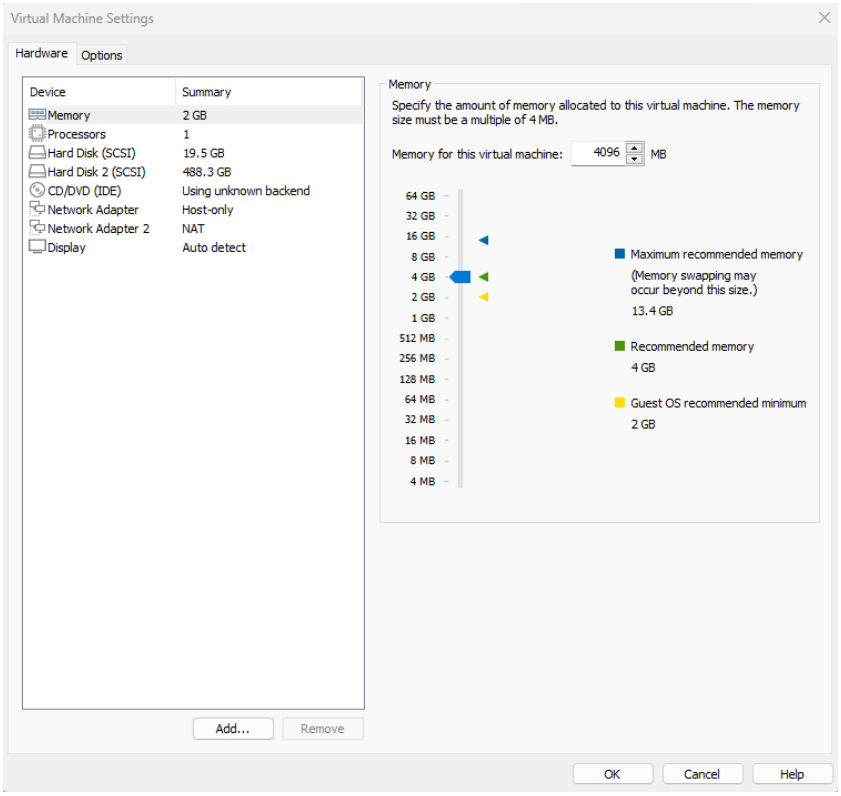


Fig. 33. VMware. Configurar VM. Memoria.

Y por último, seleccionaremos *Processors*, donde aumentaremos estos por lo menos a 2 y deshabilitaremos la opción *Virtualize Intel VT-x/EPT*. Seleccionaremos *OK* para guardar los cambios en la configuración.

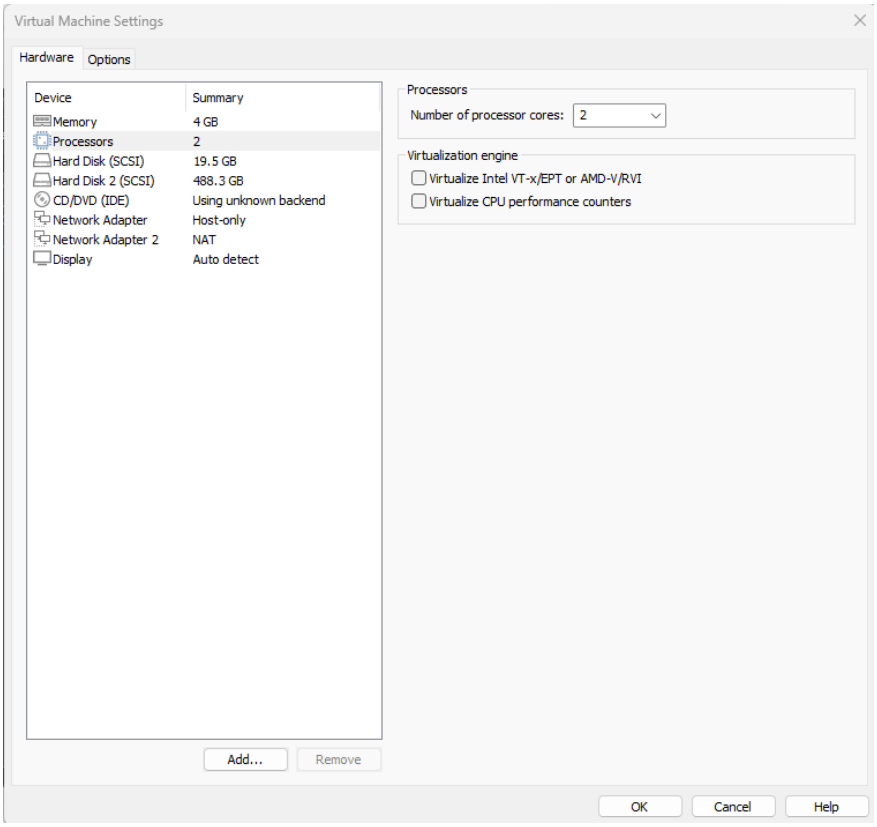


Fig. 34. VMware. Configurar VM. Memoria. Processors.

Ahora ya podemos iniciar de nuevo la máquina virtual seleccionando *Play virtual machine*.

Una vez se cargue la máquina virtual, tendremos que ver la siguiente interfaz gráfica:

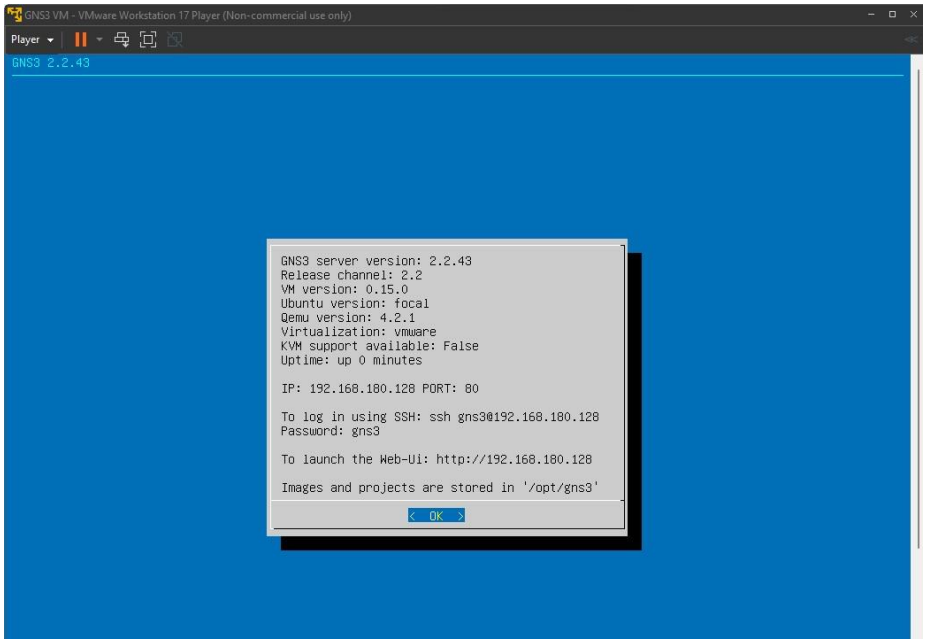


Fig. 35. Máquina Virtual GNS3 iniciada.

Además, para no tener problemas al iniciar los routers Mikrotik, tendremos que añadir dos líneas de código en el archivo gns_server.conf desde la máquina virtual.

Pero antes de esto tendremos que cambiar el idioma del teclado dentro de la máquina virtual. Para ello seleccionaremos *OK* usando el Enter y nos desplazaremos por los menús hasta *Keyboard.*

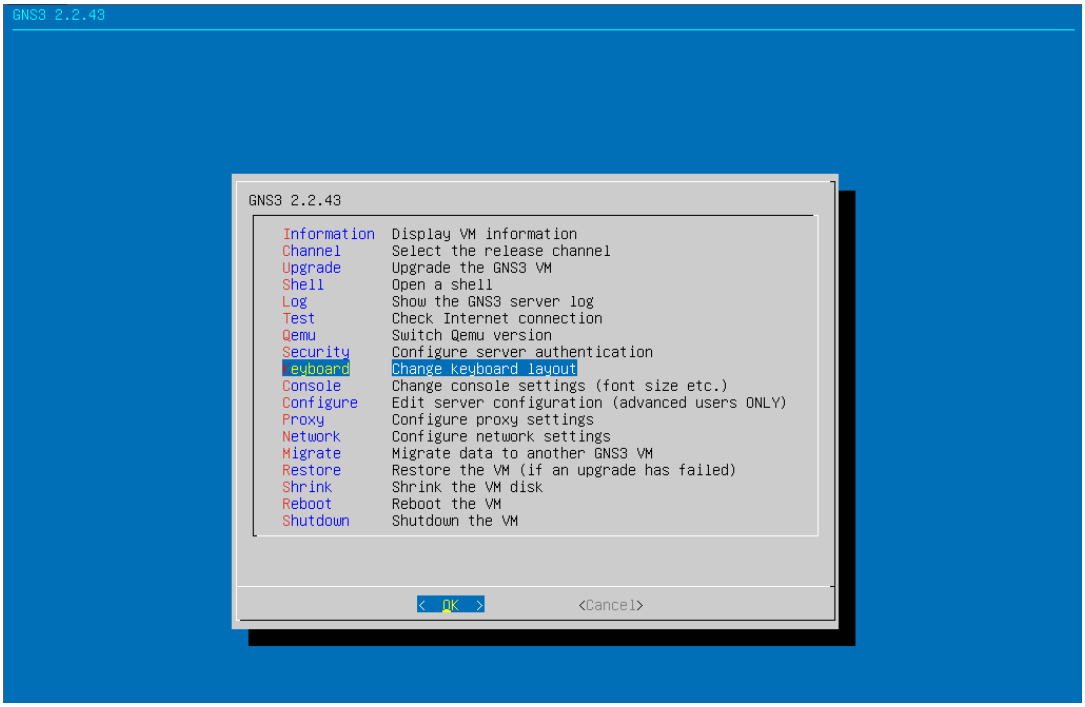


Fig. 36.VMware. Keyboard.

Ahora seleccionaremos el teclado *Generic 101-key PC*. Una vez hecho esto indicaremos que queremos cambiar el idioma, para ello seleccionamos *NO*.

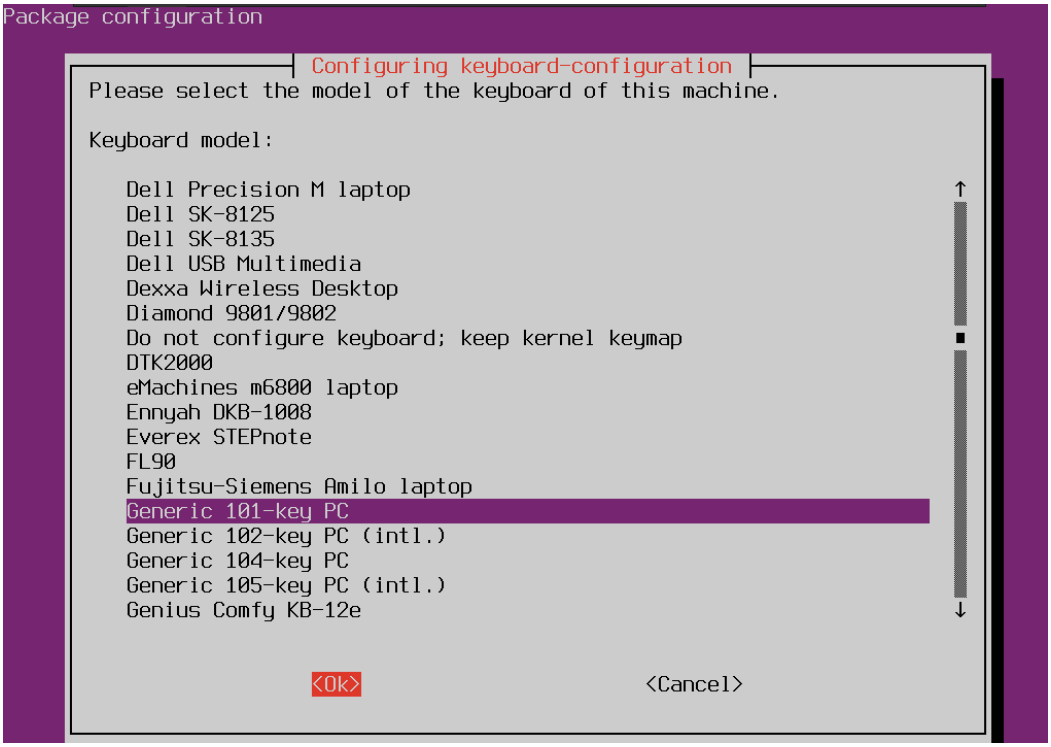


Fig. 37.VMware. Keyboard. Selección de teclado.

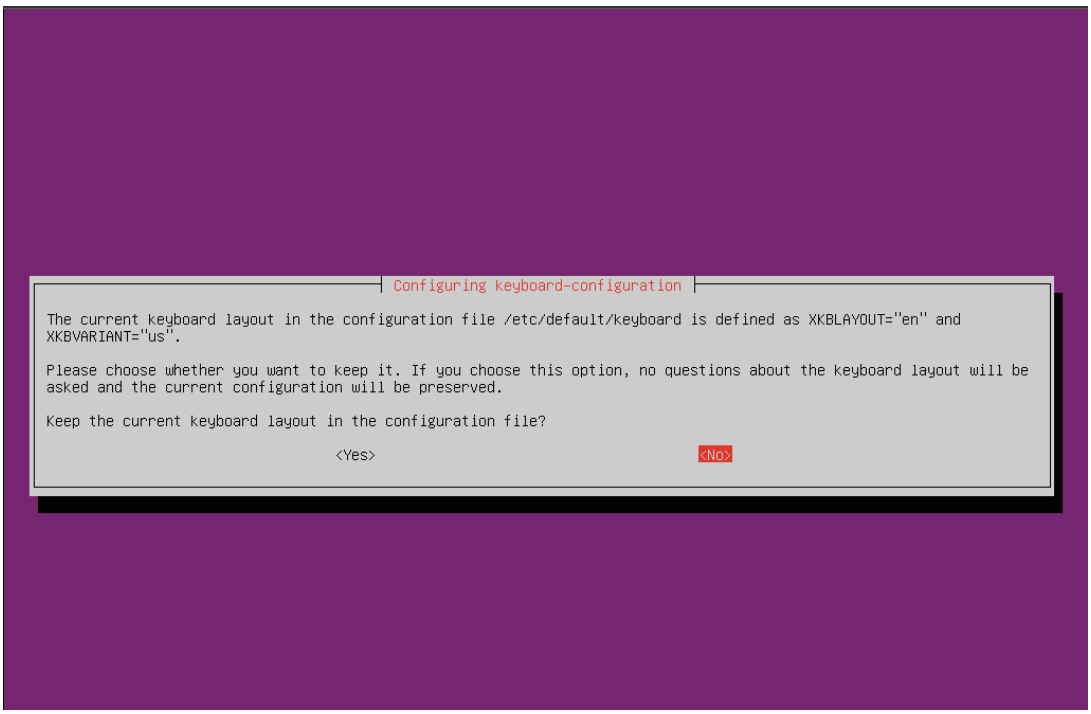


Fig. 38. VMware. Keyboard. Cambiar idioma.

Ahora nos desplazaremos por el selector hasta localizar el idioma *Spanish*, que lo seleccionaremos. A continuación, seleccionaremos el teclado de *Windows.*

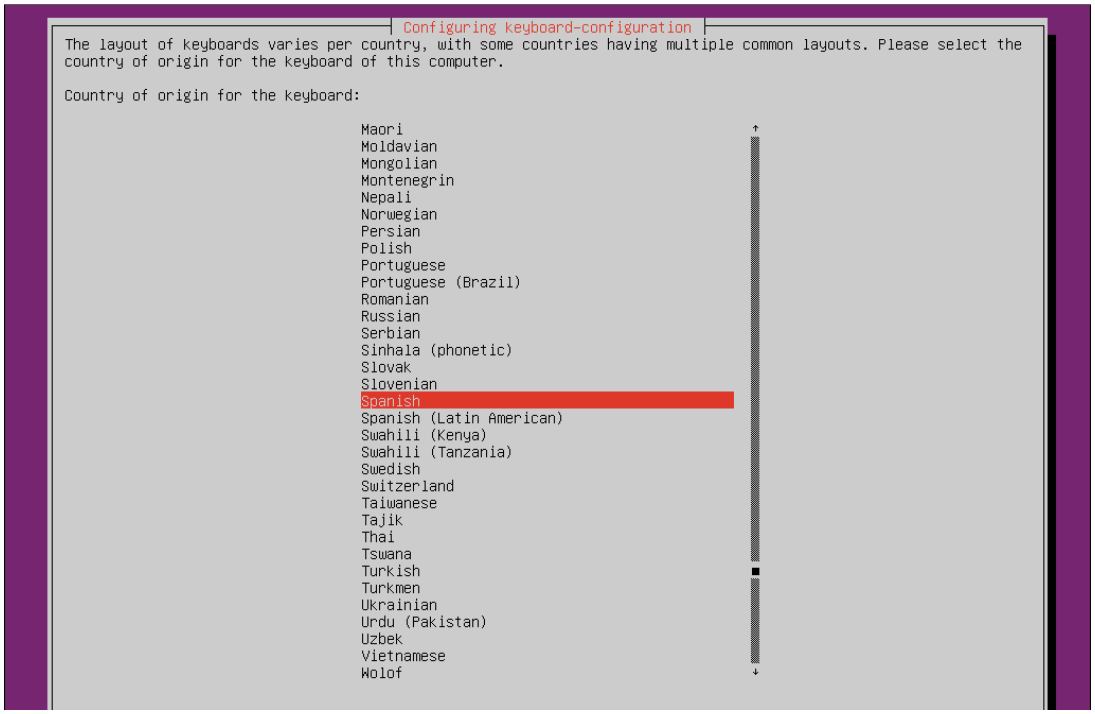


Fig. 39. VMware. Keyboard. Seleccionar idioma.

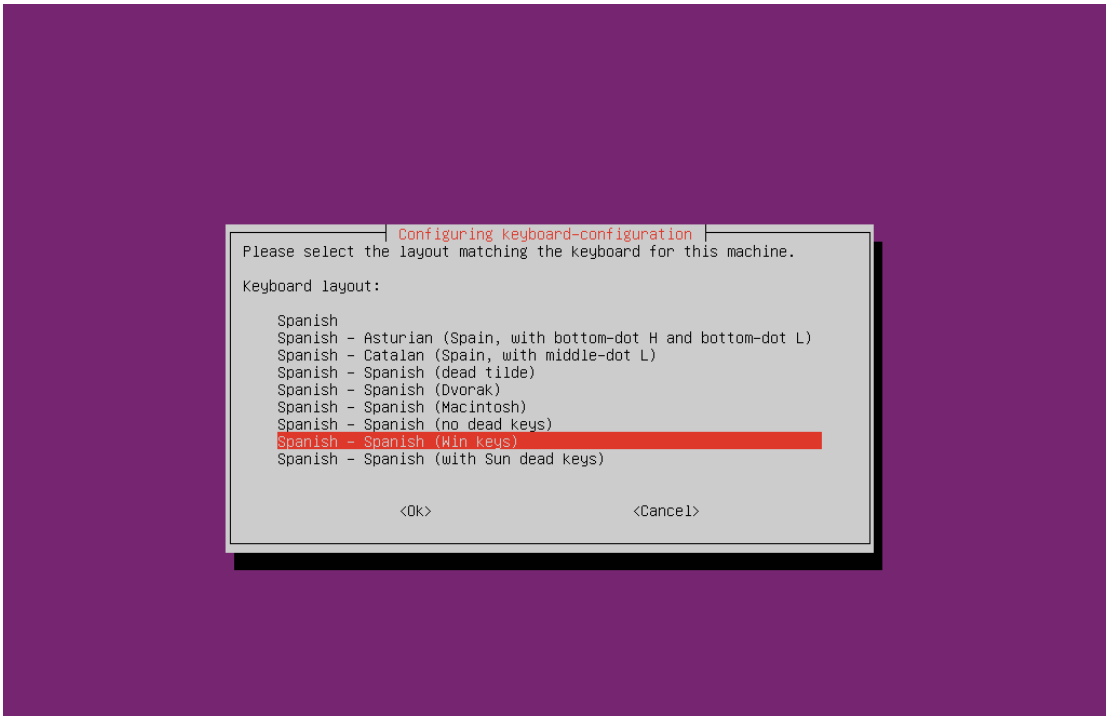


Fig. 40. VMware. Keyboard. Teclado de Windows.

Para el teclado de Windows nos pedirán que seleccionemos la tecla que se usará como *AltGr* y la que se utilizará para *comandos combinados*. Para la primera se selecciona la propia *AltGr* y para los comandos combinados no es necesario seleccionar ninguna.

Fig. 41. VMware. Keyboard. Teclado de Windows. AltGr.

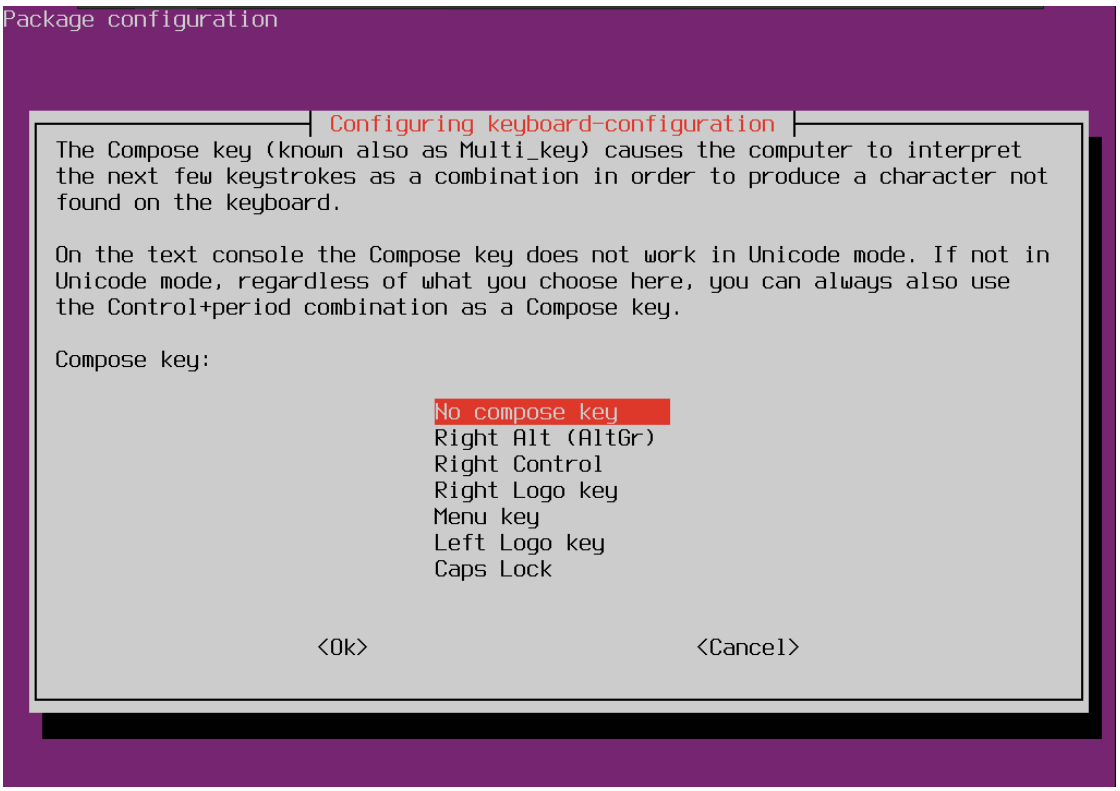


Fig. 42. VMware. Keyboard. Teclado de Windows. Compose Key.

Con estos pasos tendremos el teclado configurado para poder añadir las líneas necesarias, pero para que sea efectivo tendremos que reiniciar la máquina virtual seleccionando *Reboot* en el menú principal de la VM.

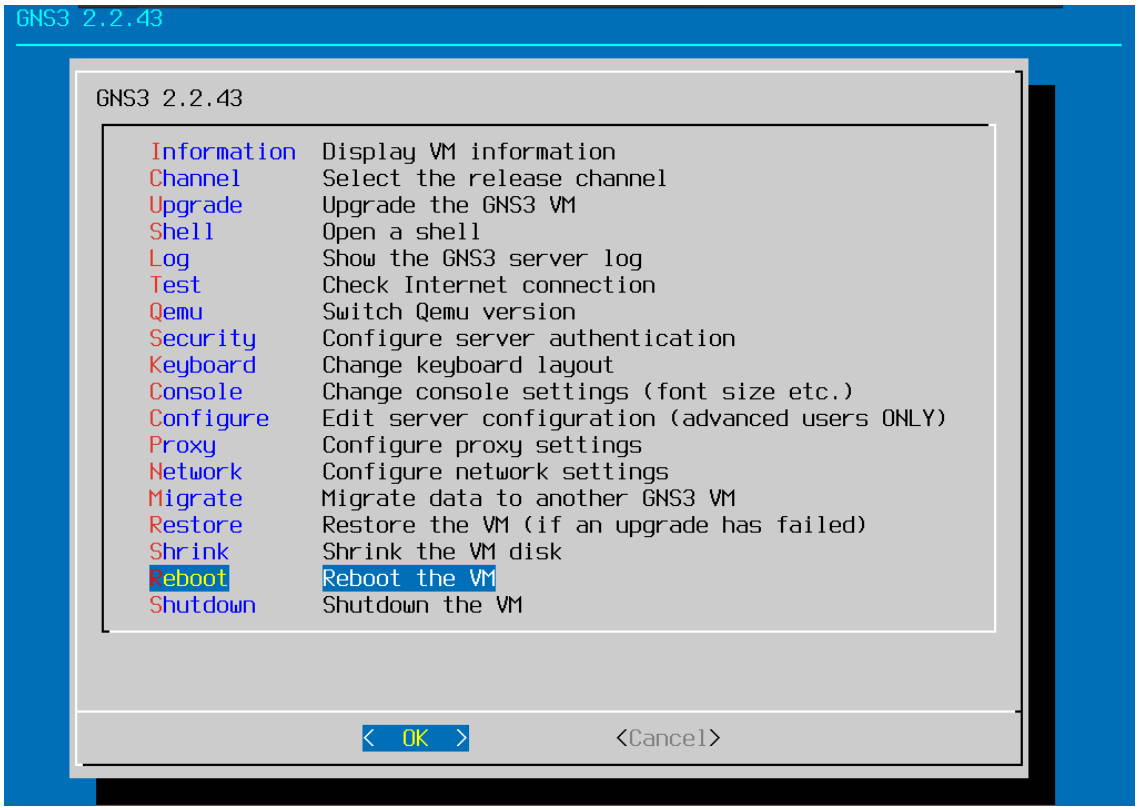


Fig. 43. VMware. Reboot.

Ahora ya podemos añadir las dos líneas que comentábamos, para ello primero en el menú de la VM seleccionamos *Configure*, que abrirá el archivo que debemos modificar para que quede como se muestra en la imagen.

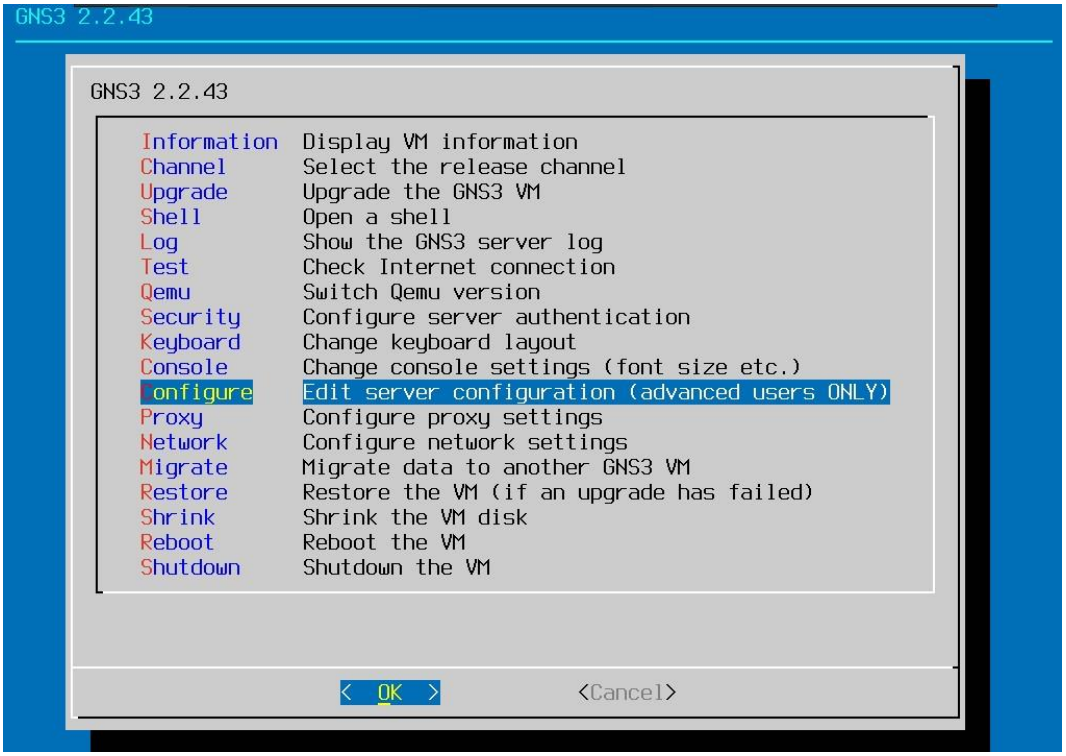


Fig. 44. VMware. Configure.

Fig. 45. Introducir Qemu en gns3_server.conf.

Guardaremos usando ***ctrl+X*** y confirmaremos pulsando la tecla ***Y*** y ***Enter***, y por último reiniciaremos la máquina para asegurarnos que los cambios se realicen.

2.4.3. GNS3

2.4.3.1. Descarga e instalación

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

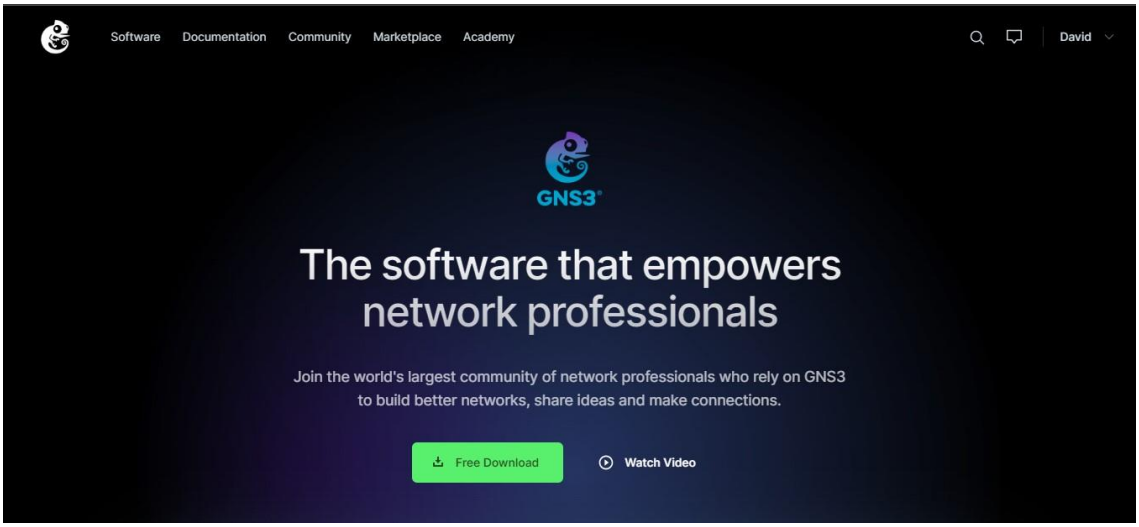


Fig. 46. Inicio de gns3.com.

Los requerimientos de hardware mínimos son:

- Sistema Operativo: Windows 7 (64 bit) o superior, Apple MAC OS o Linux.
- Procesador: 2 o más núcleos.
- Memoria: 8 GB RAM.
- Almacenamiento: 200 MB de espacio disponible para la instalación en Windows.

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

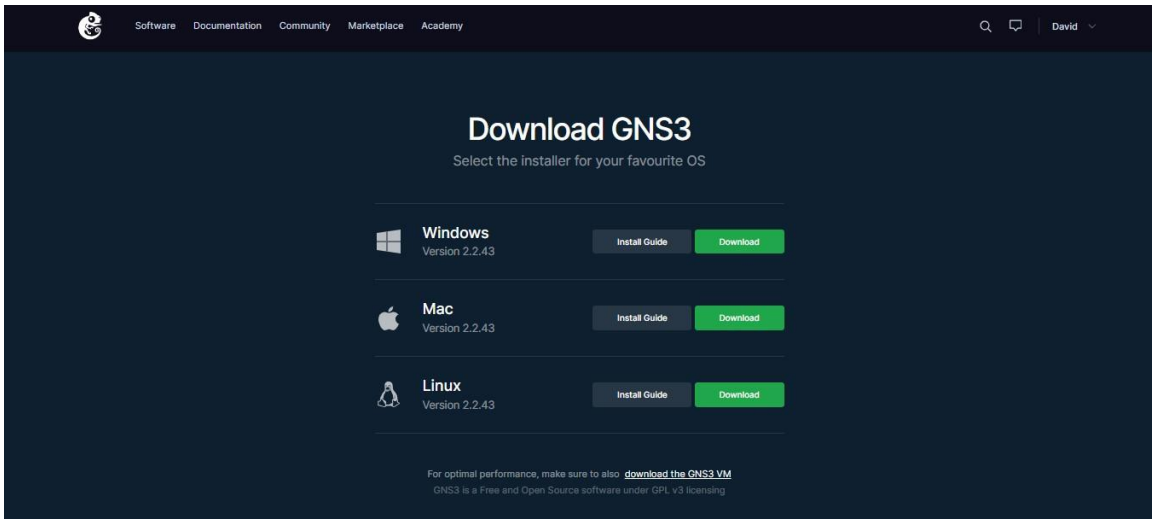


Fig. 47. Descarga GNS3.

Seleccione la versión para su sistema operativo, en este caso se trabajará sobre Windows 11 con la versión de *GNS3 2.2.43*

Presionando sobre *Download*, comenzará la descarga de un fichero denominado *GNS3-2.2.43 - all-in-one-regular*. Una vez completada la descarga del fichero, ejecútelo haciendo doble click sobre el mismo y accederá a la siguiente pantalla de Setup.

Fig. 48. Setup GNS3. Instalación.

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

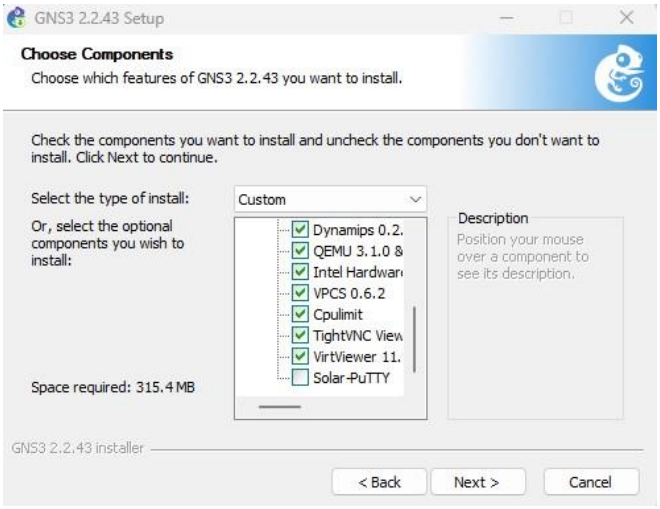


Fig. 49. Setup GNS3. Componentes.

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

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

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

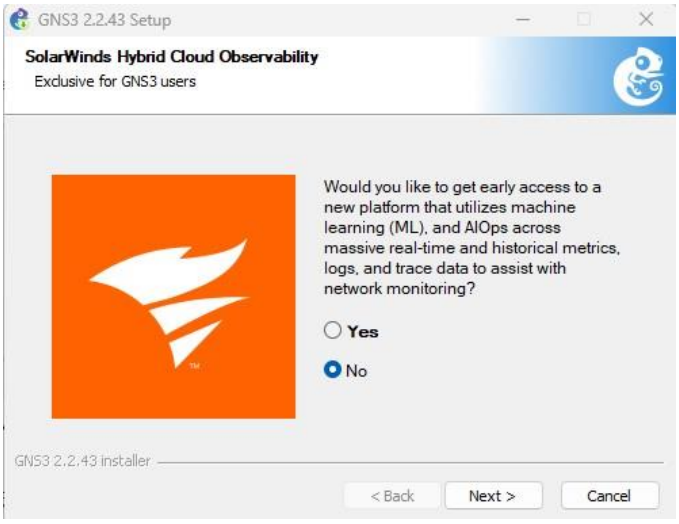


Fig. 50. Setup GNS3. Solarwinds Standard Toolset.

Por último, presione *Finish* y se iniciará GNS3.

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

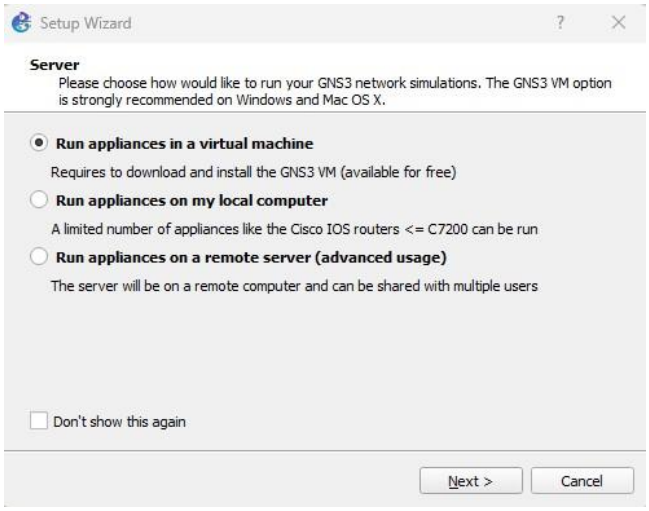


Fig. 51. GNS3. Setup Wizard.

Se tendrá que escoger el servidor sobre el que se ejecutará la simulación de la red. Escogeremos la primera opción, ya que hemos realizado la instalación de la máquina virtual con anterioridad.

En la pantalla de *Local server configuration*, seleccione en el desplegable *Host binding* la opción *localhost.*

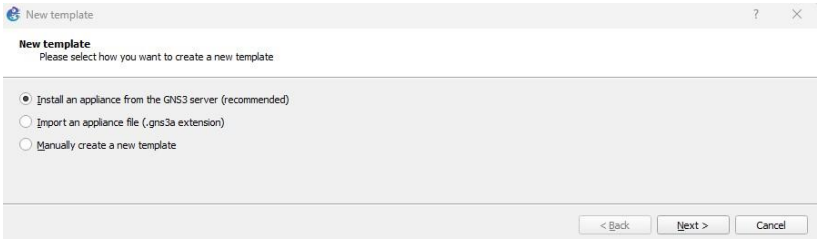


Fig. 52. GNS3. Local server configuration.

Después de elegir el servidor local, habrá que elegir el servido de la VM, para ello seleccionaremos *VMware* en *Virtualization Software*. Si todo ha ido bien y la VM está en funcionamiento, se autoseleccionará.

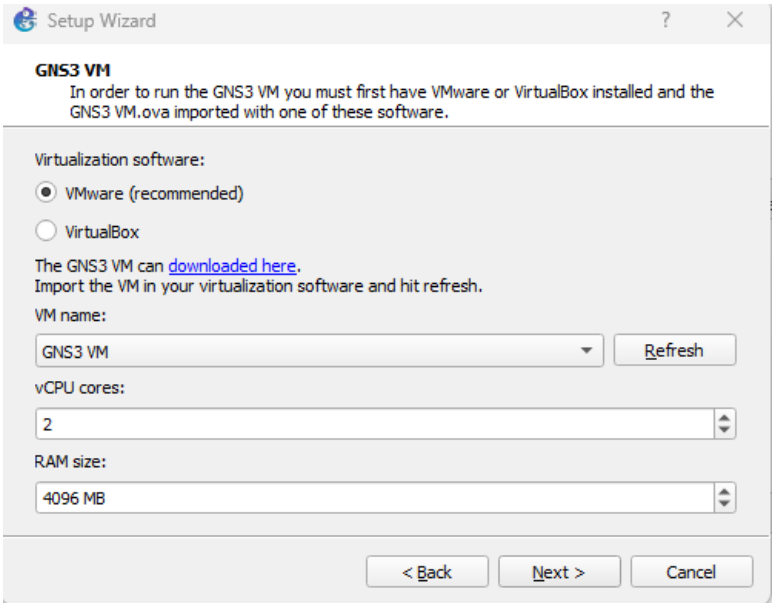


Fig. 53. GNS3. Setup Wizard. Virtualization Software.

Terminaremos con el proceso y ya tendremos GNS3 en funcionamiento.

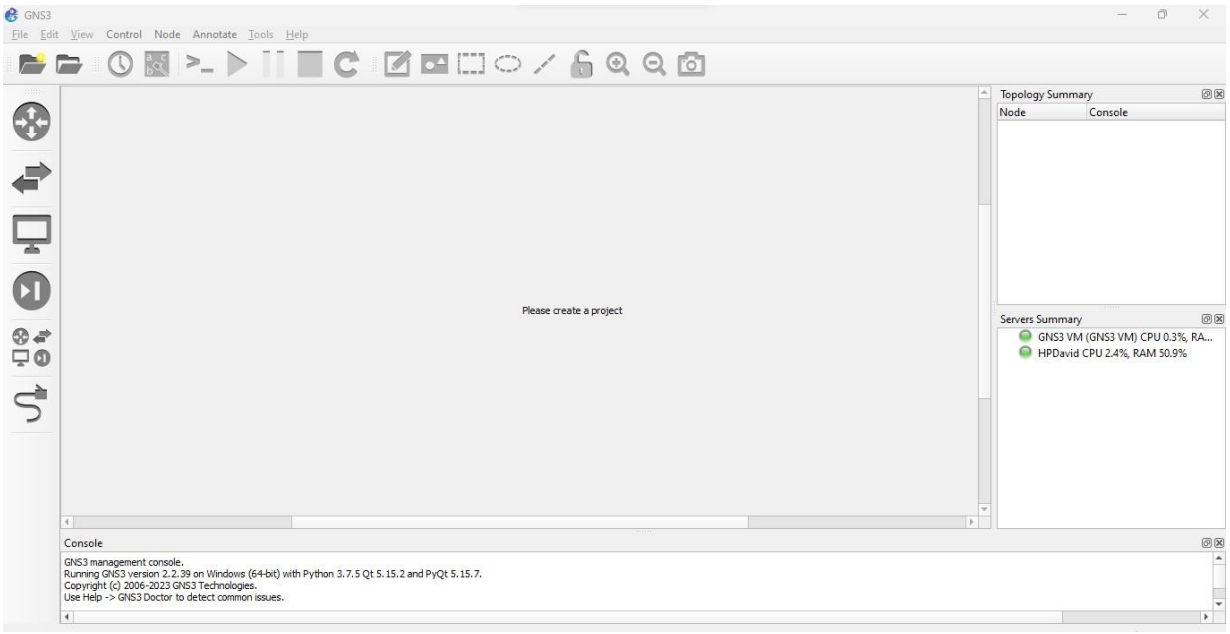


Fig. 54. GNS3. Interfaz Gráfica.

2.4.3.2. Interfaz gráfica y configuración de elementos

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

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

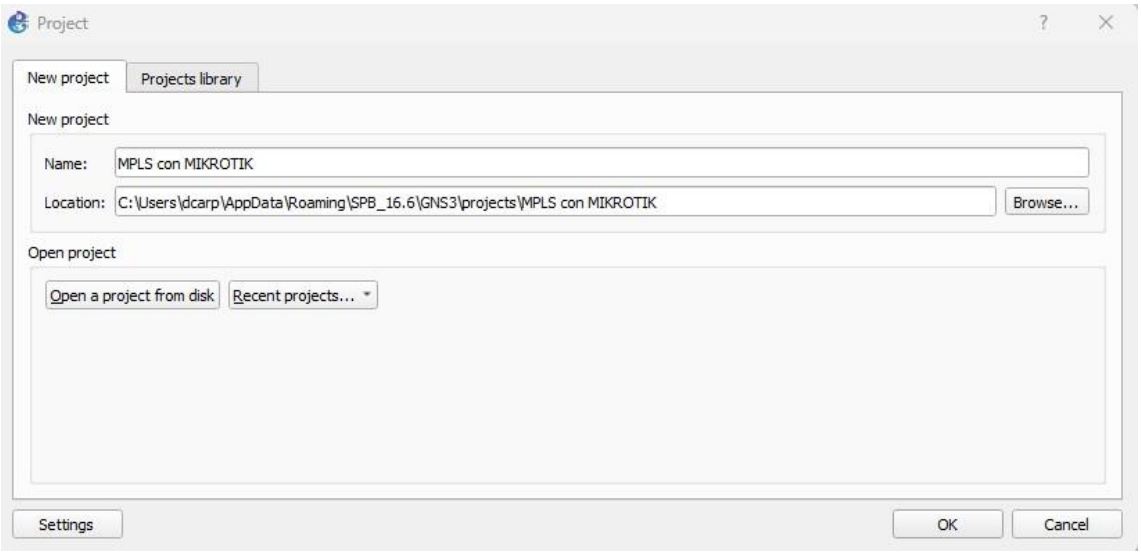


Fig. 55. GNS3. Interfaz Gráfica. Crear proyecto.

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

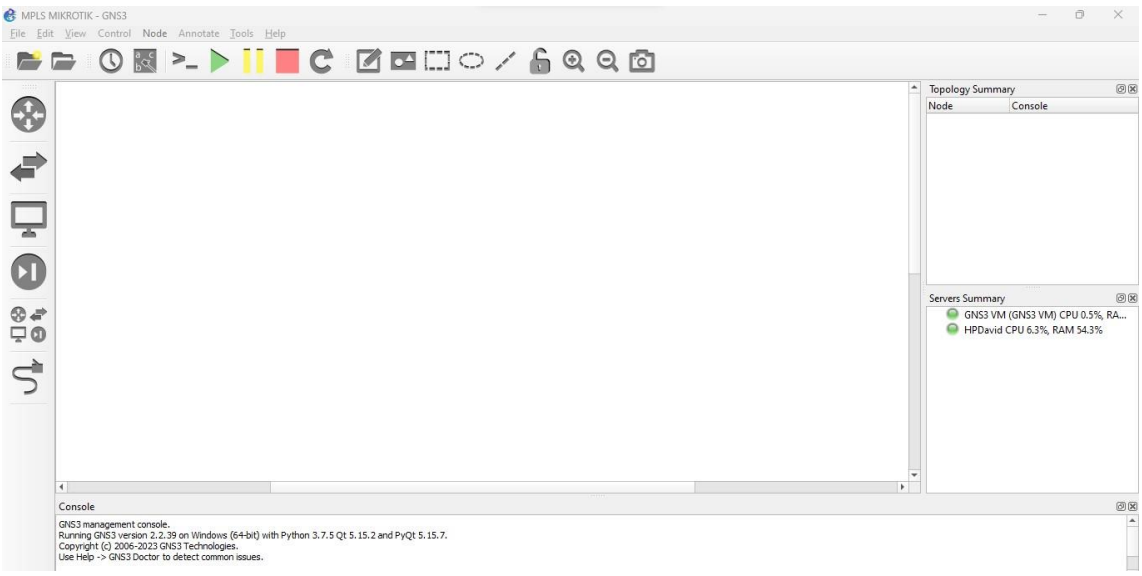


Fig. 56. GNS3. Interfaz Gráfica. Proyecto abierto.

La ventana principal de GNS3 se divide en varias subventanas:

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

Fig. 57. GNS3. Interfaz Gráfica. Barra de elementos.

Como podemos observar, no tenemos ningún router disponible para crear nuestra arquitectura, por ello vamos a realizar los pasos necesarios para disponer del router MikroTik. En este caso utilizaremos el MikroTik CRS328-24P-4S+RM.

Seleccionaremos *New Template* que abrirá una ventana para seleccionar el dispositivo, aunque primero habrá que seleccionar *Update from online Registry* para que se muestren todos los componentes disponibles.

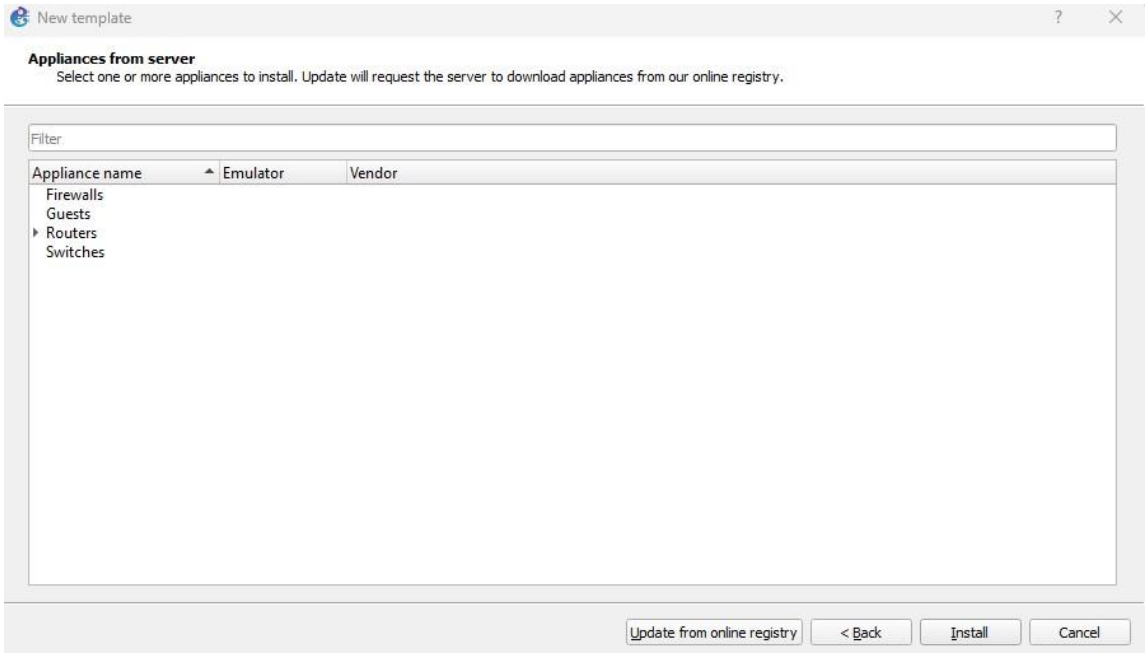


Fig. 58. GNS3. Interfaz Gráfica. New Template.

El Mikrotik que utilizaremos se encuentra dentro de los *Switches*, por eso desplegaremos esta categoría, seleccionaremos el Mikrotik CRS328-24P-4S+RM y clicaremos en *Install*.

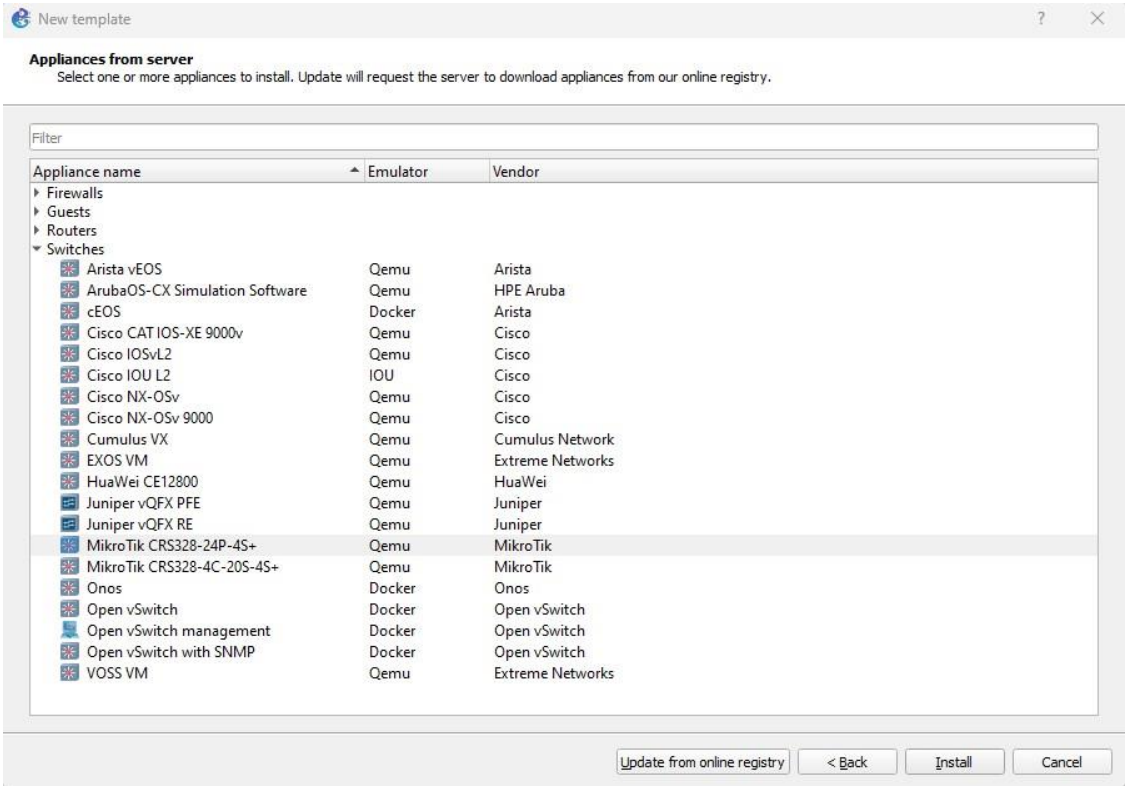


Fig. 59. GNS3. Interfaz Gráfica. New Template. Lista de Switches.

Tras este paso avanzaremos por el flujo de instalación. Primero seleccionaremos que instalaremos el *appliance* desde GNS VM.

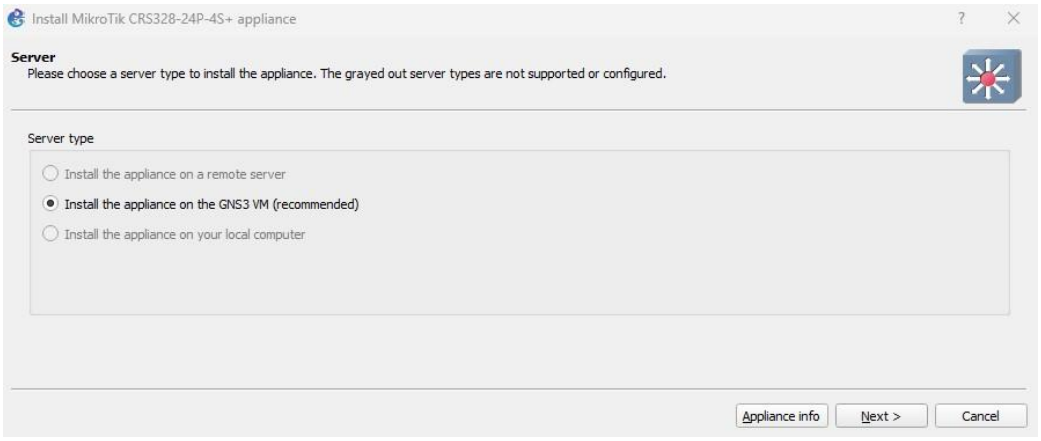


Fig. 60. Interfaz Gráfica. New Template. MikroTik CRS328-24P-4S. Server.

En la ventana ***Qemu settings*** se reconoce el Qemu automáticamente.

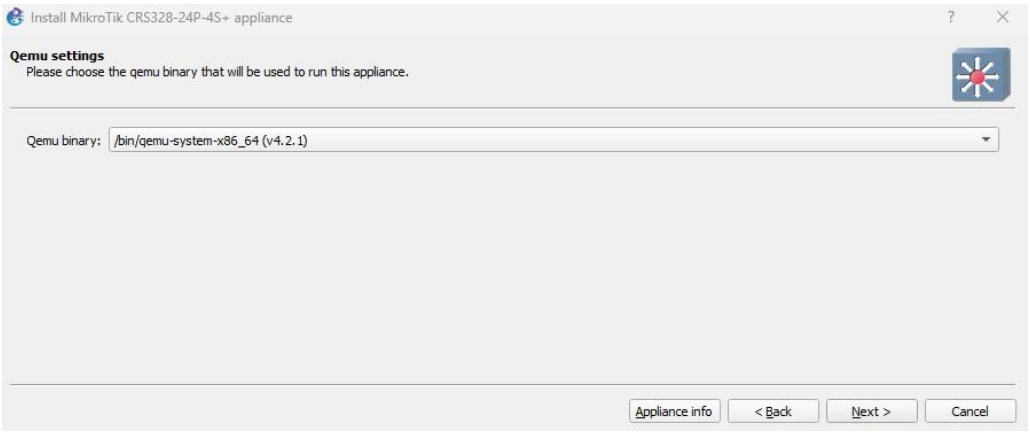


Fig. 61. Interfaz Gráfica. New Template. MikroTik CRS328-24P-4S. Quemu settings.

Luego llegaremos a la lista de versiones del MikroTik teniendo que seleccionar la imagen de la última versión, que al menos tendrá que ser la **7.10.**

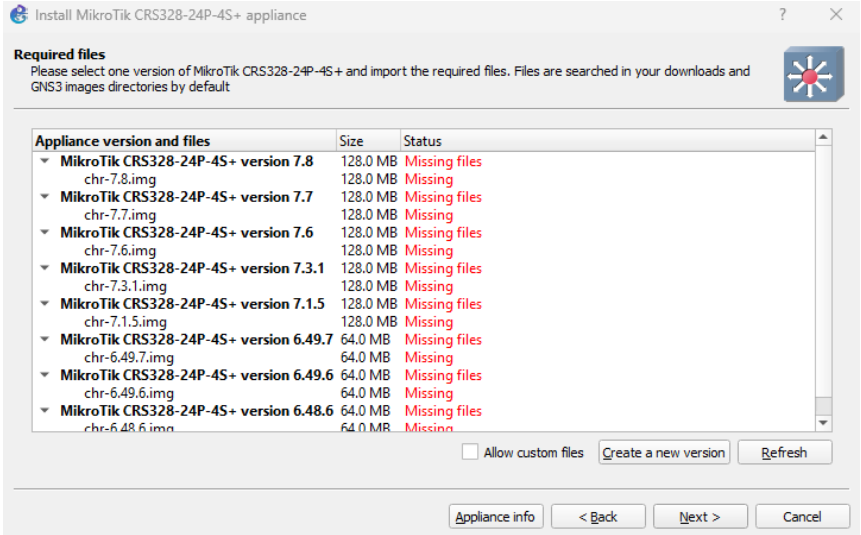


Fig. 62. Interfaz Gráfica. New Template. MikroTik CRS328-24P-4S. Lista versiones.

Si no está disponible la agregaremos manualmente, como se explica a continuación.

Desde la lista de *versiones*, seleccionaremos *Create new versión*, de tal forma que la versión quede de la siguiente manera:

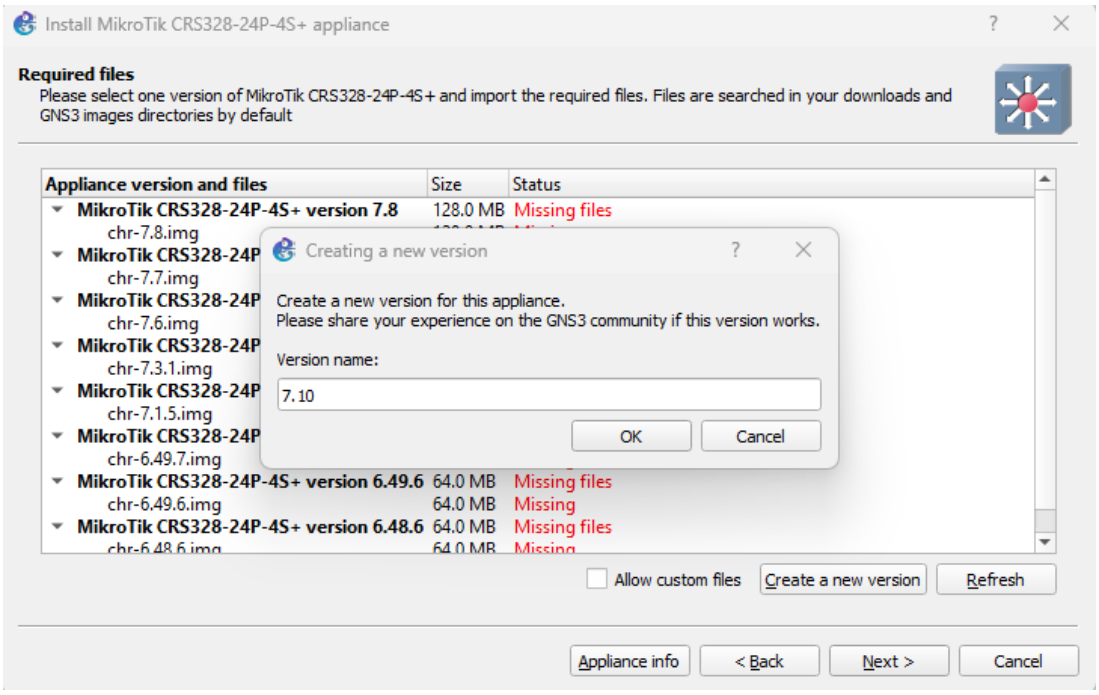


Fig. 63. Interfaz Gráfica. New Template. MikroTik CRS328-24P-4S. Nueva versión (1).

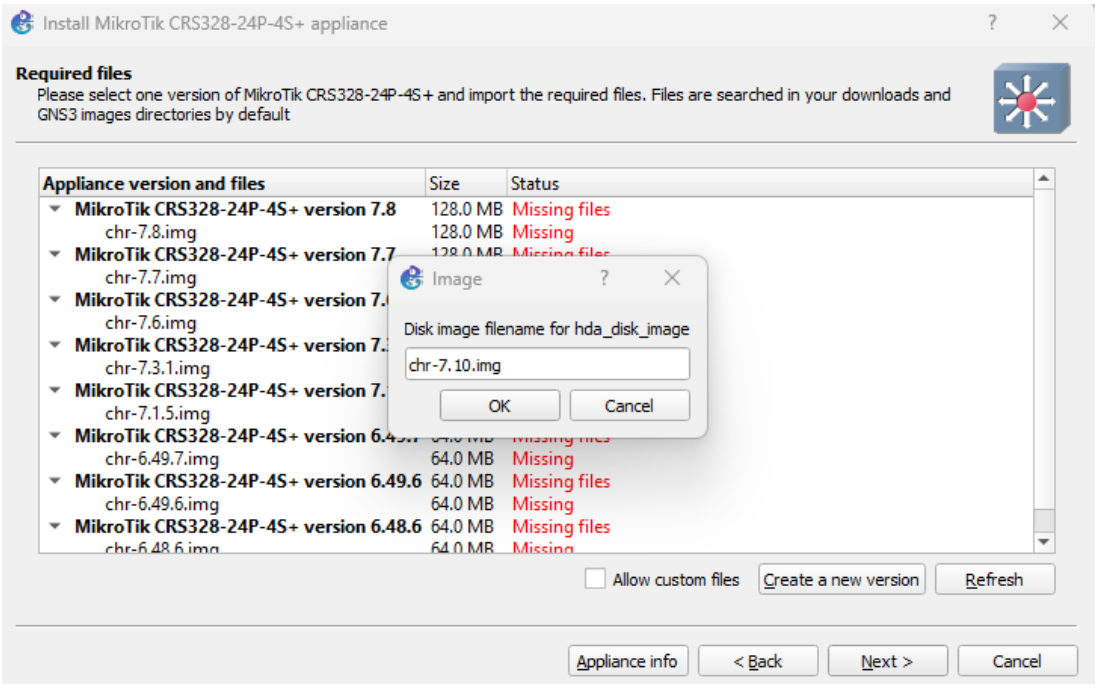


Fig. 64. Interfaz Gráfica. New Template. MikroTik CRS328-24P-4S. Nueva versión (2).

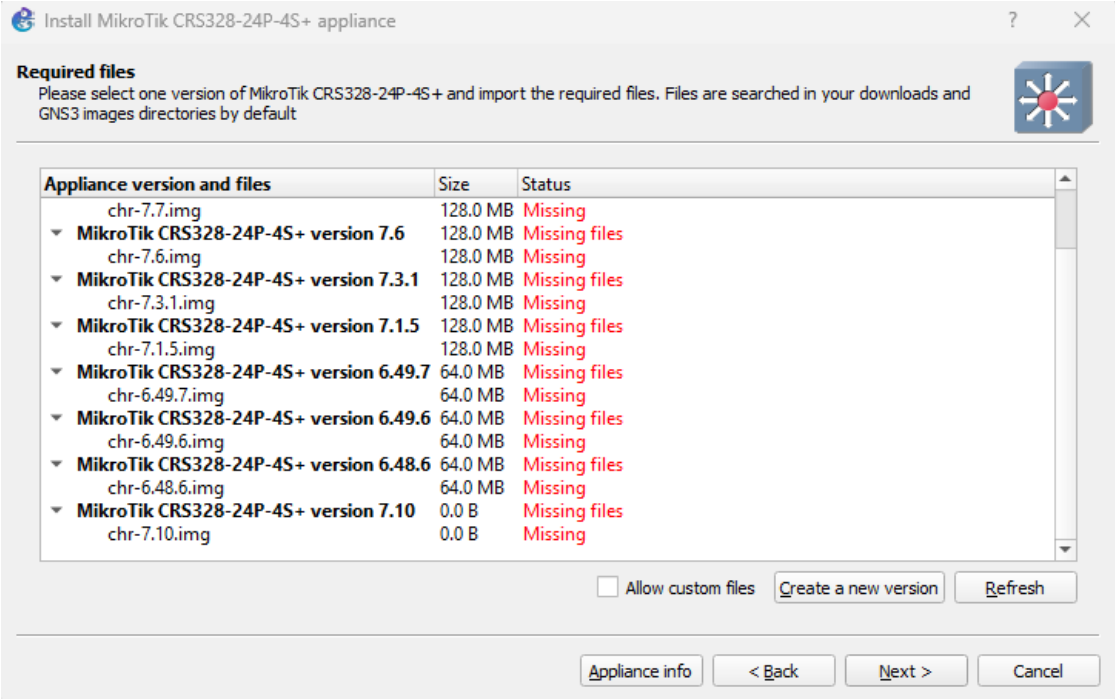


Fig. 65. Interfaz Gráfica. New Template. MikroTik CRS328-24P-4S. Nueva versión (3).

Ahora tendremos que acceder a la web de MikroTik, concretamente a la pestaña de descargas de software, donde seleccionaremos sobre el archivo de *Raw disk image* de la versión que nos interesa.

En este caso el archivo concreto lo podemos encontrar en el siguiente enlace:

https://download.mikrotik.com/routeros/7.10/chr-7.10.img.zip

Se descargará un archivo comprimido que tendremos que descomprimir y abrir desde GNS3 con *chr-7.10.img* seleccionado y clicando en *import* para buscar el archivo en cuestión.

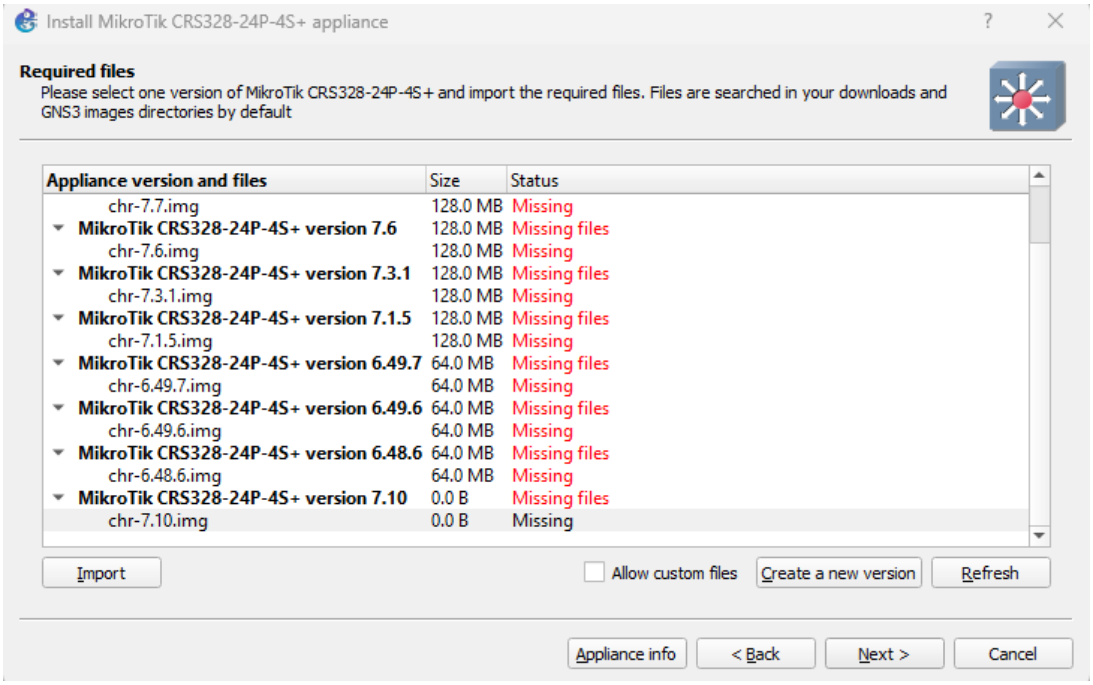


Fig. 66. Interfaz Gráfica. New Template. MikroTik CRS328-24P-4S. Required files. Importar imagen.

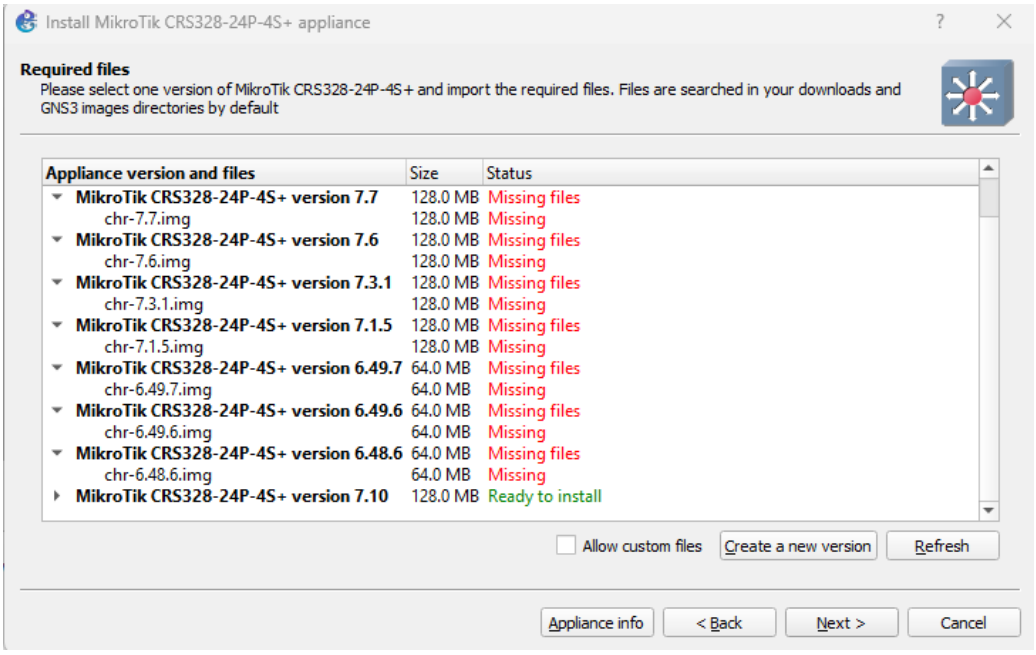


Fig. 67. Interfaz Gráfica. New Template. MikroTik CRS328-24P-4S. Required files. Imagen cargada.

Ahora ya tenemos lista la nueva versión para seleccionarla y seleccionar *Next* para completar la configuración.

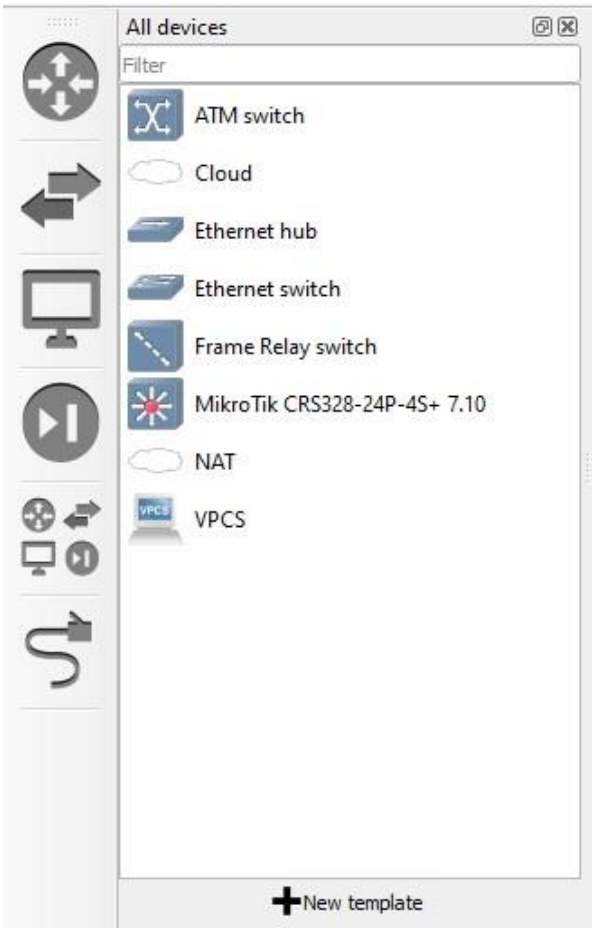


Fig. 68. GNS3. Interfaz Gráfica. Barra de elementos. MikroTik configurado.

2.4.4. Wireshark. Captura de tráfico

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

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

Utilizando los equipos físicos del laboratorio, utilizaremos este mismo software disponible en todos los PCs del laboratorio. Para su utilización es necesario hacer un uso correcto de las interfaces físicas de los PCs y que se detalla en profundidad en el desarrollo de cada práctica.

3. Prácticas en el laboratorio con Routers MikroTik

3.1. Práctica 1 "Configuración Básica de una Red MPLS"

3.1.1. Introducción

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

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

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

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

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

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

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

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

3.1.2. Etiqueta MPLS

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

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

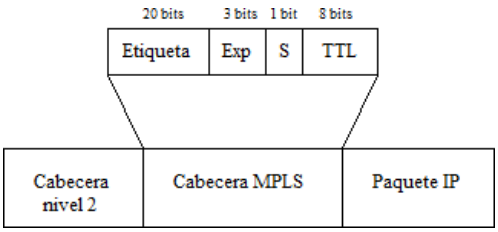


Fig. 69. Formato de cabecera MPLS.

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

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

3.1.3. Elementos MPLS

3.1.3.1. Forwarding Equivalence Class (FEC)

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

3.1.3.2. Label Switched Path (LSP)

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

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

3.1.3.3. Label Switch Routers (LSR)

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

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

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

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

3.1.3.4. Label Edge Routers (LER)

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

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

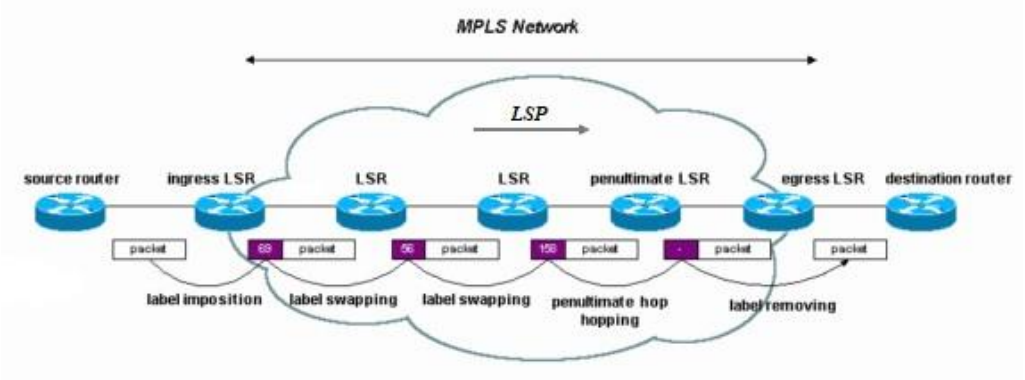


Fig. 70. Topología red MPLS.

3.1.4. Distribución de etiquetas

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

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

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

Existen varios mecanismos para la distribución:

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

En el caso de la primera práctica, utilizaremos LDP.

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

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

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

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

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

Los LSRs anuncian las direcciones de sus interfaces con mensajes "Address", o retiran las ya anunciadas con "Address Withdraw". Tras estos mensajes, se envían entre ellos "Label Request" para solicitar el mapeado de un FEC (un FEC puede ser una IP de un LSR) y responden con "Label Mapping", anunciando el mapeado de una etiqueta al FEC.

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

Los mensajes LDP se pueden clasificar en cuatro tipos:

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

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

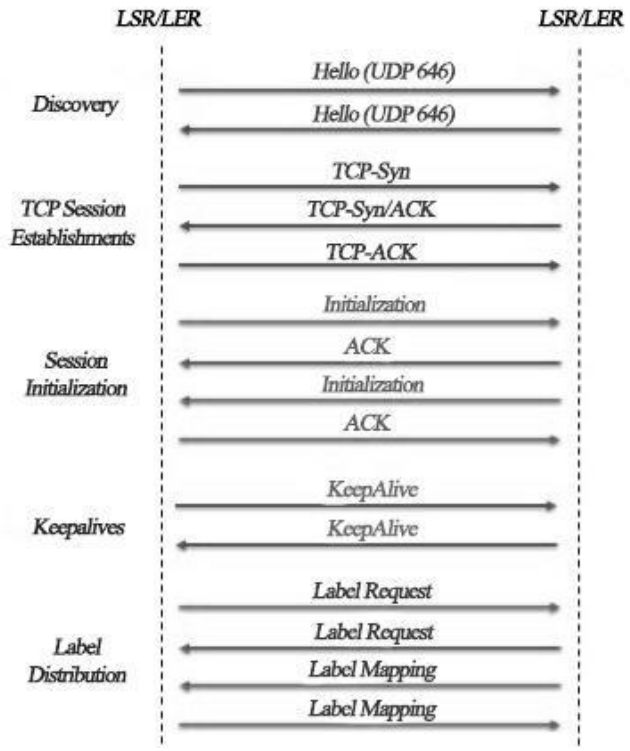


Fig. 71. Operaciones LDP.

3.1.5. Objetivos

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 MikroTik.

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 por medio de WireShark.

3.1.6. Elementos necesarios

Para la realización de la presente práctica se utilizarán los routers MikroTik disponibles en el laboratorio. En la *figura 72* podemos ver una imagen del mencionado router y en la *tabla 1* la descripción de cada uno de los elementos presentes en el mismo.

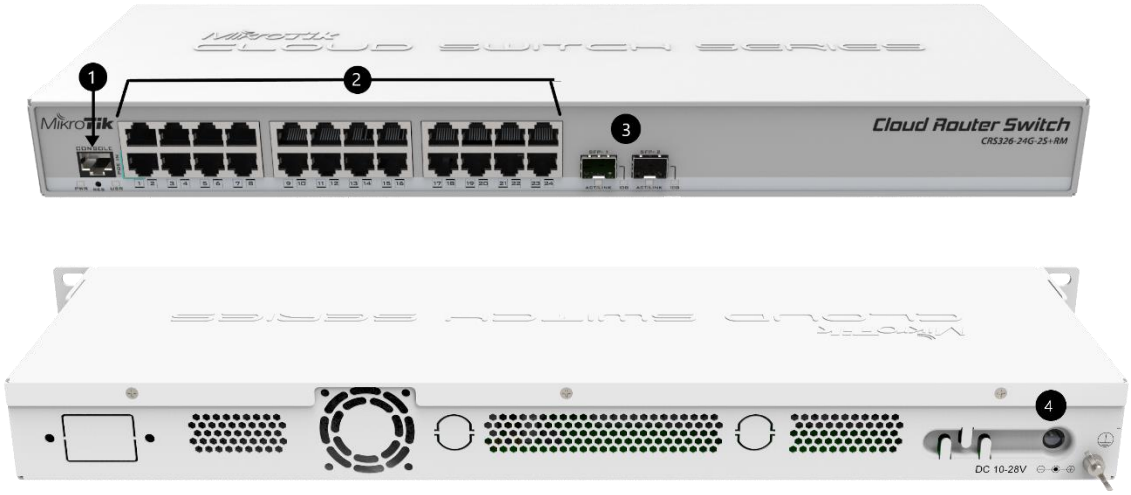


Fig. 72. Imagenes Router MikroTik.

ID	DESCRIPCIÓN
1	Puerto de consola
2	24 puertos Gigabit Ethernet
3	2 puertos 10G SPF+
4	Entrada de alimentación

Tabla 1. Descripción elementos del router

3.1.7. Topología de red

La topología de red a montar es la siguiente:

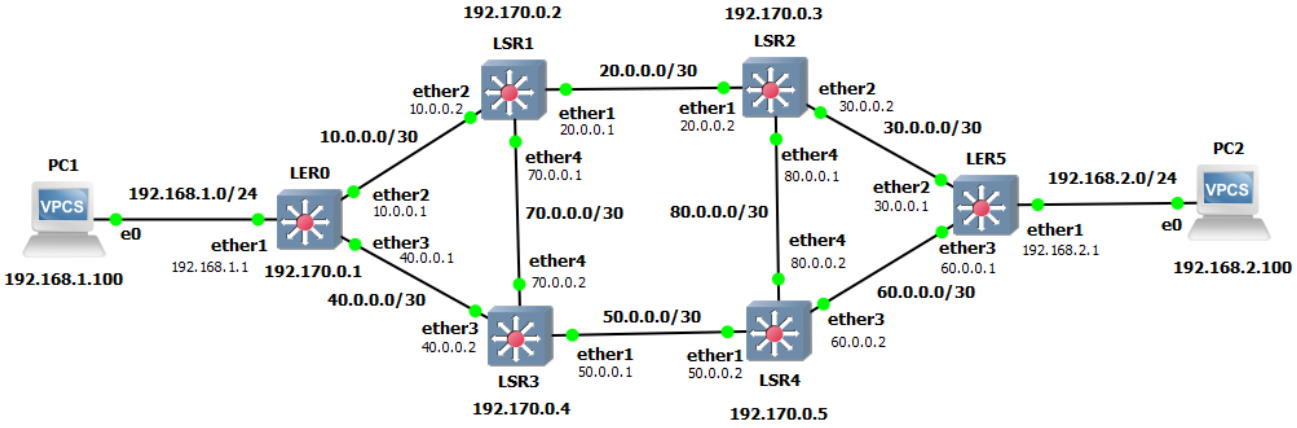


Fig. 73. Topología red MPLS Práctica 1.

En la *tabla 2* se muestran las direcciones IP y las máscaras de red de cada interfaz de los distintos equipos de la red.

Como se puede observar en la tabla, todos los routers utilizan 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.

Dispositivo	Interfaz	Dirección IP	Máscara de red	Gateway predeterminado
LER0	Lo0	192.170.0.1	255.255.255.255	N/A
	Ether1	192.168.1.1	255.255.255.0	N/A
	Ether2	10.0.0.1	255.255.255.252	N/A
	Ether3	40.0.0.1	255.255.255.252	N/A
LSR1	Lo0	192.170.0.2	255.255.255.255	N/A
	Ether1	20.0.0.1	255.255.255.252	N/A
	Ether2	10.0.0.2	255.255.255.252	N/A
	Ether4	70.0.0.1	255.255.255.252	N/A
LSR2	Lo0	192.170.0.3	255.255.255.255	N/A
	Ether1	20.0.0.2	255.255.255.252	N/A
	Ether2	30.0.0.2	255.255.255.252	N/A
	Ether4	80.0.0.1	255.255.255.252	N/A

LSR3	Lo0	192.170.0.4	255.255.255.255	N/A
	Ether1	50.0.0.1	255.255.255.252	N/A
	Ether3	40.0.0.2	255.255.255.252	N/A
	Ether4	70.0.0.2	255.255.255.252	N/A
LSR4	Lo0	192.170.0.5	255.255.255.255	N/A
	Ether1	50.0.0.2	255.255.255.252	N/A
	Ether3	60.0.0.2	255.255.255.252	N/A
	Ether4	80.0.0.2	255.255.255.252	N/A
LER5	Lo0	192.170.0.6	255.255.255.255	N/A
	Ether1	192.168.2.1	255.255.255.0	N/A
	Ether2	30.0.0.1	255.255.255.252	N/A
	Ether3	60.0.0.1	255.255.255.252	N/A
PC1	E0	192.168.1.100	255.255.255.0	192.168.1.1
PC2	E0	192.168.2.100	255.255.255.0	192.168.2.1

Tabla 2. Tabla de direccionamiento.

Antes de montar la topología con los equipos del laboratorio es recomendable apuntarse el rango de direcciones MAC de cada router, ya que será necesario para acceder a ellos a través de Winbox. Este rango de direcciones MAC lo podemos encontrar en una pegatina localizada en la parte posterior de cada router.

Fig. 74. Rango de direcciones MAC.

3.1.8. Configuración de la red

3.1.8.1. Montaje

Teniendo la información de la topología, el direccionamiento IP y el rango de MACs, procedemos a realizar el montaje que nos permitirá iniciar la configuración de la red para esta práctica.

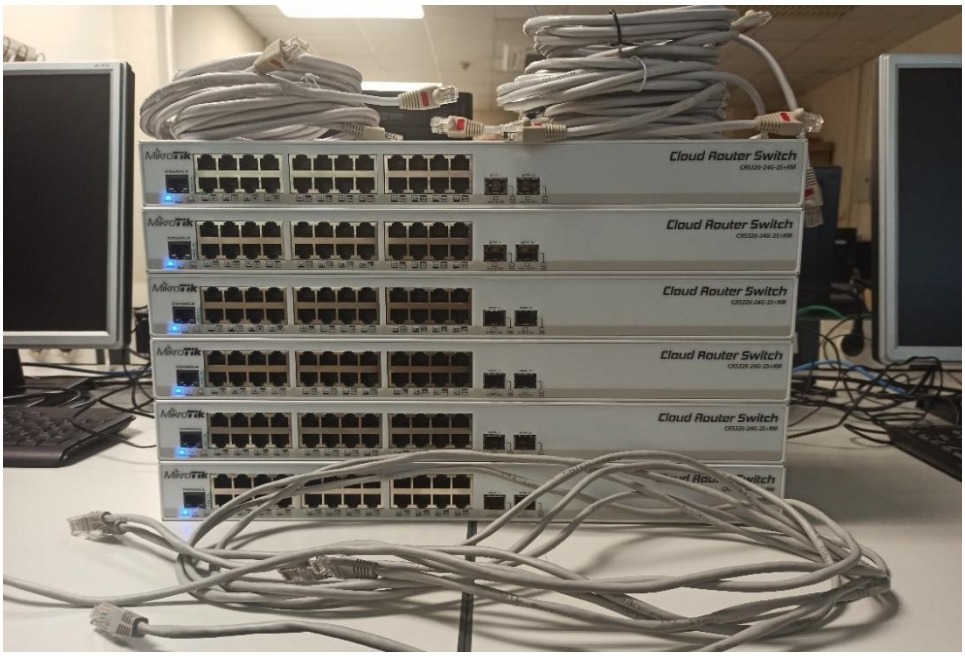


Fig. 75. Equipos y cableado previo al montaje.

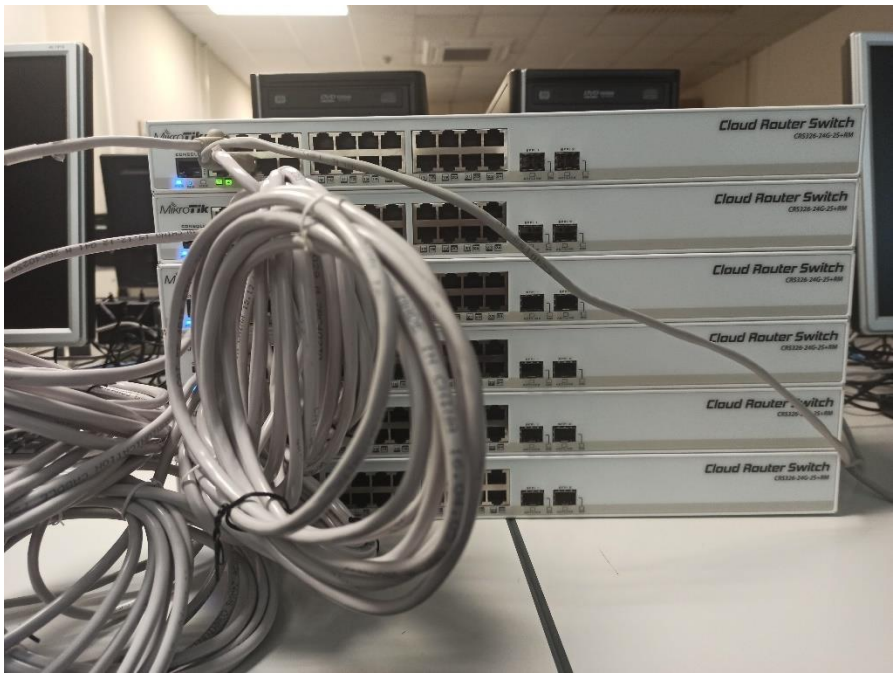


Fig. 76. Montaje de la red (I).

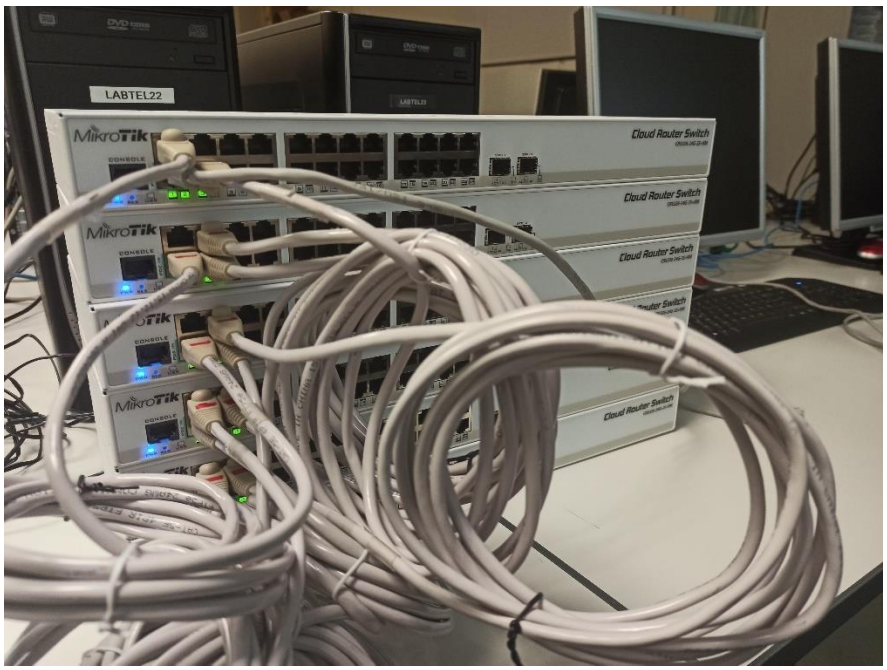


Fig. 77. Montaje de la red (II).

3.1.8.2. Acceso y borrado de la configuración

Una vez tengamos la topología montada y los routers operativos, procederemos a configurar los routers desde Winbox. Para ello conectaremos un cable Ethernet entre un PC y cualquiera de los puertos Gigabit Ethernet del router, aunque es recomendable usar siempre el mismo puerto para poder guardar la MAC del puerto asociándola con el nombre del router en WinBox, por ejemplo, usar el puerto 8 físico del router que está libre en los 6 equipos MikroTik.

Al entrar a Winbox, seleccionaremos la pestaña Neighbors donde veremos los 6 routers. Podemos observar que todos los routers utilizan la misma dirección IP, la 192.168.88.1 y que se diferencian por la dirección MAC del puerto por el que se accede a dicho router.

En el caso del LER0, la MAC que se observará para este router es del puerto ether8 que tenemos conectado, sin embargo, para el resto de routers, las direcciones MAC corresponden con los puertos que conectan el router LER0 con el resto. Por ello, recomendamos empezar a realizar la configuración con el LER 0 y terminar con el LER5 buscando siempre la dirección MAC correspondiente al ether 8, que será la 8ª dirección del rango observado en la pegatina del router.

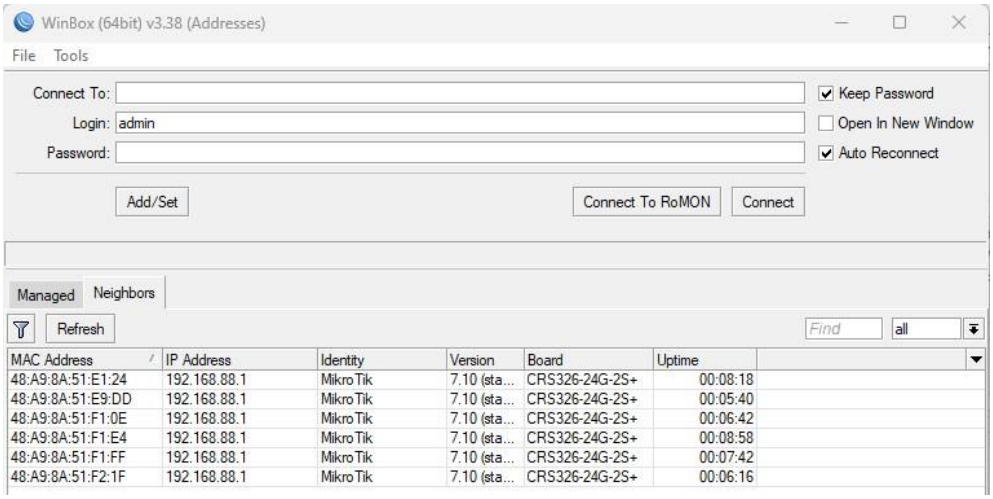


Fig. 78. Interfaz acceso a WinBox.

Antes de nada, seleccionaremos el menú *Tools* y habilitaremos el modo avanzado clicando en *Advanced Mode*.

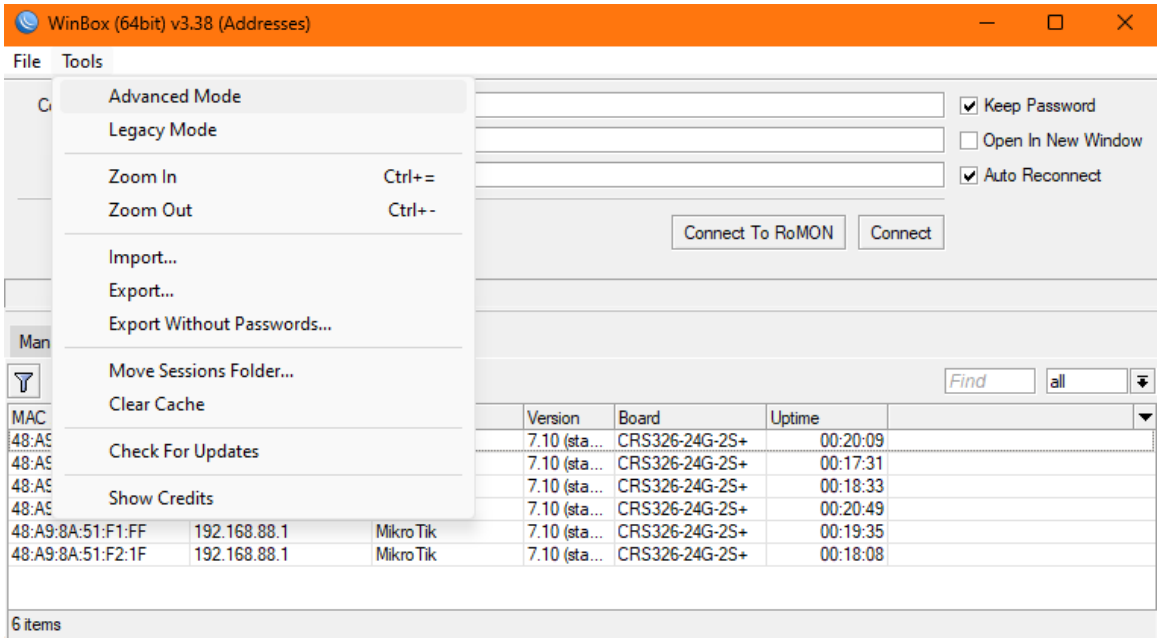


Fig. 79. Interfaz acceso a WinBox. Seleccionar Advanced Mode.

Luego buscamos la MAC que corresponde con el equipo que estamos configurando, asignándole un nombre y añadiéndolo a nuestra lista *Managed* al clicar sobre *Add/Set*. Una vez añadido seleccionaremos *Connect* para acceder al router.

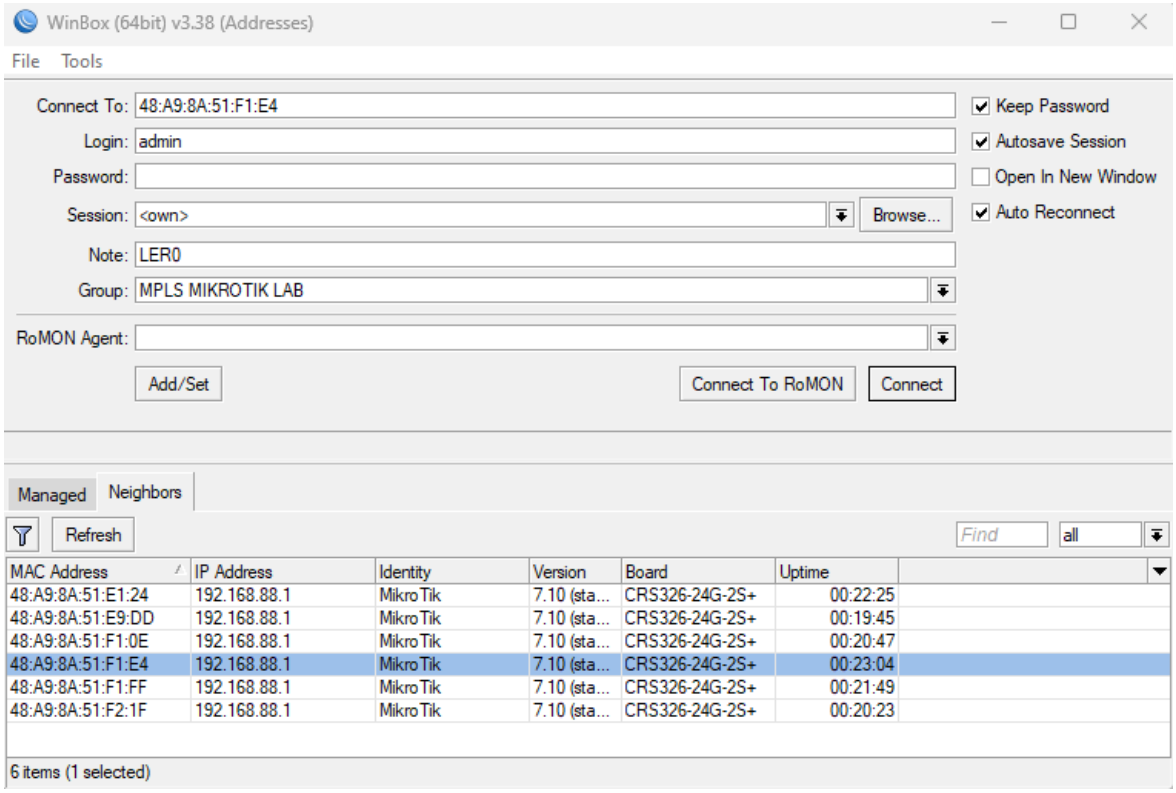


Fig. 80. Modo avanzodo de acceso al router.

El usuario es *admin* y la contraseña está en blanco por defecto. Es posible que el router tenga una configuración con una contraseña desconocida. Para poder acceder deberemos resetear el router de manera física manteniendo pulsado el botón de reset (se encuentra bajo el puerto de cosola) durante unos segundos hasta que todos los leds de los puertos se enciendan y apaguen de forma simultánea.

Al entrar al router nos preguntará si queremos borrar toda la configuración y lo haremos clicando en *Remove Configuration*. Seguidamente nos pedirá cambiar la contraseña, se recomienda utilizar *1234* para todos los routers. Esta contraseña la tendremos que actualizar en las direcciones guardadas en la pestaña *Managed* la próxima vez que accedamos.

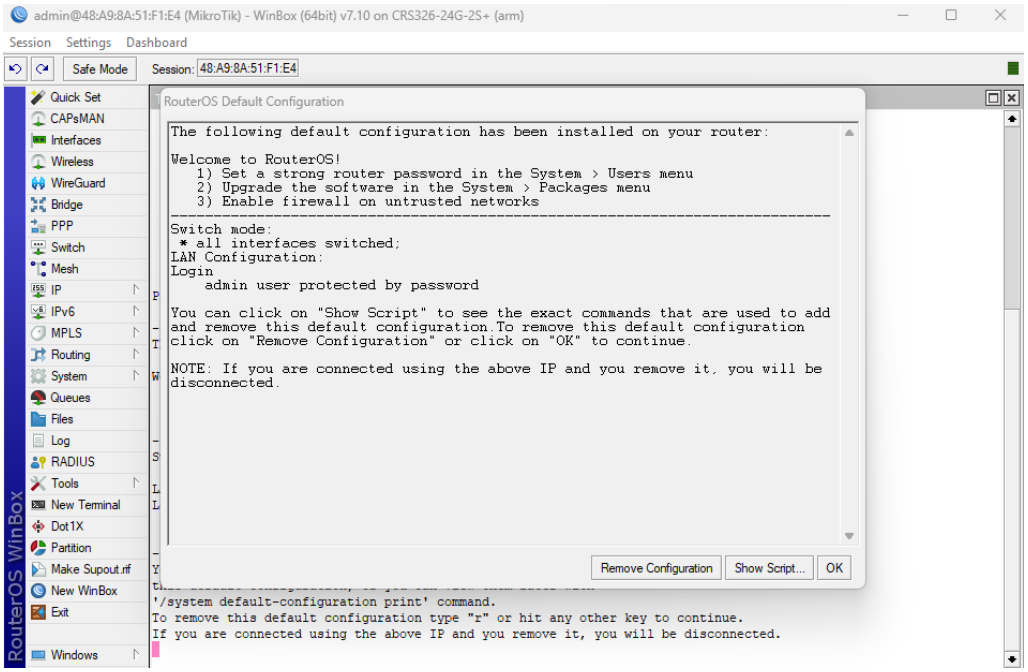


Fig. 81. Mensaje primer acceso en Winbox.

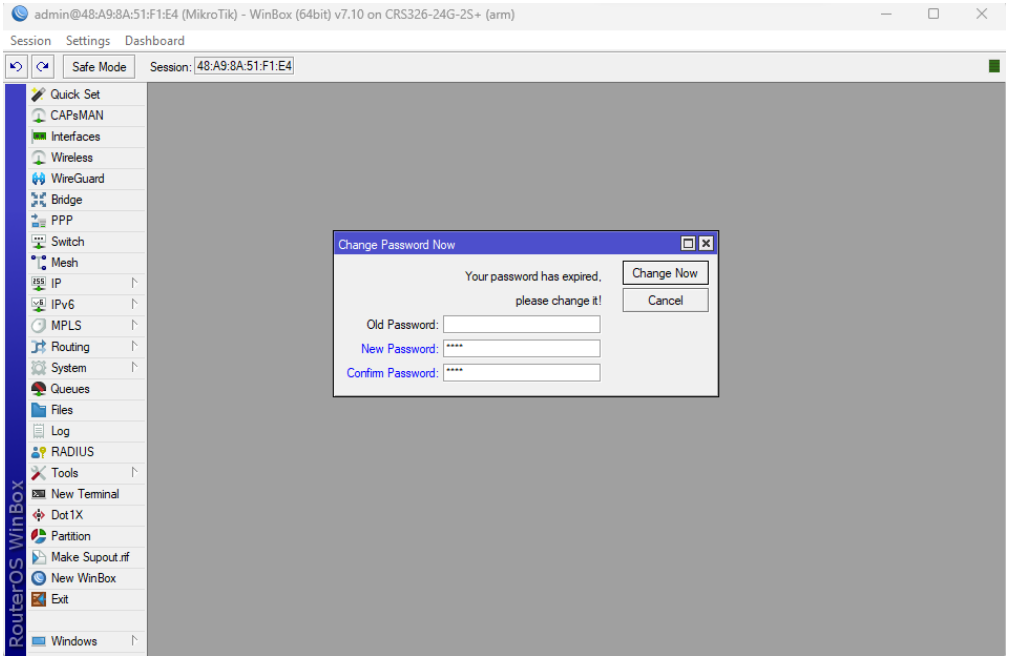


Fig. 82. Nueva contraseña del router.

Antes de configurar nada en el router LER0, realizaremos estos pasos en el resto de routers de forma que queden todos guardados en la pestaña *Manage* de Winbox y con la posible configuración anterior borrada. No hay que olvidar que hay que cambiar el cable *ethernet* al *puerto 8 físico* del router correspondiente.

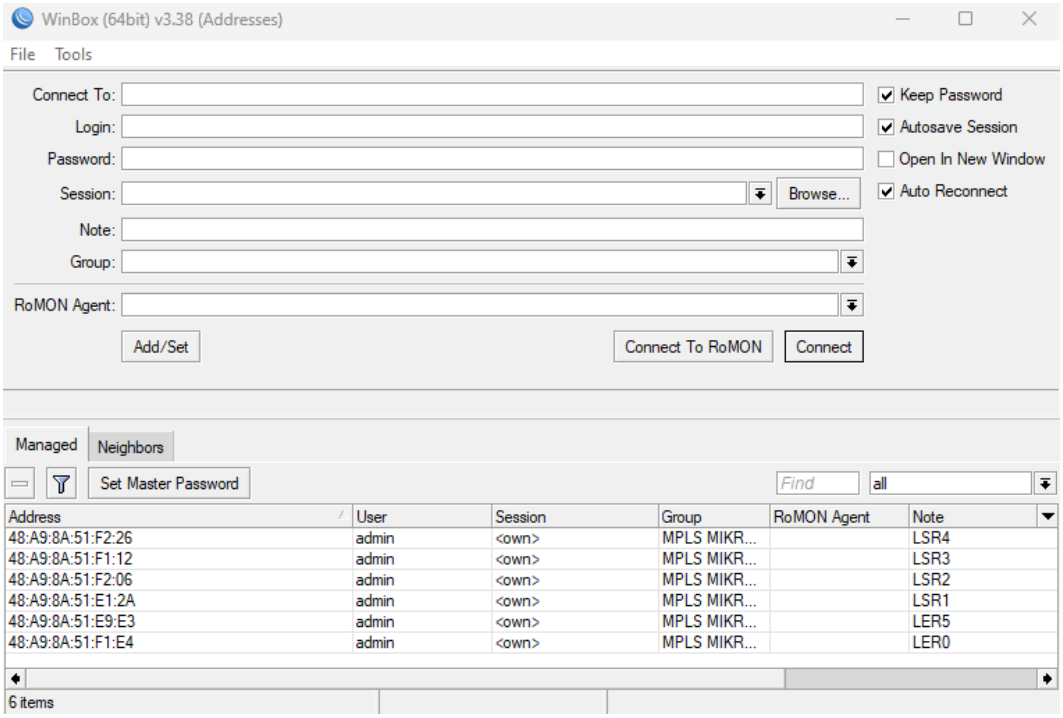


Fig. 83. Lista de routers guardados.

Con estos pasos realizados, como hemos dicho, volvemos a conectar el cable *ethernet* al *puerto 8 físico* del router LER0 y desde la pestaña *Managed* volvemos a seleccionar el router LER0, cambiamos el password a 1234 y la guardamos seleccionando *Add/Set.* Ahora ya podemos seleccionar en *Connect* y proceder a realizar las primeras configuraciones sobre el LER0 desde la ventana *Terminal* que se abrirá directamente sobre WinBox.

Para configurar el resto de routers realizaremos lo mismo, pero cambiando al *puerto 8 físico* del router correspondiete a configurar.

3.1.8.3. Crear Interfaz Loopback y asignar IP's

En primer lugar, vamos a crear la interfaz *Loopback* a través de la pestaña Bridge de Interfaces y que llamaremos *lo0.* Ejecutaremos los siguientes comandos en el router LER0:

```
[admin@MikroTik] > /interface bridge
[admin@MikroTik] /interface/bridge > add name=lo0
```

Con la interfaz *Loopback* creada, asignaremos las direcciones IP a cada interfaz del router con los siguientes comandos:

```
[admin@MikroTik] > /ip address
[admin@MikroTik] /ip/address > add
address: 192.170.0.1/32
interface: lo0
[admin@MikroTik] /ip/address > add
address: 192.168.1.1/24
interface: ether1
[admin@MikroTik] /ip/address > add
address: 10.0.0.1/30
interface: ether2
[admin@MikroTik] /ip/address > add
address: 40.0.0.1/30
interface: ether3
```

Si hemos realizado correctamente los pasos, en el terminal de Winbox debería quedar algo similar a la imagen siguiente:

Fig. 84. Configuraciónm de IP's en MikroTik.

Podemos comprobar que se han asignado bien las IP's ejecutando el comando *print:*

```
[admin@MikroTik] /ip/address> print
Columns: ADDRESS, NETWORK, INTERFACE
# ADDRESS          NETWORK      INTERFACE
0 192.170.0.1/32   192.170.0.1  lo0
1 192.168.1.1/24   192.168.1.0  ether1
2 10.0.0.1/30      10.0.0.0     ether2
3 40.0.0.1/30      40.0.0.0     ether3
[admin@MikroTik] /ip/address>
```

Fig. 85. IP's asignadas en MikroTik.

Habrá que repetir de manera análoga estos pasos para el resto de routers.

3.1.8.4. Configurar OSPF

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.

Primero crearemos una *instancia* única en el router que permite que existan múltiples instancias OSPF en un enrutador, lo que permite separar y gestionar diferentes áreas de tu red de manera independiente.

También crearemos un *área* OSPF, que es un grupo de routers que comparten información del estado de enlace. A las diferentes subredes a las que se conecta cada router le asignaremos dicha área.

A continuación, se muestra la configuración para el router LER0:

```
[admin@MikroTik] > /routing ospf instance
[admin@MikroTik] /routing/ospf/instance > add name=backbone router-id=192.170.0.1
[admin@MikroTik] /routing/ospf/instance > .. area
[admin@MikroTik] /routing/ospf/area > add name=backbone area-id=0.0.0.0 instance=backbone
[admin@MikroTik] /routing/ospf/area > .. interface-template
[admin@MikroTik] /routing/ospf/interface-template > add interface=lo0 networks=192.170.0.1/32 area=backbone
[admin@MikroTik] /routing/ospf/interface-template > add int=ether1 net=192.168.1.0/24 ar=backbone
[admin@MikroTik] /routing/ospf/interface-template > add int=ether2 net=10.0.0.0/30 ar=backbone
[admin@MikroTik] /routing/ospf/interface-template > add int=ether3 net=40.0.0.0/30 ar=backbone
```

Si se han seguido bien los pasos, en el terminal de Winbox debería quedar algo similar a la siguiente imagen:

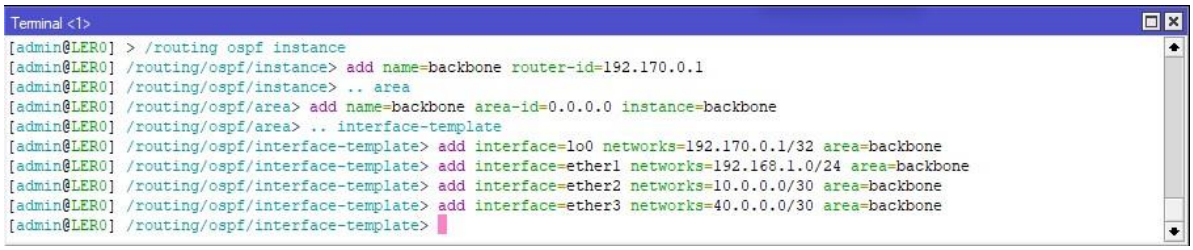


Fig. 86. Configuración OSPF en MikroTik.

Análogamente configuraremos el resto de routers con el protocolo OSPF.

Para comprobar que se ha configurado el protocolo OSPF con normalidad se puede acceder a */ip route* y ejecutar el comando *print* para comprobar los destinos aprendidos a través de OSPF y su puerta de enlace correspondiente. Así mismo también podemos ver los destinos directamente conectados.

```
[admin@MikroTik] /ip/route> print
Flags: D - DYNAMIC; A - ACTIVE; c, o, y - BGP-MPLS-VPN; + - ECMP
Columns: DST-ADDRESS, GATEWAY, DISTANCE
     DST-ADDRESS      GATEWAY           DISTANCE
DAc  10.0.0.0/30      ether2                   0
DAo  20.0.0.0/30      10.0.0.2%ether2        110
DAo  30.0.0.0/30      10.0.0.2%ether2        110
DAc  40.0.0.0/30      ether3                   0
DAo  50.0.0.0/30      40.0.0.2%ether3        110
DAo  60.0.0.0/30      40.0.0.2%ether3        110
DAo+ 70.0.0.0/30      10.0.0.2%ether2        110
DAo+ 70.0.0.0/30      40.0.0.2%ether3        110
DAo+ 80.0.0.0/30      10.0.0.2%ether2        110
DAo+ 80.0.0.0/30      40.0.0.2%ether3        110
DAc  192.168.1.0/24   ether1                   0
DAo+ 192.168.2.0/24   10.0.0.2%ether2        110
DAo+ 192.168.2.0/24   40.0.0.2%ether3        110
DAc  192.170.0.1/32   lo0                      0
DAo  192.170.0.2/32   10.0.0.2%ether2        110
DAo  192.170.0.3/32   10.0.0.2%ether2        110
DAo  192.170.0.4/32   40.0.0.2%ether3        110
DAo  192.170.0.5/32   40.0.0.2%ether3        110
DAo+ 192.170.0.6/32   10.0.0.2%ether2        110
DAo+ 192.170.0.6/32   40.0.0.2%ether3        110
```

Fig. 87. Lista rutas aprendidas mediante OSPF.

En la primera columna podemos observar las diferentes subredes configuradas en nuestra red, mientras que en la segunda columna observamos la dirección IP y/o la *interface* por la que se accede a dicha subred desde el router en el que se ejecuta la tabla, en este caso el LER0.

También podemos observas como se han aprendido. Por ejemplo, DAc indica que se ha aprendido dinámicamente y que el enlace está activo y directamente conectado. Por otro lado, DAo indica que se ha aprendido dinámicamente por medio de OSPF y que el enlace está activo.

En la última columna, podemos observar la distancia, esta nos evidencia de nuevo como se ha aprendido cada subred, ya que la distancia 110 es la distancia administrativa asociada al protocolo OSPF. Mientras que la distancia 0, ausencia de distancia, es la asociada a una conexión directa.

3.1.8.5. Configurar PC's

Con el protocolo OSPF funcionando ya deberíamos poder tener acceso a cualquier equipo de la red desde cualquier equipo de la misma red. Para comprobarlo haremos *ping* y *tracer* entre los dos PCs.

Para configurar los PC's tendremos que acceder al *Panel de Control* del ordenador y modificar la *dirección IP* con su *máscara* y la *puerta de enlace predeterminada* del *Adaptador Ethernet del laboratorio.*

Para hacer ***ping*** y ***tracer*** accederemos al terminal *cmd de Windows*. También podremos hacer ***ping*** y ***traceroute*** desde el terminal de WinBox de cualquier router con los comandos que se especifican a continuación:

```
[admin@MikroTik] > /tool
[admin@MikroTik] /tool > ping 192.168.2.100
[admin@MikroTik] /tool > traceroute 192.168.2.100
```

3.1.8.6. Configurar etiquetado LDP

Una vez tenemos OSPF configurado y todos los equipos accesibles, configuraremos la red MPLS. En esta primera práctica nos limitaremos a configurar el etiquetado por medio de LDP.

Primero tendremos que indicar el tipo de IPs que utilizaremos, en este caso IPv4, que lo haremos con la variable *afi=ip*. También identificaremos el LSR con la dirección IP y la dirección que utilizará el router para transportar los paquetes LDP. Ambas direcciones serán la misma, la asignada a la interfaz *LoopBack*.

Para terminar de configurar el etiquetado, asignaremos los puertos Ethernet que utilizarán el protocolo LDP con el resto de routers.

A continuación, se muestra la configuración para el router LER0:

```
[admin@MikroTik] > /mpls ldp
[admin@MikroTik] /mpls/ldp > add afi=ip lsr-id=192.170.0.1 transport-addresses=192.170.0.1
[admin@MikroTik] /mpls/ldp > interface
[admin@MikroTik] /mpls/ldp/interface > add
interface: ether2
[admin@MikroTik] /mpls/ldp/interface > add
interface: ether3
```

Si hemos realizado bien la configuración, el terminal de Winbox debería quedar de forma similar a la siguiente imagen:

Fig. 88. Configuración MPLS en MikroTik.

También habrá que asignar un rango de etiquetas a cada router para poder observar con más claridad el funcionamiento del protocolo, sabiendo en cada momento que etiqueta es de cada router.

Hay que tener en cuenta que las primeras 15 etiquetas están reservadas y que el rango mínimo es de 1024 labels. Con esta información realizaremos la siguiente asignación:

ROUTER	RANGO ETIQUETAS
LER0	16-9999 (min 16)
LSR1	10000-19999
LSR2	20000-29999
LSR3	30000-39999
LSR4	40000-49999
LER5	50000-59999

Fig. 89. Rango de etiquetas por router.

El comando para el LER0 es el siguiente:

```
[admin@MikroTik] > /mpls settings
[admin@MikroTik] /mpls/settings > set dynamic-label-range=16-9999
```

Habrá que realizar esta misma configuración de manera análoga con el resto de routers. Y una vez lo tengamos podremos hacer diferentes comprobaciones del buen funcionamiento de la configuración.

Con todo esto hecho, podremos consultar los routers vecinos y las direcciones IP's de las interfaces de estos con el siguiente comando desde el contexto ***/mpls/ldp***:

```
[admin@MikroTik] /mpls/ldp > nei
[admin@MikroTik] /mpls/ldp/neighbor > print
```

A continuación, podemos ver la ejecución del comando en el router LSR1:

```
[admin@MikroTik] > mpls ldp
[admin@MikroTik] /mpls/ldp> neighbor print
Flags: D, I - INACTIVE; O, T - THROTTLED; p - PASSIVE
Columns: TRANSPORT, LOCAL-TRANSPORT, PEER, ADDRESSES
#     TRANSPORT    LOCAL-TRANSPORT  PEER           ADDRESSES
0 DO  192.170.0.1  192.170.0.2      192.170.0.1:0  10.0.0.1        } LER0
                                                   40.0.0.1
                                                   192.168.1.1
                                                   192.170.0.1
1 DOp 192.170.0.3  192.170.0.2      192.170.0.3:0  20.0.0.2        } LSR2
                                                   30.0.0.2
                                                   80.0.0.1
                                                   192.170.0.3
2 DOp 192.170.0.4  192.170.0.2      192.170.0.4:0  40.0.0.2        } LSR3
                                                   50.0.0.1
                                                   70.0.0.2
                                                   192.170.0.4
[admin@MikroTik] /mpls/ldp>
```

Fig. 90. Tabla neighbor de LDP.

Podemos observar la dirección de trasporte local que hemos asignado para LSR1 que corresponde a la dirección *loopback* y las que hemos asignado en el resto de routers con los que se conecta. También podemos ver las direcciones asignadas a las diferentes interfaces de los routers a los que está conectado el LSR1.

También una vez configurada la red es posible acceder a la table LIB, aunque para tener toda la información de dicha tabla hay que ejecutar dos comandos. Ambos comandos se ejecutarán desde el contexto ***/mpls/ldp***. Con el comando ***local-mapping print*** visualizaremos el mapeo local del protocolo LDP. Y con el comando ***remote-mapping print*** visualizaremos el mapeo remoto del protocolo LDP.

A continiación se muestran ambos mapeos del router LSR1:

```
[admin@MikroTik] /mpls/ldp> local-mapping print
Flags: I - INACTIVE; D - DYNAMIC; E - EGRESS; G - GATEWAY; L - LOCAL
Columns: VRF, DST-ADDRESS, LABEL, PEERS
 #       VRF   DST-ADDRESS      LABEL       PEERS
 0 IDE L main  10.0.0.0/30      impl-null   192.170.0.1:0
                                            192.170.0.3:0
                                            192.170.0.4:0
 1 IDE L main  20.0.0.0/30      impl-null   192.170.0.1:0
                                            192.170.0.3:0
                                            192.170.0.4:0
 2   D G  main  30.0.0.0/30      10000       192.170.0.1:0
                                            192.170.0.3:0
                                            192.170.0.4:0
 3   D G  main  40.0.0.0/30      10001       192.170.0.1:0
                                            192.170.0.3:0
                                            192.170.0.4:0
 4   D G  main  50.0.0.0/30      10002       192.170.0.1:0
                                            192.170.0.3:0
                                            192.170.0.4:0
 5   D G  main  60.0.0.0/30      10003       192.170.0.1:0
                                            192.170.0.3:0
                                            192.170.0.4:0
 6 IDE L main  70.0.0.0/30      impl-null   192.170.0.1:0
                                            192.170.0.3:0
                                            192.170.0.4:0
 7   D G  main  80.0.0.0/30      10004       192.170.0.1:0
                                            192.170.0.3:0
                                            192.170.0.4:0
 8   D G  main  192.168.1.0/24   10005       192.170.0.1:0
                                            192.170.0.3:0
                                            192.170.0.4:0
 9   D G  main  192.168.2.0/24   10006       192.170.0.1:0
                                            192.170.0.3:0
                                            192.170.0.4:0
10   D G  main  192.170.0.1      10007       192.170.0.1:0
                                            192.170.0.3:0
                                            192.170.0.4:0
11 IDE L main  192.170.0.2      impl-null   192.170.0.1:0
                                            192.170.0.3:0
                                            192.170.0.4:0
12   D G  main  192.170.0.3      10008       192.170.0.1:0
                                            192.170.0.3:0
                                            192.170.0.4:0
13   D G  main  192.170.0.4      10009       192.170.0.1:0
                                            192.170.0.3:0
                                            192.170.0.4:0
14   D G  main  192.170.0.5      10010       192.170.0.1:0
                                            192.170.0.3:0
                                            192.170.0.4:0
15   D G  main  192.170.0.6      10011       192.170.0.1:0
                                            192.170.0.3:0
                                            192.170.0.4:0
```

Fig. 91. Tabla local-mapping.

```
[admin@MikroTik] /mpls/ldp> remote-mapping print
Flags: I - INACTIVE; D - DYNAMIC
Columns: VRF, DST-ADDRESS, NE/THOP, LABEL, PEER
 #    VRF   DST-ADDRESS      NE/THOP    LABEL       PEER
 0 ID main  20.0.0.0/30                 16          192.170.0.1:0
 1 ID main  10.0.0.0/30                 impl-null   192.170.0.1:0
 2  D main  40.0.0.0/30     10.0.0.1   impl-null   192.170.0.1:0
 3 ID main  50.0.0.0/30                 21          192.170.0.1:0
 4 ID main  30.0.0.0/30                 24          192.170.0.1:0
 5 ID main  192.170.0.2                 17          192.170.0.1:0
 6  D main  192.170.0.1     10.0.0.1   impl-null   192.170.0.1:0
 7 ID main  192.170.0.3                 18          192.170.0.1:0
 8 ID main  192.170.0.4                 20          192.170.0.1:0
 9 ID main  192.170.0.5                 23          192.170.0.1:0
10 ID main  192.170.0.6                 26          192.170.0.1:0
11 ID main  70.0.0.0/30                 19          192.170.0.1:0
12 ID main  80.0.0.0/30                 22          192.170.0.1:0
13 ID main  60.0.0.0/30                 25          192.170.0.1:0
14  D main  192.168.1.0/24  10.0.0.1   impl-null   192.170.0.1:0
15 ID main  192.168.2.0/24              27          192.170.0.1:0
16 ID main  192.170.0.2                 20008       192.170.0.3:0
17 ID main  192.170.0.1                 20007       192.170.0.3:0
18  D main  192.170.0.3     20.0.0.2   impl-null   192.170.0.3:0
19 ID main  192.170.0.4                 20009       192.170.0.3:0
20  D main  192.170.0.5     20.0.0.2   20010       192.170.0.3:0
21 ID main  20.0.0.0/30                 impl-null   192.170.0.3:0
22 ID main  10.0.0.0/30                 20000       192.170.0.3:0
23 ID main  70.0.0.0/30                 20004       192.170.0.3:0
24 ID main  40.0.0.0/30                 20001       192.170.0.3:0
25 ID main  50.0.0.0/30                 20002       192.170.0.3:0
26  D main  80.0.0.0/30     20.0.0.2   impl-null   192.170.0.3:0
27  D main  30.0.0.0/30     20.0.0.2   impl-null   192.170.0.3:0
28  D main  60.0.0.0/30     20.0.0.2   20003       192.170.0.3:0
29 ID main  192.168.1.0/24              20005       192.170.0.3:0
30  D main  192.168.2.0/24  20.0.0.2   20006       192.170.0.3:0
31  D main  192.170.0.6     20.0.0.2   20011       192.170.0.3:0
32 ID main  192.170.0.2                 30008       192.170.0.4:0
33 ID main  192.170.0.1                 30007       192.170.0.4:0
34 ID main  192.170.0.3                 30009       192.170.0.4:0
35  D main  192.170.0.4     70.0.0.2   impl-null   192.170.0.4:0
36  D main  192.170.0.5     70.0.0.2   30010       192.170.0.4:0
37 ID main  192.170.0.6                 30011       192.170.0.4:0
38 ID main  20.0.0.0/30                 30001       192.170.0.4:0
39 ID main  10.0.0.0/30                 30000       192.170.0.4:0
40 ID main  70.0.0.0/30                 impl-null   192.170.0.4:0
41  D main  40.0.0.0/30     70.0.0.2   impl-null   192.170.0.4:0
42  D main  50.0.0.0/30     70.0.0.2   impl-null   192.170.0.4:0
43 ID main  80.0.0.0/30                 30004       192.170.0.4:0
44 ID main  30.0.0.0/30                 30002       192.170.0.4:0
45  D main  60.0.0.0/30     70.0.0.2   30003       192.170.0.4:0
46 ID main  192.168.1.0/24              30005       192.170.0.4:0
47 ID main  192.168.2.0/24              30006       192.170.0.4:0
```

Recibido de LER0

Recibido de LSR2

Recibido de LSR3

Fig. 92. Tabla remote-mapping.

En la tabla ***Local Mapping*** podemos encontrar las direcciones destino y la etiqueta local que se añadirá en la cabecera del paquete que salga hacia el destino en cuestión, así como los peers con los que está conectado el router.

Podemos observar tantas entradas en esta tabla, ya que desde el router LSR1 se asigna una etiqueta para cada subred, ya que esta etiqueta será utilizada para que el LSR1 reciba los paquetes procedentes del resto de subredes.

La columna de peers indica las direcciones de loopback de los routers que hay conectados al router LSR1, por eso podemos observar que siempre son las mismas direcciones de loopback para todas las entradas de la tabla.

Para las subredes 10.0.0.0, 20.0.0.0 y 70.0.0.0 observamos que no se asigna etiqueta ya que están directamente conectadas al router LSR1 y no precisa de esta para recibir paquetes desde dichas subredes.

En la tabla ***Remote Mapping*** podemos localizar junto con la red de destino, la etiqueta que añadirá el router remoto a los paquetes que envíe, el siguiente salto para alcanzar su destino y el peer que corresponde con el camino elegido de los que sean posibles.

Podemos observar tantas entradas en la tabla ya que para un mismo destino existe la posibilidad de acceder a él por diferentes rutas, es decir teniendo un siguiente salto diferente y por tanto una etiqueta asignada también diferente.

En la columna de las etiquetas podemos observar la variedad de estas en cuanto a rango, ya que estas etiquetas son las asignadas por los diferentes routers, que como se ha comentado ya son de diferente rango para una mejor visualización de la trazabilidad de los paquetes hacia un router concreto en cuestión.

Desde el contexto ***/mpls*** podemos ejecutar el comando ***forwarding-table print*** que nos permitirá visualizar la tabla LFIB. Esta tabla es la encargada de realizar la conmutación de paquetes a través de la red.

En la siguiente imagen se muestra la correspondiente al router LSR1:

```
[admin@MikroTik] /mpls/ldp> .. forwarding print
Flags: L, V - VPLS
Columns: LABEL, VRF, PREFIX, NEXTHOPS
 #    LABEL  VRF   PREFIX          NEXTHOPS
 0 L 10001  main  40.0.0.0/30     { label=impl-null; nh=70.0.0.2; interface=ether4 }
                                  { label=impl-null; nh=10.0.0.1; interface=ether2 }
 1 L 10007  main  192.170.0.1     { label=impl-null; nh=10.0.0.1; interface=ether2 }
 2 L 10005  main  192.168.1.0/24  { label=impl-null; nh=10.0.0.1; interface=ether2 }
 3 L 10010  main  192.170.0.5     { label=30010; nh=70.0.0.2; interface=ether4 }
                                  { label=20010; nh=20.0.0.2; interface=ether1 }
 4 L 10003  main  60.0.0.0/30     { label=30003; nh=70.0.0.2; interface=ether4 }
                                  { label=20003; nh=20.0.0.2; interface=ether1 }
 5 L 10008  main  192.170.0.3     { label=impl-null; nh=20.0.0.2; interface=ether1 }
 6 L 10004  main  80.0.0.0/30     { label=impl-null; nh=20.0.0.2; interface=ether1 }
 7 L 10000  main  30.0.0.0/30     { label=impl-null; nh=20.0.0.2; interface=ether1 }
 8 L 10006  main  192.168.2.0/24  { label=20006; nh=20.0.0.2; interface=ether1 }
 9 L 10011  main  192.170.0.6     { label=20011; nh=20.0.0.2; interface=ether1 }
10 L 10009  main  192.170.0.4     { label=impl-null; nh=70.0.0.2; interface=ether4 }
11 L 10002  main  50.0.0.0/30     { label=impl-null; nh=70.0.0.2; interface=ether4 }
```

Fig. 93. Tabla fowarding de LDP.

Podemos observar de izquierda a derecha la etiqueta local, el identificador de destino, la etiqueta de salida, el siguiente salto y la interfaz de salida.

Las etiquetas de la columna LABEL son de un mismo rango, ya que como venimos diciendo son las asignadas localmente por el router para que lleguen a este los paquetes desde el resto de destinos.

En la columna PREFIX observamos los diferentes destinos a los que tiene acceso el router LSR1. Los destinos son las diferentes subredes creadas en la topología, aunque también se pueden observar las direcciones loopback de los routers de la red.

En la columna NEXTHOP se presentan tres datos referentes al siguiente salto para llegar a los destinos ya comentados. Primero se observa la etiqueta del siguiente salto que se añade a la cabecera del paquete. Junto a esta etiqueta se muestra el interfaz de salida a modo de dirección IP y a modo de nombre del propio interfaz.

3.1.8.7. Guardar configuración

Para guardar la configuración ejecutamos el siguiente comando, que generará un archivo en la memoria flash con el script de todo lo que hemos configurado:

```
[admin@MikroTik]  > export file=flash/myconfig.rsc
```

Para volver a la configuración deseada ante un reseteo del router, en el comando anterior habrá que sustituir *export* por *import*.

3.1.9. Ejercicios propuestos

1. **Utilizando el comando traceroute, determinar cuántas rutas pueden seguir los paquetes desde el PC-1 hasta el PC-2. Una vez obtenido filtrar las rutas para verlas individualmente y determinar el camino que siguen, esto es posible añadiendo interface = etherx al comando original siendo x el número de la interfaz de salida desde LER0.**

```
[admin@MikroTik] /tool> traceroute 192.168.2.100
Columns: ADDRESS, LOSS, SENT, LAST, AVG, BEST, WORST, STD-DEV,
STATUS
# ADDRESS      LOSS SENT LAST  AVG BEST WORST STD-DEV STATUS
1 40.0.0.2     0%   320 0.5ms 0.5 0.4  0.8   0       <MPLS:L=30006,E=0>
  10.0.0.2
2 20.0.0.2     0%   320 0.3ms 0.4 0.3  1.2   0.1     <MPLS:L=20006,E=0>
  50.0.0.2
3 30.0.0.1     0%   320 0.3ms 0.3 0.2  0.5   0
  60.0.0.1
4 192.168.2.100 0%   320 0.5ms 0.5 0.4  3.1   0.2
```

Utilizando el comando traceroute podemos determinar que existen dos posibles rutas ya que observamos dos entradas en cada salto.

```
[admin@MikroTik] /tool> traceroute interface = ether2 192.168.2.100
Columns: ADDRESS, LOSS, SENT, LAST, AVG, BEST, WORST, STD-DEV,
STATUS
# ADDRESS      LOSS SENT LAST  AVG BEST WORST STD-DEV STATUS
1 10.0.0.2     0%   320 0.5ms 0.5 0.4  0.8   0       <MPLS:L=10006,E=0>
2 20.0.0.2     0%   320 0.3ms 0.4 0.3  1.2   0.1     <MPLS:L=20006,E=0>
3 30.0.0.1     0%   320 0.3ms 0.3 0.2  0.5   0

4 192.168.2.100 0%   320 0.5ms 0.5 0.4  3.1   0.2
```

Por la interfaz ether2, los paquetes de PC1 a PC2 tendrán el primer salto en LSR1 usando la etiqueta 10006, siendo el siguiente salto el LSR2 cambiando la etiqueta 10006 por la 20006, siendo el último salto de la red MPLS el router LER5 que elimna la etiqueta al ser el PHP.

```
[admin@MikroTik] /tool> traceroute interface = ether3 192.168.2.100
Columns: ADDRESS, LOSS, SENT, LAST, AVG, BEST, WORST, STD-DEV,
STATUS
# ADDRESS      LOSS SENT LAST   AVG  BEST WORST STD-DEV STATUS
1 40.0.0.2     0%   320 0.5ms  0.5  0.4  0.8   0       <MPLS:L=30006,E=0>
2 50.0.0.2     0%   320 0.3ms  0.4  0.3  1.2   0.1     <MPLS:L=40006,E=0>
3 60.0.0.1     0%   320 0.3ms  0.3  0.2  0.5   0

4 192.168.2.100 0%   320 0.5ms  0.5  0.4  3.1   0.2
```

Por la interfaz ether3, los paquetes de PC1 a PC2 tendrán el primer salto en LSR3 usando la etiqueta 30006, siendo el siguiente salto el LSR4 cambiando la etiqueta 30006 por la 40006, siendo el último salto de la red MPLS el router LER5 que elimna la etiqueta al ser el PHP.

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. Ambas rutas que hemos observado con traceroute, servirán para repartir la carga en la red.

3. **Ejecuta en el contexto /mpls/ldp de alguno de los routers los comandos "local-mapping" y "remote-mapping", ¿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", "local-mapping" y "remote-mapping" del router LSR1, construir las tablas RIB, FIB, LIB y LFIB de forma similar a los ejercicios realizados en la teoría de la asignatura.**

```
[admin@LSR1] /mpls/ldp> local-mapping/print
Flags: I - INACTIVE; D - DYNAMIC; E - EGRESS; G - GATEWAY; L - LOCAL
Columns: VRF, DST-ADDRESS, LABEL, PEERS
 #      VRF   DST-ADDRESS     LABEL      PEERS
 0 IDE L main  10.0.0.0/30     impl-null  192.170.0.1:0
                                          192.170.0.3:0
                                          192.170.0.4:0
 1 IDE L main  20.0.0.0/30     impl-null  192.170.0.1:0
                                          192.170.0.3:0
                                          192.170.0.4:0
 2  D G  main  40.0.0.0/30     10000      192.170.0.1:0
                                          192.170.0.3:0
                                          192.170.0.4:0
 3 IDE L main  70.0.0.0/30     impl-null  192.170.0.1:0
                                          192.170.0.3:0
                                          192.170.0.4:0
 4  D G  main  192.168.1.0/24  10001       192.170.0.1:0
                                          192.170.0.3:0
                                          192.170.0.4:0
```

```
 5  D G  main  192.170.0.1     10002      192.170.0.1:0
                                          192.170.0.3:0
                                          192.170.0.4:0
 6 IDE L main  192.170.0.2     impl-null  192.170.0.1:0
                                          192.170.0.3:0
                                          192.170.0.4:0
 7  D G  main  30.0.0.0/30     10003      192.170.0.1:0
                                          192.170.0.3:0
                                          192.170.0.4:0
 8  D G  main  80.0.0.0/30     10004      192.170.0.1:0
                                          192.170.0.3:0
                                          192.170.0.4:0
 9  D G  main  192.170.0.3     10005      192.170.0.1:0
                                          192.170.0.3:0
                                          192.170.0.4:0
10  D G  main  50.0.0.0/30     10006      192.170.0.3:0
                                          192.170.0.1:0
                                          192.170.0.4:0
11  D G  main  192.170.0.4     10007      192.170.0.3:0
                                          192.170.0.1:0
                                          192.170.0.4:0
12  D G  main  60.0.0.0/30     10008      192.170.0.3:0
                                          192.170.0.4:0
                                          192.170.0.1:0
13  D G  main  192.170.0.5     10009      192.170.0.4:0
                                          192.170.0.1:0
                                          192.170.0.3:0
14  D G  main  192.170.0.6     10010      192.170.0.1:0
                                          192.170.0.3:0
                                          192.170.0.4:0
15  D G  main  192.168.2.0/24  10011      192.170.0.3:0
                                          192.170.0.1:0
                                          192.170.0.4:0


[admin@LSR1] /mpls/ldp> remote-mapping print
Flags: I - INACTIVE; D - DYNAMIC
Columns: VRF, DST-ADDRESS, NEXTHOP, LABEL, PEER
 #    VRF   DST-ADDRESS     NEXTHOP   LABEL      PEER
 0 ID main  10.0.0.0/30                impl-null  192.170.0.1:0
 1 ID main  20.0.0.0/30                16         192.170.0.1:0
 2 ID main  70.0.0.0/30                17         192.170.0.1:0
 3  D main  40.0.0.0/30     10.0.0.1   impl-null  192.170.0.1:0
 4 ID main  192.170.0.2                18         192.170.0.1:0
 5  D main  192.170.0.1     10.0.0.1   impl-null  192.170.0.1:0
 6  D main  192.168.1.0/24  10.0.0.1   impl-null  192.170.0.1:0
 7 ID main  30.0.0.0/30                20         192.170.0.1:0
 8 ID main  80.0.0.0/30                21         192.170.0.1:0
 9 ID main  192.170.0.3                19         192.170.0.1:0
10 ID main  10.0.0.0/30                20000      192.170.0.3:0
11 ID main  20.0.0.0/30                impl-null  192.170.0.3:0
12 ID main  70.0.0.0/30                20002      192.170.0.3:0
13 ID main  40.0.0.0/30                20001      192.170.0.3:0
14  D main  30.0.0.0/30     20.0.0.2   impl-null  192.170.0.3:0
15  D main  80.0.0.0/30     20.0.0.2   impl-null  192.170.0.3:0
```

```
16 ID main  192.170.0.2                20005      192.170.0.3:0
17 ID main  192.170.0.1                20004      192.170.0.3:0
18  D main  192.170.0.3      20.0.0.2  impl-null  192.170.0.3:0
19 ID main  192.168.1.0/24             20003      192.170.0.3:0
20 ID main  50.0.0.0/30                20006      192.170.0.3:0
21 ID main  192.170.0.4                20007      192.170.0.3:0
22 ID main  50.0.0.0/30                23         192.170.0.1:0
23 ID main  192.170.0.4                22         192.170.0.1:0
24 ID main  10.0.0.0/30                30000      192.170.0.4:0
25 ID main  20.0.0.0/30                30001      192.170.0.4:0
26 ID main  70.0.0.0/30                impl-null  192.170.0.4:0
27  D main  40.0.0.0/30      70.0.0.2  impl-null  192.170.0.4:0
28 ID main  30.0.0.0/30                30002      192.170.0.4:0
29 ID main  80.0.0.0/30                30003      192.170.0.4:0
30  D main  50.0.0.0/30      70.0.0.2  impl-null  192.170.0.4:0
31 ID main  192.170.0.2                30006      192.170.0.4:0
32 ID main  192.170.0.1                30005      192.170.0.4:0
33 ID main  192.170.0.3                30007      192.170.0.4:0
34  D main  192.170.0.4      70.0.0.2  impl-null  192.170.0.4:0
35 ID main  192.168.1.0/24             30004      192.170.0.4:0
36  D main  60.0.0.0/30      20.0.0.2  20008      192.170.0.3:0
37  D main  60.0.0.0/30      70.0.0.2  30009      192.170.0.4:0
38 ID main  60.0.0.0/30                25         192.170.0.1:0
39  D main  192.170.0.5      70.0.0.2  30008      192.170.0.4:0
40 ID main  192.170.0.5                24         192.170.0.1:0
41  D main  192.170.0.5      20.0.0.2  20009      192.170.0.3:0
42 ID main  192.170.0.6                26         192.170.0.1:0
43  D main  192.170.0.6      20.0.0.2  20010      192.170.0.3:0
44  D main  192.168.2.0/24   20.0.0.2  20011      192.170.0.3:0
45 ID main  192.168.2.0/24             27         192.170.0.1:0
46 ID main  192.170.0.6                30010      192.170.0.4:0
47 ID main  192.168.2.0/24             30011      192.170.0.4:0


[admin@LSR1] /mpls/forwarding-table> print
Flags: L, V - VPLS
Columns: LABEL, VRF, PREFIX, NEXTHOPS
 #   LABEL  VRF   PREFIX         NEXTHOPS
0 L 10001  main  40.0.0.0/30     { label=impl-null; nh=10.0.0.1; interface=ether2 }
                                 { nh=70.0.0.2; interface=ether4 }
1 L 10002  main  50.0.0.0/30     { nh=70.0.0.2; interface=ether4 }
2 L 10003  main  60.0.0.0/30     { nh=70.0.0.2; interface=ether4 }
                                 { label=20008; nh=20.0.0.2; interface=ether1 }
3 L 10007  main  192.170.0.1     { label=impl-null; nh=10.0.0.1; interface=ether2 }
4 L 10009  main  192.170.0.4     { nh=70.0.0.2; interface=ether4 }
5 L 10010  main  192.170.0.5     { nh=70.0.0.2; interface=ether4 }
                                 { label=20009; nh=20.0.0.2; interface=ether1 }
6 L 10005  main  192.168.1.0/24  { label=impl-null; nh=10.0.0.1; interface=ether2 }
7 L 10000  main  30.0.0.0/30     { label=impl-null; nh=20.0.0.2; interface=ether1 }
8 L 10004  main  80.0.0.0/30     { label=impl-null; nh=20.0.0.2; interface=ether1 }
9 L 10008  main  192.170.0.3     { label=impl-null; nh=20.0.0.2; interface=ether1 }
10 L 10011  main  192.170.0.6    { label=20010; nh=20.0.0.2; interface=ether1 }
11 L 10006  main  192.168.2.0/24  { label=20011; nh=20.0.0.2; interface=ether1 }
```

RIB

RED	SALTO
10.0.0.0/30	Dir. conectado
20.0.0.0/30	Dir. conectado
70.0.0.0/30	Dir. conectado
30.0.0.0/30	LSR2
40.0.0.0/30	LSR3
40.0.0.0/30	LER0
50.0.0.0/30	LSR3
60.0.0.0/30	LSR3
60.0.0.0/30	LSR2
80.0.0.0/30	LSR2
192.168.1.0/24	LER0
192.168.2.0/24	LSR2

Tabla 3. Tabla RIB LSR1.

FIB

RED	SALTO	LABEL
10.0.0.0/30	Dir. conectado	-
20.0.0.0/30	Dir. conectado	-
70.0.0.0/30	Dir. conectado	-
30.0.0.0/30	LSR2	-
40.0.0.0/30	LSR3	-
40.0.0.0/30	LER0	-
50.0.0.0/30	LSR3	-
60.0.0.0/30	LSR3	30003
60.0.0.0/30	LSR2	20003
80.0.0.0/30	LSR2	-
192.168.1.0/24	LER0	-
192.168.2.0/24	LSR2	20006

Tabla 4. Tabla FIB LSR1.

LIB

RED	LSR	LABEL
10.0.0.0/30	Dir. conectado	-
20.0.0.0/30	Dir. conectado	-
70.0.0.0/30	Dir. conectado	-
30.0.0.0/30	LOCAL	10000
40.0.0.0/30	LOCAL	10001
50.0.0.0/30	LOCAL	10002
60.0.0.0/30	LOCAL	10003
	LSR3	30003
60.0.0.0/30	LOCAL	10003
	LSR2	20003
80.0.0.0/30	LOCAL	10004
192.168.1.0/24	LOCAL	10005
192.168.2.0/24	LOCAL	10006
	LSR2	20006

Tabla 5. Tabla LIB LSR1.

LFIB

LABEL IN	LABEL OUT	ACTION	SALTO
10000	-	POP	LSR2
10001	-	POP	LSR3
10001	-	POP	LSR3
10002	-	POP	LSR3
10003	30003	SWAP	LSR3
10003	20003	SWAP	LSR2
10004	-	POP	LSR2
10005	-	POP	LER0
10006	20006	SWAP	LSR2

Tabla 6. Tabla LFIB LSR1.

En la última parte de la práctica pasaremos a utilizar el analizador de redes, para ello previamente nos valdremos de una de las funciones de nuestro router, "*port mirroring*". Esto permite enviar una copia del tráfico de unos de los puertos del router hacia otro y así poder conectar en este último un PC con "Wireshark" instalado y monitorizar el citado tráfico.

Capturaremos el tráfico de la interfaz *ether1*, tanto del router LSR1 como del LSR3, ya que los paquetes podrán elegir una ruta u otra para comunicar un PC con el otro, incluso enviar el *request* por un enlace y el *reply* por el otro. Utilizaremos la interfaz *ether7* para captar el tráfico.

Ejecutaremos las siguientes líneas tanto en LSR1 como en LSR3:

```
[admin@MikroTik] > /interface/ethernet/switch
[admin@MikroTik] /interface/ethernet/switch > set 0 name=switch mirror-source=ether1 mirror-target=ether7
```

Después de esto conectaremos los puertos a dos ordenadores que ejecutarán el *Wireshark* con el filtro *icmp* para mostrar los mensajes que se envían en un *ping*.

5. Ejecutar un ping desde PC1 a PC2. ¿Cómo aparecen encapsulados estos paquetes? ¿Puedes reconocer los campos MPLS vistos en teoría?

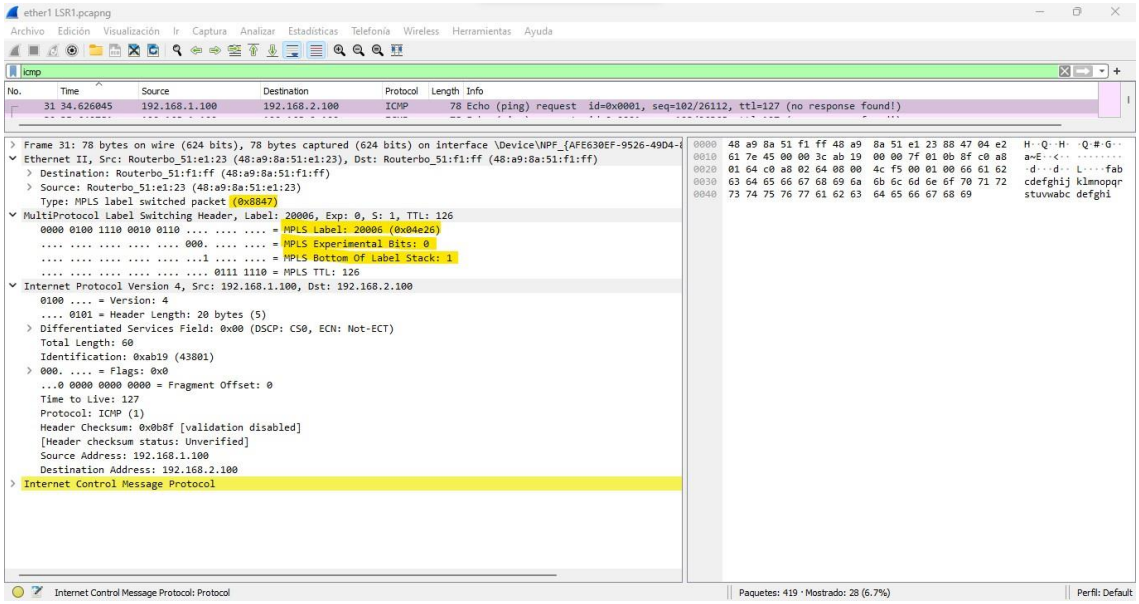


Fig. 94. Captura WireShark de ping entre PC1 y PC2

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 20006.
- 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

3.2. Práctica 2 "Configuración de una Red L3 MPLS VPN"

3.2.1. Introducción

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

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

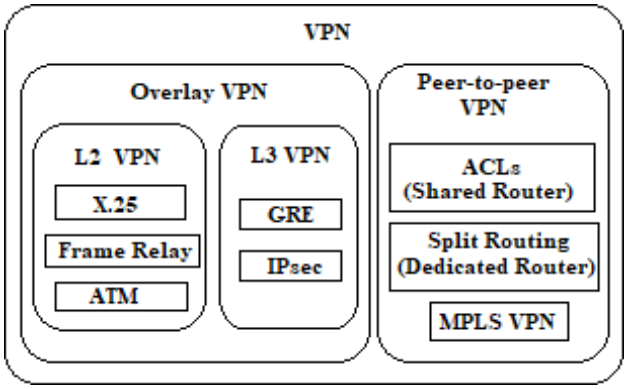


Fig. 95. Modelos VPN.

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

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

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

3.2.2. Componentes y arquitectura L3VPN

3.2.2.1. Customer Edge (CE)

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

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

3.2.2.2. Provider Edge (PE)

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

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

3.2.2.3. Provider (P)

Router MPLS en el *backbone* de la red. Nunca está conectado a la red cliente. No lleva rutas VPN, ya que solo posee información de la red del proveedor en sus tablas de routing.

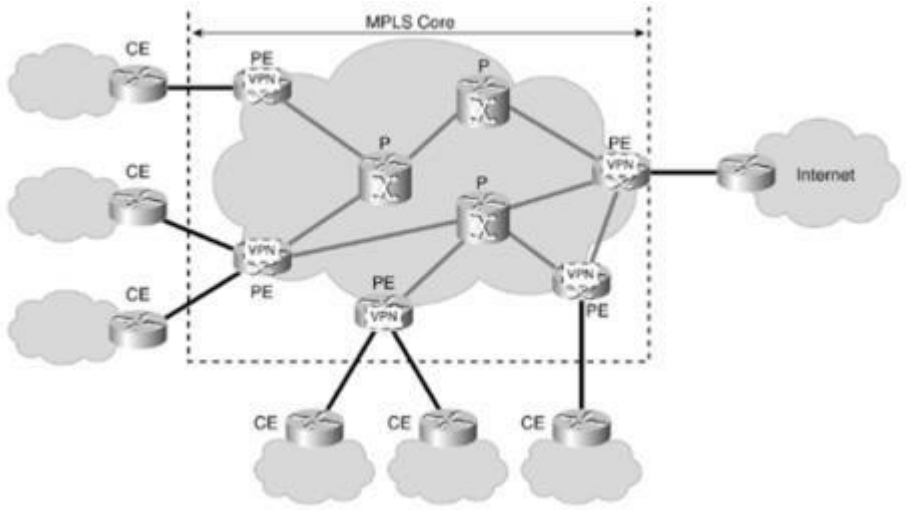


Fig. 96. Topología red L3 MPLS VPN

3.2.2.4. Virtual Routing Forwarding (VRF)

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

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

3.2.2.5. Route Distinguishers (RD)

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

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

3.2.2.6. Route Targets (RT)

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

Dos tipos de RT:

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

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

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

3.2.3. MP-BGP

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

Se puede clasificar de dos formas en función del AS:

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

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

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

3.2.4. Propagación de rutas y envío de paquetes en MPLS VPN

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

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

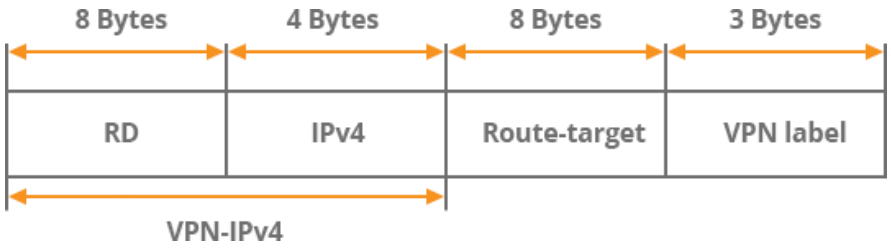


Fig. 97. Mensaje de actualización MP-BGP

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

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

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

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

3.2.5. Objetivos

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 MikroTik.

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.

3.2.6. Elementos necesarios

Para la realización de la presente práctica se utilizarán los routers MikroTik disponibles en el laboratorio. En la *figura 98* podemos ver una imagen del mencionado router y en la *tabla 7* la descripción de cada uno de los elementos presentes en el mismo.

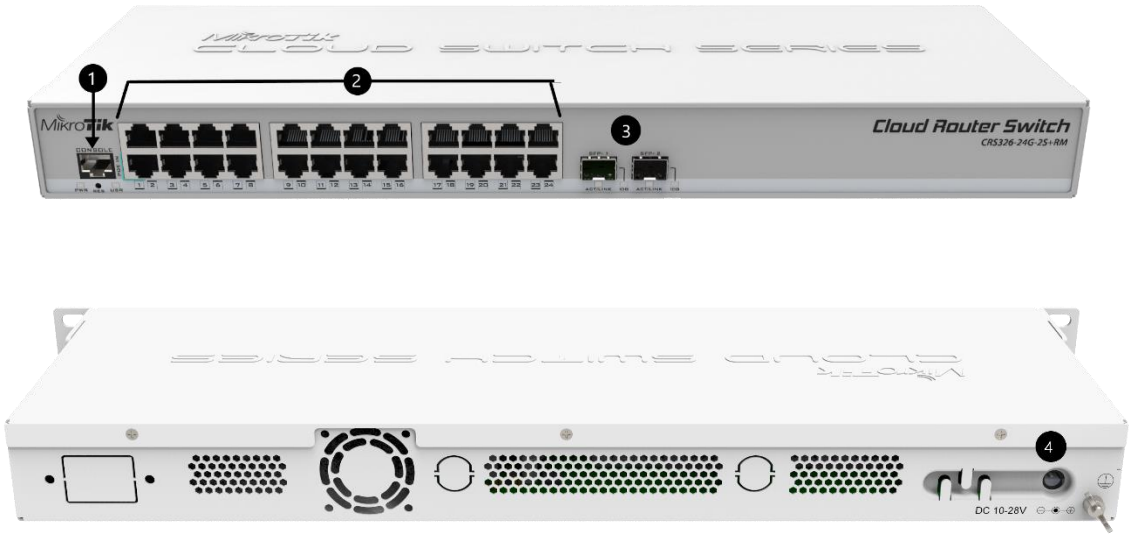


Fig. 98. Imágenes Router MikroTik.

ID	DESCRIPCIÓN
1	Puerto de consola
2	24 puertos Gigabit Ethernet
3	2 puertos 10G SPF+
4	Entrada de alimentación

Tabla 7. Descripción elementos del router.

3.2.7. Topología de red

Vamos a crear una red L3 VPN-MPLS formada por 5 routers Mikrotik CRS326-24G-2S+RM con las mismas propiedades que en las prácticas anteriores. En ella estableceremos una VPN con dos routers cliente CE, los cuales queremos comunicar.

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

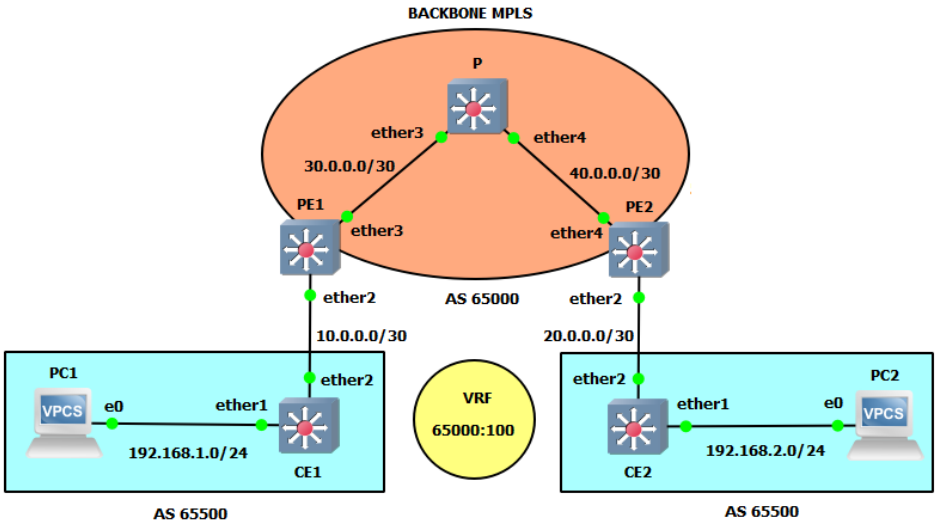


Tabla 8. Diagrama de la red L3 MPLS VPN.

Dispositivo	Interfaz	Dirección IP	Máscara de red	Gateway
CE1	Lo0	192.170.0.1	255.255.255.255	-
	Ether2	10.0.0.1	255.255.255.252	-
	Ether1	192.168.1.1	255.255.255.0	-
PE1	Lo0	192.170.0.2	255.255.255.255	-
	Ether2	10.0.0.2	255.255.255.252	-
	Ether3	30.0.0.2	255.255.255.252	-
P	Lo0	192.170.0.3	255.255.255.255	-
	Ether3	30.0.0.1	255.255.255.252	-
	Ether4	40.0.0.1	255.255.255.252	-
PE2	Lo0	192.170.0.4	255.255.255.255	-
	Ether4	40.0.0.2	255.255.255.252	-
	Ether2	20.0.0.2	255.255.255.252	-
CE2	Lo0	192.170.0.5	255.255.255.255	-
	Ether2	20.0.0.1	255.255.255.252	-
	Ether1	192.168.2.1	255.255.255.0	-
PC1	NIC	192.168.1.100	255.255.255.0	192.168.1.1
PC2	NIC	192.168.2.100	255.255.255.0	192.168.2.1

Tabla 9. Tabla de direccionamiento IP.

3.2.8. Configuración de la red

3.2.8.1. Nuevo montaje y borrado

Una vez tengamos el montaje de la nueva topología, procederemos a configurar nuestra red. De igual forma que en la Práctica 1, realizamos el borrado de la antigua configuración de los routers. Muy importante, más aún si cabe que en la primera práctica.

También podemos aprovechar los nombres guardados a través de las direcciones MAC de las prácticas anteriores (siempre que se hagan de seguido). Si no es posible volveremos a nombrar los routers a partir del rango de MACs de cada router.

3.2.8.2. Crear Interfaz Loopback y asignar IP's

Al igual que en las anteriores prácticas, primero asignaremos cada una de las direcciones de los interfaces y loopbacks de cada router. Ejecutaremos los siguientes comandos para el router PE1:

```
[admin@PE1] > int br
[admin@PE1] /interface/bridge > add name=lo0
[admin@PE1] /interface/bridge > /ip address
[admin@PE1] /ip/address > add
address: 192.170.0.2/32
interface: lo0
[admin@PE1] /ip/address > add
address: 10.0.0.2/30
interface: ether2
[admin@PE1] /ip/address > add
address: 30.0.0.2/30
interface: ether3
```

Para el resto de routers se replicará la configuración análogamente.

3.2.8.3. OSPF Backbone MPLS

Empezaremos aplicando el protocolo OSPF para aprender de forma dinámica las direcciones de los routers que pertenecen a MPLS, es decir, PE1, PE2 y P. Por tanto, las interfaces ether2 de los PE no se incluirán.

Router P:

```
[admin@P] > /routing ospf instance
[admin@P] /routing/ospf/instance > add name=backbone router-id=192.170.0.3
[admin@P] /routing/ospf/instance > .. area
[admin@P] /routing/ospf/area > add name=backbone area-id=0.0.0.0 inst=backbone
[admin@P] /routing/ospf/area > .. interface-template
[admin@P] /routing/ospf/interface-template > add int=lo0 net=192.170.0.3/32 area=backbone
[admin@P] /routing/ospf/interface-template > add int=ether3 net=30.0.0.1/30 ar=backbone
[admin@P] /routing/ospf/interface-template > add int=ether4 net=40.0.0.1/30 ar=backbone
```

Repetiremos análogamente para los routers PE.

3.2.8.4. LDP Backbone MPLS

A continuación, aplicaremos el protocolo de distribución de etiquetas (LDP) sobre estos mismos routers, los que pertenecen a la red MPLS. De nuevo no hay que configurar las interfaces que se comunican con los routers de fuera de la backbone.

Router PE1:

```
[admin@PE1] > /mpls ldp
[admin@PE1] /mpls/ldp > add afi=ip lsr-id=192.170.0.2 t=192.170.0.2
[admin@PE1] /mpls/ldp > interface
[admin@PE1] /mpls/ldp/interface > add
interface: ether3
```

Repetimos análogamente para PE2 y P.

Al igual que en la práctica anterior, podemos cambiar el rango de etiquetas dinámicas para una mayor claridad a la hora de entender el funcionamiento y estructura de los paquetes que circulan por la red.

ROUTER	RANGO ETIQUETAS
PE1	10000-19999
PE2	20000-29999
P	30000-39999

Tabla 10. Rango de etiquetas por routers.

Recuerda que el comando utilizado será el siguiente:

/mpls/settings set dy=<rango>

Se puede comprobar que todo se ha realizado correctamente realizando *traceroute* desde PE1 a PE2. Se tendrá que ver la asignación de una etiqueta, ya que en el segundo salto actúa el PHP. También se pueden consultar las tablas *local-mapping, remote-mapping, forwarding-table* y *neighbor.*

3.2.8.5. MP-BGP

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

Primero actualizaremos la plantilla *default* de BGP en el router, acorde al AS a utilizar y a las familias de direcciones que enrutaran nuestros routers PE. Un AS o Sistema Autónomo consiste en un conjunto de redes con su propia política de enrutamiento. En este caso usaremos el AS 65000, el cual es uno de los más utilizados y extendidos.

```
[admin@PE1] > rou bgp template
[admin@PE1] /routing/bgp/template > set default address-families=ip,vpnv4 as=65000
```

Con la plantilla configurada, pasaremos a configurar la conexión con el otro router PE. Como ya tenemos algunos valores preconfigurados de la plantilla, únicamente habrá que configurar la dirección local, la dirección remota, el AS remoto, el role local de bgp (en este caso ibgp al ser una conexión en el mismo AS) y habilitar tanto la escucha como la conexión.

```
[admin@PE1] /routing/bgp/template > .. connection
[admin@PE1] /routing/bgp/connection > add name=toPE2 temp=default local.add=\
192.170.0.2 .role=ibgp remote.add=192.170.0.4 .as=65000 conn=yes list=yes
```

Con la conexión realizada entre PE1 y PE2, haremos análogamente lo mismo entre PE2 y PE1. Una vez hecho podemos comprobar que se ha establecido la sesión ejecutando el siguiente comando en ambos routers PE:

```
[admin@PE1] > rou bgp session
[admin@PE1] /routing/bgp/session > print
```

Si todo ha ido bien, deberíamos ver una sesión en cada router PE como se muestra en las imágenes siguientes:

```
[admin@PE1] /routing/bgp/session> print
Flags: E - established
 0 E name="toPE2-1"
     remote.address=192.170.0.4 .as=65000 .id=192.170.0.4
     .capabilities=mp,rr,gr,as4 .afi=ip,vpnv4 .messages=18 .bytes=342 .eor=""
     local.role=ibgp .address=192.170.0.2 .as=65000 .id=192.170.0.2
     .capabilities=mp,rr,gr,as4 .afi=ip,vpnv4 .messages=18 .bytes=342 .eor=""
     output.procid=20
     input.procid=20 ibgp
     multihop=yes hold-time=3m keepalive-time=1m uptime=17m55s600ms
     last-started=2023-11-20 18:31:26 prefix-count=0
[admin@PE1] /routing/bgp/session>
```

Fig. 99. MP-BGP. Session PE1-PE2.

```
[admin@PE2] /routing/bgp/session> print
Flags: E - established
 0 E name="toPE1-1"
     remote.address=192.170.0.2 .as=65000 .id=192.170.0.2
     .capabilities=mp,rr,gr,as4 .afi=ip,vpnv4 .messages=20 .bytes=380 .eor=""
     local.role=ibgp .address=192.170.0.4 .as=65000 .id=192.170.0.4
     .capabilities=mp,rr,gr,as4 .afi=ip,vpnv4 .messages=20 .bytes=380 .eor=""
     output.procid=20
     input.procid=20 ibgp
     multihop=yes hold-time=3m keepalive-time=1m uptime=19m19s940ms
     last-started=2023-11-20 18:31:26 prefix-count=0
[admin@PE2] /routing/bgp/session>
```

Fig. 100. MP-BGP. Session PE1-PE2.

3.2.8.6. VRF y VPN

El cliente se situará en una tabla VRF configurada con un RD y RT de importación y exportación. Por ello, primero tendremos que crear esa tabla VRF en nuestros routers PE, que servirán para consultar el direccionamiento de cada cliente conectado a los routers CE. En nuestro caso como solo tenemos un cliente, solo crearemos una tabla VRF por router PE de la siguiente manera:

```
[admin@PE1] > ip vrf
[admin@PE1] /ip/vrf > add interfaces=ether2 name=CE1
```

Le daremos un nombre e indicaremos el interfaz que el router PE utilizará para la comunicación con el cliente a través del router CE.

Este mismo proceso se llevará a cabo análogamente en el router PE2.

A continuación, se configurará la VPN a través de BGP en los routers PE.

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

También definiremos los *Route Targets*, que indican qué rutas se distribuirán al peer PE según la VPN que identifique. En este caso se ha elegido el mismo valor para RD y RT por simpleza, pero no es necesario.

Por último, se indicará la política de asignación de etiquetas, la tabla VRF que se utilizará y el tipo de rutas que se distribuirán de VRF a *VPNv4*. A parte de ***static*** y ***connected*** se habilitará ***bgp*** ya que es el protocolo que utilizaremos entre los PE y los CE en su versión *external BGP*

```
[admin@PE1] > rou bgp vpn
[admin@PE1] /routing/bgp/vpn > add route-distinguisher=65000:100 import.route-targets=65000:100\
vrf=CE1 label-allocation-policy=per-vrf export.route-targets=65000:100 .redistribute=connected,static,bgp
```

Repetir análogamente en PE2.

3.2.8.7. Routing CE-PE

Para establecer la comunicación entre los routers PE y CE configuraremos el protocolo EBGP, configurando un número de AS diferente al del *backbone* MPLS, concretamente el 65500.

3.2.8.7.1. Configuración en PE

En el caso de los routers PE, crearemos una nueva conexión, pero en este caso con el router CE. Para ello, esta vez no utilizaremos ninguna plantilla y configuraremos todos los campos desde la propia conexión.

Necesitaremos indicar la dirección local, el AS local, el role local, el AS remoto, la dirección remota y habilitar la conexión y la escucha al igual que hacíamos en la conexión BGP dentro de la *backbone*. Esta vez no es necesario activar la familia *VPNv4*, ya que no habrá tráfico de este tipo.

Pero adicionalmente para esta conexión, tenemos que indicar el ID del router (la interfaz loopback), el vrf que se utilizará y la tabla de routing asociada al vrf.

Por último, debemos permitir que cualquier ruta se pueda anunciar a través de la red MPLS, esto se configura con ***output.default-originate=always.*** Así se creará la ruta por defecto en el router CE aprendida a través de eBGP y que hará posible la comunicación entre direcciones privadas a través de la red MPLS.

```
[admin@PE1] > rou bgp con
[admin@PE1] /routing/bgp/connection > add name=toCE1 router-id=192.170.0.2 as=65000\
local.address=10.0.0.2 .role=ebgp remote.address=10.0.0.1 .as=65500 routing-table=CE1 vrf=CE1\
connect=yes listen=yes  output.default-originate=always
```

3.2.8.7.2. Configuración en CE

En los routers CE, aparte de realizar la conexión BGP, primero habrá que crear una lista de direcciones en *ip/firewall*. Esta lista de direcciones la asignaremos a la conexión BGP para permitir que los paquetes recibidos en el CE puedan ser enviados a la red privada del cliente en cuestión.

```
[admin@CE1] > ip firewall address-list
[admin@CE1] /ip/firewall/address-list > add address=192.168.1.0/24 list=BGP_OUT
```

Con la lista creada, procedemos a configurar la conexión BGP indicando ID del router, AS, dirección local, role local, dirección remota, AS remoto y asignando la lista de direcciones como red de salida como bien hemos mencionada. Tampoco hay que olvidar activar la conexión y la escucha.

```
[admin@CE1] /ip/firewall/address-list > /routing/bgp/connection
[admin@CE1] /routing/bgp/connection > add na=toPE1 router-id=192.170.0.1 as=65500 local.add=\
10.0.0.1 .role=ebgp remote.addr=10.0.0.2 .as=65000 output.network=BGP_OUT conn=yes list=yes
```

No hay que olvidar llevar a cabo la configuración análoga en PE2 y en CE2.

Con las conexiones establecidas se podrá comprobar con ***/routing/bgp/session print*** si las sesiones están activas y por tanto el proceso se ha realizado correctamente.

3.2.8.8. Verificación final

Para verificar la correcta conectividad de los routers del cliente realizaremos un ping entre el PC1 y el PC2.

No hay que olvidar asignar las IPs correspondientes a los ordenadores a través del adaptador de red, como se hacía en la Práctica 1.

3.2.9. Actividades propuestas

Al igual que en las prácticas anteriores, para realizar las actividades nos valdremos de la función *"Port Mirroring"* de los routers Ejecutaremos las siguientes líneas en el Router P para capturar el tráfico en el enlace PE1-P:

```
[admin@P] > /interface/ethernet/switch
[admin@P] /interface/ethernet/switch > set 0 name=switch mirror-s=ether3 mirror-t=ether7
```

Después de esto conectaremos el puerto al ordenador que ejecutará el Wireshark con el filtro icmp para mostrar los mensajes que se envían en un ping.

1. Realice un ping desde el PC1 al PC2 ¿Qué diferencias observa en los paquetes respecto a la práctica 1? Compare también las diferencias entre los paquetes de *request* y de *reply*.

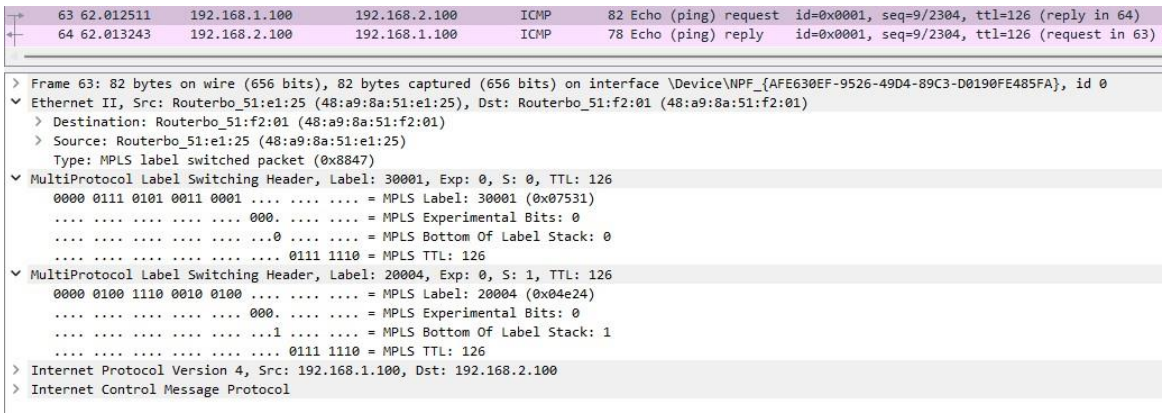


Fig. 101. Paquete request enlace PE1-P.

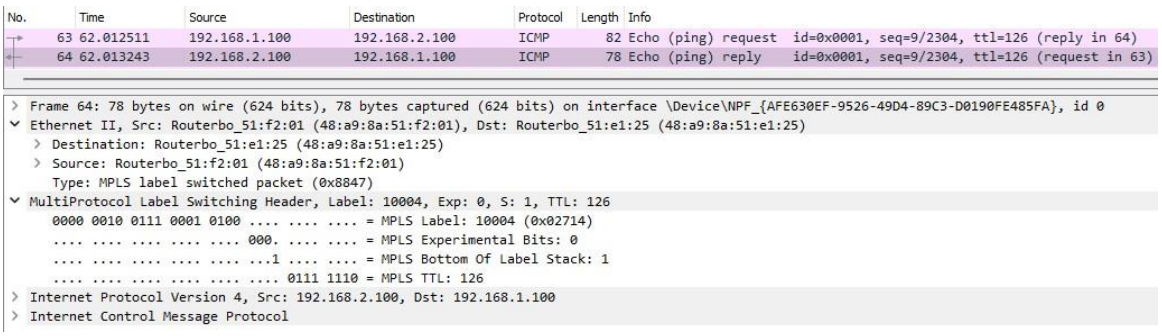


Fig. 102. Paquete reply enlace PE1-P.

Observamos que el paquete request contiene los mismos campos que un paquete MPLS, pero en este caso aparecen dos etiquetas anidadas: 30001 y 20004.

La etiqueta 20004, lleva el bit S o stack a 1, lo que indica que se trata de la última etiqueta, utilizada para la ruta encaminada mediante OSPF en el backbone MPLS.

La etiqueta 30001 pertenece a la VPN y la identifica para que el PE sepa a dónde reenviar el paquete.

En el paquete reply observamos que únicamente contiene una etiqueta, en este caso la 10004. Esto se debe a que el router P ya ha encaminado el paquete a la VPN en cuestión y ha eliminado la etiqueta, quedando solo la que el router PE1 anunció como la etiqueta para recibir los paquetes.

2. Ahora queremos comprobar el establecimiento de la sesión BGP, por tanto, configura el Port Mirroring bien en el ether3 de PE1 o bien en el ether4 de PE2. Reinicia el router PE con /system reboot. que está conectado al WireShark. ¿Qué tipo de paquetes BGP aparecen?

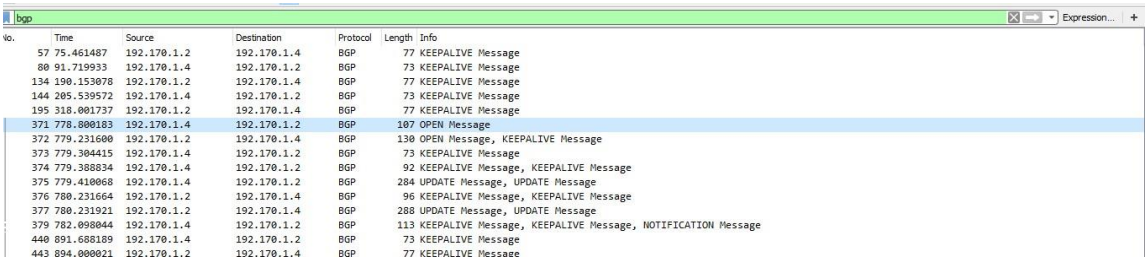


Fig. 103. Captura paquetes BGP.

Se identifican tres tipos de paquetes: KeepAlive Message, Open Message y Update Message

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

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

- Número de versión de BGP usada. Es importante que las versiones de ambos peer coincidan.
- Identificador BGP el cual corresponde con la IP del router con el que establece la sesión, en este caso el PE2 (192.170.0.4)
- Número del AS local, en este caso 65000.

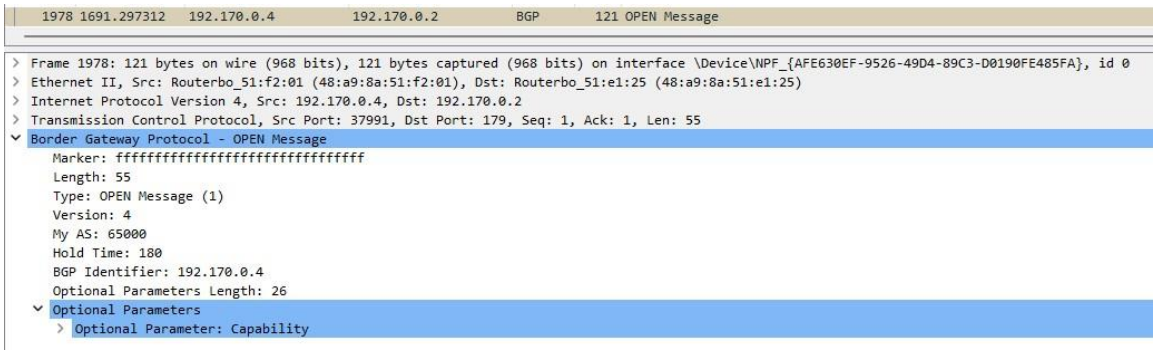


Fig. 104. Captura OPEN Message.

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

- ORIGIN: Indica mediante qué proceso ha sido aprendida la ruta. En este caso es IGP al haber sido aprendida por el protocolo interno OSPF.
- AS_PATH: Indica el número del Sistema Autónomo (AS) en el que se encuentra el router con el que se está estableciendo la sesión BGP para enrutar paquetes entre CE y PE.
- LOCAL_PREF: Atributo para influir en la selección de ruta preferida dentro de un AS cuando hay múltiples rutas disponibles para un destino específico, en nuestro caso solo hay una ruta.
- EXTENDED_COMMUNITIES: Los destinos son agrupados en comunidades. Contiene la información de los RTs y AS. En este caso la hemos definido como extendida y se indica el RD.
- MP_REACH_NLRI: Información de la familia ipv4 y del RD, también se puede observar la Label Stack.

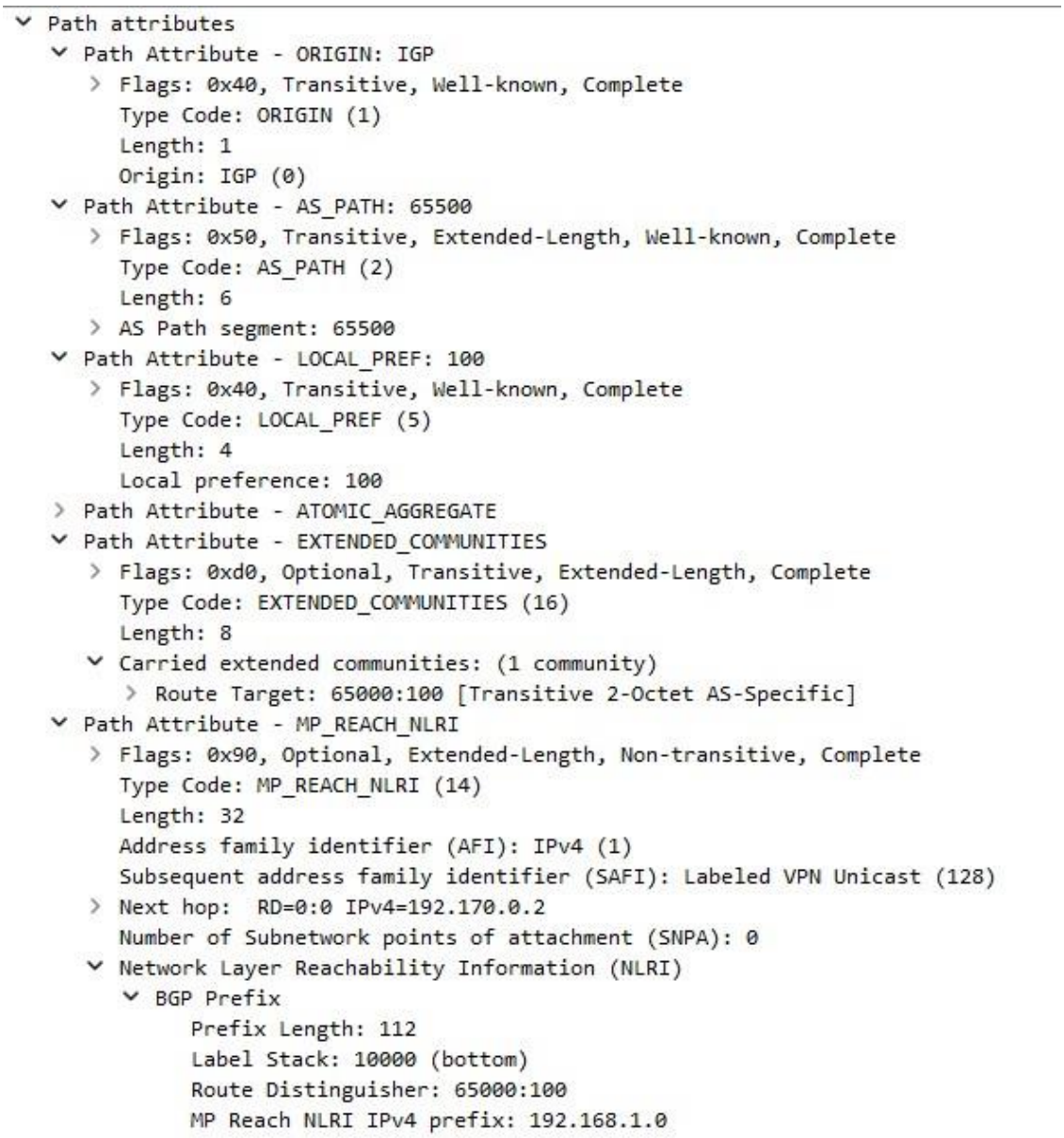


Fig. 105. Captura UPDATE Message.

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

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

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

Posteriormente, añadir mediante el protocolo de routing BGP la nueva red que conecta el router CE al router PE con un identificador AS distinto.

Por último, modificar la familia ipv4 y redistribuir las nuevas rutas creadas.

4. La red actual puede ser perfectamente una implementación para una empresa cos dos sedes separadas geográficamente. Pero ahora la empresa decide comprar una nueva filial y la quiere ubicar en el mismo edificio de otra filial, creando una red local aparte pero que a la vez ambas estén accesibles desde la sede central. ¿Es esto posible utilizando únicamente un router más en la red que ya tenemos montada? Justifica experimentalmente.

Esto sí es posible dado que podemos utilizar el router adicional como router CE conectado al PE2. En este router CE, que llamaremos CE3, se puede montar la red local de la nueva filiar y utilizar una nueva tabla vrf independiente en PE2. Desde PE2 se conectará a la sede central en CE1 con una nueva VPN a través del protocolo BGP.

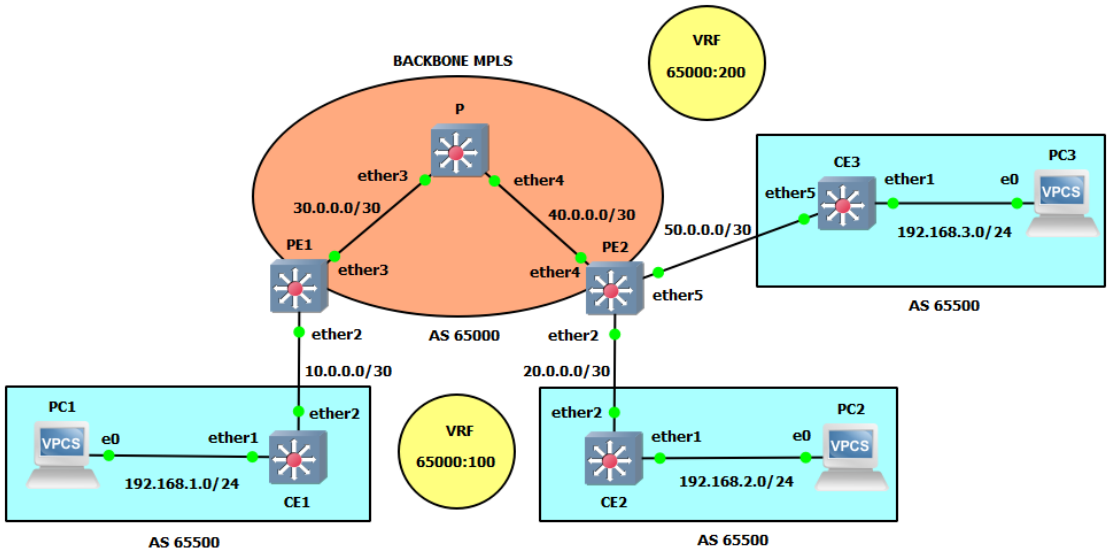


Fig. 106. Topología de red con 3 routers CE.

Realmente como sabemos la red es escalable y solo se necesita de ciertas configuraciones en CE3, PE2 y PE1.

En PE2, primero asignaremos la nueva dirección IP a la interfaz que se conecta al CE3, crearemos la nueva vrf, estableceremos la conexión eBGP con CE3 y la VPN con CE1:

```
[admin@PE2] > /ip add
[admin@PE2] > /ip/address > add
address: 50.0.0.2/30
interface: ether5
[admin@PE2] /ip/address > .. vrf
[admin@PE2] /ip/vrf > add name=CE3 int=ether5
[admin@PE2] /ip/vrf > /rou bgp con
[admin@PE2] /routing/bgp/connection > add name=toCE3 router-id=192.170.0.4 as=65000\
local.address=50.0.0.2 .role=ebgp remote.address=50.0.0.1 .as=65500 routing-table=CE3\
vrf=CE3 connect=yes listen=yes output.default-originate=always
[admin@PE2] /routing/bgp/connection > .. vpn
[admin@PE2] /routing/bgp/vpn > add route-disting=65000:200 import.route-targ=65000:200\
vrf=CE3 label-allocation-poli=per-vrf export.route-t=65000:200 .redist=connected,static,bgp
```

Una vez hecho esto, en CE3 asignaremos las direcciones IP y estableceremos la conexión BGP con PE2, no sin antes crear la lista de direcciones de firewall:

```
[admin@CE3] > /int br
[admin@CE3] /interface/bridge > add name=lo0
[admin@CE3] /interface/bridge > /ip addr
[admin@CE3] /ip/address > add
address: 192.170.0.6/32
interface: lo0
[admin@CE3] /ip/address > add
address: 192.168.3.1/24
interface: ether1
[admin@CE3] /ip/address > add
address: 50.0.0.1/30
interface: ether5
[admin@CE3] /ip/address > .. fire addr
[admin@CE3] /ip/firewall/address-list > add addr=192.168.3.0/30 list=BGP_OUT
[admin@CE3] /ip/firewall/address-list > /rou bgp conn
[admin@CE3] /routing/bgp/connection > add name=toPE2 router-id=192.170.0.6 as=65500\
local.add=50.0.0.1 .role=ebgp remote.addr=50.0.0.2 .as=65000 output.network=BGP_OUT\
conn=yes list=yes
```

Por último, en PE1 crearemos la VPN que conectará la sede centra con la nueva filiar. Haremos uso del mismo RD, Import RT y Export RT que en PE1 para CE3 y que será deferente al de la conexión entre CE1 y CE3:

```
[admin@PE1] > /rou bgp vpn
[admin@PE1] /routing/bgp/vpn > add route-disting=65000:200 import.route-t=65000:200\
vrf=CE1 label-allocation-po=per-vrf export.route-t=65000:200 .redist=connected,static,bgp
```

3.3. Práctica 3 "Configuración de una Red L2 MPLS VPLS"

3.3.1. Introducción

VPLS (Servicio de LAN Privada Virtual) es una tecnología capaz de proporcionar Ethernet multipunto a multipunto basado en la comunicación sobre redes IP/MPLS. Permite a sitios dispersos geográficamente compartir un dominio de difusión Ethernet mediante la conexión de estos a través de pseudowires (PW).

VPLS es una Red Privada Virtual (VPN), que en contraste con L3 MPLS VPN, que solo permite túneles punto a punto en capa 2, VPLS permite conectividad multipunto. En una VPLS la LAN de cada sitio se extiende hasta el borde de la red del proveedor (MPLS).

VPLS al emular una LAN, necesita de una conectividad completa de malla. Para ello, se cuenta con dos métodos diferentes para el establecimiento de la malla VPLS completa:

- LDP (Protocolo de Distribución de Etiquetas): cada PE debe configurarse para participar en un VPLS determinado.
- BGP (Protocolo de puerta de Enlace Fronterizo): proporciona autodescubrimiento y señalización.

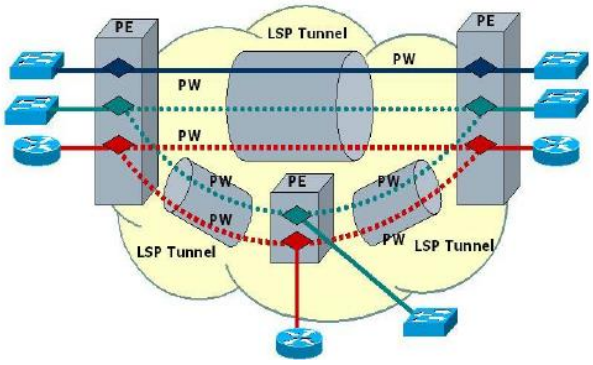


Fig. 107. Topología con 3 redes VPLS.

Permite una conectividad transparente a través de una red de proveedor de servicios (backbone MPLS), pudiéndose enviar paquetes IP entre ellos. La red proveedora puede ofrecer conectividad a redes de clientes distintas, apareciendo cada una de ellas como un PW distinto.

3.3.2. Componentes y arquitectura L2VPLS

3.3.2.1. Customer Edge (CE)

Los routers CE son los equipos de borde del cliente que están conectados a la red L2VPLS. Estos dispositivos se utilizan para enviar y recibir tráfico de capa 2 a través de la L2VPLS.

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

3.3.2.2. Provider Edge (PE)

Router frontera de la red del proveedor de servicios conectado al router CE. Encapsulan y envían los paquetes de capa 2 a través de la red MPLS hacia los demás routers PE.

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

3.3.3. Protocolos de Señalización

Los protocolos de señalización se utilizan para establecer y controlar las conexiones L2VPLS entre los dispositivos PE. Estos protocolos pueden incluir BGP (Border Gateway Protocol) para la señalización y el intercambio de información de enrutamiento, y LDP (Label Distribution Protocol) para el establecimiento y mantenimiento de las etiquetas MPLS utilizadas en la red MPLS.

3.3.3.1. LDP

LDP permite la señalización de la red VPLS, pero requiere del establecimiento de interfaces VPLS en ambos extremos del túnel, es decir, en ambos routers PE que formarán parte de la malla VPLS. En una red de 4 routers PE, se necesitarán 12 interfaces VPLS.

3.3.3.2. BGP

BGP permite señalización y autodescubriento, para ello, se hace uso de conexiones MP-BGP entre routers PE. Sobre BGP se establecen las VPN's desde cada PE, logrando autodescubrirse al resto de PE's de la malla.

3.3.4. Componentes funcionales de VPLS

3.3.4.1. Pseudowire (PW)

Es un circuito virtual VC o circuito Ethernet emulado. Desde la óptica de MPLS un pseudowire es un camino LSP con una etiqueta VC. Es bidireccional e interconecta dos LSRs (PEs), cada dirección va sobre un LSP. Entre dos PEs pueden existir múltiples pseudowires, y cada LSP puede llevar uno o más pseudowires que pertenezcan a uno o varios clientes.

Posibilitan el transporte de cualquier tipo de tráfico como Ethernet, FR, ATM, TDM de baja velocidad T1, T3, E1, E3 y SONET/SDH sobre una red MPLS.

3.3.4.2. VSI (Instancia de Switch Virtual)

VSI equivale a un Bridge transparente, que interconecta una malla de PWs para reenviar tramas Ethernet entre routers PE's.

VSI es similar a VRF en L3VPN, que entregaba tráfico conmutado a IP/MPLS. VSI entrega tráfico L2 VPN a IP/MPLS usando VPLS.

3.3.5. Envío de Tramas

VPLS permite que el PE actúe como un bridge con una tabla de MACs por VSI. El VSI por tanto tiene una tabla que se construye mediante las MAC fuente que circulan por el AC (attachment circuit) o PW, funcionado como un switch convencional.

Cuando una trama Ethernet accede por medio de un AC de entrada al PE, la MAC destino es buscada en la tabla y la trama enviada por el PW adecuado para alcanzar el PE remoto correspondiente.

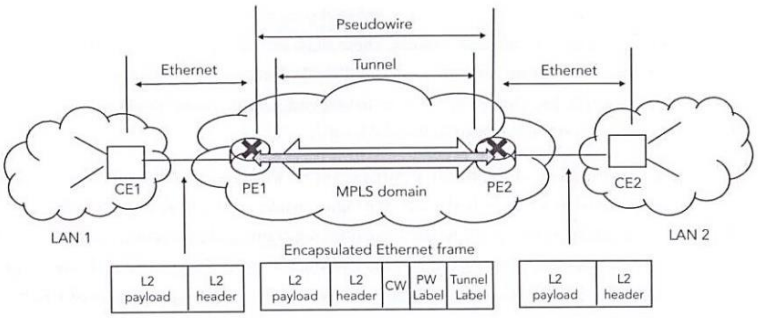


Fig. 108. Envió de tramas entre dos LAN's.

Destinos Broadcast, Multicast o desconocidos son inundados excepto por el AC de llegada, por todos los Pseudowires disponibles. El VSI actualiza su tabla con las tramas que le llegan y las entradas que no se utilizan son eliminadas de la misma.

Esta inundación puede producir bucles de tráfico, para evitar esto, los PEs utilizan Bridge Horizon sobre los PW para asegurar la no existencia de bucles.

3.3.5.1. Bridge Horizon

La idea básica de Bridge Horizon es hacer que el tráfico que llega por algún puerto nunca se envíe a algún conjunto de puertos. Esto significaría nunca enviar paquetes que llegaron a través de un túnel VPLS.

Bridge Horizon permite configurar los puertos dentro de un bridge para que el paquete recibido a través del puerto con valor horizonte X no se reenvíe a ningún otro puerto con el mismo valor de horizonte X.

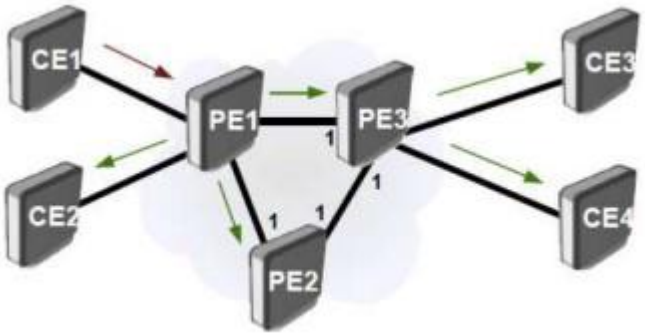


Fig. 109. Topología Bridge Horizon.

3.3.6. Objetivos

El objetivo de la presente práctica es familiarizarse con los conceptos de túneles VPLS sobre MPLS, así como su configuración en una red implementada con routers Mikrotik.

Para ello, se deberán realizar las siguientes actividades:

- introducir en los routers los comandos necesarios para configurar una red L2 VPLS (LDP) y L2 VPLS (BGP).
- verificar el correcto funcionamiento de los túneles VPLS establecidos en la red.
- visualizar los diferentes paquetes que circulan por la red e identificar los campos pertenecientes a los diferentes protocolos utilizados en VPLS.
- Diferenciar y contrastar la escalabilidad de una red VPLS sobre LDP y BGP.

3.3.7. Elementos necesarios

Para la realización de la presente práctica se utilizarán los routers MikroTik disponibles en el laboratorio. En la *figura 110* podemos ver una imagen del mencionado router y en la *tabla 11* la descripción de cada uno de los elementos presentes en el mismo.

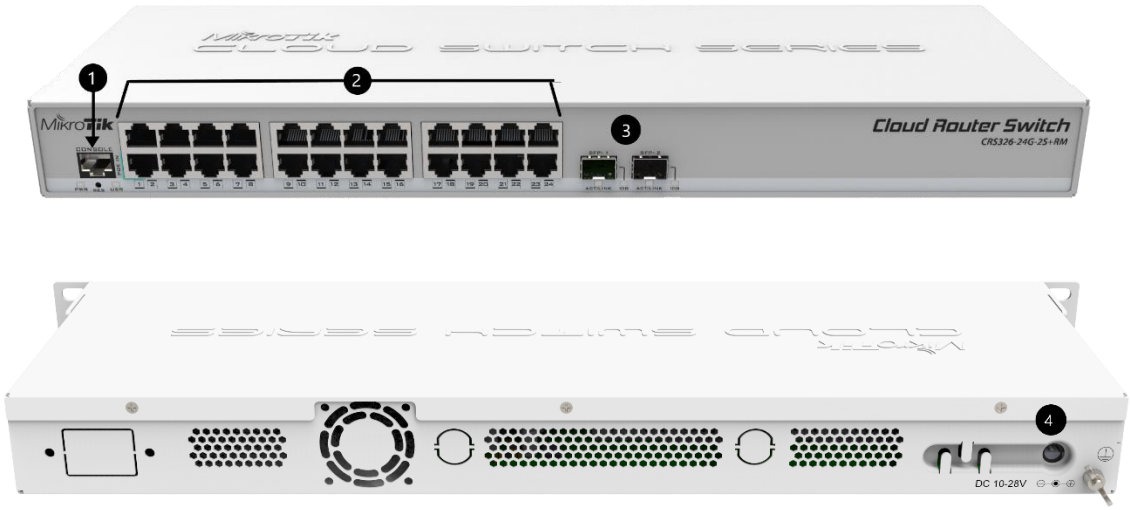


Fig. 110. Imágenes Router MikroTIk.

ID	DESCRIPCIÓN
1	Puerto de consola
2	24 puertos Gigabit Ethernet
3	2 puertos 10G SPF+
4	Entrada de alimentación

Tabla 11. Descripción elementos del router.

3.3.8. Topología de red

Vamos a crear una red L2 MPLS VPLS sobre LDP formada por 6 routers Mikrotik CRS326-24G-2S+RM con las mismas propiedades que en las prácticas anteriores. En ella estableceremos una maya de túneles VPLS con 4 routers LER, los cuales queremos comunicar.

Utilizaremos los protocolos OSPF y LDP para intercambiar información de direccionamiento en la red MPLS.

Como observamos a continuación, hemos prescindido de los routers CE para centrarnos en el establecimiento de las conexiones y túneles entre routers PE (LER´s). Así pues, solo conectamos un PC directamente a cada LER para comprobar el funcionamiento.

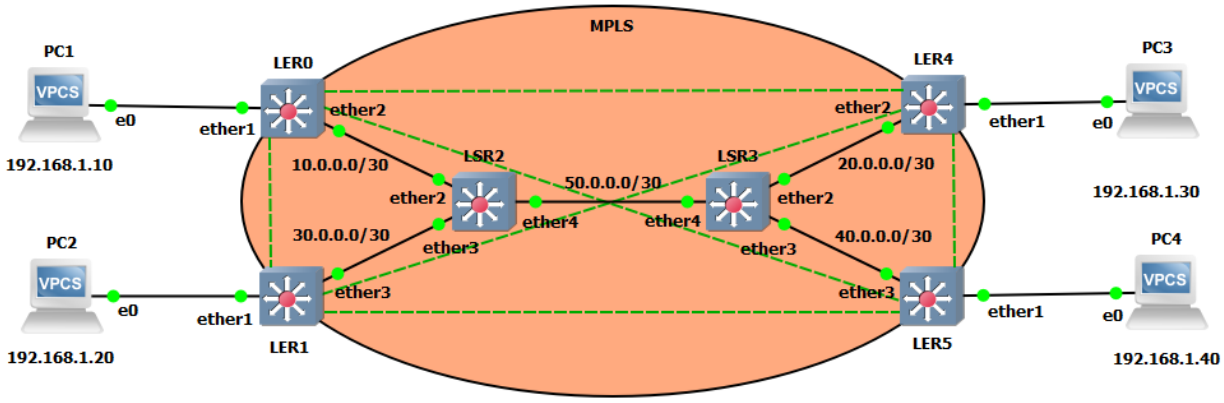


Fig. 111. Topología de red.

Dispositivo	Interfaz	Dirección IP	Máscara de red	Gateway
LER0	lo0	192.170.0.1	255.255.255.255	-
	ether2	10.0.0.1	255.255.255.252	-
	VPLS	192.168.1.1	255.255.255.0	-
LER1	lo0	192.170.0.2	255.255.255.255	-
	ether3	30.0.0.1	255.255.255.252	-
	VPLS	192.168.1.2	255.255.255.0	-
LSR2	lo0	192.170.0.3	255.255.255.255	-
	ether2	10.0.0.2	255.255.255.252	-
	ether3	30.0.0.2	255.255.255.252	
	ether4	50.0.0.2	255.255.255.252	-
LSR3	lo0	192.170.0.4	255.255.255.255	-
	ether2	20.0.0.1	255.255.255.252	-
	ether3	40.0.0.1	255.255.255.252	
	ether4	50.0.0.1	255.255.255.252	-
LER2	lo0	192.170.0.5	255.255.255.255	-
	ether2	20.0.0.2	255.255.255.252	-
	VPLS	192.168.1.3	255.255.255.0	-
CE2	lo0	192.170.0.6	255.255.255.255	-
	ether3	40.0.0.2	255.255.255.252	-
	VPLS	192.168.1.4	255.255.255.0	-
PC1	NIC	192.168.1.10	255.255.255.0	192.168.1.1
PC2	NIC	192.168.1.20	255.255.255.0	192.168.1.2
PC3	NIC	192.168.1.30	255.255.255.0	192.168.1.3
PC4	NIC	192.168.1.40	255.255.255.0	192.168.1.4

Tabla 12. Tabla direccionamiento IP.

3.3.9. Configuración de la red

3.3.9.1. Nuevo montaje y borrado

Una vez tengamos el montaje de la nueva topología, procederemos a configurar nuestra red. De igual forma que en las prácticas anteriores, realizamos el borrado de la antigua configuración de los routers.

También podemos aprovechar los nombres guardados a través de las direcciones MAC de las prácticas anteriores (siempre que se hagan de seguido). Si no es posible volveremos a nombrar los routers a partir del rango de MACs de cada router.

3.3.9.2. Crear Interfaz Loopback y Bridge VPLS

Vamos a crear la interfaz *loopback* como venimos haciendo en las prácticas anteriores, pero esta vez también crearemos un *bridge* adicional, el VSI que hemos visto en la teoría. Este bridge que llamaremos VPLS nos servirá para asociar a una interfaz física los diferentes túneles vpls.

```
[admin@LER0] > int br
[admin@LER0] /interface/bridge > add name=lo0
[admin@ LER0] /interface/bridge > add name=VPLS
```

Para el resto de routers LER se replicará la configuración análogamente. Para el caso de los routers LSR no se creará en Bridge VPLS.

3.3.9.3. Asignar IP's

A continuación, asignaremos las direcciones IP's. En este caso hay que tener en cuenta que al bridge VPLS se le asignará la dirección IP que los PC's utilizarán como puerta de enlace predeterminada, la que anteriormente asignábamos al puerto físico.

```
[admin@ LER0] /interface/bridge > /ip address
[admin@ LER0] /ip/address > add
address: 192.170.0.1/32
interface: lo0
[admin@ LER0] /ip/address > add
address: 192.168.1.1/24
interface: VPLS
[admin@ LER0] /ip/address > add
address: 10.0.0.1/30
interface: ether2
```

Repetir coherentemente en el resto de routers.

3.3.9.4. OSPF Backbone MPLS

Ahora aplicaremos el protocolo OSPF para aprender de forma dinámica las direcciones de los routers que pertenecen a MPLS, excluyendo por tanto el interfaz ether1 de los routers LER.

```
[admin@LSR2] > /routing ospf instance
[admin@ LSR2] /routing/ospf/instance > add name=backbone router-id=192.170.0.3
[admin@ LSR2] /routing/ospf/instance > .. area
[admin@ LSR2] /routing/ospf/area > add name=backbone area-id=0.0.0.0 inst=backbone
[admin@ LSR2] /routing/ospf/area > .. int
[admin@ LSR2] /routing/ospf/interface-template > add int=lo0 net=192.170.0.3/32 ar=backbone
[admin@ LSR2] /routing/ospf/interface-template > add int=ether2 net=10.0.0.2/30 ar=backbone
[admin@ LSR2] /routing/ospf/interface-template > add int=ether3 net=30.0.0.2/30 ar=backbone
[admin@ LSR2] /routing/ospf/interface-template > add int=ether4 net=50.0.0.2/30 ar=backbone
```

Repetiremos análogamente para el resto de routers.

3.3.9.5. LDP Backbone MPLS

A continuación, aplicaremos el protocolo de distribución de etiquetas (LDP). De nuevo no hay que configurar las interfaces que se comunican con los routers de fuera de la backbone.

```
[admin@LER1] > /mpls ldp
[admin@ LER1] /mpls/ldp > add afi=ip lsr-id=192.170.0.2 t=192.170.0.2
[admin@ LER1] /mpls/ldp > interface
[admin@ LER1] /mpls/ldp/interface > add
interface: ether3
```

Repetimos análogamente los demás routers.

Al igual que en las prácticas anteriores, podemos cambiar el rango de etiquetas dinámicas para una mayor claridad a la hora de entender el funcionamiento y estructura de los paquetes que circulan por la red.

ROUTER	RANGO ETIQUETAS
LER0	16-9999
LER1	10000-19999
LSR2	20000-29999
LSR3	30000-39999
LER4	40000-49999
LER5	50000-59999

Tabla 13. Rango de etiquetas por routers.

Recuerda que el comando utilizado será el siguiente:

/mpls/settings set dy=<rango>

Se puede comprobar que todo se ha realizado correctamente realizando traceroute entre dos routers cualesquiera de la MPLS. La asignación de etiquetas variará en función de los saltos que hay entre los routers que se ha probado.

3.3.9.6. Configuración de los PC's

No hay que olvidar configurar todos los PC's conectados a los routers LER dentro de la misma subred, como se especifica en la tabla de direcciones.

Por supuesto, a parte de la dirección IP y la máscara, hay que configurar la puerta de enlace predeterminada. En este caso será la dirección IP asignada al bridge del router LER en cuestión.

3.3.9.7. VPLS (LDP)

3.3.9.7.1. Configuración

Con la red MPLS ya configurada, nos centraremos en la configuración propia de esta práctica, los túneles VPLS. Para ello, hay que tener en cuenta que tendremos que crear interfaces VPLS en ambos extremos de cada túnel.

Como queremos comunicar los 4 equipos de la misma red a través de la red MPLS, necesitamos crear un total de 6 túneles VPLS. Como se ha comentado en la teoría, cada túnel tiene un identificativo único que se indica en la tabla junto a cada túnel necesario:

TÚNEL	VPLS-ID
LER0-LER1	1:1
LER0-LER4	2:1
LER0-LER5	3:1
LER1-LER4	4:1
LER1-LER5	5:1
LER4-LER5	6:1

Tabla 14. Identificación por túneles.

Creamos los interfaces con el peer remoto del túnel y el vpls-id:

```
[admin@LER0] > int vpls
[admin@LER0] /interface/vpls > add name=vpls-to-ler1 peer=192.170.0.2 vpls-id=1:1
[admin@LER0] /interface/vpls > add name=vpls-to-ler4 peer=192.170.0.5 vpls-id=2:1
[admin@LER0] /interface/vpls > add name=vpls-to-ler5 peer=192.170.0.6 vpls-id=3:1
```

Con las interfaces configuradas tenemos un extremo de los enlaces virtuales, pero para conseguir el enlace transparente entre los extremos del túnel, tenemos que asociar las interfaces del túnel con la interfaz física a través del bridge creado inicialmente con los siguientes comandos:

```
[admin@LER0] /interface/vpls > .. br port
[admin@ LER0] /interface/bridge/port> add bridge=VPLS interface=ether1
[admin@ LER0] /interface/bridge/port> add bridge=VPLS interface=vpls-to-ler1
[admin@ LER0] /interface/bridge/port> add bridge=VPLS interface=vpls-to-ler4
[admin@ LER0] /interface/bridge/port> add bridge=VPLS interface=vpls-to-ler5
```

Repetiremos estos dos pasos análogamente en el resto de routers LER para tener los túneles completamente operativos, y por tanto la malla VPLS.

3.3.9.7.2. Verificación final

Para verificar el correcto funcionamiento, podemos realizar una serie de procesos que nos cerciorarán que todo está perfectamente operativo.

Primeramente, podemos realizar pings entre todos los PC's de la red y ver que se obtiene las respuestas a los paquetes icmp enviados.

Con esto vemos que hay conexión entre equipos, pero para comprobar que se establece a través de los túneles, podemos monitorizar los interfaces VPLS. Con ello veremos la etiqueta local, que servirá para que el otro extremo envíe la información; la etiqueta remota, que es la asignada por el otro extremo para la comunicación hacia dicho peer; y los diferentes atributos relacionados con el siguiente salto, etiqueta, interface y.nexthop.

```
[admin@LER0] > int vpls
[admin@LER0] /interface/vpls> monitor vpls-to-ler1
   remote-label: 10002
    local-label: 18
  remote-status:
       nexthops: { label=20003; nh=10.0.0.2%ether2; interface=ether2 }

[admin@LER0] /interface/vpls> monitor vpls-to-ler4
   remote-label: 40002
    local-label: 17
  remote-status:
       nexthops: { label=20005; nh=10.0.0.2%ether2; interface=ether2 }

[admin@LER0] /interface/vpls> monitor vpls-to-ler5
   remote-label: 50002
    local-label: 16
  remote-status:
       nexthops: { label=20006; nh=10.0.0.2%ether2; interface=ether2 }
```

Tabla 15. Interfaces VPLS.

3.3.9.7.3. Actividades propuestas

1. **Realice *traceroute* desde LER0 a PC4 y *tracer* desde PC1 a PC4. Comente que diferencias observa respecto a la Práctica 1.**

   Podemos observar en ambos casos que hay un único salto entre los extremos. En cuanto a esta casuística respecto a la práctica 1, podemos decir que en ella cuando realizábamos el traceroute desde el router LER observábamos cada uno de los saltos por la red MPLS con sus respectivas etiquetas. Aunque realizando el tracert entre PCs no se observaban las etiquetas, si que se podían ver los diferentes saltos por la red.

   Sin embargo, esto no está ocurriendo en la configuración VPLS por lo que se comenta en la teoría y hemos aplicado en la configuración, el enlace transparente entre equipos finales de la red hace que a nivel lógico los routers LER se consideren como vecinos, es decir, directamente conectados. Esto se puede observar en la tabla *neighbor* de ldp como vecinos dinámicos.

```
[admin@LER0] /mpls/ldp/neighbor> print
Flags: D, I - INACTIVE; O, T - THROTTLED; t - SENDING-TARGETED-HELLO; v - VPLS; p - PASSIVE
Columns: TRANSPORT, LOCAL-TRANSPORT, PEER, ADDRESSES
#       TRANSPORT     LOCAL-TRANSPORT  PEER            ADDRESSES
0 DOtvp 192.170.0.6   192.170.0.1      192.170.0.6:0   40.0.0.2
                                                       192.168.1.4
                                                       192.170.0.6
1 DOtvp 192.170.0.5   192.170.0.1      192.170.0.5:0   20.0.0.2
                                                       192.168.1.3
                                                       192.170.0.5
2 DOtvp 192.170.0.2   192.170.0.1      192.170.0.2:0   30.0.0.1
                                                       192.168.1.2
                                                       192.170.0.2
3 DO  p 192.170.0.3   192.170.0.1      192.170.0.3:0   10.0.0.2
                                                       30.0.0.2
                                                       50.0.0.2
                                                       192.170.0.3
```

Fig. 112. Tabla neighbor LDP.

2. **Ahora ejecute el software WireShark en los 4 PC's seleccionando la interfaz Ethernet_Labo y aplicando el filtro arp. Una vez hecho esto, realice un ping a la dirección IP de red de nuestra subred desde cualquiera de los 4 PCs. En este caso sería la dirección 192.168.1.0. Comente los paquetes que se observan en los diferentes PC's.**

Tras realizar el ping desde uno de los PC's, podemos observar como una serie de paquetes ARP llegan a cada uno de los PC's en busca de la dirección IP solicitada, pero como es obvio, no se obtiene ninguna respuesta ya que está dirección está reservada y no se ha asignado a ningún equipo.

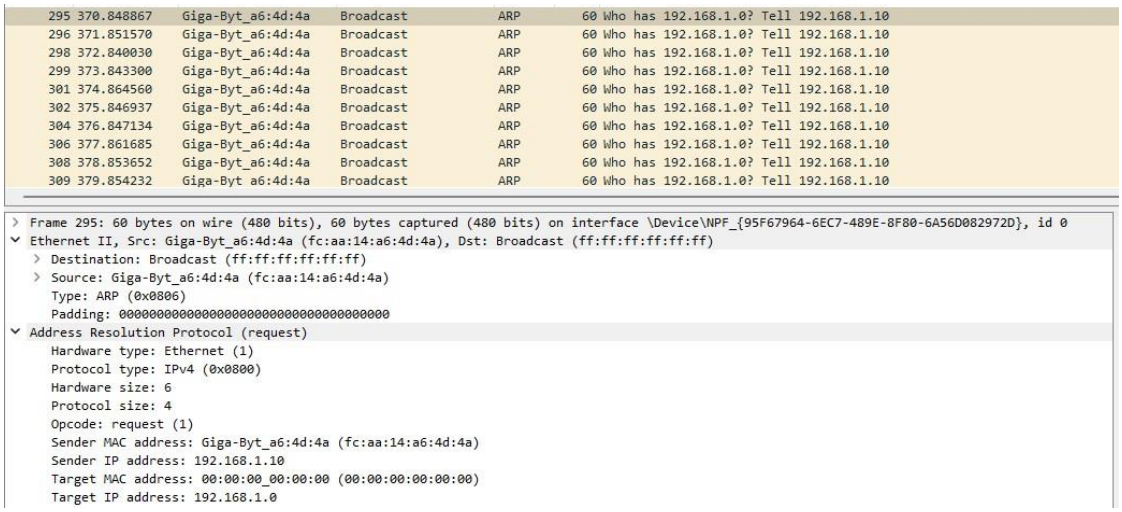


Fig. 113. Capturas paquetes ARP.

En cada una de las solicitudes podemos ver que se está realizando un envío a la dirección MAC de broadcast (FF:FF:FF:FF:FF:FF) desde la dirección MAC del PC en cuestión.

También podemos ver en la cabecera Ethertype el tipo de protocolo en su denominación en hexadecimal***, 0x0806*** que hace referencia a ARP.

Con esto podemos determinar lo que ya sabíamos, que toda nuestra red está comunicada creando una malla VPLS.

Si por el contrario el *ping* se hubiera hecho a una dirección IP asignada a cualesquiera de los PC's, observaríamos los paquetes *request* en el equipo destino del *ping* y los paquetes *reply* en el equipo que ejecuta dicho *ping*.

Al igual que en las prácticas anteriores, vamos a utilizar la función *"Port Mirroring"* de los routers Ejecutaremos las siguientes líneas en el Router LSR2 para capturar el tráfico en el enlace LSR2-LSR3:

```
[admin@LSR2] > /interface/ethernet/switch
[admin@LSR2] /interface/ethernet/switch > set 0 na=switch mirror-s=ether4 mirror-t=ether7
```

Después de esto conectaremos el puerto al ordenador que ejecutará el Wireshark con el filtro icmp para mostrar los mensajes que se envían en un ping. Necesitaremos utilizar un 5º PC o bien desconectar el PC1 o el PC3 que no actuarán en este ping.

También haremos lo propio sobre el enlce LER1-LSR2 y LSR3-LER5 para observar todo el túnel VPLS. Se sugiere la siguiente configuración:

```
[admin@LER1] > /interface/ethernet/switch
[admin@LER1] /interface/ethernet/switch > set 0 na=switch mirror-s=ether3 mirror-t=ether7
```

```
[admin@LER5] > /interface/ethernet/switch
[admin@LER5] /interface/ethernet/switch > set 0 na=switch mirror-s=ether3 mirror-t=ether7
```

3. Realice un ping desde el PC2 al PC4 para cada enlace, guardando la captura de WireShark para comparar los 3 tramos del túnel. ¿Qué diferencias observa en los paquetes respecto a la práctica 3? Compare también las diferencias entre los paquetes de *request* y de *reply*.

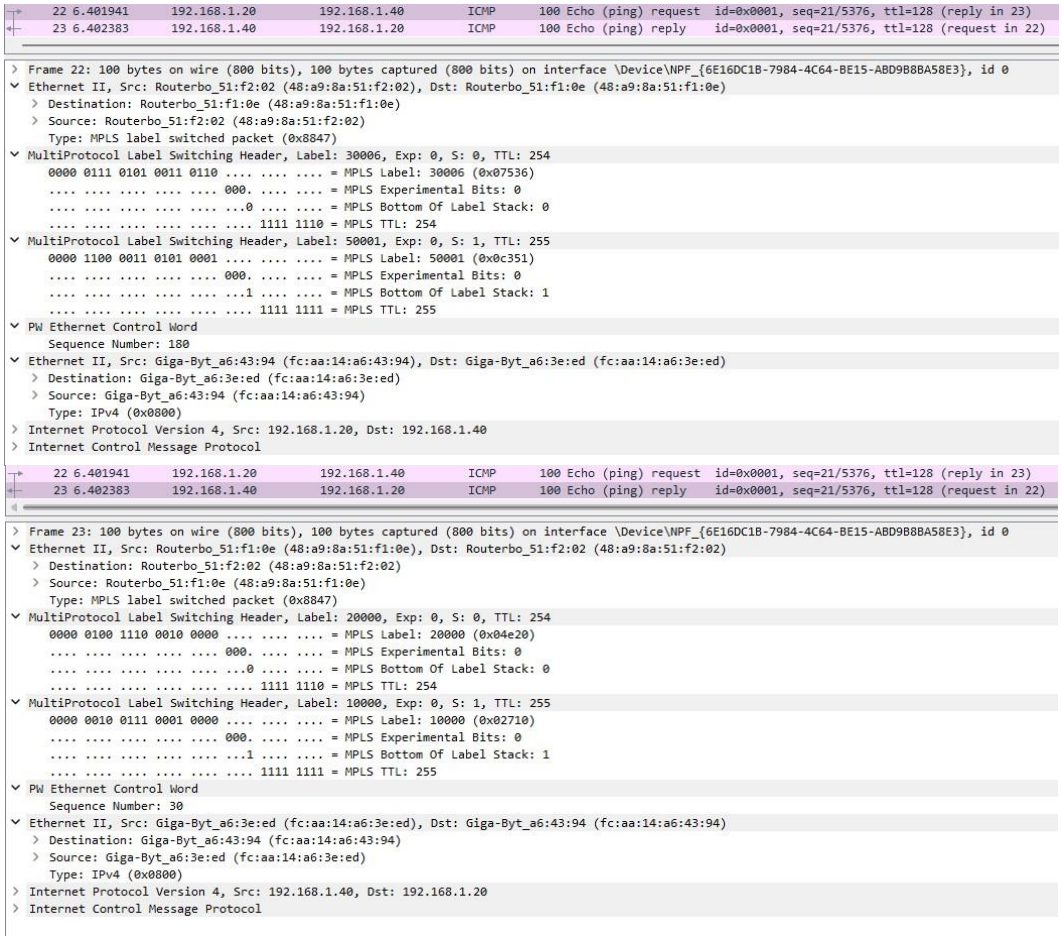


Fig. 114. Capturas ICMP enlace LSR2-LSR3.

Observamos que el paquete request contiene dos etiquetas anidadas al igual que ocurría en la práctica 3: 30006 y 50001.

La etiqueta 50001, lleva el bit S o stack a 1, lo que indica que se trata de la última etiqueta, utilizada en este caso para indicar la ruta hacia el router final del túnel, por tanto es la etiqueta del pseudowire.

La etiqueta 30006 indica el siguiente salto dentro de la red MPLS utilizando OSPF.

En el paquete reply observamos lo mismo, pero de manera análoga con las etiquetas 20000 y 10000.

También se observa respecto a la práctica 3, que hay un nuevo campo en la cabecera del paquete, PW Ethernet Control Word. Este campo nos indica que el direccionamiento IP es trasportado a través de un pseudowire (PW), ese paquete IP lo observamos seguidamente.

Por lo tanto, a parte de la pila de etiquetas, en los paquetes ICMP se observan dos paquetes Ethernet, uno para indicar el transporte sobre MPLS y otro para indicar el trasporte IP, que viene encapsulado sobre el pseudowire mencionado.

Ahora vamos a comparar los diferentes enlaces del túnel.

LER1-LSR2:

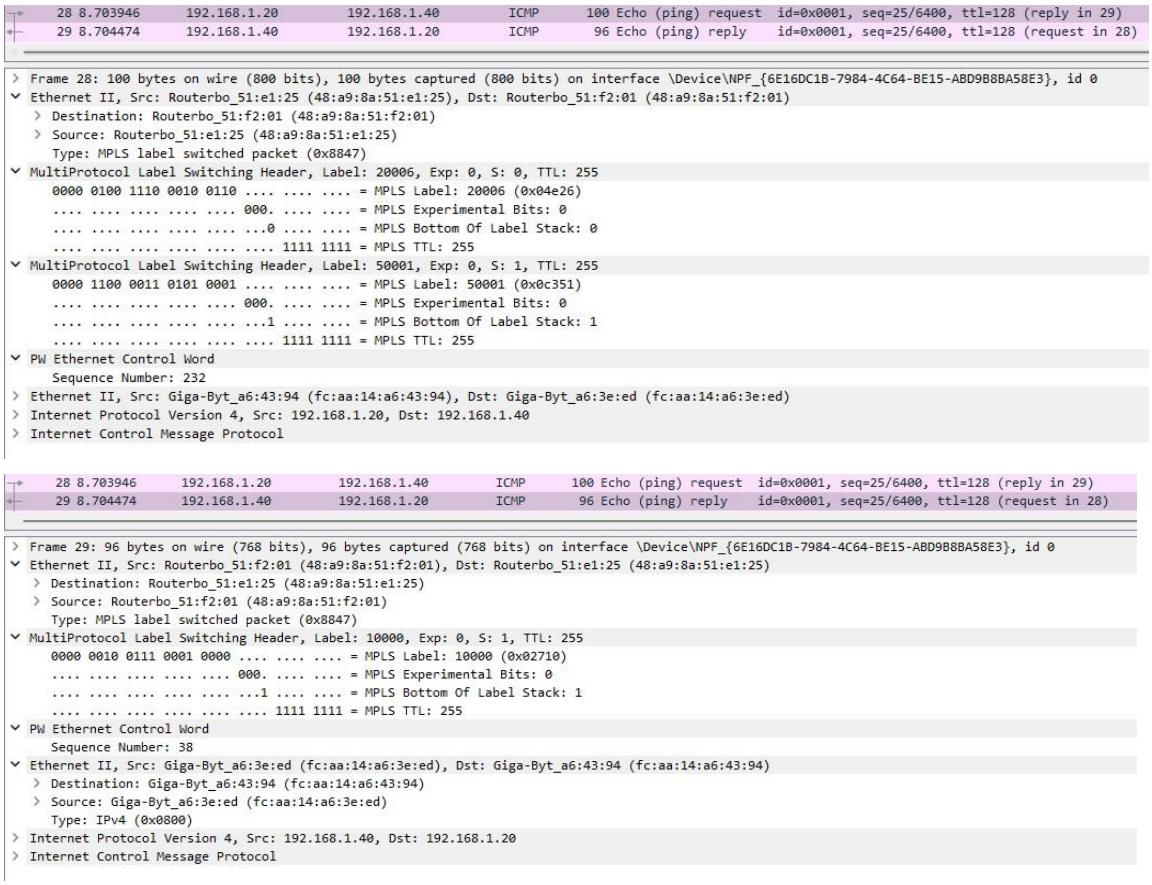


Fig. 115. Capturas ICMP enlace LER1-LSR2.

En este enlace podemos ver en el paquete request, de nuevo, 2 etiquetas. Se puede apreciar que la 50001 es la misma que aparece en el enlace LSR2-LSR3 ya que se trata de la etiqueta del pseudowire para todo el túnel. Bajo esta etiqueta, se encuentra la 2006, que es la asociada a la pila de etiquetas para enviar los paquetes dentro de la MPLS hacia LSR2.

En el paquete reply solo vemos una etiqueta ya que el router LSR2 es penúltimo salto de la red MPLS en el mensaje hacia PC2 Por tanto, solo se observa la etiqueta 10000 correspondiente al pseudowire hacia el PC2.

LER5-LSR3:

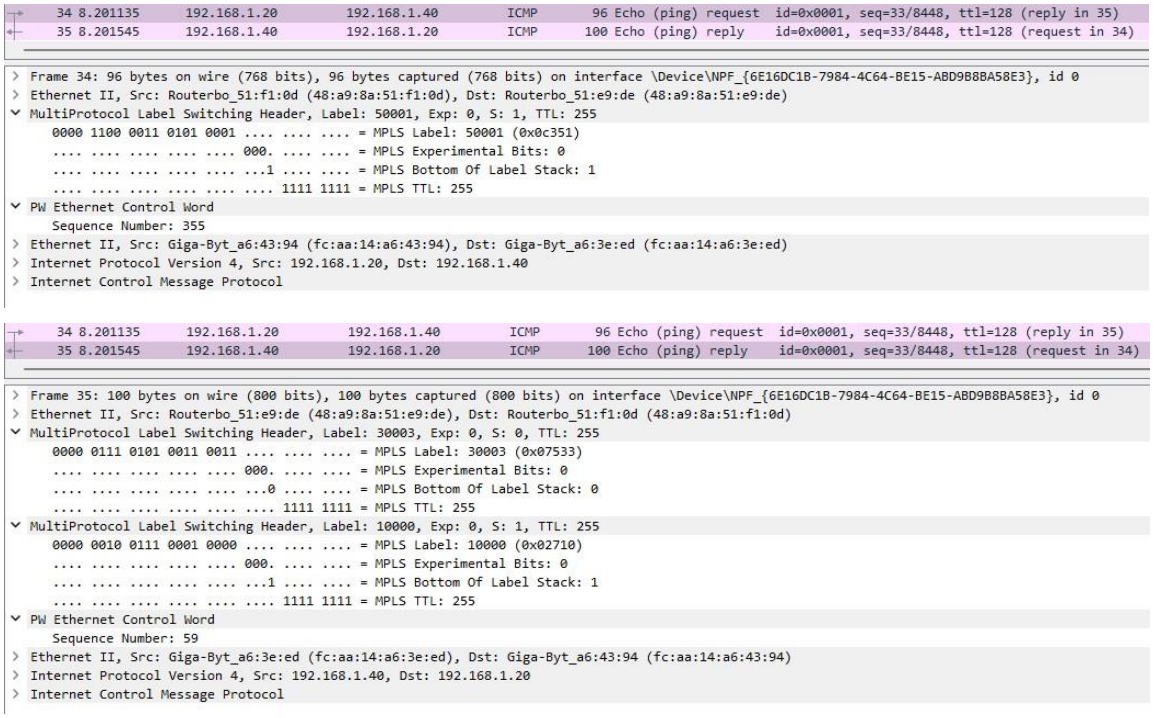


Fig. 116. Capturas ICMP enlace LSR3-LER5.

En este enlace podemos ver que el paquete que tiene 2 etiquetas es el reply, observando de nuevo la correspondiente al pseudowire, la 10000. Mientras que en este caso es la 30003 la asociada al envío de paquetes hacia LSR3.

En ente caso es el router LSR3 el que es penúltimo salto de la red MPLS en el mensaje hacia PC4 Por tanto, solo se observa la etiqueta 50001 correspondiente al pseudowire hacia el PC4.

4. Queremos incluir un nuevo equipo a nuestra red desde una ubicación geográficamente alejada de cualquiera de las ubicaciones que ya tenemos configuradas. ¿Qué pasos deberíamos realizar? ¿Qué conclusiones sacas de la escalabilidad de esta tecnología?

Para poder incluir un equipo a nuestra red transparente VPLS, primero tendremos que conectar a cualquiera de los routers LSR, el router de la nueva ubicación que actuará como un nuevo router LER. En dicho router LER habría que configurar tanto OSPF como MPLS, teniendo en cuenta que tiene que estar en la misma área que la red MPLS original. Obviamente habrá que realizar lo propio sobre el nuevo enlace en el router LSR.

Una vez hecho esto, habría que configurar las interfaces VPLS con el resto de routers LER, implicando esto tener que configurar una nueva interfaz VPLS en cada uno de los routers LER restantes.

Con las interfaces configuradas, habría que incluirlas en los Bridges para formar la malla. También habría que crear un nuevo Bridge en el nuevo router LER con sus interfaces VPLS y la

interfaz física elegida, igual que se hizo con el resto de routers LER.

Como se puede observar, utilizar VPLS sobre LDP no es escalable, ya que requiere de configuración muy específica en cada uno de los routers LER de la red.

5. Ahora queremos utilizar la misma red MPLS para conectar de forma transparente otra subred diferente pero que comparte ubicación con la que ya tenemos configurada. ¿Qué pasos deberíamos realizar?

Para poder tener una nueva subred sobre la misma topología, primero tendremos que elegir la subred a configurar de forma que no haya solapamiento con la original. Por ejemplo, podríamos elegir la subred 192.168.2.0/24.

Una vez teniendo esto claro, crearemos un nuevo Bridge para cada router LER y le asignaremos una dirección IP dentro del rango de la subred y que nos servirá de puerta de enlace predeterminada para los equipos que conectaremos.

Ahora crearemos las interfaces VPLS que se usarán para crear los túneles. Hay que tener en cuenta que los vpls-id deben ser únicos entre los propios enlaces de la subred y entre los de la subred original. Una vez los tengamos creados los asociaremos al bridge en cuestión junto a una nueva interfaz física que debe estar libre y ser distinta a la de la otra subred.

Ya por último conectaremos los PC's a las interfaces físicas asociadas con el Bridge y les daremos a los PC's una dirección IP dentro de la subred. No hay que olvidar configurar la puerta de enlace predeterminada con la IP del Bridge.

3.3.9.8. VPLS (BGP)

3.3.9.8.1. Configuración

Como se ha explicado en la teoría y como se ha podido comprobar con los equipos, la configuración de una red transparente VPLS a través de LDP no es escalable. Esto se debe a que el autodescubrimiento no es posible y se tiene que realizar de forma manual.

Sin embargo, si utilizamos conexiones BGP entre los routers que están conectados a los equipos finales, será posible el autodescubrimiento a través de estas conexiones configurando únicamente una instancia VPLS BGP. De esta forma se autodescubrirán todos los pseudowires necesarios para crear la malla de nuestra red.

3.3.9.8.1.1. Borrar interfaces VPLS

Lo primero de todo será eliminar las interfaces VPLS creadas en el anterior diseño y quitar la asociación de estas con el Bridge VPLS en todos los routers LER:

```
[admin@LER0] > int vpls
[admin@LER0] /interface/vpls > remove vpls-to-ler1,vpls-to-ler4,vpls-to-ler5
[admin@LER0] /interface/vpls > .. br port
[admin@LER0] /interface/bridge/port > print
Flags: I - INACTIVE; H - HW-OFFLOAD
Columns: INTERFACE, BRIDGE, HW, PVID, PRIORITY, PATH-COST, INTERNAL-PATH-COST, HORIZON
#   INTERFACE BRIDGE HW  PVID PRIORITY PATH-COST INTERNAL-PATH-COST HORIZON
0  H ether1    VPLS   yes   1 0x80          10                10 none
1   I *24      VPLS          1 0x80          10                10 none
2   I *25      VPLS          1 0x80          10                10 none
3   I *26      VPLS          1 0x80          10                10 none
[admin@LER0] /interface/bridge/port > remove 1,2,3
```

Como se puede observar, al eliminar las interfaces VPLS, que estas no se reconocen como puertos del Bridge. Para eliminarlas hay que hacer uso del número de filas con la precaución de no eliminar también el interfaz físico.

3.3.9.8.1.2. MP-BGP

Tal y como hemos comentado, el autodescubrimiento se realizará a través de sesiones BGP que tendremos que establecer entre los routers LER. El establecimiento de estas sesiones es similar al realizado en la Práctica 2 entre los routers PE.

Antes de realizar esto, como hacíamos en la práctica anterior, modificaremos la plantilla *default* de BGP para una mayor facilidad y rapidez para configurar las tres conexiones por router:

```
[admin@LER0] > rou bgp template
[admin@LER0] /routing/bgp/template> set default address-fam=ip,l2vpn as=65000 router-id=192.170.0.1
```

Como podemos ver, en este caso hemos añadido a la familia de direcciones el direccionamiento *l2vpn* que permite encapsular la información de capa 2 enviada en los pseudowires.

Con la plantilla creada, procedemos a establecer las conexiones BGP con los siguientes comandos:

```
[admin@LER0] /routing/bgp/template> .. con
[admin@LER0] /routing/bgp/connection> add name=toLER1 temp=default local.add=\
192.170.0.1 .role=ibgp remote.add=192.170.0.2 .as=65000 conn=yes list=yes
[admin@LER0] /routing/bgp/connection> add name=toLER4 temp=default local.add=\
192.170.0.1 .role=ibgp remote.add=192.170.0.5 .as=65000 conn=yes list=yes
[admin@LER0] /routing/bgp/connection> add name=toLER5 temp=default local.add=\
192.170.0.1 .role=ibgp remote.add=192.170.0.6 .as=65000 conn=yes list=yes
```

Repetiremos análogamente estos dos pasos en el resto de routers LER y comprobaremos que se han establecido las conexiones utilizando el comando ***/routing/bgp/session print*** sobre el terminal de los diferentes routers LER. A continuación, se muestra el ejemplo para LER0:

```
[admin@LER0] > /routing/bgp/session print
Flags: E - established
 0 E name="toLER1-1"
     remote.address=192.170.0.2 .as=65000 .id=192.170.0.2 .capabilities=mp,rr,gr,as4
     .afi=ip,l2vpn,l2vpn-cisco .messages=27 .bytes=584 .eor=""
     local.role=ibgp .address=192.170.0.1 .as=65000 .id=192.170.0.1 .capabilities=mp,rr,gr,as4
     .afi=ip,l2vpn .messages=27 .bytes=584 .eor=""
     output.procid=20
     input.procid=20 ibgp
     multihop=yes hold-time=3m keepalive-time=1m uptime=25m14s940ms
     last-started=1970-01-02 00:06:32 prefix-count=1

 1 E name="toLER4-1"
     remote.address=192.170.0.5 .as=65000 .id=192.170.0.5 .capabilities=mp,rr,gr,as4
     .afi=ip,l2vpn,l2vpn-cisco .messages=27 .bytes=584 .eor=""
     local.role=ibgp .address=192.170.0.1 .as=65000 .id=192.170.0.1 .capabilities=mp,rr,gr,as4
     .afi=ip,l2vpn .messages=27 .bytes=584 .eor=""
     output.procid=21
     input.procid=21 ibgp
     multihop=yes hold-time=3m keepalive-time=1m uptime=25m18s670ms
     last-started=1970-01-02 00:06:33 prefix-count=1

 2 E name="toLER5-1"
     remote.address=192.170.0.6 .as=65000 .id=192.170.0.6 .capabilities=mp,rr,gr,as4
     .afi=ip,l2vpn,l2vpn-cisco .messages=27 .bytes=584 .eor=""
     local.role=ibgp .address=192.170.0.1 .as=65000 .id=192.170.0.1 .capabilities=mp,rr,gr,as4
     .afi=ip,l2vpn .messages=27 .bytes=584 .eor=""
     output.procid=22
     input.procid=22 ibgp
     multihop=yes hold-time=3m keepalive-time=1m uptime=25m18s570ms
     last-started=1970-01-02 00:06:33 prefix-count=1
```

Fig. 117. Conexiones BGP establecidas.

3.3.9.8.1.3. BGP-VPLS

Con las sesiones BGP ya establecidas, podemos centrarnos en configurar el autodescubrimiento de los túneles VPLS a través de estas conexiones. Para ello debemos acceder a /routing/bgp/vpls y crear una única instancia que generará el autodescubrimiento.

Esta instancia además de tenerle que dar un nombre y asociarle el Bridge VPLS, tendrá configurados el *Route Distinguisher*, el *Export Route Targets* y el *Import Route Targets* únicos para toda la subred, por tanto, el mismo en todos los routers LER. Eso sí, para diferenciar los diferentes LER, cada uno de ellos tendrá un único identificador llamado *site-id*.

```
[admin@LER0] /routing/bgp/vpls> add bridge=VPLS export-route-t=1:1 import-route-t=1:1/
name=VPLS rd=1:1 site-id=1
```

Repetiremos análogamente esta ejecución en el resto de routers LER teniendo en cuenta la pertinente modificación del *site-id.*

Si ejecutamos ***/interface/vpls print*** en el terminal, podremos ver que las inerfaces VPLS se han aprendido dinámicamente. De igual forma, ejecutando ***/interface/bridge/port print*** podemos ver que dichas interfaces se han asociado también dinámicamente al Bridge VPLS.

```
[admin@LER0] > int vpls print
Flags: R - RUNNING; D - DYNAMIC
Columns: NAME, PEER, BGP-VPLS
#    NAME   PEER          BGP-VPLS
0 RD vpls1  192.170.0.2   VPLS
1 RD vpls2  192.170.0.5   VPLS
2 RD vpls3  192.170.0.6   VPLS
[admin@LER0] > /int br port print
Flags: D - DYNAMIC; H - HW-OFFLOAD
Columns: INTERFACE, BRIDGE, HW, PVID, PRIORITY, PATH-COST, INTERNAL-PATH-COST, HORIZON
#    INTERFACE  BRIDGE  HW   PVID  PRIORITY  PATH-COST  INTERNAL-PATH-COST  HORIZON
0  H ether1     VPLS    yes     1  0x80             10                  10  none
1 D  vpls1      VPLS            1  0x80             10                  10  none
2 D  vpls2      VPLS            1  0x80             10                  10  none
3 D  vpls3      VPLS            1  0x80             10                  10  none
```

Fig. 118. Interfaces VPLS aprendidas dinámicamente.

3.3.9.8.2. Verificación final

Para verificar el correcto funcionamiento, podemos realizar una serie de procesos que nos cerciorarán que todo está perfectamente operativo.

Primeramente, podemos realizar pings entre todos los PC's de la red y ver que se obtiene las respuestas a los paquetes icmp enviados.

Con esto vemos que hay conexión entre equipos, pero para comprobar que se establece a través de los túneles, podemos monitorizar los interfaces VPLS que se han generado dinámicamente. Con ello veremos la etiqueta local, que servirá para que el otro extremo envíe la información; la etiqueta remota, que es la asignada por el otro extremo para la comunicación hacia dicho peer; y los diferentes atributos relacionados con el siguiente salto, etiqueta, interface y.nexthop.

```
[admin@LER0] /interface/vpls> monitor vpls1
  remote-label: 10001
   local-label: 18
      nexthops: { label=20003; nh=10.0.0.2%ether2; interface=ether2 }

[admin@LER0] /interface/vpls> monitor vpls2
  remote-label: 40001
   local-label: 19
      nexthops: { label=20005; nh=10.0.0.2%ether2; interface=ether2 }

[admin@LER0] /interface/vpls> monitor vpls3
  remote-label: 50001
   local-label: 20
      nexthops: { label=20006; nh=10.0.0.2%ether2; interface=ether2 }
```

Fig. 119. Monitorización interfaces VPLS.

3.3.9.8.3. Actividades propuestas

6. Realice un ping desde el PC1 al PC4 ¿Hay alguna diferencia en algún enlace del túnel VPLS respecto a VPLS (LDP)? Recuerde que tiene que conectar el PC que ejecuta el WireShark al puerto ether7 de los diferentes routers según se configuró el Port Mirroring en ellos.

LER1-LSR2:

```
   45 9.187371    192.168.1.20    192.168.1.40    ICMP     100 Echo (ping) request  id=0x0001, seq=5/1280, ttl=128 (reply in 46)
   46 9.187828    192.168.1.40    192.168.1.20    ICMP      96 Echo (ping) reply    id=0x0001, seq=5/1280, ttl=128 (request in 45)

> Frame 45: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface \Device\NPF_{6E16DC1B-7984-4C64-BE15-ABD9B8BA58E3}, id 0
v Ethernet II, Src: Routerbo_51:e1:25 (48:a9:8a:51:e1:25), Dst: Routerbo_51:f2:01 (48:a9:8a:51:f2:01)
    > Destination: Routerbo_51:f2:01 (48:a9:8a:51:f2:01)
    > Source: Routerbo_51:e1:25 (48:a9:8a:51:e1:25)
      Type: MPLS label switched packet (0x8847)
v MultiProtocol Label Switching Header, Label: 20006, Exp: 0, S: 0, TTL: 255
      0000 0100 1110 0010 0110 .... .... .... = MPLS Label: 20006 (0x04e26)
      .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
      .... .... .... .... .... ...0 .... .... = MPLS Bottom Of Label Stack: 0
      .... .... .... .... .... .... 1111 1111 = MPLS TTL: 255
v MultiProtocol Label Switching Header, Label: 50002, Exp: 0, S: 1, TTL: 255
      0000 1100 0011 0101 0010 .... .... .... = MPLS Label: 50002 (0x0c352)
      .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
      .... .... .... .... .... ...1 .... .... = MPLS Bottom Of Label Stack: 1
      .... .... .... .... .... .... 1111 1111 = MPLS TTL: 255
v PW Ethernet Control Word
      Sequence Number: 4905
v Ethernet II, Src: Giga-Byt_a6:43:94 (fc:aa:14:a6:43:94), Dst: Giga-Byt_a6:3e:ed (fc:aa:14:a6:3e:ed)
    > Destination: Giga-Byt_a6:3e:ed (fc:aa:14:a6:3e:ed)
    > Source: Giga-Byt_a6:43:94 (fc:aa:14:a6:43:94)
      Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 192.168.1.20, Dst: 192.168.1.40
> Internet Control Message Protocol
```

```
   45 9.187371    192.168.1.20    192.168.1.40    ICMP     100 Echo (ping) request  id=0x0001, seq=5/1280, ttl=128 (reply in 46)
   46 9.187828    192.168.1.40    192.168.1.20    ICMP      96 Echo (ping) reply    id=0x0001, seq=5/1280, ttl=128 (request in 45)

> Frame 46: 96 bytes on wire (768 bits), 96 bytes captured (768 bits) on interface \Device\NPF_{6E16DC1B-7984-4C64-BE15-ABD9B8BA58E3}, id 0
v Ethernet II, Src: Routerbo_51:f2:01 (48:a9:8a:51:f2:01), Dst: Routerbo_51:e1:25 (48:a9:8a:51:e1:25)
    > Destination: Routerbo_51:e1:25 (48:a9:8a:51:e1:25)
    > Source: Routerbo_51:f2:01 (48:a9:8a:51:f2:01)
      Type: MPLS label switched packet (0x8847)
v MultiProtocol Label Switching Header, Label: 10004, Exp: 0, S: 1, TTL: 255
      0000 0010 0111 0001 0100 .... .... .... = MPLS Label: 10004 (0x02714)
      .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
      .... .... .... .... .... ...1 .... .... = MPLS Bottom Of Label Stack: 1
      .... .... .... .... .... .... 1111 1111 = MPLS TTL: 255
v PW Ethernet Control Word
      Sequence Number: 614
v Ethernet II, Src: Giga-Byt_a6:3e:ed (fc:aa:14:a6:3e:ed), Dst: Giga-Byt_a6:43:94 (fc:aa:14:a6:43:94)
    > Destination: Giga-Byt_a6:43:94 (fc:aa:14:a6:43:94)
    > Source: Giga-Byt_a6:3e:ed (fc:aa:14:a6:3e:ed)
      Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 192.168.1.40, Dst: 192.168.1.20
> Internet Control Message Protocol
```

Fig. 120. Capturas ICMP enlace LER1-LSR2.

LSR2-LSR3:

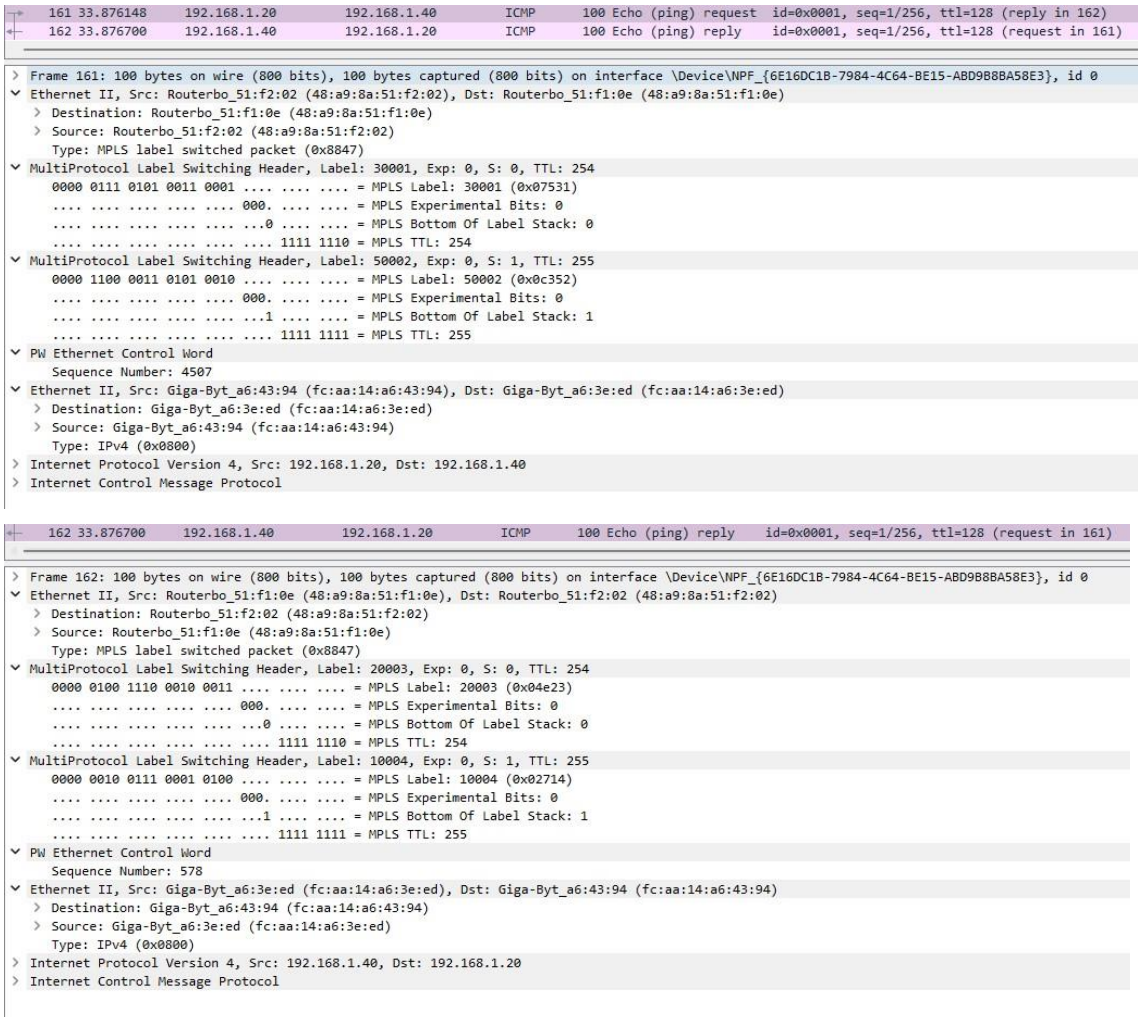
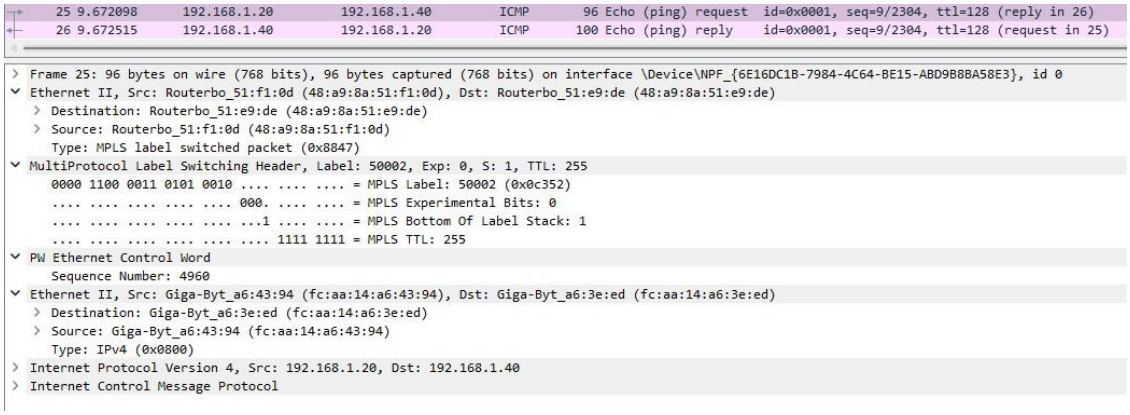


Fig. 121. Capturas ICMP enlace LSR2-LSR3.

LSR3-LER5:



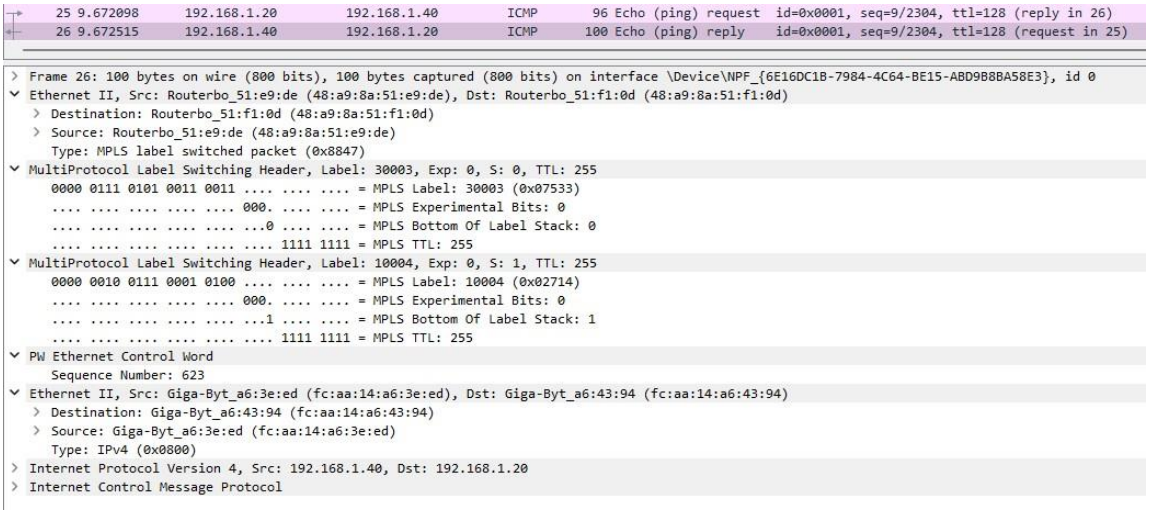


Fig. 122. Capturas ICPM enlace LSR3-LER5.

Podemos observar de nuevo la misma dinámica, una pila de dos etiquetas con la etiqueta del pseudowire como última etiqueta y la de direccionamiento en la red MPLS por debajo. Vemos que las etiquetas toman otros valores distintos.

Al fin y al cabo, vemos que la aplicación es la misma ya que no se aprecian diferencias y es más escalable a través de BGP.

7. ¿Qué pasos deberíamos realizar en la configuración VPLS (BGP) respecto a VPLS (LDP) para conectar un nuevo equipo a la red desde una ubicación geográficamente alejada?

Para poder incluir un equipo a nuestra red transparente VPLS, primero tendremos que conectar a cualquiera de los routers LSR, el router de la nueva ubicación que actuará como un nuevo router LER.

En dicho router LER habría que configurar tanto OSPF como MPLS, teniendo en cuenta que tiene que estar en la misma área que la red MPLS original. Sin olvidar hacer lo propio con el nuevo enlace en el router LSR.

Estos dos primeros pasos no difieren variación respecto a VPLS (LDP). Ahora crearemos el Bridge VPLS en el LER como hacíamos con el resto y le asignaremos la IP de puerta de enlace predeterminada, que tendrá que estar dentro de la subred.

Una vez hecho esto, habría que establecer conexiones BGP entre el nuevo router LER y el resto de routers de este tipo que ya forman parte de la red. Con las sesiones establecidas se creará una instancia BGP-VPLS con los mismos parámetros que en el resto de LER's (*Route Distinguisher*, *Export Route Targets*, *Import Route Targets*) a excepción del identificador *site-id* que será único.

Como hemos visto, a partir de BGP_VPLS se crean dinámicamente las interfaces y se asocian al Bridge, por tanto, solo nos quedará añadir la interfaz física a la que ser conectará el nuevo PC al Bridge VPLS.

Por último, habrá que cerciorarse de que la IP del PC pertenece a la misma subred y puede hacer *ping* con el resto de PC's de nuestra subred.

8. ¿Cómo deberíamos proceder en VPLS (BGP) para configurar una subred diferente sobre la misma topología MPLS?

Para poder tener una nueva subred sobre la misma topología, primero tendremos que elegir la subred a configurar de forma que no haya solapamiento con la original. Por ejemplo, podríamos elegir la subred 192.168.2.0/24.

Una vez teniendo esto claro, crearemos un nuevo Bridge para cada router LER y le asignaremos una dirección IP dentro del rango de la subred y que nos servirá de puerta de enlace predeterminada para los equipos que conectaremos.

Estos pasos son idénticos a una configuración VPLS sobre LDP, pero ahora en vez de crear directamente las interfaces, crearemos una nueva instancia VPLS-BGP para cada router LER. El *site-id* será el mismo que la otra instancia creada en el router, sin embargo, el *Route Distinguisher*, el *Export Route Targets* y el *Import Route Targets* serán diferentes respecto a la otra instancia, pero idénticos a todos los routers LER para esta subred. Por ejemplo, 2:2. No hay que olvidar que estas instancias se asocian con el nuevo Bridge creado.

Como sabemos ya de sobra, las interfaces VPLS y las asociaciones al Bridge se crean dinámicamente. Por tanto, lo único que tenemos que hacer sobre el Bridge es asociar una interfaz física diferente y que esté libre.

Ya por último conectaremos los PC's a las interfaces físicas asociadas con el Bridge y les daremos a los PC's una dirección IP dentro de la subred. No hay que olvidar configurar la puerta de enlace predeterminada con la IP del Bridge.

4. Prácticas en el entorno de simulación GNS3

4.1. Práctica 1 "Configuración Básica de una Red MPLS"

4.1.1. Introducción

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

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

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

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

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

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

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

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

4.1.2. Etiqueta MPLS

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

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

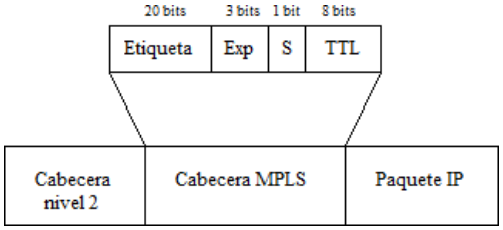


Fig. 123. Formato de cabecera MPLS.

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

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

4.1.3. Elementos MPLS

4.1.3.1. Forwarding Equivalence Class (FEC)

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

4.1.3.2. Label Switched Path (LSP)

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

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

4.1.3.3. Label Switch Routers (LSR)

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

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

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

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

4.1.3.4. Label Edge Routers (LER)

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

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

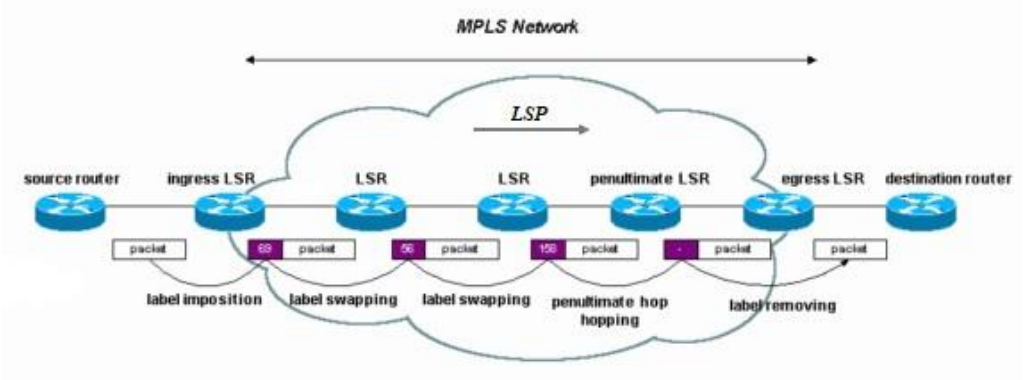


Fig. 124. Topología red MPLS.

4.1.4. Distribución de etiquetas

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

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

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

Existen varios mecanismos para la distribución:

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

En el caso de la primera práctica, utilizaremos LDP.

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

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

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

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

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

Los LSRs anuncian las direcciones de sus interfaces con mensajes "Address", o retiran las ya anunciadas con "Address Withdraw". Tras estos mensajes, se envían entre ellos "Label Request" para solicitar el mapeado de un FEC (un FEC puede ser una IP de un LSR) y

responden con “Label Mapping”, anunciando el mapeado de una etiqueta al FEC.

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

Los mensajes LDP se pueden clasificar en cuatro tipos:

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

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

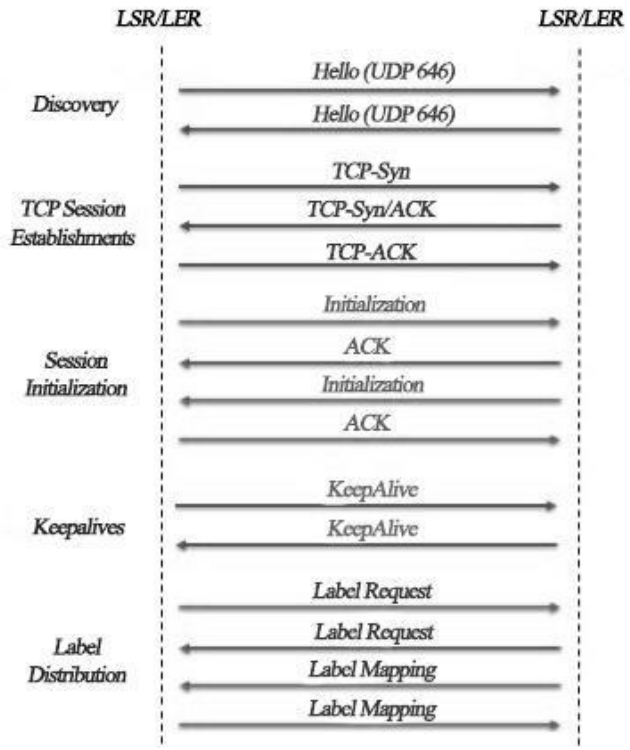


Fig. 125. Operaciones LDP.

4.1.5. Objetivos

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 Mikrotik.

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 por medio de WireShark.

4.1.6. Elementos necesarios

Para la realización de la presente práctica se utilizará el entorno de simulación GNS3 con el router MikroTik CRS328-24G-4S previamente instalado a través de la VM y los PC's virtuales disponibles, VPCS.

4.1.7. Topología de red

La topología de red a montar es la siguiente:

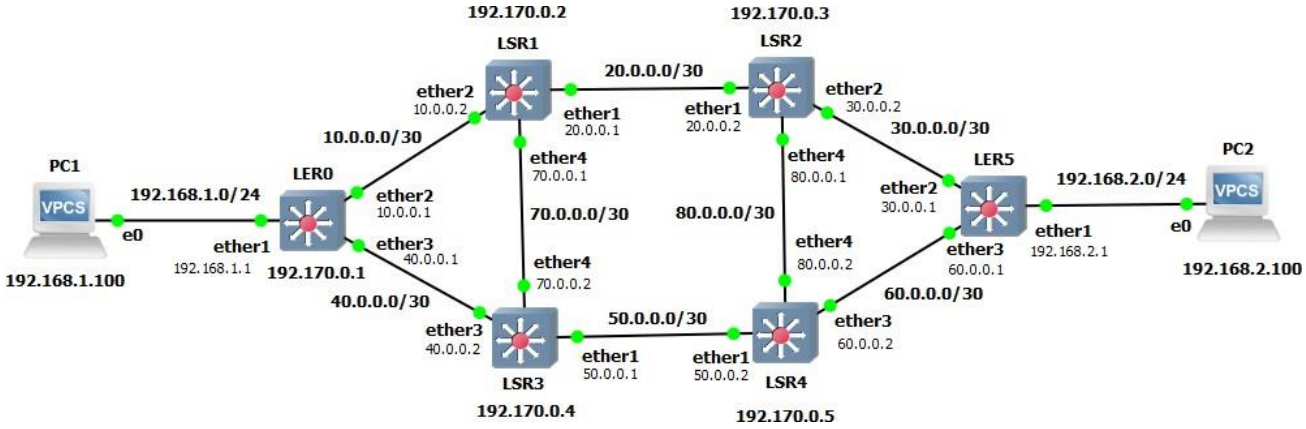


Fig. 126. Topología red MPLS Práctica 1.

En la *tabla 16* se muestran las direcciones IP y las máscaras de red de cada interfaz de los distintos equipos de la red.

Como se puede observar en la tabla, todos los routers utilizan 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.

Dispositivo	Interfaz	Dirección IP	Máscara de red	Gateway predeterminado
LER0	Lo0	192.170.0.1	255.255.255.255	N/A
	Ether1	192.168.1.1	255.255.255.0	N/A
	Ether2	10.0.0.1	255.255.255.252	N/A
	Ether3	40.0.0.1	255.255.255.252	N/A
LSR1	Lo0	192.170.0.2	255.255.255.255	N/A
	Ether1	20.0.0.1	255.255.255.252	N/A
	Ether2	10.0.0.2	255.255.255.252	N/A
	Ether4	70.0.0.1	255.255.255.252	N/A
LSR2	Lo0	192.170.0.3	255.255.255.255	N/A
	Ether1	20.0.0.2	255.255.255.252	N/A
	Ether2	30.0.0.2	255.255.255.252	N/A
	Ether4	80.0.0.1	255.255.255.252	N/A
LSR3	Lo0	192.170.0.4	255.255.255.255	N/A
	Ether1	50.0.0.1	255.255.255.252	N/A
	Ether3	40.0.0.2	255.255.255.252	N/A
	Ether4	70.0.0.2	255.255.255.252	N/A
LSR4	Lo0	192.170.0.5	255.255.255.255	N/A
	Ether1	50.0.0.2	255.255.255.252	N/A
	Ether3	60.0.0.2	255.255.255.252	N/A
	Ether4	80.0.0.2	255.255.255.252	N/A
LER5	Lo0	192.170.0.6	255.255.255.255	N/A
	Ether1	192.168.2.1	255.255.255.0	N/A
	Ether2	30.0.0.1	255.255.255.252	N/A
	Ether3	60.0.0.1	255.255.255.252	N/A
PC1	E0	192.168.1.100	255.255.255.0	192.168.1.1
PC2	E0	192.168.2.100	255.255.255.0	192.168.2.1

Tabla 16. Tabla de direccionamiento.

4.1.8. Configuración de la red

4.1.8.1.1. Crear proyecto y montar topología

Primero crearemos un nuevo proyecto al que le daremos un nombre que podamos asociar con esta práctica. Lo haremos como ya se ha explicado anteriormente.

Luego realizaremos la configuración de los distintos equipos que conforman la red. Para ello, desplazaremos los routers y los VPCS desde sus correspondientes grupos en la barra de elementos al área de trabajo. Haciendo doble click encima de cada nodo o botón derecho y *Configure*, se abrirá la ventana de configuración de sus propiedades donde procederemos a cambiar los nombres.

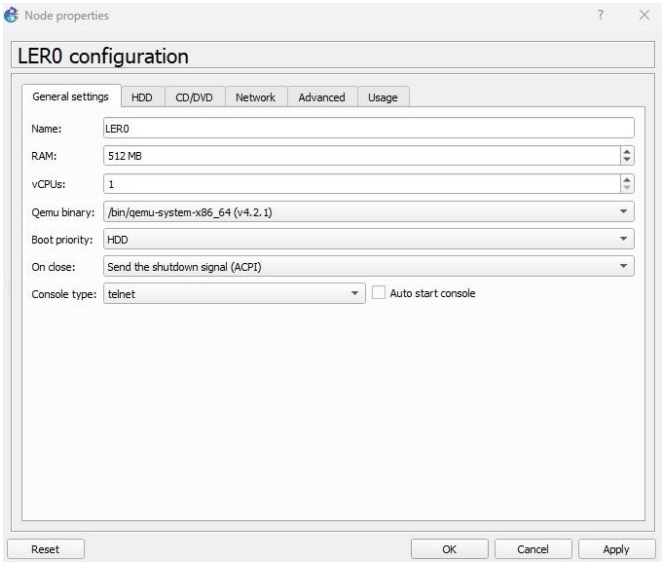


Fig. 127. GNS3. Configure LER0.

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

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

4.1.8.1.2. Iniciar routers

Con la topología montada, iniciaremos la red clicando en *Start all nodes* (triángulo verde de la barra de herramientas). Una vez esté la red inicializada (todas las luces de color verde), podremos hacer doble click en cada uno de los equipos para abrir el *terminal PuTTY* donde se introducirán los comandos que se explican en esta práctica.

Hay que tener en cuenta que la iniciación del router hasta que pide introducir el *login* y el *password* lleva un tiempo.

El login por defecto es *admin* y el *password* en blanco. Esto nos permite acceder a la configuración, pero antes nos pedirá cambiar el *password*. Recomendamos usar *1234* para todos los routers y evitar así problemas.

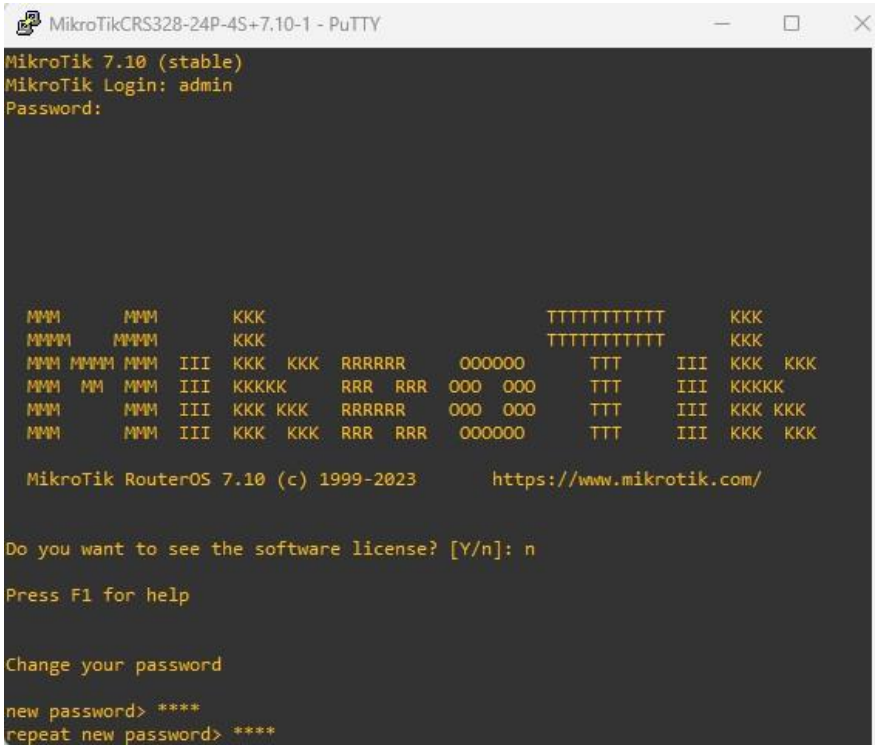


Fig. 128. GNS3. PuTTY. Iniciar router.

Como el proyecto es nuevo y los equipos acaban de ser instalados en la red, no es necesario borrar sus configuraciones. En el caso de que la red se diese ya creada, se tendría que borrar cualquier configuración existente en los routers con el comando /*system reset-configuration* para asegurarnos de que no haya interferencias con la configuración a realizar.

4.1.8.1.3. Crear Interfaz Loopback y asignar IP's

En primer lugar, vamos a crear la interfaz Loopback a través de la pestaña Bridge de Interfaces y que llamaremos lo0. Ejecutaremos los siguientes comandos en el router LER0:

```
[admin@MikroTik] > /interface bridge
[admin@MikroTik] /interface/bridge > add name=lo0
```

Con la interfaz Loopback creada, asignaremos las direcciones IP a cada interfaz del router con los siguientes comandos:

```
[admin@MikroTik] > /ip address
[admin@MikroTik] /ip/address > add
address: 192.170.0.1/32
interface: lo0
[admin@MikroTik] /ip/address > add
address: 192.168.1.1/24
interface: ether1
[admin@MikroTik] /ip/address > add
address: 10.0.0.1/30
interface: ether2
[admin@MikroTik] /ip/address > add
address: 40.0.0.1/30
interface: ether3
```

Si hemos realizado correctamente los pasos, en el terminal PuTTY debería quedar algo similar a la imagen siguiente:

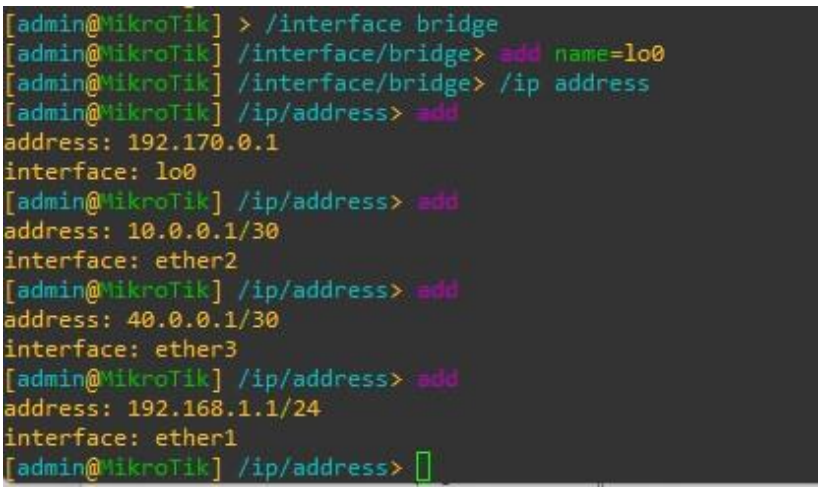


Fig. 129. Configuraciónm de IP's en MikroTik.

Podemos comprobar que se han asignado bien las IP's ejecutando el comando *print:*

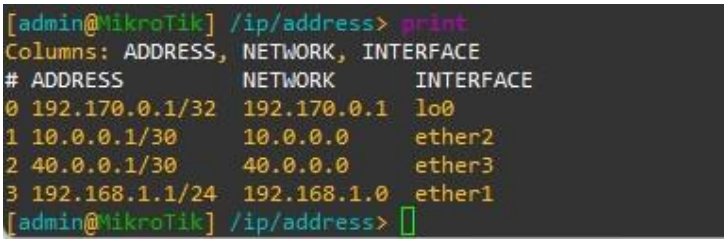


Fig. 130. IP's asignadas en MikroTik.

Habrá que repetir de manera análoga estos pasos para el resto de routers.

4.1.8.1.4. Configurar OSPF

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.

Primero crearemos una instancia única en el router que permite que existan múltiples instancias OSPF en un enrutador, lo que permite separar y gestionar diferentes áreas de tu red de manera independiente.

También crearemos un área OSPF, que es un grupo de routers que comparten información del estado de enlace. A las diferentes subredes a las que se conecta cada router le asignaremos dicha área.

A continuación, se muestra la configuración para el router LER0:

```
[admin@MikroTik] > /routing ospf instance
[admin@MikroTik] /routing/ospf/instance > add name=backbone router-id=192.170.0.1
[admin@MikroTik] /routing/ospf/instance > .. area
[admin@MikroTik] /routing/ospf/area > add name= backbone area-id=0.0.0.0 instance= backbone
[admin@MikroTik] /routing/ospf/area > .. interface-template
[admin@MikroTik] /routing/ospf/interface-template > add interface=lo0 networks=192.170.0.1/32 area= backbone
[admin@MikroTik] /routing/ospf/interface-template > add int=ether1 net=192.168.1.0/24 ar= backbone
[admin@MikroTik] /routing/ospf/interface-template > add int=ether2 net=10.0.0.0/30 ar= backbone
[admin@MikroTik] /routing/ospf/interface-template > add int=ether3 net=40.0.0.0/30 ar= backbone
```

Si se han seguido bien los pasos, en el terminal PuTTY debería quedar algo similar a la siguiente imagen:

```
[admin@MikroTik] > /routing ospf instance
[admin@MikroTik] /routing/ospf/instance> add name=backbone router-id=192.170.0.1
[admin@MikroTik] /routing/ospf/instance> .. area
[admin@MikroTik] /routing/ospf/area> add name=backbone area-id=0.0.0.0 instance=backbone
[admin@MikroTik] /routing/ospf/area> ..interface-template
[admin@MikroTik] /routing/ospf/interface-template> add interface=lo0 networks=192.170.0.1/32 area=backbone
[admin@MikroTik] /routing/ospf/interface-template> add interface=ether1 networks=192.168.1.0/24 area=backbone
[admin@MikroTik] /routing/ospf/interface-template> add interface=ether2 networks=10.0.0.0/30 area=backbone
[admin@MikroTik] /routing/ospf/interface-template> add interface=ether3 networks=40.0.0.0/30 area=backbone
[admin@MikroTik] /routing/ospf/interface-template>
```

Fig. 131. Configuración OSPF en MikroTik.

Análogamente configuraremos el resto de routers con el protocolo OSPF.

Para comprobar que se ha configurado el protocolo OSPF con normalidad se puede acceder a */ip route* y ejecutar el comando *print* para comprobar los destinos aprendidos a través de OSPF y su puerta de enlace correspondiente. Así mismo también podemos ver los destinos directamente conectados.

```
[admin@LER0] /ip/route> print
Flags: D - DYNAMIC; A - ACTIVE; c, o, y - BGP-MPLS-VPN; + - ECMP
Columns: DST-ADDRESS, GATEWAY, DISTANCE
     DST-ADDRESS     GATEWAY          DISTANCE
DAc  10.0.0.0/30     ether2                  0
DAo  20.0.0.0/30     10.0.0.2%ether2       110
DAo  30.0.0.0/30     10.0.0.2%ether2       110
DAc  40.0.0.0/30     ether3                  0
DAo  50.0.0.0/30     40.0.0.2%ether3       110
DAo  60.0.0.0/30     40.0.0.2%ether3       110
DAo+ 70.0.0.0/30     40.0.0.2%ether3       110
DAo+ 70.0.0.0/30     10.0.0.2%ether2       110
DAo+ 80.0.0.0/30     40.0.0.2%ether3       110
DAo+ 80.0.0.0/30     10.0.0.2%ether2       110
DAc  192.168.1.0/24  ether1                  0
DAo+ 192.168.2.0/24  40.0.0.2%ether3       110
DAo+ 192.168.2.0/24  10.0.0.2%ether2       110
DAc  192.170.0.1/32  lo0                     0
DAo  192.170.0.2/32  10.0.0.2%ether2       110
DAo  192.170.0.3/32  10.0.0.2%ether2       110
DAo  192.170.0.4/32  40.0.0.2%ether3       110
DAo  192.170.0.5/32  40.0.0.2%ether3       110
DAo+ 192.170.0.6/32  40.0.0.2%ether3       110
DAo+ 192.170.0.6/32  10.0.0.2%ether2       110
[admin@LER0] /ip/route>
```

Fig. 132. Lista rutas aprendidas mediante OSPF.

En la primera columna podemos observar las diferentes subredes configuradas en nuestra red, mientras que en la segunda columna observamos la dirección IP y/o la interface por la que se accede a dicha subred desde el router en el que se ejecuta la tabla, en este caso el LER0.

También podemos observas como se han aprendido. Por ejemplo, DAc indica que se ha aprendido dinámicamente y que el enlace está activo y directamente conectado. Por otro lado DAo indica que se ha aprendido dinámicamente por medio de OSPF y que el enlace está activo.

En la última columna, podemos observar la distancia, esta nos evidencia de nuevo como se ha aprendido cada subred, ya que la distancia 110 es la distancia administrativa asociada al protocolo OSPF. Mientras que la distancia 0, ausencia de distancia, es la asociada a una conexión directa.

4.1.8.1.5. Configurar PC's

Con el protocolo OSPF funcionando ya deberíamos poder tener acceso a cualquier equipo de la red desde cualquier equipo de la misma red. Para comprobarlo haremos *ping* y *trace* entre los dos PCs.

La asignación IP es algo diferente a lo que estamos acostumbrados a hacer en un equipo físico, por lo que a continuación se muestra el ejemplo para el PC1. Para poder ejecutar las líneas que se muestran, primero haremos doble clic sobre el VPCS que abrirá el terminal de comandos del PC.

```
PC1> ip 192.168.1.100 255.255.255.0 192.168.1.1
PC1> save
```

Para hacer ***ping*** y ***trace*** solo hay que ejecutar los comandos habituales sobre el mismo *terminal.* También podremos hacer ***ping*** y ***traceroute*** desde el terminal PuTTY de cualquier router con los comandos que se especifican a continuación:

```
[admin@MikroTik] > /tool
[admin@MikroTik] /tool > ping 192.168.2.100
[admin@MikroTik] /tool > traceroute 192.168.2.100
```

4.1.8.1.6. Configurar etiquetado LDP

Una vez tenemos OSPF configurado y todos los equipos accesibles, configuraremos la red MPLS. En esta primera práctica nos limitaremos a configurar el etiquetado por medio de LDP.

Primero tendremos que indicar el tipo de IPs que utilizaremos, en este caso IPv4, que lo haremos con la variable *afi=ip*. También identificaremos el LSR con la dirección IP y la dirección que utilizará el router para transportar los paquetes LDP. Ambas direcciones serán la misma, la asignada a la interfaz LoopBack.

Para terminar de configurar el etiquetado, asignaremos los puertos Ethernet que utilizarán el protocolo LDP con el resto de routers.

A continuación, se muestra la configuración para el router LER0:

```
[admin@MikroTik] > /mpls ldp
[admin@MikroTik] /mpls/ldp > add afi=ip lsr-id=192.170.0.1 transport-addresses=192.170.0.1
[admin@MikroTik] /mpls/ldp > interface
[admin@MikroTik] /mpls/ldp/interface > add
interface: ether2
[admin@MikroTik] /mpls/ldp/interface > add
interface: ether3
```

Si hemos realizado bien la configuración, el terminal PuTTY debería quedar de forma similar a la siguiente imagen:

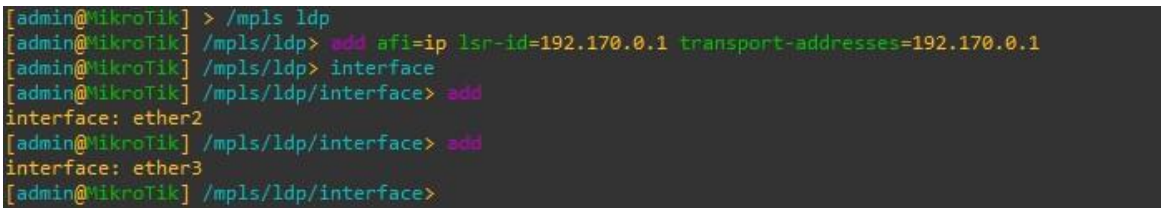


Fig. 133. Configuración MPLS en MikroTik.

También habrá que asignar un rango de etiquetas a cada router para poder observar con más claridad el funcionamiento del protocolo, sabiendo en cada momento que etiqueta es de cada router.

Hay que tener en cuenta que las primeras 15 etiquetas están reservadas y que el rango mínimo es de 1024 labels. Con esta información realizaremos la siguiente asignación:

ROUTER	RANGO ETIQUETAS
LER0	16-9999 (min 16)
LSR1	10000-19999
LSR2	20000-29999
LSR3	30000-39999
LSR4	40000-49999
LER5	50000-59999

Fig. 134. Rango de etiquetas por router.

El comando para el LER0 es el siguiente:

```
[admin@MikroTik] > /mpls settings
[admin@MikroTik] /mpls/settings > set dynamic-label-range=16-9999
```

Habrá que realizar esta misma configuración de manera análoga con el resto de routers. Y una vez lo tengamos podremos hacer diferentes comprobaciones del buen funcionamiento de la configuración.

Con todo esto hecho, podremos consultar los routers vecinos y las direcciones IP's de las interfaces de estos con el siguiente comando desde el contexto ***/mpls/ldp***:

```
[admin@MikroTik] /mpls/ldp > nei
[admin@MikroTik] /mpls/ldp/neighbor > print
```

A continuación, podemos ver la ejecución del comando en el router LSR1:

Fig. 135. Tabla neighbor de LDP.

Podemos observar la dirección de trasporte local que hemos asignado para LSR1 que corresponde a la dirección loopback y las que hemos asignado en el resto de routers con los que se conecta. También podemos ver las direcciones asignadas a las diferentes interfaces de los routers a los que está conectado el LSR1.

También una vez configurada la red es posible acceder a la table LIB, aunque para tener toda la información de dicha tabla hay que ejecutar dos comandos. Ambos comandos se ejecutarán desde el contexto ***/mpls/ldp***. Con el comando ***local-mapping print*** visualizaremos el mapeo local del protocolo LDP. Y con el comando ***remote-mapping print*** visualizaremos el mapeo remoto del protocolo LDP.

A continuación, se muestran ambos mapeos del router LSR1:

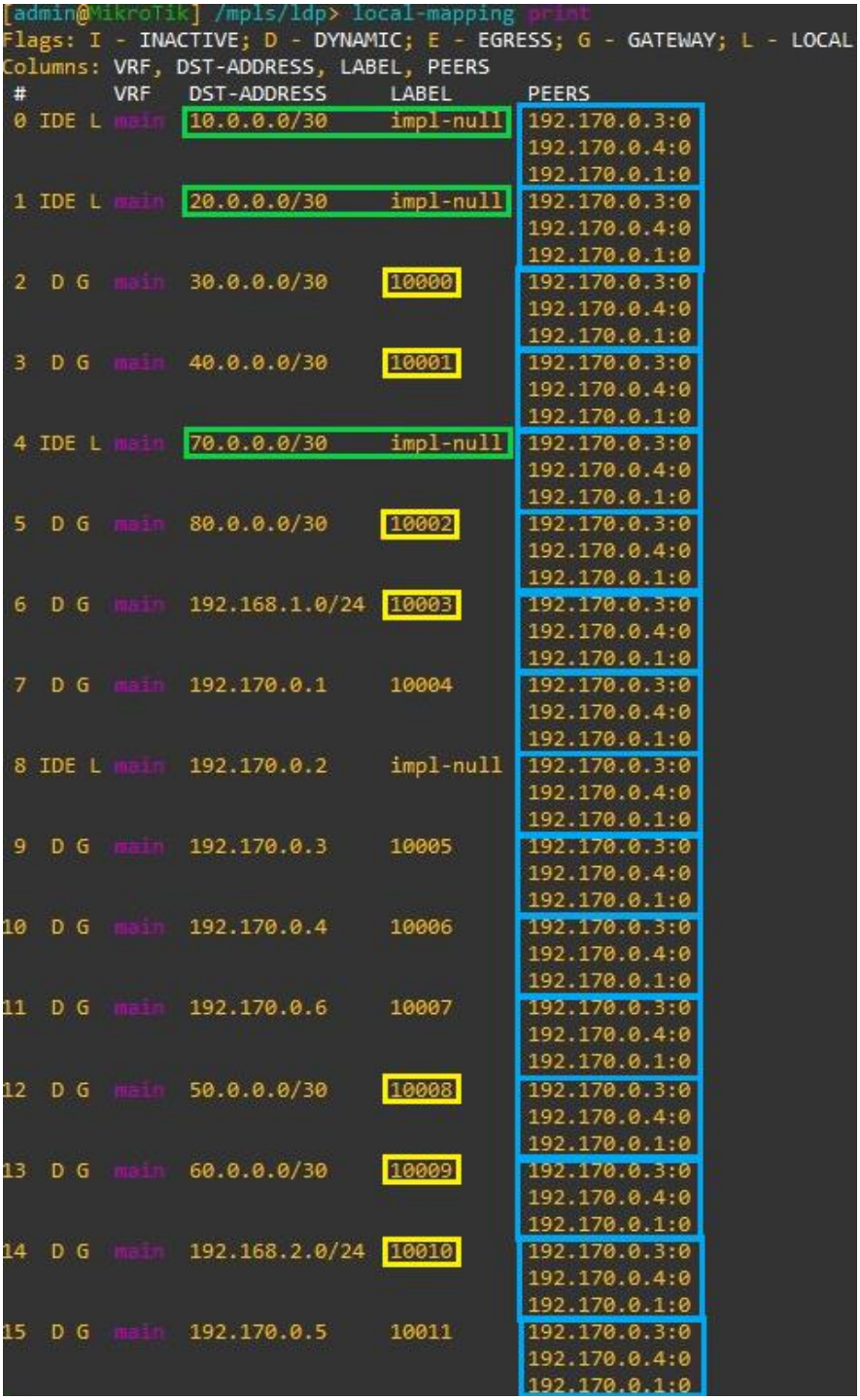


Fig. 136. Tabla local-mapping.

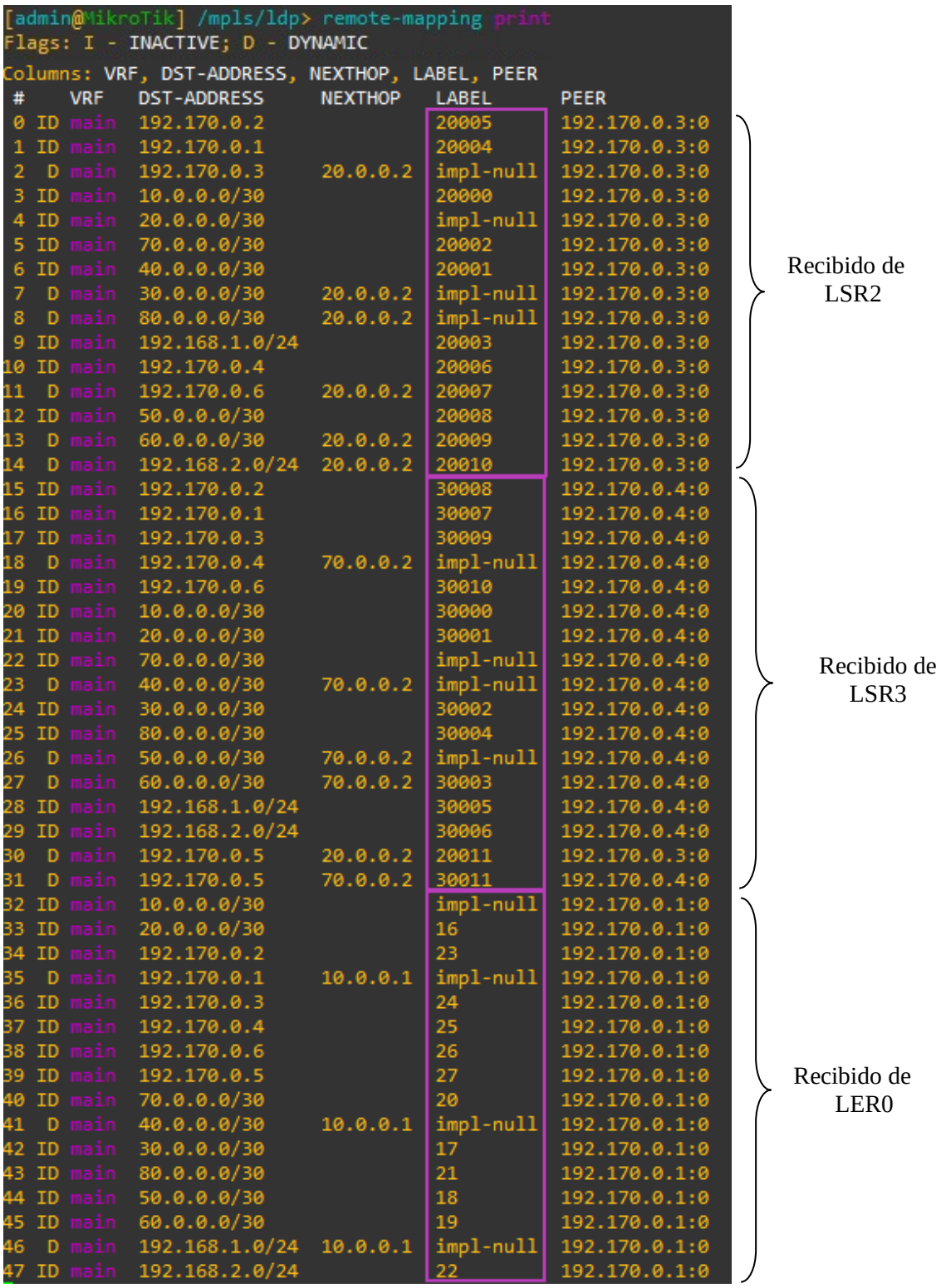


Fig. 137. Tabla remote-mapping.

En la tabla ***Local Mapping*** podemos encontrar las direcciones destino y la etiqueta local que se añadirá en la cabecera del paquete que salga hacia el destino en cuestión, así como los peers con los que está conectado el router.

Podemos observar tantas entradas en esta tabla, ya que desde el router LSR1 se asigna una etiqueta para cada subred, ya que esta etiqueta será utilizada para que el LSR1 reciba los paquetes procedentes del resto de subredes.

La columna de peers indica las direcciones de loopback de los routers que hay conectados al router LSR1, por eso podemos observar que siempre son las mismas direcciones de loopback para todas las entradas de la tabla.

Para las subredes 10.0.0.0, 20.0.0.0 y 70.0.0.0 observamos que no se asigna etiqueta ya que están directamente conectadas al router LSR1 y no precisa de esta para recibir paquetes desde dichas subredes.

En la tabla ***Remote Mapping*** podemos localizar junto con la red de destino, la etiqueta que añadirá el router remoto a los paquetes que envíe, el siguiente salto para alcanzar su destino y el peer que corresponde con el camino elegido de los que sean posibles.

Podemos observar tantas entradas en la tabla ya que para un mismo destino existe la posibilidad de acceder a él por diferentes rutas, es decir teniendo un siguiente salto diferente y por tanto una etiqueta asignada también diferente.

En la columna de las etiquetas podemos observar la variedad de estas en cuanto a rango, ya que estas etiquetas son las asignadas por los diferentes routers, que como se ha comentado ya son de diferente rango para una mejor visualización de la trazabilidad de los paquetes hacia un router concreto en cuestión.

Desde el contexto ***/mpls*** podemos ejecutar el comando ***forwarding-table print*** que nos permitirá visualizar la tabla LFIB. Esta tabla es la encargada de realizar la conmutación de paquetes a través de la red.

En la siguiente imagen se muestra la correspondiente al router LSR1:

```
[admin@MikroTik] /mpls/ldp> .. forwarding-table print
Flags: L, V - VPLS
Columns: LABEL, VRF, PREFIX, NEXTHOPS
 #   LABEL  VRF   PREFIX          NEXTHOPS
 0 L 10001  main  40.0.0.0/30     { label=impl-null; nh=10.0.0.1; interface=ether2 }
                                  { label=impl-null; nh=70.0.0.2; interface=ether4 }
 1 L 10003  main  60.0.0.0/30     { label=30003; nh=70.0.0.2; interface=ether4 }
                                  { label=20000; nh=20.0.0.2; interface=ether1 }
 2 L 10005  main  192.168.1.0/24  { label=impl-null; nh=10.0.0.1; interface=ether2 }
 3 L 10007  main  192.170.0.1     { label=impl-null; nh=10.0.0.1; interface=ether2 }
 4 L 10010  main  192.170.0.5     { label=30008; nh=70.0.0.2; interface=ether4 }
                                  { label=20003; nh=20.0.0.2; interface=ether1 }
 5 L 10002  main  50.0.0.0/30     { label=impl-null; nh=70.0.0.2; interface=ether4 }
 6 L 10009  main  192.170.0.4     { label=impl-null; nh=70.0.0.2; interface=ether4 }
 7 L 10000  main  30.0.0.0/30     { label=impl-null; nh=20.0.0.2; interface=ether1 }
 8 L 10004  main  80.0.0.0/30     { label=impl-null; nh=20.0.0.2; interface=ether1 }
 9 L 10006  main  192.168.2.0/24  { label=20001; nh=20.0.0.2; interface=ether1 }
10 L 10008  main  192.170.0.3     { label=impl-null; nh=20.0.0.2; interface=ether1 }
11 L 10011  main  192.170.0.6     { label=20002; nh=20.0.0.2; interface=ether1 }
[admin@MikroTik] /mpls/ldp>
```

Fig. 138. Tabla fowarding de LDP.

Podemos observar de izquierda a derecha la etiqueta local, el identificador de destino, la etiqueta de salida, el siguiente salto y la interfaz de salida.

Las etiquetas de la columna LABEL son de un mismo rango, ya que como venimos diciendo son las asignadas localmente por el router para que lleguen a este los paquetes desde el resto de destinos.

En la columna PREFIX observamos los diferentes destinos a los que tiene acceso el router LSR1. Los destinos son las diferentes subredes creadas en la topología, aunque también se pueden observar las direcciones *loopback* de los routers de la red.

En la columna NEXTHOP se presentan tres datos referentes al siguiente salto para llegar a los destinos ya comentados. Primero se observa la etiqueta del siguiente salto que se añade a la cabecera del paquete. Junto a esta etiquete se muestra el interfaz de salida a modo de dirección IP y a modo de nombre del propio intrfaz.

4.1.8.1.7. Guardar configuración

Para guardar la configuración ejecutamos el siguiente comando, que generará un archivo con el script de todo lo que hemos configurado:

```
[admin@MikroTik]  > export file=flash/myconfig.rsc
```

A priori, no podremos guardar el archivo en local, pero nos servirá para recuperar la configuración si se hace algún borrado erróneo o se quiere volver a la configuración guardada tras ciertas modificaciones.

4.1.9. Ejercicios propuestos

1. **Utilizando el comando traceroute, determinar cuántas rutas pueden seguir los paquetes desde el PC-1 hasta el PC-2. Una vez obtenido filtrar las rutas para verlas individualmente y determinar el camino que siguen, esto es posible añadiendo interface = etherx al comando original siendo x el número de la interfaz de salida desde LER0.**

```
[admin@MikroTik] /tool> traceroute 192.168.2.100
Columns: ADDRESS, LOSS, SENT, LAST, AVG, BEST, WORST, STD-DEV,
STATUS
# ADDRESS      LOSS SENT LAST  AVG BEST WORST STD-DEV STATUS
1 40.0.0.2     0%   320 0.5ms 0.5 0.4  0.8   0       <MPLS:L=30011,E=0>
  10.0.0.2
2 50.0.0.2     0%   320 0.3ms 0.4 0.3  1.2   0.1     <MPLS:L=40011,E=0>
  20.0.0.2
3 60.0.0.1     0%   320 0.3ms 0.3 0.2  0.5   0
  30.0.0.1
4 192.168.2.100 0%   320 0.5ms 0.5 0.4  3.1   0.2
```

Utilizando el comando traceroute podemos determinar que existen dos posibles rutas ya que observamos dos entradas en cada salto.

```
[admin@MikroTik] /tool> traceroute interface = ether2 192.168.2.100
Columns: ADDRESS, LOSS, SENT, LAST, AVG, BEST, WORST, STD-DEV,
STATUS
# ADDRESS      LOSS SENT LAST  AVG BEST WORST STD-DEV STATUS
1 10.0.0.2     0%   320 0.5ms 0.5 0.4  0.8   0       <MPLS:L=10011,E=0>
2 20.0.0.2     0%   320 0.3ms 0.4 0.3  1.2   0.1     <MPLS:L=20011,E=0>
3 30.0.0.1     0%   320 0.3ms 0.3 0.2  0.5   0

4 192.168.2.100 0%   320 0.5ms 0.5 0.4  3.1   0.2
```

Por la interfaz ether2, los paquetes de PC1 a PC2 tendrán el primer salto en LSR1 usando la etiqueta 10011, siendo el siguiente salto el LSR2 cambiando la etiqueta 10011 por la 20011, siendo el último salto de la red MPLS el router LER5 que elimna la etiqueta al ser el PHP.

```
[admin@MikroTik] /tool> traceroute interface = ether3 192.168.2.100
Columns: ADDRESS, LOSS, SENT, LAST, AVG, BEST, WORST, STD-DEV,
STATUS
# ADDRESS      LOSS SENT LAST  AVG BEST WORST STD-DEV STATUS
1 40.0.0.2     0%   320 0.5ms 0.5 0.4  0.8   0       <MPLS:L=30011,E=0>
2 50.0.0.2     0%   320 0.3ms 0.4 0.3  1.2   0.1     <MPLS:L=40011,E=0>
3 60.0.0.1     0%   320 0.3ms 0.3 0.2  0.5   0

4 192.168.2.100 0%   320 0.5ms 0.5 0.4  3.1   0.2
```

Por la interfaz ether3, los paquetes de PC1 a PC2 tendrán el primer salto en LSR3 usando la etiqueta 30011, siendo el siguiente salto el LSR4 cambiando la etiqueta 30011 por la 40011, siendo el último salto de la red MPLS el router LER5 que elimina la etiqueta al ser el PHP.

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. Ambas rutas que hemos observado con traceroute, servirán para repartir la carga en la red.

3. **Ejecuta en el contexto /mpls/ldp de alguno de los routers los comandos "local-mapping print" y "remote-mapping print", ¿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", "local-mapping" y "remote-mapping" del router LSR1, construir las tablas RIB, FIB, LIB y LFIB de forma similar a los ejercicios realizados en la teoría de la asignatura.**

```
[admin@MikroTik] /mpls/ldp> local-mapping print
Flags: I - INACTIVE; D - DYNAMIC; E - EGRESS; G - GATEWAY; L - LOCAL
Columns: VRF, DST-ADDRESS, LABEL, PEERS
 #          VRF  DST-ADDRESS     LABEL      PEERS
 0 IDE L  main     10.0.0.0/30     impl-null  192.170.0.1:0
                                              192.170.0.3:0
                                              192.170.0.4:0
 1 IDE L  main     20.0.0.0/30     impl-null  192.170.0.1:0
                                              192.170.0.3:0
                                              192.170.0.4:0
 2  D G   main     30.0.0.0/30     10000      192.170.0.1:0
                                              192.170.0.3:0
                                              192.170.0.4:0
 3  D G   main     40.0.0.0/30     10001      192.170.0.1:0
                                              192.170.0.3:0
                                              192.170.0.4:0
 4  D G   main     50.0.0.0/30     10002      192.170.0.1:0
                                              192.170.0.3:0
                                              192.170.0.4:0
 5  D G   main     60.0.0.0/30     10003      192.170.0.1:0
                                              192.170.0.3:0
                                              192.170.0.4:0
 6 IDE L  main     70.0.0.0/30     impl-null  192.170.0.1:0
                                              192.170.0.3:0
                                              192.170.0.4:0
 7  D G   main     80.0.0.0/30     10004      192.170.0.1:0
                                              192.170.0.3:0
                                              192.170.0.4:0
 8  D G   main     192.168.1.0/24  10005      192.170.0.1:0
```

```
                                                  192.170.0.3:0
                                                  192.170.0.4:0
 9  D G  main     192.168.2.0/24      10006       192.170.0.1:0
                                                  192.170.0.3:0
                                                  192.170.0.4:0
10  D G  main     192.170.0.1         10007       192.170.0.1:0
                                                  192.170.0.3:0
                                                  192.170.0.4:0
11 IDE L main     192.170.0.2         impl-null   192.170.0.1:0
                                                  192.170.0.3:0
                                                  192.170.0.4:0
12  D G  main     192.170.0.3         10008       192.170.0.1:0
                                                  192.170.0.3:0
                                                  192.170.0.4:0
13  D G  main     192.170.0.4         10009       192.170.0.1:0
                                                  192.170.0.3:0
                                                  192.170.0.4:0
14  D G  main     192.170.0.5         10010       192.170.0.1:0
                                                  192.170.0.3:0
                                                  192.170.0.4:0
15  D G  main     192.170.0.6         10011       192.170.0.1:0
                                                  192.170.0.3:0
                                                  192.170.0.4:0

[admin@MikroTik] /mpls/ldp> remote-mapping print
Flags: I - INACTIVE; D - DYNAMIC
Columns: VRF, DST-ADDRESS, NEXTHOP, LABEL, PEER
 #     VRF    DST-ADDRESS      NEXTHOP    LABEL       PEER
 0 ID main      20.0.0.0/30                 16       192.170.0.1:0
 1 ID main      10.0.0.0/30                 impl-null  192.170.0.1:0
 2  D main      40.0.0.0/30      10.0.0.1   impl-null  192.170.0.1:0
 3 ID main      50.0.0.0/30                 21       192.170.0.1:0
 4 ID main      30.0.0.0/30                 24       192.170.0.1:0
 5 ID main      192.170.0.2                 17       192.170.0.1:0
 6  D main      192.170.0.1      10.0.0.1   impl-null  192.170.0.1:0
 7 ID main      192.170.0.3                 18       192.170.0.1:0
 8 ID main      192.170.0.4                 20       192.170.0.1:0
 9 ID main      192.170.0.5                 23       192.170.0.1:0
10 ID main      192.170.0.6                 26       192.170.0.1:0
11 ID main      70.0.0.0/30                 19       192.170.0.1:0
12 ID main      80.0.0.0/30                 22       192.170.0.1:0
13 ID main      60.0.0.0/30                 25       192.170.0.1:0
14  D main      192.168.1.0/24   10.0.0.1   impl-null  192.170.0.1:0
15 ID main      192.168.2.0/24              27       192.170.0.1:0
16 ID main      192.170.0.2                 20008    192.170.0.3:0
17 ID main      192.170.0.1                 20007    192.170.0.3:0
18  D main      192.170.0.3      20.0.0.2   impl-null  192.170.0.3:0
19 ID main      192.170.0.4                 20009    192.170.0.3:0
20  D main      192.170.0.5      20.0.0.2   20010    192.170.0.3:0
21 ID main      20.0.0.0/30                 impl-null  192.170.0.3:0
22 ID main      10.0.0.0/30                 20000    192.170.0.3:0
23 ID main      70.0.0.0/30                 20004    192.170.0.3:0
24 ID main      40.0.0.0/30                 20001    192.170.0.3:0
25 ID main      50.0.0.0/30                 20002    192.170.0.3:0
26  D main      80.0.0.0/30      20.0.0.2   impl-null  192.170.0.3:0
```

```
27  D main         30.0.0.0/30          20.0.0.2   impl-null   192.170.0.3:0
28  D main         60.0.0.0/30          20.0.0.2    20003      192.170.0.3:0
29 ID main        192.168.1.0/24                     20005      192.170.0.3:0
30  D main        192.168.2.0/24        20.0.0.2     20006     192.170.0.3:0
31  D main         192.170.0.6          20.0.0.2     20011     192.170.0.3:0
32 ID main         192.170.0.2                       30008     192.170.0.4:0
33 ID main         192.170.0.1                       30007     192.170.0.4:0
34 ID main         192.170.0.3                       30009     192.170.0.4:0
35  D main         192.170.0.4          70.0.0.2   impl-null   192.170.0.4:0
36  D main         192.170.0.5          70.0.0.2     30010     192.170.0.4:0
37 ID main  192.170.0.6                              30011     192.170.0.4:0
38 ID main  20.0.0.0/30                              30001     192.170.0.4:0
39 ID main  10.0.0.0/30                              30000     192.170.0.4:0
40 ID main  70.0.0.0/30                            impl-null   192.170.0.4:0
41  D main  40.0.0.0/30                70.0.0.2   impl-null   192.170.0.4:0
42  D main  50.0.0.0/30                70.0.0.2   impl-null   192.170.0.4:0
43 ID main  80.0.0.0/30                              30004     192.170.0.4:0
44 ID main  30.0.0.0/30                              30002     192.170.0.4:0
45  D main  60.0.0.0/30                70.0.0.2      30003    192.170.0.4:0
46 ID main  192.168.1.0/24                           30005     192.170.0.4:0
47 ID main  192.168.2.0/24                           30006     192.170.0.4:0


[admin@MikroTik] /mpls/ldp> .. forwarding print
Flags: L, V - VPLS
Columns: LABEL, VRF, PREFIX, NEXTHOPS
 #   LABEL  VRF   PREFIX          NEXTHOPS
 0 L 10001  main  40.0.0.0/30     { label=impl-null; nh=70.0.0.2; interface=ether4 }
                                  { label=impl-null; nh=10.0.0.1; interface=ether2 }
1 L 10007  main  192.170.0.1      { label=impl-null; nh=10.0.0.1; interface=ether2 }
2 L 10005  main  192.168.1.0/24   { label=impl-null; nh=10.0.0.1; interface=ether2 }
3 L 10010  main  192.170.0.5      { label=30010; nh=70.0.0.2; interface=ether4 }
                                  { label=20010; nh=20.0.0.2; interface=ether1 }
4 L 10003  main  60.0.0.0/30      { label=30003; nh=70.0.0.2; interface=ether4 }
                                  { label=20003; nh=20.0.0.2; interface=ether1 }
5 L 10008  main  192.170.0.3      { label=impl-null; nh=20.0.0.2; interface=ether1 }
6 L 10004  main  80.0.0.0/30      { label=impl-null; nh=20.0.0.2; interface=ether1 }
7 L 10000  main  30.0.0.0/30      { label=impl-null; nh=20.0.0.2; interface=ether1 }
8 L 10006  main  192.168.2.0/24   { label=20006; nh=20.0.0.2; interface=ether1 }
9 L 10011  main  192.170.0.6      { label=20011; nh=20.0.0.2; interface=ether1 }
10 L 10009  main  192.170.0.4     { label=impl-null; nh=70.0.0.2; interface=ether4 }
11 L 10002  main  50.0.0.0/30     { label=impl-null; nh=70.0.0.2; interface=ether4 }
```

RIB

RED	SALTO
10.0.0.0/30	Dir. conectado
20.0.0.0/30	Dir. conectado
70.0.0.0/30	Dir. conectado
30.0.0.0/30	LSR2
40.0.0.0/30	LSR3
40.0.0.0/30	LER0
50.0.0.0/30	LSR3
60.0.0.0/30	LSR3
60.0.0.0/30	LSR2
80.0.0.0/30	LSR2
192.168.1.0/24	LER0
192.168.2.0/24	LSR2

Tabla 17. Tabla RIB LSR1.

FIB

RED	SALTO	LABEL
10.0.0.0/30	Dir. conectado	-
20.0.0.0/30	Dir. conectado	-
70.0.0.0/30	Dir. conectado	-
30.0.0.0/30	LSR2	-
40.0.0.0/30	LSR3	-
40.0.0.0/30	LER0	-
50.0.0.0/30	LSR3	-
60.0.0.0/30	LSR3	30003
60.0.0.0/30	LSR2	20003
80.0.0.0/30	LSR2	-
192.168.1.0/24	LER0	-
192.168.2.0/24	LSR2	20006

Tabla 18. Tabla FIB LSR1.

LIB

RED	LSR	LABEL
10.0.0.0/30	Dir. conectado	-
20.0.0.0/30	Dir. conectado	-
70.0.0.0/30	Dir. conectado	-
30.0.0.0/30	LOCAL	10000
40.0.0.0/30	LOCAL	10001
50.0.0.0/30	LOCAL	10002
60.0.0.0/30	LOCAL	10003
	LSR3	30003
60.0.0.0/30	LOCAL	10003
	LSR2	20003
80.0.0.0/30	LOCAL	10004
192.168.1.0/24	LOCAL	10005
192.168.2.0/24	LOCAL	10006
	LSR2	20006

Tabla 19. Tabla LIB LSR1.

LFIB

LABEL IN	LABEL OUT	ACTION	SALTO
10000	-	POP	LSR2
10001	-	POP	LSR3
10001	-	POP	LSR3
10002	-	POP	LSR3
10003	30003	SWAP	LSR3
10003	20003	SWAP	LSR2
10004	-	POP	LSR2
10005	-	POP	LER0
10006	20006	SWAP	LSR2

Tabla 20. Tabla LFIB LSR1.

En la última parte de la práctica pasaremos a utilizar el analizador de redes, para ello únicamente nos tenemos que poner sobre el enlace que queremos capturar. Clicando sobre él con el botón derecho podremos seleccionar la opción *Start capture* y después seleccionando *OK* se abrirá el WireShark.

Capturaremos tanto el enlace LER0-LSR1 como el enlace LER0-LSR3, ya que los paquetes podrán elegir una ruta u otra para comunicar un PC con el otro, incluso enviar el *request* por un enlace y el *reply* por el otro.

En ambas capturas, deberemos ejecutar el filtro *icmp* para mostrar los mensajes que se envían en un *ping*.

5. Ejecutar un ping desde PC1 a PC2. ¿Cómo aparecen encapsulados estos paquetes? ¿Puedes reconocer los campos MPLS vistos en teoría?

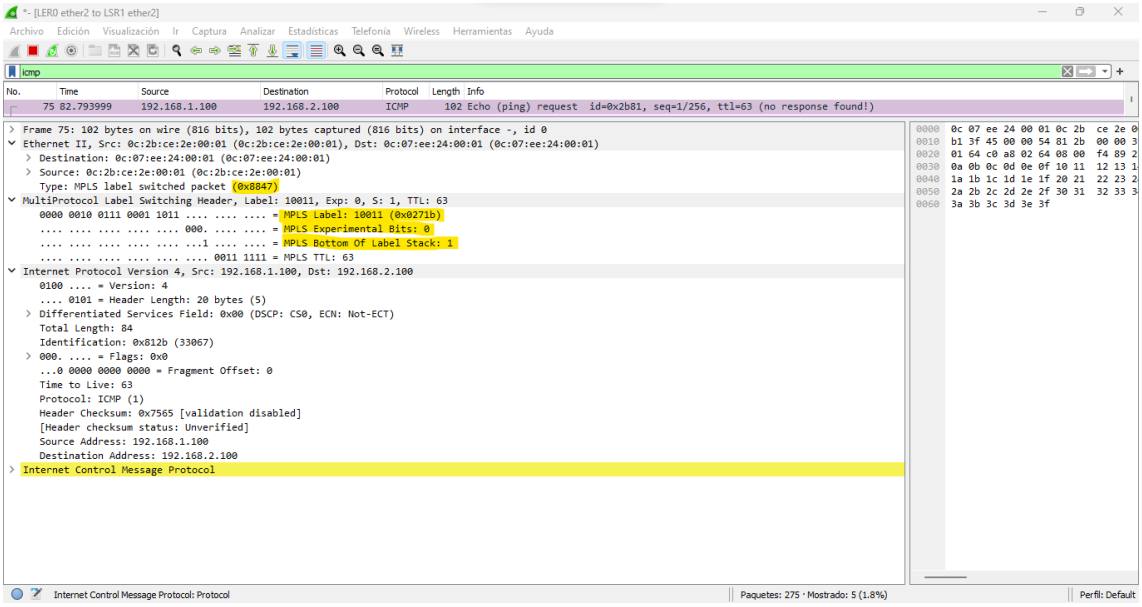


Fig. 139. Captura WireShark de ping entre PC1 y PC2

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 10011.
- 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

4.2. Práctica 2 "Configuración de una Red L3 MPLS VPN"

4.2.1. Introducción

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

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

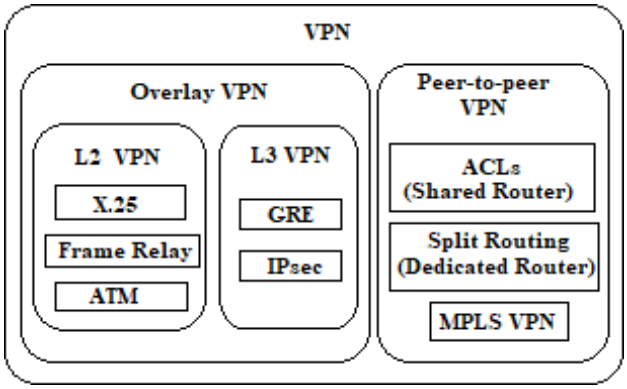


Fig. 140. Modelos VPN.

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

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

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

4.2.2. Componentes y arquitectura L3VPN

4.2.2.1. Customer Edge (CE)

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

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

4.2.2.2. Provider Edge (PE)

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

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

4.2.2.3. Provider (P)

Router MPLS en el backbone de la red. Nunca está conectado a la red cliente. No lleva rutas VPN, ya que solo posee información de la red del proveedor en sus tablas de routing.

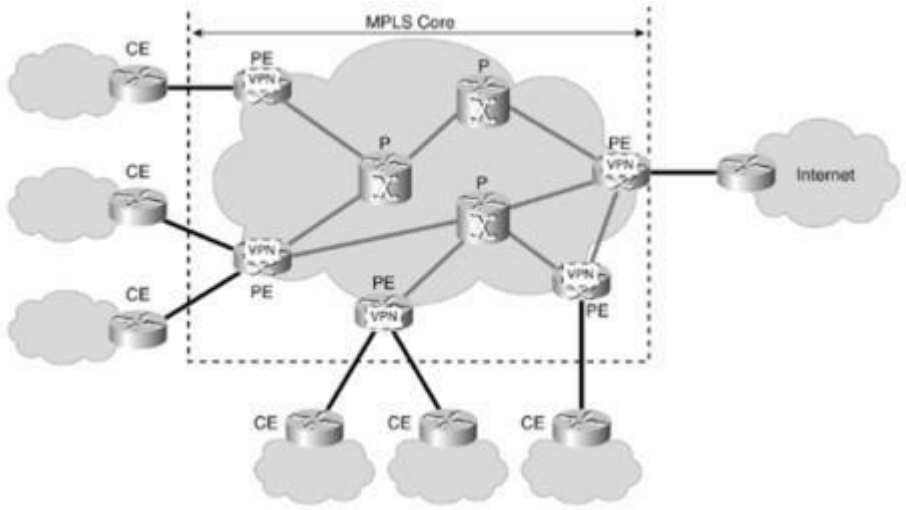


Fig. 141. Topología red L3 MPLS VPN

4.2.2.4. Virtual Routing Forwarding (VRF)

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

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

4.2.2.5. Route Distinguishers (RD)

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

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

4.2.2.6. Route Targets (RT)

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

Dos tipos de RT:

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

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

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

4.2.3. MP-BGP

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

Se puede clasificar de dos formas en función del AS:

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

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

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

4.2.4. Propagación de rutas y envío de paquetes en MPLS VPN

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

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

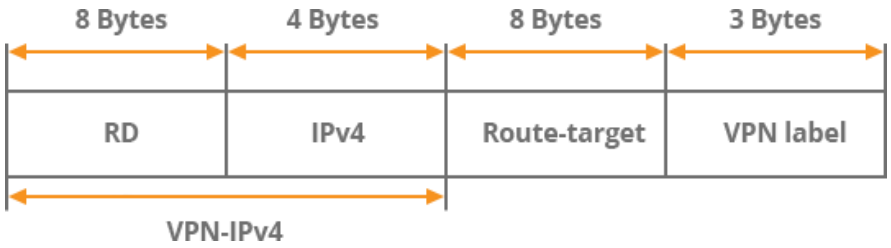


Fig. 142. Mensaje de actualización MP-BGP

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

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

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

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

4.2.5. Objetivos

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 Mikrotik.

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.

4.2.6. Elementos necesarios

Para la realización de la presente práctica se utilizará el entorno de simulación GNS3 con el router MikroTik CRS328-24G-4S previamente instalado a través de la VM y los PC's virtuales disponibles, VPCS.

4.2.7. Topología de red

Vamos a crear una red L3 VPN-MPLS formada por 5 routers Mikrotik CRS328-24G-4S con las mismas propiedades que en las prácticas anteriores. En ella estableceremos una VPN con dos routers cliente CE, los cuales queremos comunicar.

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

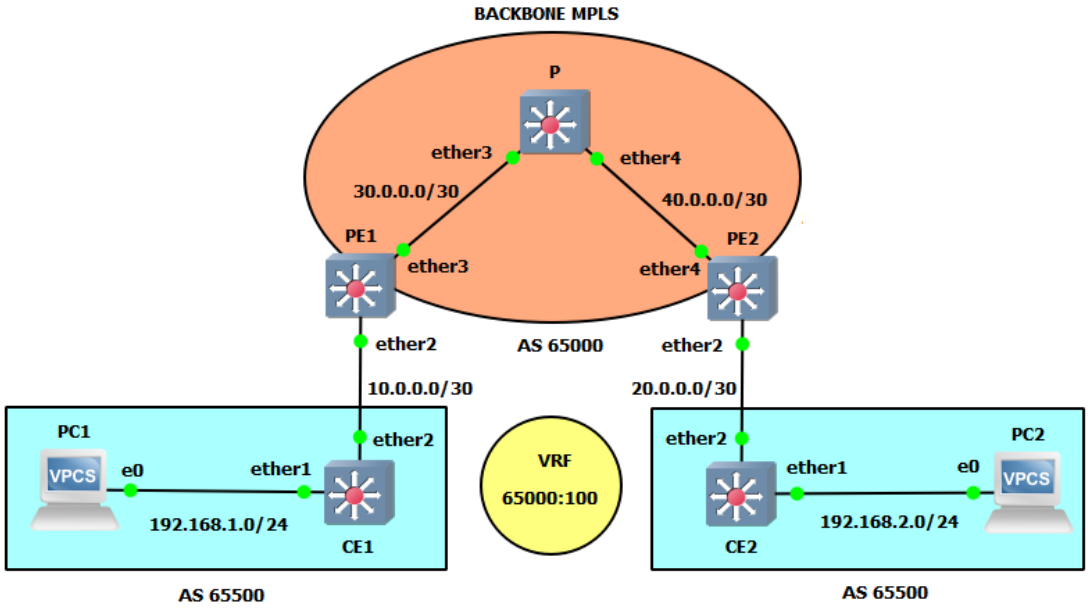


Fig. 143. Diagrama de la red L3 MPLS VPN.

Dispositivo	Interfaz	Dirección IP	Máscara de red	Gateway
CE1	Lo0	192.170.0.1	255.255.255.255	-
	Ether2	10.0.0.1	255.255.255.252	-
	Ether1	192.168.1.1	255.255.255.0	-
PE1	Lo0	192.170.0.2	255.255.255.255	-
	Ether2	10.0.0.2	255.255.255.252	-
	Ether3	30.0.0.2	255.255.255.252	-
P	Lo0	192.170.0.3	255.255.255.255	-
	Ether3	30.0.0.1	255.255.255.252	-
	Ether4	40.0.0.1	255.255.255.252	-
PE2	Lo0	192.170.0.4	255.255.255.255	-
	Ether4	40.0.0.2	255.255.255.252	-
	Ether2	20.0.0.2	255.255.255.252	-
CE2	Lo0	192.170.0.5	255.255.255.255	-
	Ether2	20.0.0.1	255.255.255.252	-
	Ether1	192.168.2.1	255.255.255.0	-
PC1	NIC	192.168.1.100	255.255.255.0	192.168.1.1
PC2	NIC	192.168.2.100	255.255.255.0	192.168.2.1

Tabla 21. Tabla de direccionamiento IP.

4.2.8. Configuración de la red

4.2.8.1. Crear proyecto y montar topología

De igual forma que en la práctica anterior, crearemos un proyecto con un nombre que nos resulte fácil de asociar a dicha práctica.

Luego montaremos la topología e iniciaremos los routers de la misma forma que se explicó en la primera práctica. Así ya tendremos nuestra red lista para la configuración.

4.2.8.2. Crear Interfaz Loopback y asignar IP's

Al igual que en la anterior práctica, primero asignaremos cada una de las direcciones de los interfaces y loopbacks de cada router. Ejecutaremos los siguientes comandos para el router PE1:

```
[admin@PE1] > int br
[admin@PE1] /interface/bridge > add name=lo0
[admin@PE1] /interface/bridge > /ip address
[admin@PE1] /ip/address > add
address: 192.170.0.2/32
interface: lo0
[admin@PE1] /ip/address > add
```

```
address: 10.0.0.2/30
interface: ether2
[admin@PE1] /ip/address > add
address: 30.0.0.2/30
interface: ether3
```

Para el resto de routers se replicará la configuración análogamente.

4.2.8.3. OSPF Backbone MPLS

Empezaremos aplicando el protocolo OSPF para aprender de forma dinámica las direcciones de los routers que pertenecen a MPLS, es decir, PE1, PE2 y P. Por tanto, las interfaces ether2 de los PE no se incluirán.

Router P:

```
[admin@P] > /routing ospf instance
[admin@P] /routing/ospf/instance > add name=backbone router-id=192.170.0.3
[admin@P] /routing/ospf/instance > .. area
[admin@P] /routing/ospf/area > add name=backbone area-id=0.0.0.0 inst=backbone
[admin@P] /routing/ospf/area > .. interface-template
[admin@P] /routing/ospf/interface-template > add int=lo0 net=192.170.0.3/32 area=backbone
[admin@P] /routing/ospf/interface-template > add int=ether3 net=30.0.0.1/30 ar=backbone
[admin@P] /routing/ospf/interface-template > add int=ether4 net=40.0.0.1/30 ar=backbone
```

Repetiremos análogamente para los routers PE.

4.2.8.4. LDP Backbone MPLS

A continuación, aplicaremos el protocolo de distribución de etiquetas (LDP) sobre estos mismos routers, los que pertenecen a la red MPLS. De nuevo no hay que configurar las interfaces que se comunican con los routers de fuera de la backbone.

Router PE1:

```
[admin@PE1] > /mpls ldp
[admin@PE1] /mpls/ldp > add afi=ip lsr-id=192.170.0.2 t=192.170.0.2
[admin@PE1] /mpls/ldp > interface
[admin@PE1] /mpls/ldp/interface > add
interface: ether3
```

Repetimos análogamente para PE2 y P.

Al igual que en la práctica anterior, podemos cambiar el rango de etiquetas dinámicas para una mayor claridad a la hora de entender el funcionamiento y estructura de los paquetes que circulan por la red.

ROUTER	RANGO ETIQUETAS
PE1	10000-19999
PE2	20000-29999
P	30000-39999

Tabla 22. Rango de etiquetas por routers.

Recuerda que el comando utilizado será el siguiente:

/mpls/settings set dy=<rango>

Se puede comprobar que todo se ha realizado correctamente realizando *traceroute* desde PE1 a PE2. Se tendrá que ver la asignación de una etiqueta, ya que en el segundo salto actúa el PHP. También se pueden consultar las tablas *local-mapping, remote-mapping, forwarding-table* y *neighbor*.

Si vemos que las etiquetas no corresponden con el rango asignado a cada router, reiniciaremos los routers con el comando ***system reboot.***

4.2.8.5. MP-BGP

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

Primero actualizaremos la plantilla *default* de BGP en el router, acorde al AS a utilizar y a las familias de direcciones que enrutaran nuestros routers PE. Un AS o Sistema Autónomo consiste en un conjunto de redes con su propia política de enrutamiento. En este caso usaremos el AS 65000, el cual es uno de los más utilizados y extendidos.

```
[admin@PE1] > rou bgp template
[admin@PE1] /routing/bgp/template > set default address-families=ip,vpnv4 as=65000
```

Con la plantilla configurada, pasaremos a configurar la conexión con el otro router PE. Como ya tenemos algunos valores preconfigurados de la plantilla, únicamente habrá que configurar la dirección local, la dirección remota, el AS remoto, el role local de bgp (en este caso ibgp al ser una conexión en el mismo AS) y habilitar tanto la escucha como la conexión.

```
[admin@PE1] /routing/bgp/template > .. connection
[admin@PE1] /routing/bgp/connection > add name=toPE2 temp=default local.add=\
192.170.0.2 .role=ibgp remote.add=192.170.0.4 .as=65000 conn=yes list=yes
```

Con la conexión realizada entre PE1 y PE2, haremos análogamente lo mismo entre PE2 y PE1. Una vez hecho podemos comprobar que se ha establecido la sesión ejecutando el siguiente comando en ambos routers PE:

```
[admin@PE1] > rou bgp session
[admin@PE1] /routing/bgp/session > print
```

Si todo ha ido bien, deberíamos ver una sesión en cada router PE como se muestra en las imágenes siguientes:

```
[admin@PE1] /routing/bgp/session> print
Flags: E - established
 0 E name="toPE2-1"
     remote.address=192.170.0.4 .as=65000 .id=192.170.0.4
     .capabilities=mp,rr,gr,as4 .afi=ip,vpnv4 .messages=10 .bytes=190 .eor=""
     local.role=ibgp .address=192.170.0.2 .as=65000 .id=192.170.0.2
     .capabilities=mp,rr,gr,as4 .afi=ip,vpnv4 .messages=10 .bytes=190 .eor=""
     output.procid=20
     input.procid=20 ibgp
     multihop=yes hold-time=3m keepalive-time=1m uptime=9m14s600ms
     last-started=2023-11-20 18:31:26 prefix-count=0
[admin@PE1] /routing/bgp/session>
```

Fig. 144. MP-BGP. Session PE1-PE2.

```
[admin@PE2] /routing/bgp/session> print
Flags: E - established
 0 E name="toPE1-1"
     remote.address=192.170.0.2 .as=65000 .id=192.170.0.2
     .capabilities=mp,rr,gr,as4 .afi=ip,vpnv4 .messages=2 .bytes=38 .eor=""
     local.role=ibgp .address=192.170.0.4 .as=65000 .id=192.170.0.4
     .capabilities=mp,rr,gr,as4 .afi=ip,vpnv4 .messages=2 .bytes=38 .eor=""
     output.procid=20
     input.procid=20 ibgp
     multihop=yes hold-time=3m keepalive-time=1m uptime=1m27s940ms
     last-started=2023-11-20 18:31:26 prefix-count=0
[admin@PE2] /routing/bgp/session>
```

Fig. 145. MP-BGP. Session PE2-PE1.

4.2.8.6. VRF y VPN

El cliente se situará en una tabla VRF configurada con un RD y RT de importación y exportación. Por ello, primero tendremos que crear esa tabla VRF en nuestros routers PE, que servirán para consultar el direccionamiento de cada cliente conectado a los routers CE. En nuestro caso como solo tenemos un cliente, solo crearemos una tabla VRF por router PE de la siguiente manera:

```
[admin@PE1] > ip vrf
[admin@PE1] /ip/vrf > add interfaces=ether2 name=CE1
```

Le daremos un nombre e indicaremos el interfaz que el router PE utilizará para la comunicación con el cliente a través del router CE.

Este mismo proceso se llevará a cabo análogamente en el router PE2.

A continuación, se configurará la VPN a través de BGP en los routers PE.

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

También definiremos los *Route Targets*, que indican qué rutas se distribuirán al peer PE según la VPN que identifique. En este caso se ha elegido el mismo valor para *RD* y *RT* por simpleza, pero no es necesario.

Por último, se indicará la política de asignación de etiquetas, la tabla VRF que se utilizará y el tipo de rutas que se distribuirán de VRF a VPNv4. A parte de ***static*** y ***connected*** se habilitará ***bgp*** ya que es el protocolo que utilizaremos entre los PE y los CE en su versión *external BGP*

```
[admin@PE1] > rou bgp vpn
[admin@PE1] /routing/bgp/vpn > add route-distinguisher=65000:100 import.route-targets=65000:100\
vrf=CE1 label-allocation-policy=per-vrf export.route-targets=65000:100 .redistribute=connected,static,bgp
```

Repetir análogamente en PE2.

4.2.8.7. Routing CE-PE

Para establecer la comunicación entre los routers PE y CE configuraremos el protocolo EBGP, configurando un número de AS diferente al del backbone MPLS, concretamente el 65500.

4.2.8.7.1. Configuración en PE

En el caso de los routers PE, crearemos una nueva conexión, pero en este caso con el router CE. Para ello, esta vez no utilizaremos ninguna plantilla y configuraremos todos los campos desde la propia conexión.

Necesitaremos indicar la dirección local, el AS local, el role local, el AS remoto, la dirección remota y habilitar la conexión y la escucha al igual que hacíamos en la conexión BGP dentro de la backbone. Esta vez no es necesario activar la familia *VPNv4*, ya que no habrá tráfico de este tipo.

Pero adicionalmente para esta conexión, tenemos que indicar el ID del router (la interfaz loopback), el *vrf* que se utilizará y la tabla de routing asociada al *vrf*.

Por último, debemos permitir que cualquier ruta se pueda anunciar a través de la red MPLS, esto se configura con ***output.default-originate=always.*** Así se creará la ruta por defecto en el router CE aprendida a través de *eBGP* y que hará posible la comunicación entre direcciones privadas a través de la red MPLS.

```
[admin@PE1] > rou bgp con
[admin@PE1] /routing/bgp/connection > add name=toCE1 router-id=192.170.0.2 as=65000\
local.address=10.0.0.2 .role=ebgp remote.address=10.0.0.1 .as=65500 routing-table=CE1 vrf=CE1\
connect=yes listen=yes  output.default-originate=always
```

4.2.8.7.2. Configuración en CE

En los routers CE, aparte de realizar la conexión BGP, primero habrá que crear una lista de direcciones en *ip/firewall*. Esta lista de direcciones la asignaremos a la conexión BGP para permitir que los paquetes recibidos en el CE puedan ser enviados a la red privada del cliente en cuestión.

```
[admin@CE1] > ip firewall address-list
[admin@CE1] /ip/firewall/address-list > add address=192.168.1.0/24 list=BGP_OUT
```

Con la lista creada, procedemos a configurar la conexión BGP indicando ID del router, AS, dirección local, role local, dirección remota, AS remoto y asignando la lista de direcciones como red de salida como bien hemos mencionado. Tampoco hay que olvidar activar la conexión y la escucha.

```
[admin@CE1] /ip/firewall/address-list > /routing/bgp/connection
[admin@CE1] /routing/bgp/connection > add na=toPE1 router-id=192.170.0.1 as=65500 local.add=\
10.0.0.1 .role=ebgp remote.addr=10.0.0.2 .as=65000 output.network=BGP_OUT conn=yes list=yes
```

No hay que olvidar llevar a cabo la configuración análoga en PE2 y en CE2.

Con las conexiones establecidas se podrá comprobar con ***/routing/bgp/session print*** si las sesiones están activas y por tanto el proceso se ha realizado correctamente.

4.2.8.8. Verificación final

Para verificar la correcta conectividad de los routers del cliente realizaremos un ping entre el PC1 y el PC2.

Obviamente, es necesario asignar la IP con su respectiva máscara y su Gateway a cada PC como veíamos en la Práctica 1.

4.2.9. Actividades propuestas

Al igual que en las prácticas anteriores, para realizar las actividades nos valdremos de software WireShark. En este caso nos interesa capturar el enlace PE1-P.

De nuevo, aplicaremos el filtro icmp para ver los paquetes enviados en un ping.

1. **Realice un ping desde el PC1 al PC2 ¿Qué diferencias observa en los paquetes respecto a la práctica 1? Compare también las diferencias entre los paquetes de *request* y de *reply*.**

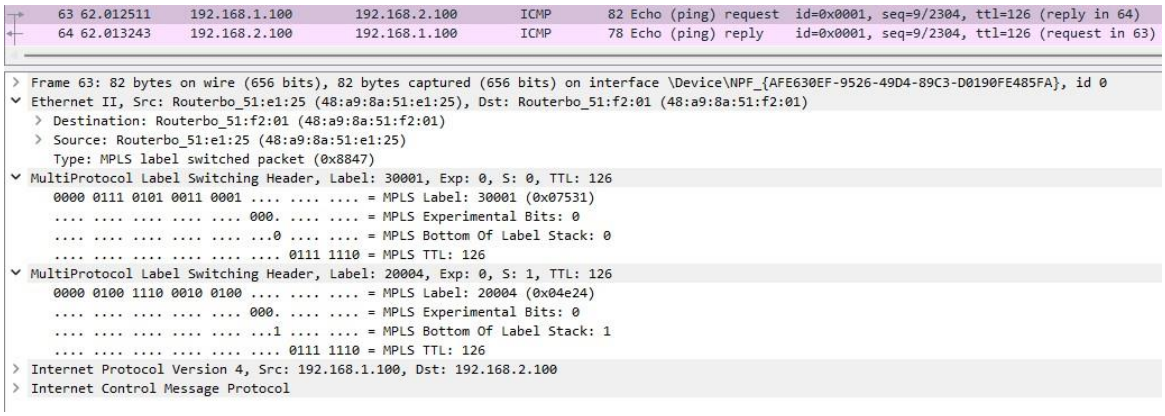


Fig. 146. Paquete request enlace PE1-P.

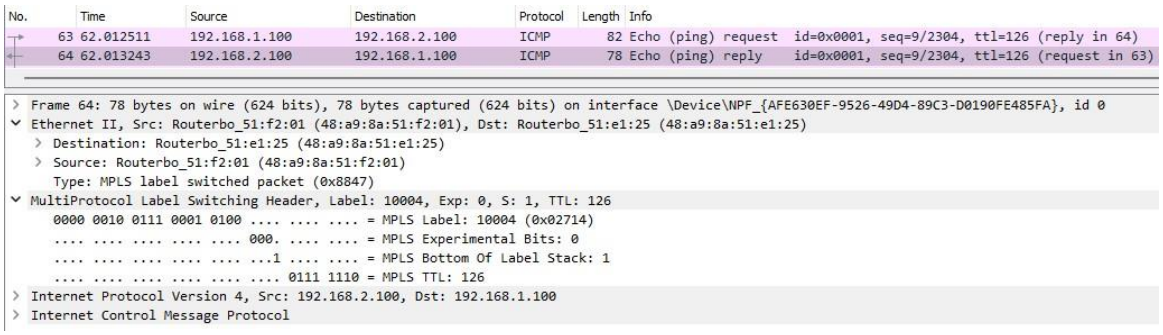


Fig. 147. Paquete reply enlace PE1-P.

Observamos que el paquete request contiene los mismos campos que un paquete MPLS, pero en este caso aparecen dos etiquetas anidadas: 30001 y 20004.

La etiqueta 20004, lleva el bit S o stack a 1, lo que indica que se trata de la última etiqueta, utilizada para la ruta encaminada mediante OSPF en el backbone MPLS.

La etiqueta 30001 pertenece a la VPN y la identifica para que el PE sepa a dónde reenviar el paquete.

En el paquete reply observamos que únicamente contiene una etiqueta, en este caso la 10004. Esto se debe a que el router P ya ha encaminado el paquete a la VPN en cuestión y ha eliminado la etiqueta, quedando solo la que el router PE1 anunció como la etiqueta para recibir los paquetes.

2. Ahora queremos comprobar el establecimiento de la sesión BGP. Reinicia el router PE que está conectado al WireShark con /system reboot. ¿Qué tipo de paquetes BGP aparecen?

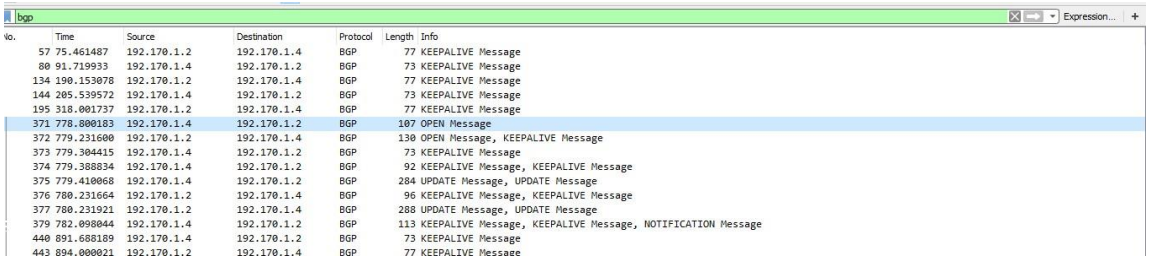


Fig. 148. Captura paquetes BGP.

Se identifican tres tipos de paquetes: KeepAlive Message, Open Message y Update Message

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

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

- Número de versión de BGP usada. Es importante que las versiones de ambos peer coincidan.
- Identificador BGP el cual corresponde con la IP del router con el que establece la sesión, en este caso el PE2 (192.170.0.4)
- Número del AS local, en este caso 65000.

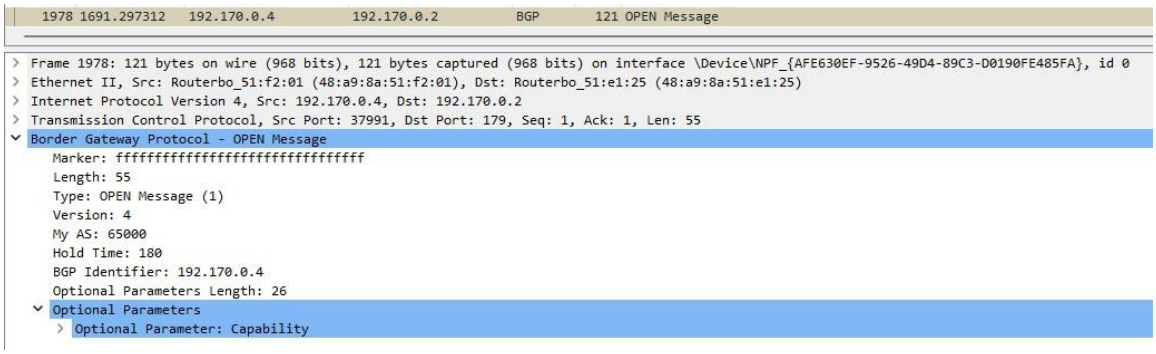


Fig. 149. Captura OPEN Message.

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

- ORIGIN: Indica mediante qué proceso ha sido aprendida la ruta. En este caso es IGP al haber sido aprendida por el protocolo interno OSPF.
- AS_PATH: Indica el número del Sistema Autónomo (AS) en el que se encuentra el router con el que se está estableciendo la sesión BGP para enrutar paquetes entre CE y PE.
- LOCAL_PREF: Atributo para influir en la selección de ruta preferida dentro de un AS cuando hay múltiples rutas disponibles para un destino específico, en nuestro caso solo hay una ruta.
- EXTENDED_COMMUNITIES: Los destinos son agrupados en comunidades. Contiene la información de los RTs y AS. En este caso la hemos definido como extendida y se indica el RD.
- MP_REACH_NLRI: Información de la familia ipv4 y del RD, también se puede observar la Label Stack.

```
Path attributes
  Path Attribute - ORIGIN: IGP
    Flags: 0x40, Transitive, Well-known, Complete
    Type Code: ORIGIN (1)
    Length: 1
    Origin: IGP (0)
  Path Attribute - AS_PATH: 65500
    Flags: 0x50, Transitive, Extended-Length, Well-known, Complete
    Type Code: AS_PATH (2)
    Length: 6
    AS Path segment: 65500
  Path Attribute - LOCAL_PREF: 100
    Flags: 0x40, Transitive, Well-known, Complete
    Type Code: LOCAL_PREF (5)
    Length: 4
    Local preference: 100
  Path Attribute - ATOMIC_AGGREGATE
  Path Attribute - EXTENDED_COMMUNITIES
    Flags: 0xd0, Optional, Transitive, Extended-Length, Complete
    Type Code: EXTENDED_COMMUNITIES (16)
    Length: 8
    Carried extended communities: (1 community)
      Route Target: 65000:100 [Transitive 2-Octet AS-Specific]
  Path Attribute - MP_REACH_NLRI
    Flags: 0x90, Optional, Extended-Length, Non-transitive, Complete
    Type Code: MP_REACH_NLRI (14)
    Length: 32
    Address family identifier (AFI): IPv4 (1)
    Subsequent address family identifier (SAFI): Labeled VPN Unicast (128)
    Next hop:  RD=0:0 IPv4=192.170.0.2
    Number of Subnetwork points of attachment (SNPA): 0
    Network Layer Reachability Information (NLRI)
      BGP Prefix
          Prefix Length: 112
          Label Stack: 10000 (bottom)
          Route Distinguisher: 65000:100
          MP Reach NLRI IPv4 prefix: 192.168.1.0
```

Fig. 150. Captura UPDATE Message.

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

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

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

Posteriormente, añadir mediante el protocolo de routing BGP la nueva red que conecta el router CE al router PE con un identificador AS distinto.

Por último, modificar la familia ipv4 y redistribuir las nuevas rutas creadas.

4. La red actual puede ser perfectamente una implementación para una empresa cos dos sedes separadas geográficamente. Pero ahora la empresa decide comprar una nueva filial y la quiere ubicar en el mismo edificio de otra filial, creando una red local aparte pero que a la vez ambas estén accesibles desde la sede central. ¿Es esto posible utilizando únicamente un router más en la red que ya tenemos montada? Justifica experimentalmente.

Esto sí es posible dado que podemos utilizar el router adicional como router CE conectado al PE2. Es este router CE, que llamaremos CE3, se puede montar la red local de la nueva filiar y utilizar una nueva tabla vrf independiente en PE2. Desde PE2 se conectará a la sede central en CE1 con una nueva VPN a través del protocolo BGP.

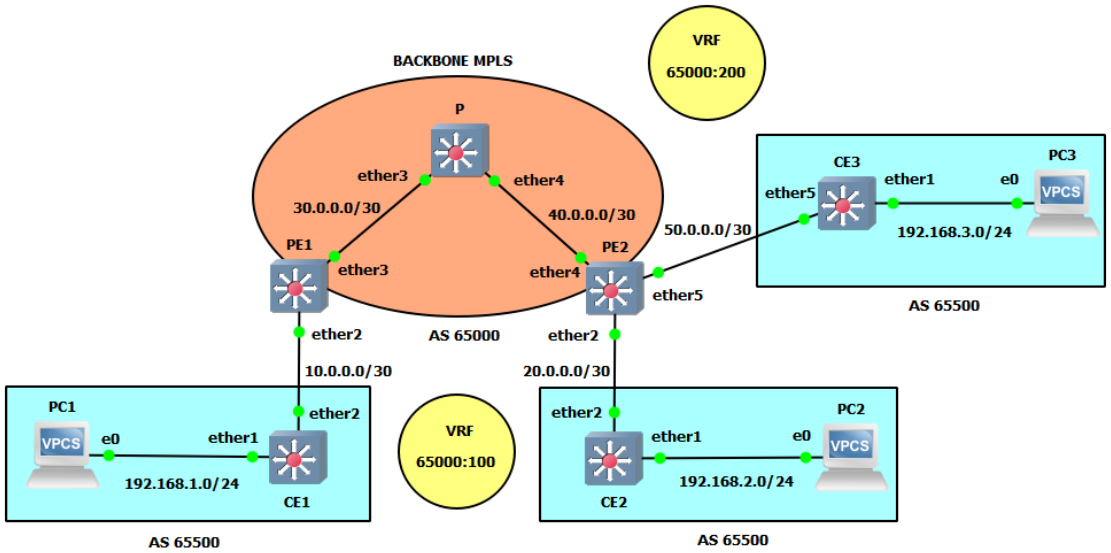


Fig. 151. Topología de red con 3 routers CE.

Realmente como sabemos la red es escalable y solo se necesita de ciertas configuraciones en CE3, PE2 y PE1.

En PE2, primero asignaremos la nueva dirección IP a la interfaz que se conecta al CE3, crearemos la nueva vrf, estableceremos la conexión eBGP con CE3 y la VPN con CE1:

```
[admin@PE2] > /ip add
[admin@PE2] > /ip/address > add
address: 50.0.0.2/30
interface: ether5
[admin@PE2] /ip/address > .. vrf
[admin@PE2] /ip/vrf > add name=CE3 int=ether5
[admin@PE2] /ip/vrf > /rou bgp con
[admin@PE2] /routing/bgp/connection > add name=toCE3 router-id=192.170.0.4 as=65000\
local.address=50.0.0.2 .role=ebgp remote.address=50.0.0.1 .as=65500 routing-table=CE3\
vrf=CE3 connect=yes listen=yes output.default-originate=always
[admin@PE2] /routing/bgp/connection > .. vpn
[admin@PE2] /routing/bgp/vpn > add route-disting=65000:200 import.route-targ=65000:200\
vrf=CE3 label-allocation-poli=per-vrf export.route-t=65000:200 .redist=connected,static,bgp
```

Una vez hecho esto, en CE3 asignaremos las direcciones IP y estableceremos la conexión BGP con PE2, no sin antes crear la lista de direcciones de firewall:

```
[admin@CE3] > /int br
[admin@CE3] /interface/bridge > add name=lo0
[admin@CE3] /interface/bridge > /ip addr
[admin@CE3] /ip/address > add
address: 192.170.0.6/32
interface: lo0
[admin@CE3] /ip/address > add
address: 192.168.3.1/24
interface: ether1
[admin@CE3] /ip/address > add
address: 50.0.0.1/30
interface: ether5
[admin@CE3] /ip/address > .. fire addr
[admin@CE3] /ip/firewall/address-list > add addr=192.168.3.0/30 list=BGP_OUT
[admin@CE3] /ip/firewall/address-list > /rou bgp conn
[admin@CE3] /routing/bgp/connection > add name=toPE2 router-id=192.170.0.6 as=65500\
local.add=50.0.0.1 .role=ebgp remote.addr=50.0.0.2 .as=65000 output.network=BGP_OUT\
conn=yes list=yes
```

Por último, en PE1 crearemos la VPN que conectará la sede central con la nueva filiar. Haremos uso del mismo RD, Import RT y Export RT que en PE1 para CE3 y que será deferente al de la conexión entre CE1 y CE3:

```
[admin@PE1] > /rou bgp vpn
[admin@PE1] /routing/bgp/vpn > add route-disting=65000:200 import.route-t=65000:200\
vrf=CE1 label-allocation-po=per-vrf export.route-t=65000:200 .redist=connected,static,bgp
```

4.3. Práctica 3 "Configuración de una Red L2 MPLS VPLS"

4.3.1. Introducción

VPLS (Servicio de LAN Privada Virtual) es una tecnología capaz de proporcionar Ethernet multipunto a multipunto basado en la comunicación sobre redes IP/MPLS. Permite a sitios dispersos geográficamente compartir un dominio de difusión Ethernet mediante la conexión de estos a través de pseudowires (PW).

VPLS es una Red Privada Virtual (VPN), que en contraste con L3 MPLS VPN, que solo permite túneles punto a punto en capa 2, VPLS permite conectividad multipunto. En una VPLS la LAN de cada sitio se extiende hasta el borde de la red del proveedor (MPLS).

VPLS al emular una LAN, necesita de una conectividad completa de malla. Para ello, se cuenta con dos métodos diferentes para el establecimiento de la malla VPLS completa:

- LDP (Protocolo de Distribución de Etiquetas): cada PE debe configurarse para participar en un VPLS determinado.
- BGP (Protocolo de puerta de Enlace Fronterizo): proporciona autodescubrimiento y señalización.

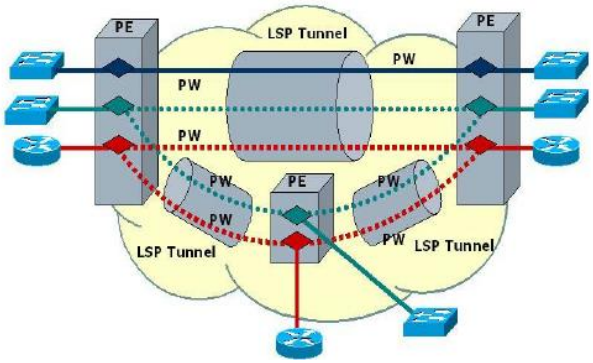


Fig. 152. Topología con 3 redes VPLS.

Permite una conectividad transparente a través de una red de proveedor de servicios (backbone MPLS), pudiéndose enviar paquetes IP entre ellos. La red proveedora puede ofrecer conectividad a redes de clientes distintas, apareciendo cada una de ellas como un PW distinto.

4.3.2. Componentes y arquitectura L2VPLS

4.3.2.1. Customer Edge (CE)

Los routers CE son los equipos de borde del cliente que están conectados a la red L2VPLS. Estos dispositivos se utilizan para enviar y recibir tráfico de capa 2 a través de la L2VPLS.

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

4.3.2.2. Provider Edge (PE)

Router frontera de la red del proveedor de servicios conectado al router CE. Encapsulan y envían los paquetes de capa 2 a través de la red MPLS hacia los demás routers PE.

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

4.3.3. Protocolos de Señalización

Los protocolos de señalización se utilizan para establecer y controlar las conexiones L2VPLS entre los dispositivos PE. Estos protocolos pueden incluir BGP (Border Gateway Protocol) para la señalización y el intercambio de información de enrutamiento, y LDP (Label Distribution Protocol) para el establecimiento y mantenimiento de las etiquetas MPLS utilizadas en la red MPLS.

4.3.3.1. LDP

LDP permite la señalización de la red VPLS, pero requiere del establecimiento de interfaces VPLS en ambos extremos del túnel, es decir, en ambos routers PE que formarán parte de la malla VPLS. En una red de 4 routers PE, se necesitarán 12 interfaces VPLS.

4.3.3.2. BGP

BGP permite señalización y autodescubriento, para ello, se hace uso de conexiones MP-BGP entre routers PE. Sobre BGP se establecen las VPN's desde cada PE, logrando autodescubrirse al resto de PE's de la malla.

4.3.4. Componentes funcionales de VPLS

4.3.4.1. Pseudowire (PW)

Es un circuito virtual VC o circuito Ethernet emulado. Desde la óptica de MPLS un pseudowire es un camino LSP con una etiqueta VC. Es bidireccional e interconecta dos LSRs (PEs), cada dirección va sobre un LSP. Entre dos PEs pueden existir múltiples pseudowires, y cada LSP puede llevar uno o más pseudowires que pertenezcan a uno o varios clientes.

Posibilitan el transporte de cualquier tipo de tráfico como Ethernet, FR, ATM, TDM de baja velocidad T1, T3, E1, E3 y SONET/SDH sobre una red MPLS.

4.3.4.2. VSI (Instancia de Switch Virtual)

VSI equivale a un Bridge transparente, que interconecta una malla de PWs para reenviar tramas Ethernet entre routers PE's.

VSI es similar a VRF en L3VPN, que entregaba tráfico conmutado a IP/MPLS. VSI entrega tráfico L2 VPN a IP/MPLS usando VPLS.

4.3.5. Envío de Tramas

VPLS permite que el PE actúe como un bridge con una tabla de MACs por VSI. El VSI por tanto tiene una tabla que se construye mediante las MAC fuente que circulan por el AC (attachment circuit) o PW, funcionado como un switch convencional.

Cuando una trama Ethernet accede por medio de un AC de entrada al PE, la MAC destino es buscada en la tabla y la trama enviada por el PW adecuado para alcanzar el PE remoto correspondiente.

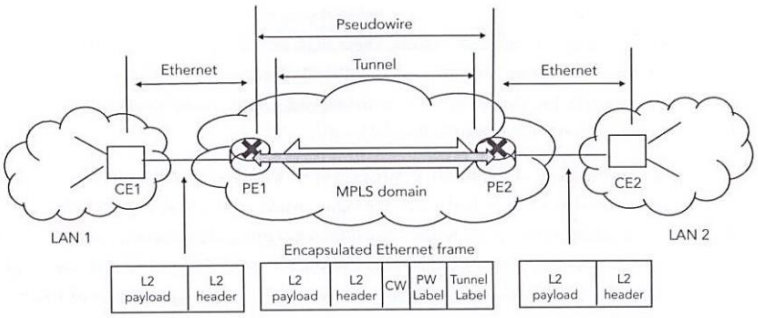


Fig. 153. Envió de tramas entre dos LAN's.

Destinos Broadcast, Multicast o desconocidos son inundados excepto por el AC de llegada, por todos los Pseudowires disponibles. El VSI actualiza su tabla con las tramas que le llegan y las entradas que no se utilizan son eliminadas de la misma.

Esta inundación puede producir bucles de tráfico, para evitar esto, los PEs utilizan Bridge Horizon sobre los PW para asegurar la no existencia de bucles.

4.3.5.1. Bridge Horizon

La idea básica de Bridge Horizon es hacer que el tráfico que llega por algún puerto nunca se envíe a algún conjunto de puertos. Esto significaría nunca enviar paquetes que llegaron a través de un túnel VPLS.

Bridge Horizon permite configurar los puertos dentro de un bridge para que el paquete recibido a través del puerto con valor horizonte X no se reenvíe a ningún otro puerto con el mismo valor de horizonte X.

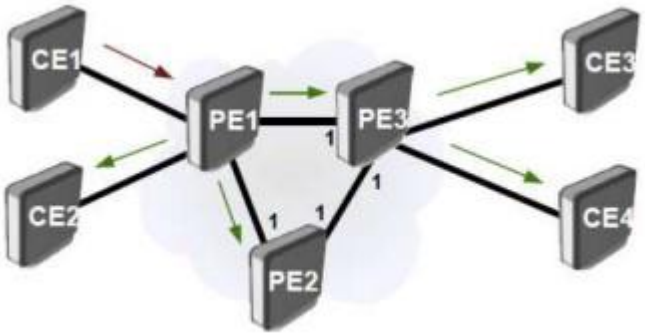


Fig. 154. Topología Bridge Horizon.

4.3.6. Objetivos

El objetivo de la presente práctica es familiarizarse con los conceptos de túneles VPLS sobre MPLS, así como su configuración en una red implementada con routers Mikrotik.

Para ello, se deberán realizar las siguientes actividades:

- introducir en los routers los comandos necesarios para configurar una red L2 VPLS (LDP) y L2 VPLS (BGP).
- verificar el correcto funcionamiento de los túneles VPLS establecidos en la red.
- visualizar los diferentes paquetes que circulan por la red e identificar los campos pertenecientes a los diferentes protocolos utilizados en VPLS.
- Diferenciar y contrastar la escalabilidad de una red VPLS sobre LDP y BGP.

4.3.7. Elementos necesarios

Para la realización de la presente práctica se utilizará el entorno de simulación GNS3 con el router MikroTik CRS328-24G-4S previamente instalado a través de la VM y los PC's virtuales disponibles, VPCS.

4.3.8. Topología de red

Vamos a crear una red L2 MPLS VPLS sobre LDP formada por 6 routers Mikrotik CRS328-24G-4S con las mismas propiedades que en las prácticas anteriores. En ella estableceremos una maya de túneles VPLS con 4 routers LER, los cuales queremos comunicar.

Utilizaremos los protocolos OSPF y LDP para intercambiar información de direccionamiento en la red MPLS.

Como observamos a continuación, hemos prescindido de los routers CE para centrarnos en el establecimiento de las conexiones y túneles entre routers PE (LER´s). Así pues, solo conectamos un PC directamente a cada LER para comprobar el funcionamiento.

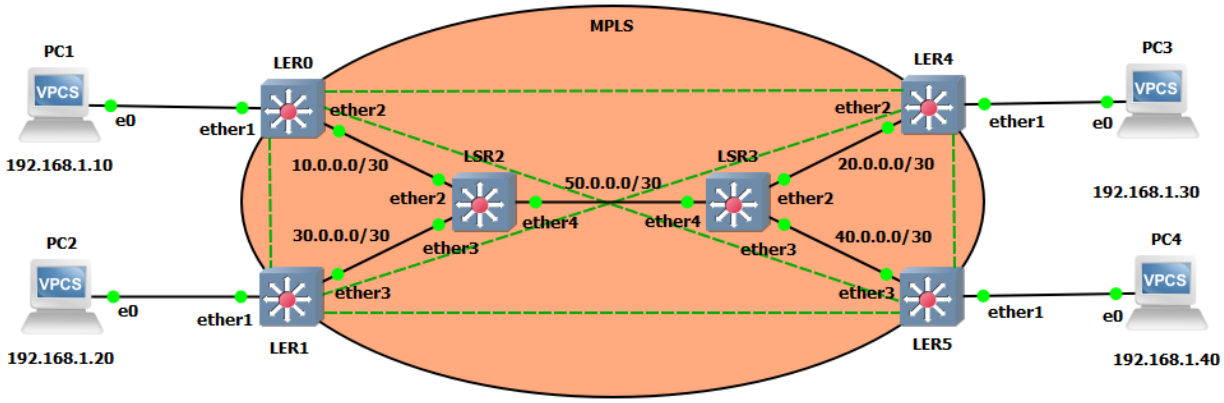


Fig. 155. Topología de red.

Dispositivo	Interfaz	Dirección IP	Máscara de red	Gateway
LER0	lo0	192.170.0.1	255.255.255.255	-
	ether2	10.0.0.1	255.255.255.252	-
	VPLS	192.168.1.1	255.255.255.0	-
LER1	lo0	192.170.0.2	255.255.255.255	-
	ether3	30.0.0.1	255.255.255.252	-
	VPLS	192.168.1.2	255.255.255.0	-
LSR2	lo0	192.170.0.3	255.255.255.255	-
	ether2	10.0.0.2	255.255.255.252	-
	ether3	30.0.0.2	255.255.255.252	
	ether4	50.0.0.2	255.255.255.252	-
LSR3	lo0	192.170.0.4	255.255.255.255	-
	ether2	20.0.0.1	255.255.255.252	-
	ether3	40.0.0.1	255.255.255.252	
	ether4	50.0.0.1	255.255.255.252	-
LER4	lo0	192.170.0.5	255.255.255.255	-
	ether2	20.0.0.2	255.255.255.252	-
	VPLS	192.168.1.3	255.255.255.0	-
LER5	lo0	192.170.0.6	255.255.255.255	-
	ether3	40.0.0.2	255.255.255.252	-
	VPLS	192.168.1.4	255.255.255.0	-
PC1	NIC	192.168.1.10	255.255.255.0	192.168.1.1
PC2	NIC	192.168.1.20	255.255.255.0	192.168.1.2
PC3	NIC	192.168.1.30	255.255.255.0	192.168.1.3
PC4	NIC	192.168.1.40	255.255.255.0	192.168.1.4

Tabla 23. Tabla direccionamiento IP.

4.3.9. Configuración de la red

4.3.9.1. Crear proyecto y montar topología

De igual forma que en las prácticas anteriores, crearemos un proyecto con un nombre que nos resulte fácil de asociar a dicha práctica.

Luego montaremos la topología e iniciaremos los routers de la misma forma que se explicó en la primera práctica. Así ya tendremos nuestra red lista para la configuración.

4.3.9.2. Crear Interfaz Loopback y Bridge VPLS

Vamos a crear la interfaz loopback como venimos haciendo en las prácticas anteriores, pero esta vez también crearemos un bridge adicional, el VSI que hemos visto en la teoría. Este bridge que llamaremos VPLS nos servirá para asociar a una interfaz física los diferentes túneles vpls.

```
[admin@LER0] > int br
[admin@LER0] /interface/bridge > add name=lo0
[admin@ LER0] /interface/bridge > add name=VPLS
```

Para el resto de routers LER se replicará la configuración análogamente. Para el caso de los routers LSR no se creará en Bridge VPLS.

4.3.9.3. Asignar IP's

A continuación, asignaremos las direcciones IP's. En este caso hay que tener en cuenta que al bridge VPLS se le asignará la dirección IP que los PC's utilizarán como puerta de enlace predeterminada, la que anteriormente asignábados al puerto físico.

```
[admin@ LER0] /interface/bridge > /ip address
[admin@ LER0] /ip/address > add
address: 192.170.0.1/32
interface: lo0
[admin@ LER0] /ip/address > add
address: 192.168.1.1/24
interface: VPLS
[admin@ LER0] /ip/address > add
address: 10.0.0.1/30
interface: ether2
```

Repetir coherentemente en el resto de routers.

4.3.9.4. OSPF Backbone MPLS

Ahora aplicaremos el protocolo OSPF para aprender de forma dinámica las direcciones de los routers que pertenecen a MPLS, excluyendo por tanto el interfaz ether1 de los routers LER.

```
[admin@LSR2] > /routing ospf instance
[admin@ LSR2] /routing/ospf/instance > add name=backbone router-id=192.170.0.3
[admin@ LSR2] /routing/ospf/instance > .. area
[admin@ LSR2] /routing/ospf/area > add name=backbone area-id=0.0.0.0 inst=backbone
[admin@ LSR2] /routing/ospf/area > .. i
[admin@ LSR2] /routing/ospf/interface-template > add int=lo0 net=192.170.0.3/32 ar=backbone
[admin@ LSR2] /routing/ospf/interface-template > add int=ether2 net=10.0.0.2/30 ar=backbone
[admin@ LSR2] /routing/ospf/interface-template > add int=ether3 net=30.0.0.2/30 ar=backbone
[admin@ LSR2] /routing/ospf/interface-template > add int=ether4 net=50.0.0.2/30 ar=backbone
```

Repetiremos análogamente para el resto de routers.

4.3.9.5. LDP Backbone MPLS

A continuación, aplicaremos el protocolo de distribución de etiquetas (LDP). De nuevo no hay que configurar las interfaces que se comunican con los routers de fuera de la backbone.

```
[admin@LER1] > /mpls ldp
[admin@ LER1] /mpls/ldp > add afi=ip lsr-id=192.170.0.2 t=192.170.0.2
[admin@ LER1] /mpls/ldp > interface
[admin@ LER1] /mpls/ldp/interface > add
interface: ether3
```

Repetimos análogamente los demás routers.

Al igual que en las prácticas anteriores, podemos cambiar el rango de etiquetas dinámicas para una mayor claridad a la hora de entender el funcionamiento y estructura de los paquetes que circulan por la red.

ROUTER	RANGO ETIQUETAS
LER0	16-9999
LER1	10000-19999
LSR2	20000-29999
LSR3	30000-39999
LER4	40000-49999
LER5	50000-59999

Tabla 24. Rango de etiquetas por routers.

Recuerda que el comando utilizado será el siguiente:

/mpls/settings set dy=<rango>

Se puede comprobar que todo se ha realizado correctamente realizando *traceroute* entre dos routers cualesquiera de la MPLS. La asignación de etiquetas variará en función de los saltos que hay entre los routers que se ha probado.

Si vemos que las etiquetas no corresponden con el rango asignado a cada router, reiniciaremos los routers con el comando ***system reboot.***

4.3.9.6. Configuración de los PC's

No hay que olvidar configurar todos los PC's conectados a los routers LER dentro de la misma subred, como se especifica en la tabla de direcciones.

Por supuesto, a parte de la dirección IP y la máscara, hay que configurar la puerta de enlace predeterminada. En este caso será la dirección IP asignada al bridge del router LER en cuestión.

4.3.9.7. VPLS (LDP)

4.3.9.7.1. Configuración

Con la red MPLS ya configurada, nos centraremos en la configuración propia de esta práctica, los túneles VPLS. Para ello, hay que tener en cuenta que tendremos que crear interfaces VPLS en ambos extremos de cada túnel.

Como queremos comunicar los 4 equipos de la misma red a través de la red MPLS, necesitamos crear un total de 6 túneles VPLS. Como se ha comentado en la teoría, cada túnel tiene un identificativo único que se indica en la tabla junto a cada túnel necesario:

TÚNEL	VPLS-ID
LER0-LER1	1:1
LER0-LER4	2:1
LER0-LER5	3:1
LER1-LER4	4:1
LER1-LER5	5:1
LER4-LER5	6:1

Tabla 25. Identificación por túneles.

Creamos los interfaces con el peer remoto del túnel y el vpls-id:

```
[admin@LER0] > int vpls
[admin@LER0] /interface/vpls > add name=vpls-to-ler1 peer=192.170.0.2 vpls-id=1:1
[admin@LER0] /interface/vpls > add name=vpls-to-ler4 peer=192.170.0.5 vpls-id=2:1
[admin@LER0] /interface/vpls > add name=vpls-to-ler5 peer=192.170.0.6 vpls-id=3:1
```

Con las interfaces configuradas tenemos un extremo de los enlaces virtuales, pero para conseguir el enlace transparente entre los extremos del túnel, tenemos que asociar las interfaces del túnel con la interfaz física a través del bridge creado inicialmente con los siguientes comandos:

```
[admin@LER0] /interface/vpls > .. br port
[admin@PE1] /interface/bridge/port> add bridge=VPLS interface=ether1
[admin@PE1] /interface/bridge/port> add bridge=VPLS interface=vpls-to-ler1
[admin@PE1] /interface/bridge/port> add bridge=VPLS interface=vpls-to-ler4
[admin@PE1] /interface/bridge/port> add bridge=VPLS interface=vpls-to-ler5
```

Repetiremos estos dos pasos análogamente en el resto de routers LER para tener los túneles completamente operativos, y por tanto la malla VPLS.

4.3.9.7.2. Verificación final

Para verificar el correcto funcionamiento, podemos realizar una serie de procesos que nos cerciorarán que todo está perfectamente operativo.

Primeramente, podemos realizar pings entre todos los PC’s de la red y ver que se obtiene las respuestas a los paquetes icmp enviados.

Con esto vemos que hay conexión entre equipos, pero para comprobar que se establece a través de los túneles, podemos monitorizar los interfaces VPLS. Con ello veremos la etiqueta local,

que servirá para que el otro extremo envíe la información; la etiqueta remota, que es la asignada por el otro extremo para la comunicación hacia dicho peer; y los diferentes atributos relacionados con el siguiente salto, etiqueta, interface y.nexthop.

```
[admin@LER0] > /interface/vpls
[admin@LER0] /interface/vpls> monitor vpls-to-ler1
   remote-label: 10009
    local-label: 18
  remote-status:
       nexthops: { label=20006; nh=10.0.0.2%ether2; interface=ether2 }

[admin@LER0] /interface/vpls> monitor vpls-to-ler4
   remote-label: 40002
    local-label: 17
  remote-status:
       nexthops: { label=20003; nh=10.0.0.2%ether2; interface=ether2 }

[admin@LER0] /interface/vpls> monitor vpls-to-ler5
   remote-label: 50002
    local-label: 16
  remote-status:
       nexthops: { label=20004; nh=10.0.0.2%ether2; interface=ether2 }
```

Tabla 26. Interfaces VPLS.

4.3.9.7.3. Actividades propuestas

1. **Realice *traceroute* desde LER0 a PC4 y *tracer* desde PC1 a PC4. Comente que diferencias observa respecto a la Práctica 1.**

   Podemos observar en ambos casos que hay un único salto entre los extremos. En cuanto a esta casuística respecto a la práctica 1, podemos decir que en ella cuando realizábamos el traceroute desde el router LER observábamos cada uno de los saltos por la red MPLS con sus respectivas etiquetas. Aunque realizando el tracer entre PCs no se observaban las etiquetas, si que se podían ver los diferentes saltos por la red.

   Sin embargo, esto no está ocurriendo en la configuración VPLS por lo que se comenta en la teoría y hemos aplicado en la configuración, el enlace transparente entre equipos finales de la red hace que a nivel lógico los routers LER se consideren como vecinos, es decir, directamente conectados. Esto se puede observar en la tabla *neighbor* de ldp como vecinos dinámicos.

```
[admin@LER0] /mpls/ldp/neighbor> print
Flags: D, I - INACTIVE; O, T - THROTTLED; t - SENDING-TARGETED-HELLO; v - VPLS; p - PASSIVE
Columns: TRANSPORT, LOCAL-TRANSPORT, PEER, ADDRESSES
#       TRANSPORT    LOCAL-TRANSPORT  PEER           ADDRESSES
0 DOtvp 192.170.0.6  192.170.0.1      192.170.0.6:0  40.0.0.2
                                                     192.168.1.4
                                                     192.170.0.6
1 DOtvp 192.170.0.5  192.170.0.1      192.170.0.5:0  20.0.0.2
                                                     192.168.1.3
                                                     192.170.0.5
2 DOtvp 192.170.0.2  192.170.0.1      192.170.0.2:0  30.0.0.1
                                                     192.168.1.2
                                                     192.170.0.2
3 DO  p 192.170.0.3  192.170.0.1      192.170.0.3:0  10.0.0.2
                                                     30.0.0.2
                                                     50.0.0.2
                                                     192.170.0.3
```

Fig. 156. Tabla neighbor LDP.

2. Ahora ejecute el software WireShark en los 4 enlaces entre los PC's y los routers LER aplicando el filtro arp. Una vez hecho esto, realice un ping a la dirección IP de red de nuestra subred desde cualquiera de los 4 PCs. En este caso sería la dirección 192.168.1.0. Comente los paquetes que se observan en los diferentes PC's.

Tras realizar el ping desde uno de los PC's, podemos observar como una serie de paquetes ARP llegan a cada uno de los PC's en busca de la dirección IP solicitada, pero como es obvio, no se obtiene ninguna respuesta ya que está dirección está reservada y no se ha asignado a ningún equipo.

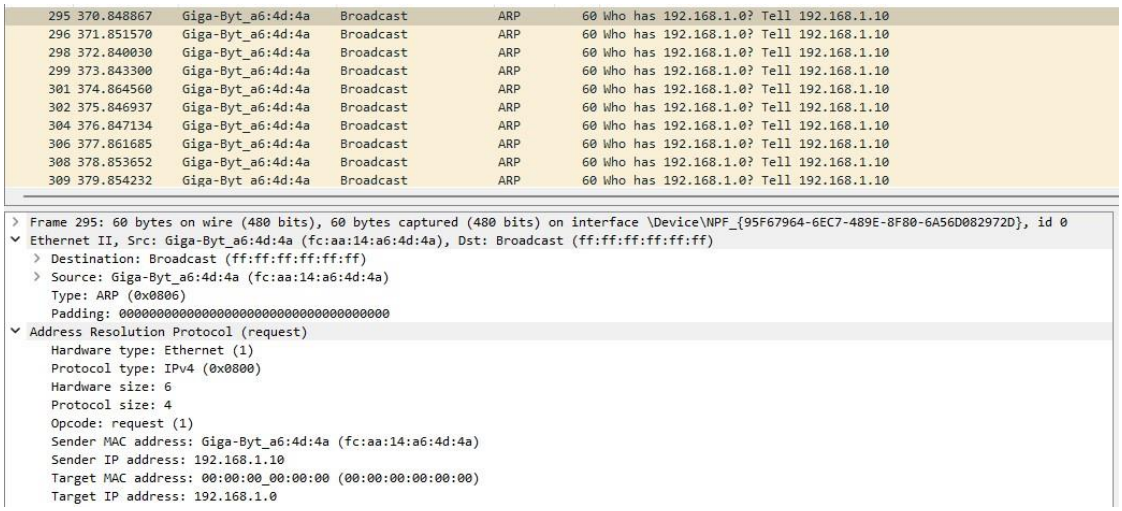


Fig. 157. Capturas paquetes ARP.

En cada una de las solicitudes podemos ver que se está realizando un envío a la dirección MAC de broadcast (FF:FF:FF:FF:FF:FF) desde la dirección MAC del PC en cuestión.

También podemos ver en la cabecera Ethertype el tipo de protocolo en su denominación en hexadecimal***, 0x0806*** que hace referencia a ARP.

Con esto podemos determinar lo que ya sabíamos, que toda nuestra red está comunicada creando una malla VPLS.

Si por el contrario el *ping* se hubiera hecho a una dirección IP asignada a cualesquiera de los PC's, observaríamos los paquetes *request* en el equipo destino del *ping* y los paquetes *reply* en el equipo que ejecuta dicho *ping*.

Ahora capturaremos el tráfico de los enlaces LSR2-LSR3, LER1-LSR2 y LSR3-LER5 con WireShark de la misma forma que venimos haciendo hasta ahora en el entorno de simulación GNS3.

3. Realice un ping desde el PC2 al PC4 para cada enlace, guardando la captura de WireShark para comparar los 3 tramos del túnel. ¿Qué diferencias observa en los paquetes respecto a la práctica 3? Compare también las diferencias entre los paquetes de *request* y de *reply*.

LSR2-LSR3:

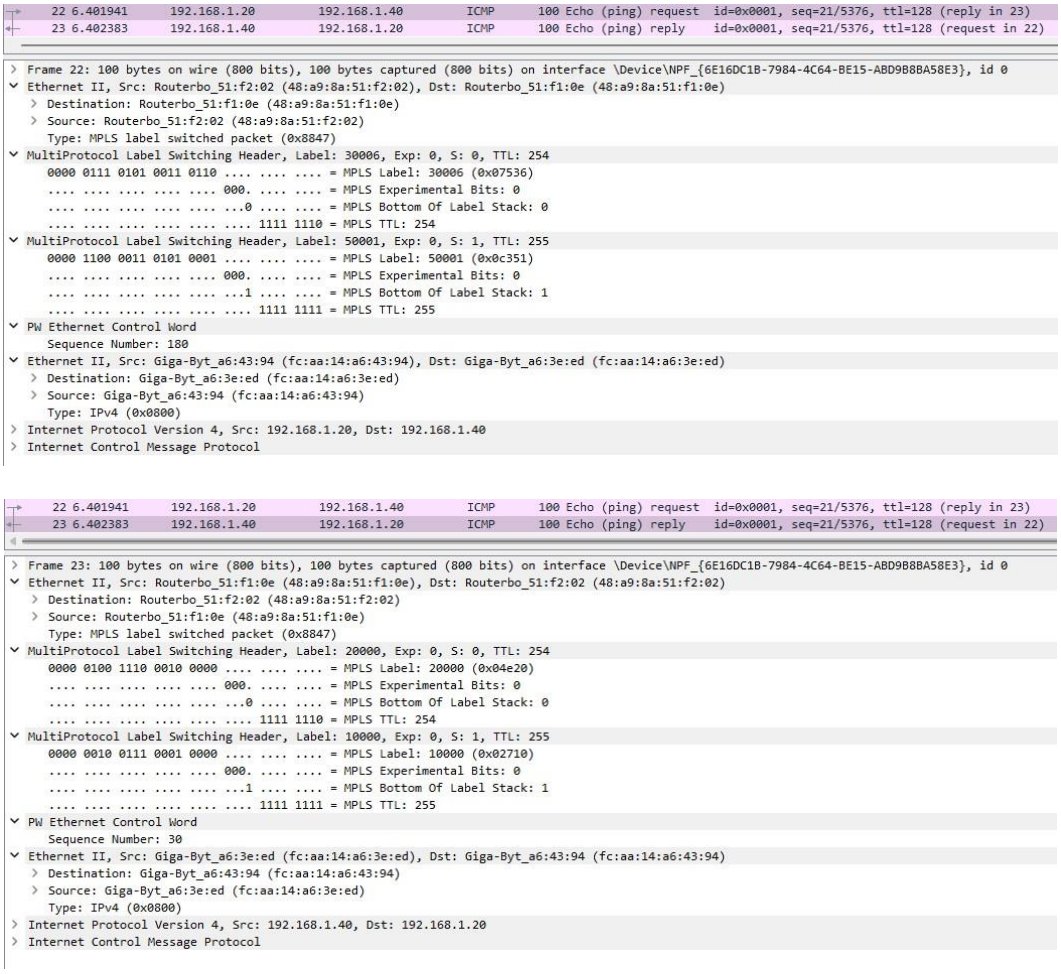


Fig. 158. Capturas ICMP enlace LSR2-LSR3.

Observamos que el paquete request contiene dos etiquetas anidadas al igual que ocurría en la práctica 3: 30006 y 50001.

La etiqueta 50001, lleva el bit S o stack a 1, lo que indica que se trata de la última etiqueta, utilizada en este caso para indicar la ruta hacia el router final del túnel, por tanto es la etiqueta del pseudowire.

La etiqueta 30006 indica el siguiente salto dentro de la red MPLS utilizando OSPF.

En el paquete reply observamos lo mismo, pero de manera análoga con las etiquetas 20000 y 10000.

También se observa respecto a la práctica 3, que hay un nuevo campo en la cabecera del paquete, PW Ethernet Control Word. Este campo nos indica que el direccionamiento IP es trasportado a través de un pseudowire (PW), ese paquete IP lo observamos seguidamente.

Por lo tanto, a parte de la pila de etiquetas, en los paquetes ICMP se observan dos paquetes Ethernet, uno para indicar el transporte sobre MPLS y otro para indicar el trasporte IP, que viene encapsulado sobre el pseudowire mencionado.

Ahora vamos a comparar los diferentes enlaces del túnel.

LER1-LSR2:

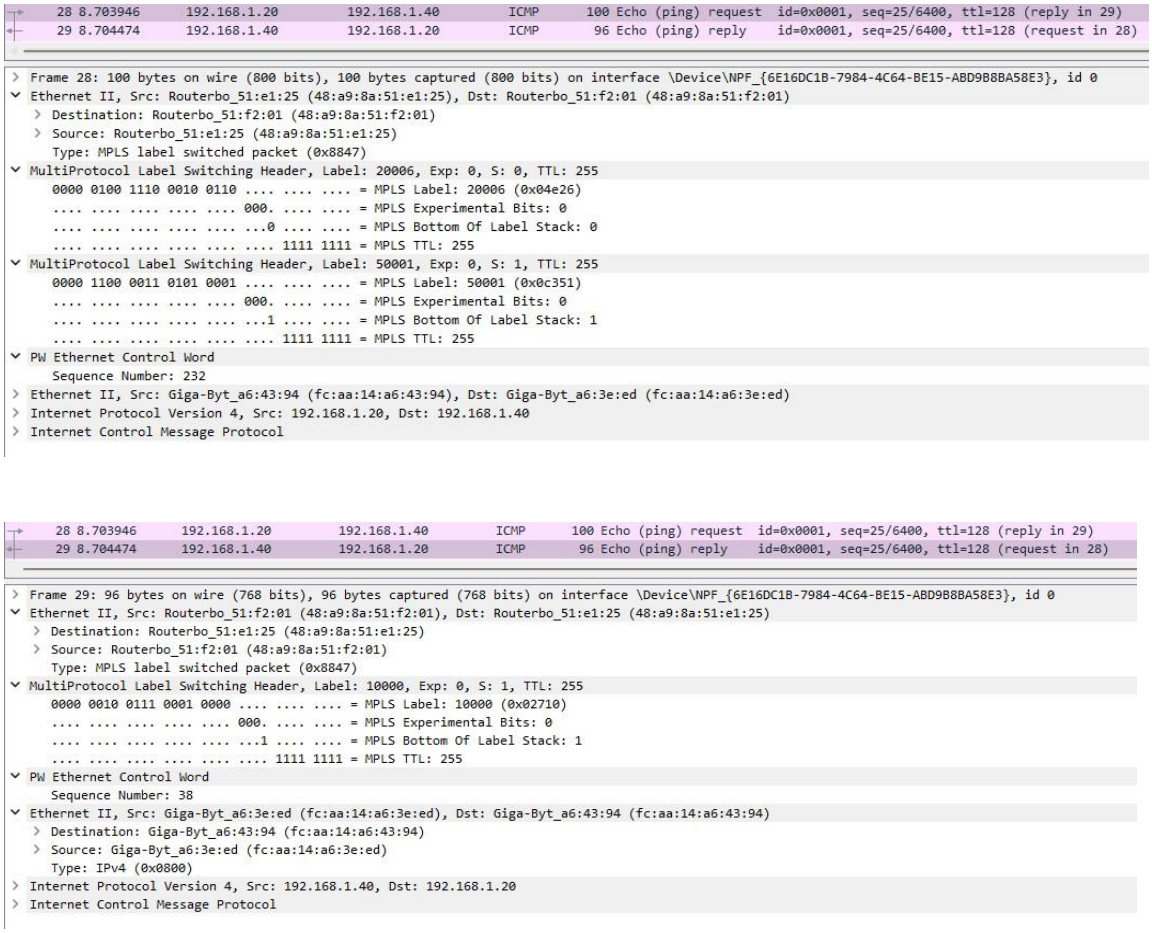
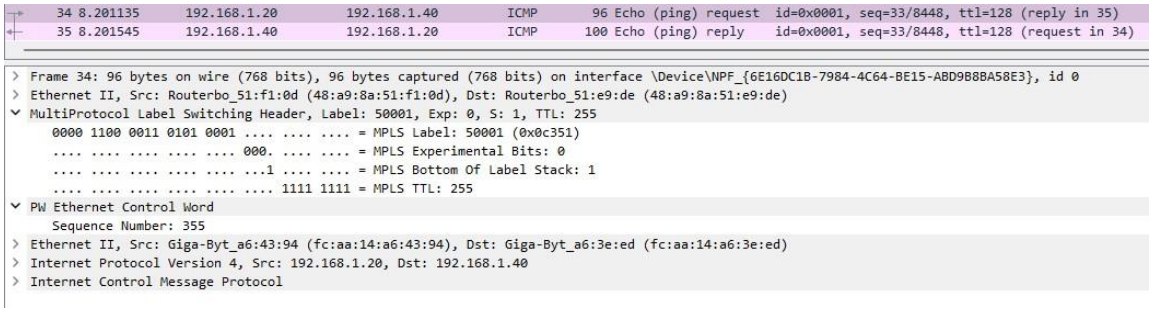


Fig. 159. Capturas ICMP enlace LER1-LSR2.

En este enlace podemos ver en el paquete request, de nuevo, 2 etiquetas. Se puede apreciar que la 50001 es la misma que aparece en el enlace LSR2-LSR3 ya que se trata de la etiqueta del pseudowire para todo el túnel. Bajo esta etiqueta, se encuentra la 2006, que es la asociada a la pila de etiquetas para enviar los paquetes dentro de la MPLS hacia LSR2.

En el paquete reply solo vemos una etiqueta ya que el router LSR2 es penúltimo salto de la red MPLS en el mensaje hacia PC2 Por tanto, solo se observa la etiqueta 10000 correspondiente al pseudowire hacia el PC2.

LER5-LSR3:



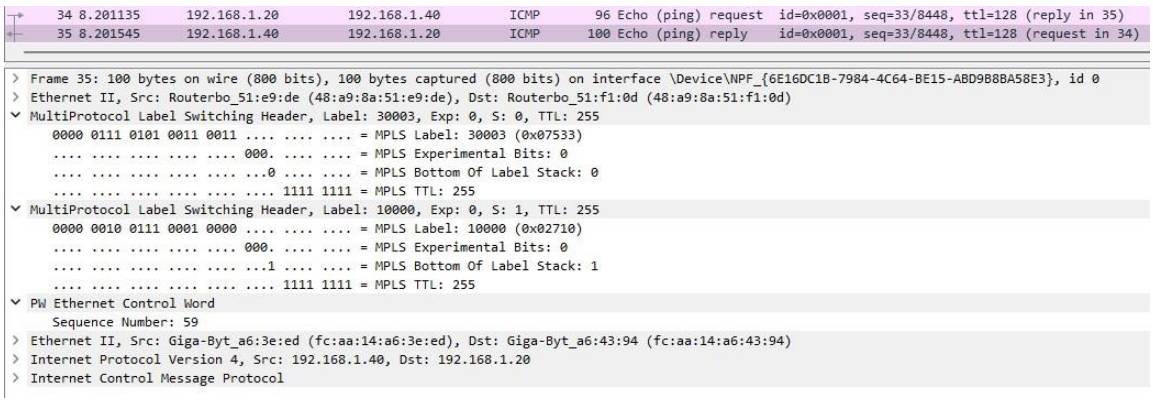


Fig. 160. Capturas ICMP enlace LSR3-LER5.

En este enlace podemos ver que el paquete que tiene 2 etiquetas es el reply, observando de nuevo la correspondiente al pseudowire, la 10000. Mientras que en este caso es la 30003 la asociada al envío de paquetes hacia LSR3.

En ente caso es el router LSR3 el que es penúltimo salto de la red MPLS en el mensaje hacia PC4 Por tanto, solo se observa la etiqueta 50001 correspondiente al pseudowire hacia el PC4.

4. Queremos incluir un nuevo equipo a nuestra red desde una ubicación geográficamente alejada de cualquiera de las ubicaciones que ya tenemos configuradas. ¿Qué pasos deberíamos realizar? ¿Qué conclusiones sacas de la escalabilidad de esta tecnología?

Para poder incluir un equipo a nuestra red transparente VPLS, primero tendremos que conectar a cualquiera de los routers LSR, el router de la nueva ubicación que actuará como un nuevo router LER. En dicho router LER habría que configurar tanto OSPF como MPLS, teniendo en cuenta que tiene que estar en la misma área que la red MPLS original. Obviamente habrá que realizar lo propio sobre el nuevo enlace en el router LSR.

Una vez hecho esto, habría que configurar las interfaces VPLS con el resto de routers LER, implicando esto tener que configurar una nueva interfaz VPLS en cada uno de los routers LER restantes.

Con las interfaces configuradas, habría que incluirlas en los Bridges para formar la malla. También habría que crear un nuevo Bridge en el nuevo router LER con sus interfaces VPLS y la interfaz física elegida, igual que se hizo con el resto de routers LER.

Como se puede observar, utilizar VPLS sobre LDP no es escalable, ya que requiere de configuración muy específica en cada uno de los routers LER de la red.

5. Ahora queremos utilizar la misma red MPLS para conectar de forma transparente otra subred diferente pero que comparte ubicación con la que ya tenemos configurada. ¿Qué pasos deberíamos realizar?

Para poder tener una nueva subred sobre la misma topología, primero tendremos que elegir la subred a configurar de forma que no haya solapamiento con la original. Por ejemplo, podríamos elegir la subred 192.168.2.0/24.

Una vez teniendo esto claro, crearemos un nuevo Bridge para cada router LER y le asignaremos una dirección IP dentro del rango de la subred y que nos servirá de puerta de enlace predeterminada para los equipos que conectaremos.

Ahora crearemos las interfaces VPLS que se usarán para crear los túneles. Hay que tener en cuenta que los vpls-id deben ser únicos entre los propios enlaces de la subred y entre los de la subred original. Una vez los tengamos creados los asociaremos al bridge en cuestión junto a una nueva interfaz física que debe estar libre y ser distinta a la de la otra subred.

Ya por último conectaremos los PC's a las interfaces físicas asociadas con el Bridge y les daremos a los PC's una dirección IP dentro de la subred. No hay que olvidar configurar la puerta de enlace predeterminada con la IP del Bridge.

4.3.9.8. VPLS (BGP)

4.3.9.8.1. Configuración

Como se ha explicado en la teoría y como se ha podido comprobar con los equipos, la configuración de una red transparente VPLS a través de LDP no es escalable. Esto se debe a que el autodescubrimiento no es posible y se tiene que realizar de forma manual.

Sin embargo, si utilizamos conexiones BGP entre los routers que están conectados a los equipos finales, será posible el autodescubrimiento a través de estas conexiones configurando únicamente una instancia VPLS BGP. De esta forma se autodescubrirán todos los pseudowires necesarios para crear la malla de nuestra red.

4.3.9.8.1.1. Borrar interfaces VPLS

Lo primero de todo será eliminar las interfaces VPLS creadas en el anterior diseño y quitar la asociación de estas con el Bridge VPLS en todos los routers LER:

```
[admin@LER0] > int vpls
[admin@LER0] /interface/vpls > remove vpls-to-ler1,vpls-to-ler4,vpls-to-ler5
[admin@LER0] /interface/vpls > .. br port
[admin@LER0] /interface/bridge/port > print
Flags: I - INACTIVE; H - HW-OFFLOAD
Columns: INTERFACE, BRIDGE, HW, PVID, PRIORITY, PATH-COST, INTERNAL-PATH-COST, HORIZON
#   INTERFACE BRIDGE HW  PVID PRIORITY PATH-COST INTERNAL-PATH-COST HORIZON
0 H ether1    VPLS   yes   1 0x80          10             10 none
1   I *24     VPLS         1 0x80          10             10 none
2   I *25     VPLS         1 0x80          10             10 none
3   I *26     VPLS         1 0x80          10             10 none
[admin@LER0] /interface/bridge/port > remove 1,2,3
```

Como se puede observar, al eliminar las interfaces VPLS, estas no se reconocen como puertos del Bridge. Para eliminarlas hay que hacer uso del número de filas con la precaución de no eliminar también el interfaz físico.

4.3.9.8.1.2. MP-BGP

Tal y como hemos comentado, el autodescubrimiento se realizará a través de sesiones BGP que tendremos que establecer entre los routers LER. El establecimiento de estas sesiones es similar al realizado en la Práctica 2 entre los routers PE.

Antes de realizar esto, como hacíamos en la práctica anterior, modificaremos la plantilla *default* de BGP para una mayor facilidad y rapidez para configurar las tres conexiones por router:

```
[admin@LER0] > rou bgp template
[admin@LER0] /routing/bgp/template> set default address-fam=ip,l2vpn as=65000 router-id=192.170.0.1
```

Como podemos ver, en este caso hemos añadido a la familia de direcciones el direccionamiento *l2vpn* que permite encapsular la información de capa 2 enviada en los pseudowires.

Con la plantilla creada, procedemos a establecer las conexiones BGP con los siguientes comandos:

```
[admin@LER0] /routing/bgp/template> .. con
[admin@LER0] /routing/bgp/connection> add name=toLER1 temp=default local.add=\
192.170.0.1 .role=ibgp remote.add=192.170.0.2 .as=65000 conn=yes list=yes
[admin@LER0] /routing/bgp/connection> add name=toLER4 temp=default local.add=\
192.170.0.1 .role=ibgp remote.add=192.170.0.5 .as=65000 conn=yes list=yes
[admin@LER0] /routing/bgp/connection> add name=toLER5 temp=default local.add=\
192.170.0.1 .role=ibgp remote.add=192.170.0.6 .as=65000 conn=yes list=yes
```

Repetiremos análogamente estos dos pasos en el resto de routers LER y comprobaremos que se han establecido las conexiones utilizando el comando ***/routing/bgp/session print*** sobre el terminal de los diferentes routers LER. A continuación, se muestra el ejemplo para LER0:

```
[admin@LER0] /routing/bgp/session> print
Flags: E - established
 0 E name="toLER4-1"
     remote.address=192.170.0.5 .as=65000 .id=192.170.0.5
     .capabilities=mp,rr,gr,as4 .afi=ip,l2vpn,l2vpn-cisco .messages=21
     .bytes=571 .eor=""
     local.role=ibgp .address=192.170.0.1 .as=65000 .id=192.170.0.1
     .capabilities=mp,rr,gr,as4 .afi=ip,l2vpn .messages=21 .bytes=571 .eor=""
     output.procid=20
     input.procid=20 ibgp
     multihop=yes hold-time=3m keepalive-time=1m uptime=17m14s130ms
     last-started=2023-07-14 08:35:36 prefix-count=1

 1 E name="toLER5-1"
     remote.address=192.170.0.6 .as=65000 .id=192.170.0.6
     .capabilities=mp,rr,gr,as4 .afi=ip,l2vpn,l2vpn-cisco .messages=21
     .bytes=571 .eor=""
     local.role=ibgp .address=192.170.0.1 .as=65000 .id=192.170.0.1
     .capabilities=mp,rr,gr,as4 .afi=ip,l2vpn .messages=21 .bytes=571 .eor=""
     output.procid=21
     input.procid=21 ibgp
     multihop=yes hold-time=3m keepalive-time=1m uptime=17m8s20ms
     last-started=2023-07-14 08:35:43 prefix-count=1

 2 E name="toLER1-1"
     remote.address=192.170.0.2 .as=65000 .id=192.170.0.2
     .capabilities=mp,rr,gr,as4 .afi=ip,l2vpn,l2vpn-cisco .messages=21
     .bytes=571 .eor=""
     local.role=ibgp .address=192.170.0.1 .as=65000 .id=192.170.0.1
     .capabilities=mp,rr,gr,as4 .afi=ip,l2vpn .messages=21 .bytes=571 .eor=""
     output.procid=22
     input.procid=22 ibgp
     multihop=yes hold-time=3m keepalive-time=1m uptime=17m3s100ms
     last-started=2023-07-14 08:35:47 prefix-count=1
```

Fig. 161. Conexiones BGP establecidas.

4.3.9.8.1.3. BGP-VPLS

Con las sesiones BGP ya establecidas, podemos centrarnos en configurar el autodescubrimiento de los túneles VPLS a través de estas conexiones. Para ello debemos acceder a /routing/bgp/vpls y crear una única instancia que generará el autodescubrimiento.

Esta instancia además de tenerle que dar un nombre y asociarle el Bridge VPLS, tendrá configurados el *Route Distinguisher*, el *Export Route Targets* y el *Import Route Targets* únicos para toda la subred, por tanto, el mismo en todos los routers LER. Eso sí, para diferenciar los diferentes LER, cada uno de ellos tendrá un único identificador llamado *site-id*.

```
[admin@LER0] /routing/bgp/vpls> add bridge=VPLS export-route-t=1:1 import-route-t=1:1/
name=VPLS rd=1:1 site-id=1
```

Repetiremos análogamente esta ejecución en el resto de routers LER teniendo en cuenta la pertinente modificación del *site-id.*

Si ejecutamos ***/interface/vpls print*** en el terminal, podremos ver que las interfaces VPLS se han aprendido dinámicamente. De igual forma, ejecutando ***/interface/bridge/port print*** podemos ver que dichas interfaces se han asociado también dinámicamente al Bridge VPLS.

```
[admin@LER0] > int vpls print
Flags: R - RUNNING; D - DYNAMIC
Columns: NAME, PEER, BGP-VPLS
#    NAME   PEER          BGP-VPLS
0 RD vpls1  192.170.0.5   vpls
1 RD vpls3  192.170.0.6   vpls
2 RD vpls5  192.170.0.2   vpls
[admin@LER0] > int bridge port print
Flags: D - DYNAMIC
Columns: INTERFACE, BRIDGE, HW, PVID, PRIORITY, PATH-COST, INTERNAL-PATH-COST, HORIZON
#   INTERFACE  BRIDGE  HW   PVID  PRIORITY  PATH-COST  IN  HORIZON
0   ether1     VPLS    yes     1  0x80             10  10  none
1 D vpls1      VPLS            1  0x80             10  10  none
2 D vpls3      VPLS            1  0x80             10  10  none
3 D vpls5      VPLS            1  0x80             10  10  none
```

Fig. 162. Interfaces VPLS aprendidas dinámicamente.

4.3.9.8.2. Verificación final

Para verificar el correcto funcionamiento, podemos realizar una serie de procesos que nos cerciorarán que todo está perfectamente operativo.

Primeramente, podemos realizar pings entre todos los PC's de la red y ver que se obtiene las respuestas a los paquetes icmp enviados.

Con esto vemos que hay conexión entre equipos, pero para comprobar que se establece a través de los túneles, podemos monitorizar los interfaces VPLS que se han generado dinámicamente. Con ello veremos la etiqueta local, que servirá para que el otro extremo envíe la información; la etiqueta remota, que es la asignada por el otro extremo para la comunicación hacia dicho peer; y los diferentes atributos relacionados con el siguiente salto, etiqueta, interface y.nexthop.

```
[admin@LER0] /interface/vpls> monitor vpls1
  remote-label: 10001
   local-label: 18
      nexthops: { label=20003; nh=10.0.0.2%ether2; interface=ether2 }

[admin@LER0] /interface/vpls> monitor vpls2
  remote-label: 40001
   local-label: 19
      nexthops: { label=20005; nh=10.0.0.2%ether2; interface=ether2 }

[admin@LER0] /interface/vpls> monitor vpls3
  remote-label: 50001
   local-label: 20
      nexthops: { label=20006; nh=10.0.0.2%ether2; interface=ether2 }
```

Fig. 163. Monitorización interfaces VPLS.

4.3.9.8.3. Actividades propuestas

6. Realice un ping desde el PC1 al PC4 ¿Hay alguna diferencia en algún enlace del túnel VPLS respecto a VPLS (LDP)? Recuerde que tiene que ejecutar el WireShark en los diferentes enlaces que hay entre LER1 y LER5.

LER1-LSR2:

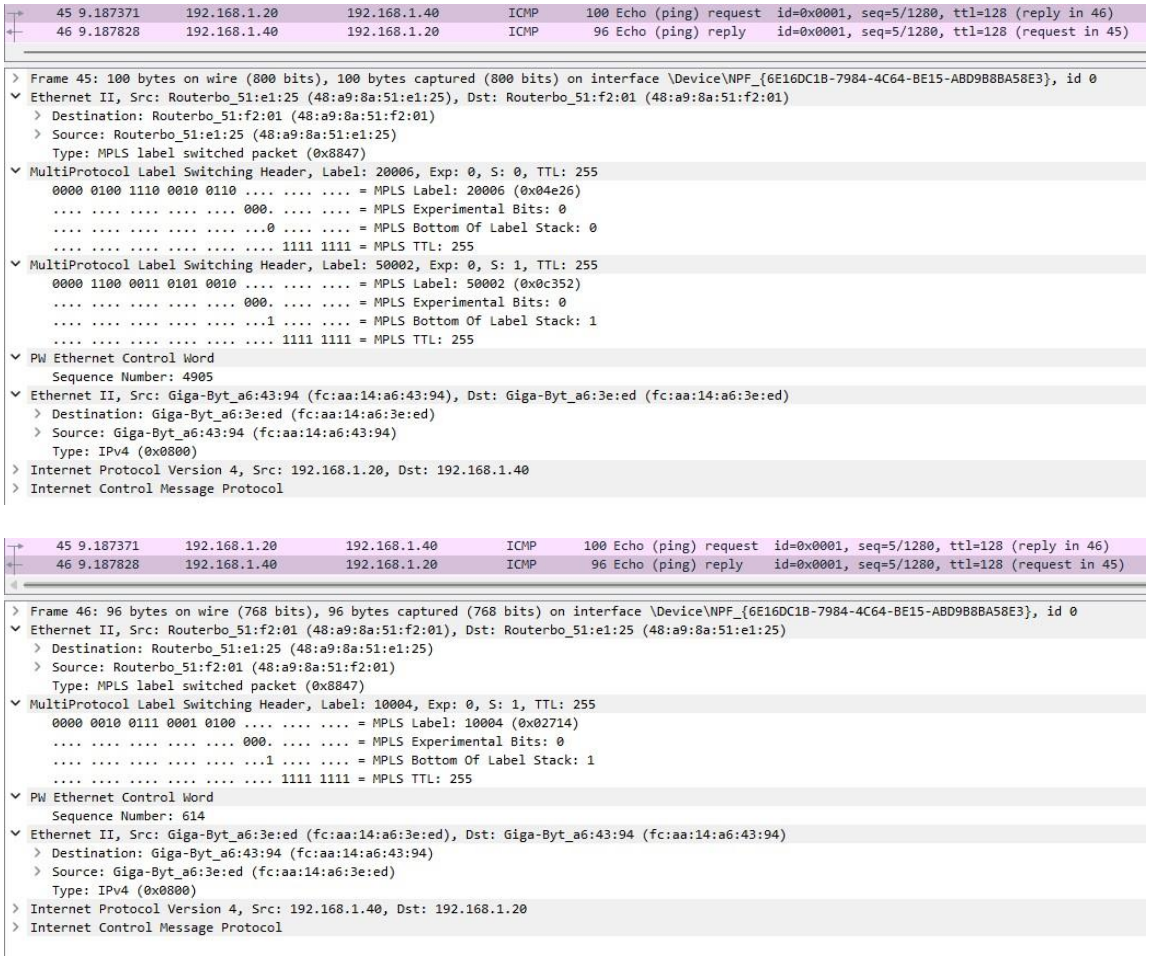
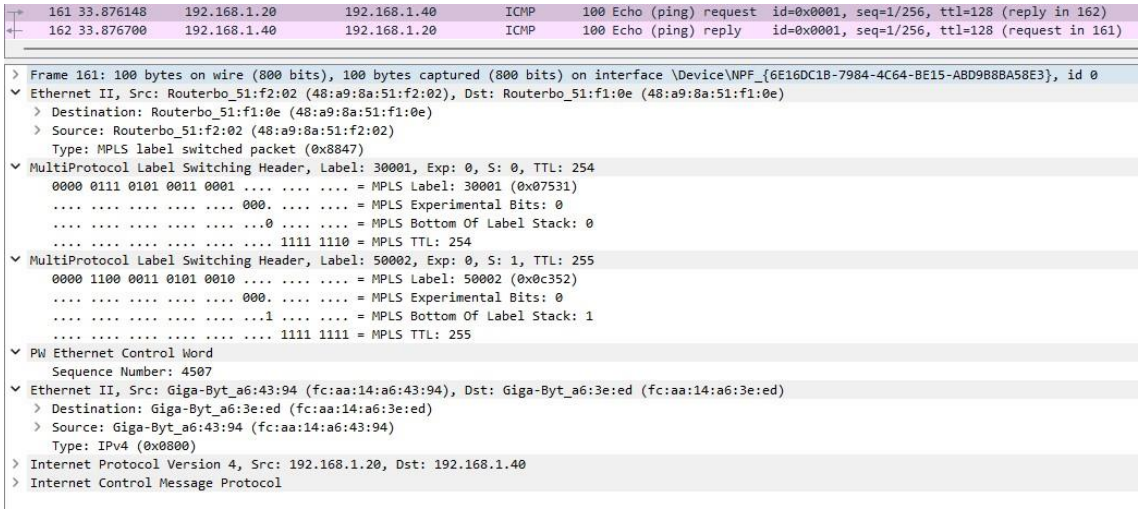


Fig. 164. Capturas ICMP enlace LER1-LSR2.

LSR2-LSR3:



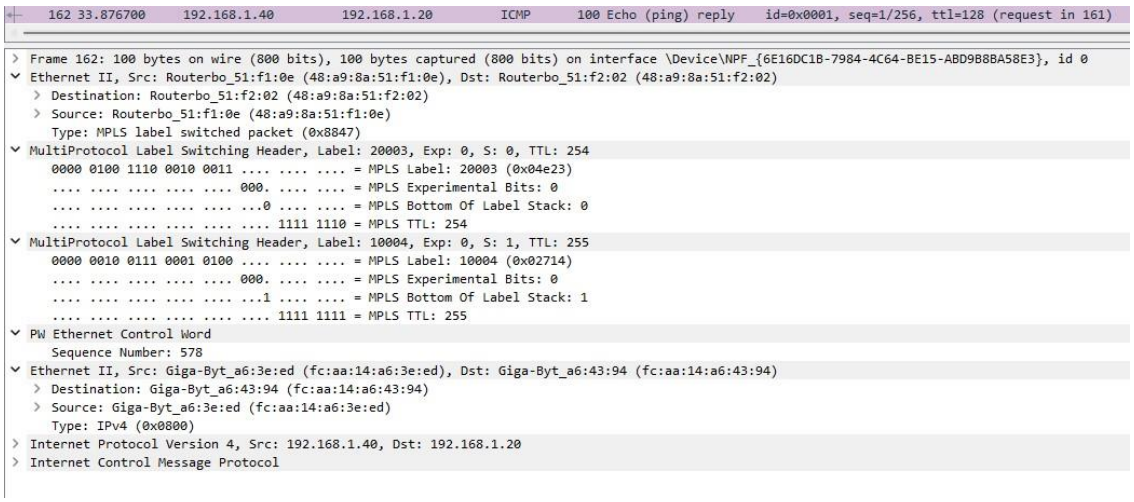


Fig. 165. Capturas ICMP enlace LSR2-LSR3.

LSR3-LER5:

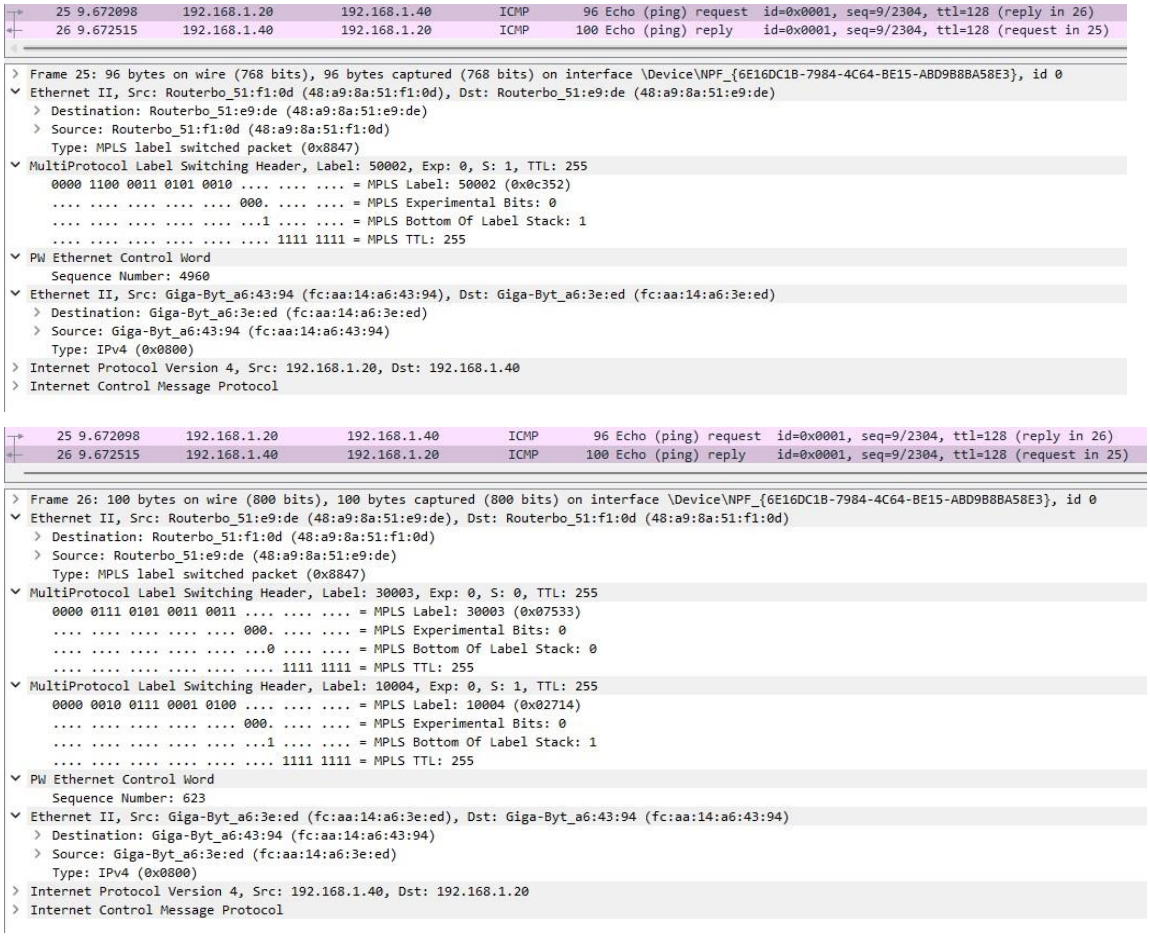


Fig. 166. Capturas ICPM enlace LSR3-LER5.

Podemos observar de nuevo la misma dinámica, una pila de dos etiquetas con la etiqueta del pseudowire como última etiqueta y la de direccionamiento en la red MPLS por debajo. Vemos que las etiquetas toman otros valores distintos.

Al fin y al cabo, vemos que la aplicación es la misma ya que no se aprecian diferencias y es más escalable a través de BGP.

7. ¿Qué pasos deberíamos realizar en la configuración VPLS (BGP) respecto a VPLS (LDP) para conectar un nuevo equipo a la red desde una ubicación geográficamente alejada?

Para poder incluir un equipo a nuestra red transparente VPLS, primero tendremos que conectar a cualquiera de los routers LSR, el router de la nueva ubicación que actuará como un nuevo router LER.

En dicho router LER habría que configurar tanto OSPF como MPLS, teniendo en cuenta que tiene que estar en la misma área que la red MPLS original. Sin olvidar hacer lo propio con el nuevo enlace en el router LSR.

Estos dos primeros pasos no difieren variación respecto a VPLS (LDP). Ahora crearemos el Bridge VPLS en el LER como hacíamos con el resto y le asignaremos la IP de puerta de enlace predeterminada, que tendrá que estar dentro de la subred.

Una vez hecho esto, habría que establecer conexiones BGP entre el nuevo router LER y el resto de routers de este tipo que ya forman parte de la red. Con las sesiones establecidas se creará una instancia BGP-VPLS con los mismos parámetros que en el resto de LER's (*Route Distinguisher*, *Export Route Targets*, *Import Route Targets*) a excepción del identificador *site-id* que será único.

Como hemos visto, a partir de BGP_VPLS se crean dinámicamente las interfaces y se asocian al Bridge, por tanto, solo nos quedará añadir la interfaz física a la que se conectará el nuevo PC al Bridge VPLS.

Por último, habrá que cerciorarse de que la IP del PC pertenece a la misma subred y puede hacer *ping* con el resto de PC's de nuestra subred.

8. ¿Cómo deberíamos proceder en VPLS (BGP) para configurar una subred diferente sobre la misma topología MPLS?

Para poder tener una nueva subred sobre la misma topología, primero tendremos que elegir la subred a configurar de forma que no haya solapamiento con la original. Por ejemplo, podríamos elegir la subred 192.168.2.0/24.

Una vez teniendo esto claro, crearemos un nuevo Bridge para cada router LER y le asignaremos una dirección IP dentro del rango de la subred y que nos servirá de puerta de enlace predeterminada para los equipos que conectaremos.

Estos pasos son idénticos a una configuración VPLS sobre LDP, pero ahora en vez de crear directamente las interfaces, crearemos una nueva instancia VPLS-BGP para cada router LER. El *site-id* será el mismo que la otra instancia creada en el router, sin embargo, el *Route Distinguisher*, el *Export Route Targets* y el *Import Route Targets* serán diferentes respecto a la otra instancia, pero idénticos a todos los routers LER para esta subred. Por ejemplo, 2:2. No hay que olvidar que estas instancias se asocian con el nuevo Bridge creado.

Como sabemos ya de sobra, las interfaces VPLS y las asociaciones al Bridge se crean dinámicamente. Por tanto, lo único que tenemos que hacer sobre el Bridge es asociar una interfaz física diferente y que esté libre.

Ya por último conectaremos los PC's a las interfaces físicas asociadas con el Bridge y les daremos a los PC's una dirección IP dentro de la subred. No hay que olvidar configurar la puerta de enlace predeterminada con la IP del Bridge.

5. Bibliografía

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

[2] MikroTik Documentation https://help.mikrotik.com/docs/display/ROS/RouterOS [Online]

[3] MikroTik Documentation https://wiki.mikrotik.com/wiki/Manual:TOC [Online]

[4] Administración Avanzada de BGP y MPLS con MikroTIk RouterOS", Academy Xperts.

[5] Documentation GNS3 http://docs.gns3.com/ [Online]

[6] VMware Workstation Player Documentation https://www.vmware.com/es [Online]