Universidad Politécnica de Cartagena

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

Construcción de una red MPLS y validación de GNS3 para su simulación.

AUTOR:

Héctor Delgado Patiño

DIRECTOR:

Pablo Antonio López-Matencio Pérez

ÍNDICE

Este proyecto está dedicado: a mi madre, sin cuyo esfuerzo y sacrificio no hubiera podido conseguir una carrera, y a Victoria, por todos los años juntos y las vivencias.

Construcción de una red MPLS y validación de GNS3 para su simulación.

Capítulo 1: INTRODUCCIÓN

1.1 ¿Por qué este trabajo?

El propósito en este proyecto es:

- Abordar un problema de ingeniería real, diseñando y construyendo una red de comunicaciones de prueba en laboratorio similar a la que pudiera tener una empresa con varias sedes.
- La tecnología MPLS ha ido ganando terreno en los últimos años y actualmente se usa en las redes de tecnología de muchos operadores. Entre los motivos está principalmente su facilidad para configurar nuevos servicios y ofrecer QoS (Calidad de Servicio).
- Utilizar para ello hardware y software CISCO, el más extendido actualmente. Esto me permitirá ganar experiencia con un fabricante de equipos con el que muy probablemente trabajaré como ingeniero.
- Al mismo tiempo, en este proyecto queremos averiguar si podemos utilizar el simulador Global Network Simulator (GNS3) para probar de forma fiable cambios en la configuración de una red MPLS.

Las principales ventajas de GNS3 y que le diferencian de otros simuladores son:

- Desarrollo hecho en Python, que es un lenguaje interpretado, por lo que fácilmente puedo utilizarlo en diversas plataformas.
- Ejecuta el sistema operativo del router con todas las funciones y comandos de un router físico.

1.2. Objetivos

a) **Estudiar exhaustivamente las capacidades de la herramienta GNS3 para la simulación y montaje de redes en entorno real.**
b) **Estudiar la transmisión de datos mediante el protocolo MPLS.**
c) **Comparativa de datos obtenidos en la simulación y en el entorno real.**
d) **Revisar las capacidades de utilización de los aparatos y software CISCO para su uso en la solución de necesidades de la comunicación.**

1.3. Contenido de la memoria.

Este Trabajo de Final de Grado se estructura en 6 capítulos: El capítulo 1, denominado *Introducción,* contiene la motivación del proyecto, los objetivos que persigue y una breve descripción del contenido de la misma. En el capítulo 2, *La tecnología MPLS* describimos exhaustivamente qué es el protocolo MPLS, así como sus fundamentos generales. Incluye también conceptos generales sobre CISCO y una descripción de la arquitectura de la herramienta GNS3 que implementa el protocolo. El capítulo 3, *El simulador GNS3*, se basa en la descripción, fundamentos y "guía de usuario" de la herramienta GNS3. El capítulo 4, denominado *Simulación de la red en la herramienta GNS3,* se trata de la simulación realizada sobre el escenario elegido y se describe cómo funcionarían los parámetros con la implementación del protocolo. En el capítulo 5, *Montaje de la red con equipos reales,* se pasa a comprobar lo medido y analizado en el capítulo anterior ya sobre el escenario real. El capítulo 6, *Conclusiones* se encarga de analizar los resultados, extraer conclusiones y estudiar posibles líneas de investigación. Por último, se lista la bibliografía empleada.

Se incluye también dos Anexos listando abreviaturas empleadas y las características del hardware utilizado en el proyecto.

CAPÍTULO 2: LA TECNOLOGÍA MPLS

2.1. Conceptos generales de MPLS

Multiprotocol Label Switching (MPLS) **[1]** es un protocolo para incrementar la velocidad y moldear los flujos de tráfico en una red. Permite a la mayoría de los paquetes ser enviados en la capa OSI 2(Nivel de enlace) de forma preferente a que suban al nivel 3 (Nivel de Red). Cada paquete es etiquetado a la entrada de la red del proveedor de servicios por el router de ingreso. Todos los conmutadores de ruta subsiguientes ponen en marcha el reenvío de paquetes basándose en estas etiquetas (no miran la cabecera IP). Finalmente, el router de salida elimina las etiquetas y envía el paquete IP original hasta su destino final.

La etiqueta determina el camino predeterminado que seguirá el paquete. Estos caminos se denominan **caminos de intercambio de etiquetas** (label-switched paths) y permiten al proveedor de servicios decidir en el acto cual será el mejor camino para ciertos flujos de tráfico, en función de si la red es privada o pública.

MPLS encuentra sus principales aplicaciones en la mejora de la calidad de servicio (QoS) por parte de los proveedores de servicios de Internet, pudiendo definirse diferentes LSP para controlar parámetros tales como: latencia, retardo, pérdida de paquetes y congestión de la red. Este protocolo también permite la creación de redes privadas virtuales (VPN), servicios LAN privados virtuales (VPLS) y líneas virtuales arrendadas (VLLS).

2.1.1. Un poco de historia. El porqué de MPLS.

Tradicionalmente, el envío de paquetes IP consistía en el análisis de la dirección IP destino contenida en la cabecera de la etiqueta de red de cada paquete, al mismo tiempo que el paquete viajaba de su origen a su destino. Esto involucraba que un router tenía que analizar la dirección IP independientemente de cada salto en la red. Para construir la tabla de enrutamiento, tanto los protocolos de enrutamiento dinámicos como las configuraciones necesitaban analizar las direcciones IP de los paquetes recibidos. A esta forma de implementación del enrutamiento, se le llamó ***encaminamiento unicast salto-a-salto basado en destino.***

Pese al éxito de esta implementación, existían ciertas restricciones que se hicieron evidentes con el incremento de los tamaños de las redes y la generalización del acceso a las mismas. La más grave era la disminución de la

flexibilidad de la red a medida que esta crecía. Para paliar esta situación, a mediados de la década de los 90(1996) un grupo de investigación de la empresa CISCO, proponía un sistema de conmutación basado en etiquetas. El propósito principal era evitar tener routers actualizando y mirando continuamente las tablas de enrutamiento IP, lo que hubiera supuesto una pérdida de tiempo fácilmente evitable.

Se tiende a pensar que este protocolo sólo se usa en redes privadas, pero en realidad, es utilizado por todas las redes de los proveedores de servicio (incluyendo los backbones). Hoy incluso se ha extendido una generalización del protocolo (GMPLS) que lo amplía incluso a redes que implementan diferentes clases de tecnologías de intercambio, más allá del simple intercambio de paquetes.

2.1.2. Ventajas y desventajas del protocolo MPLS

Concebido como un protocolo que buscaba mejorar las tecnologías existentes, podemos destacar las siguientes ventajas **[4]**:

a) Mediante MPLS, los proveedores de servicio de Internet pueden soportar servicios diferenciados o DiffServ (tal como se recoge en la norma RFC 3270). Ante el aumento de la demanda de nuevas aplicaciones, que suponen nuevos requerimientos de ancho de banda y tolerancia a retardos, MPLS ofrece una gran flexibilidad en cuanto a los diferentes servicios ofertados, lo que permite responder a esta demanda de forma óptima.
b) MPLS ofrece un mecanismo sencillo para crear VPNs, ya que permite la creación de circuitos o túneles virtuales dentro de la red IP, y esto a su vez, garantiza poder aislar el tráfico y el acceso al mismo.
c) Permite ahorrar costes entre un 10%-25% frente a otros servicios de datos, en función de la combinación específica de aplicaciones y de la configuración de red de la empresa. En los últimos años, se han efectuado diversas pruebas que incluso han alcanzado el 40 % de ahorro de costes respecto a ATM o Frame Relay.
d) Mejora del rendimiento, ya que al ser su naturaleza "muchos-a-muchos", los diseñadores de red pueden reducir el número de saltos entre puntos, permitiendo a su vez mejorar los tiempos de respuesta y rendimiento de las aplicaciones.
e) Recuperación ante desastres: Los servicios basados en MPLS permiten la recuperación de diferentes maneras. En primer lugar, permiten conectar los emplazamientos clave a la nube MPLS, y a través de ella, a otros sitios de la red. Además es posible, reconectar los sitios remotos a localizaciones que

actúen como copia de seguridad en caso de desastre. Todo ello lo hace una de las principales razones por las cuales las empresas están migrando a esta tecnología.

Ahora bien, MPLS no es perfecto, y también cuenta con algunas desventajas. En primer lugar, su aparente flexibilidad no es completa del todo, ya que ciertas características o su forma de implementación en el protocolo no han sido estandarizadas, dejándose al arbitrio de cada fabricante de red. El hecho de que sea un protocolo joven también hace que se vea infrautilizado en algunos casos, dado que al estar en continua evolución, aún se siguen especificando estándares y borradores para algunas características. Finalmente, la desventaja principal es que MPLS no posee ningún mecanismo "per se" para proteger la seguridad en las comunicaciones, teniendo que poner el proveedor de servicio sus propios medios para obtenerla.

2.1.3. La etiqueta MPLS

En este último subapartado, vamos a hablar de la etiqueta MPLS. Quedó antedicho que MPLS permite a cada nodo (ya sea switch o router) asignar una etiqueta a cada uno de los elementos de la tabla y comunicarla a sus nodos vecinos.

Esta etiqueta es únicamente un valor corto de tamaño fijo que se transporta en la cabecera del paquete para identificar un FEC (Forward Equivalence Class). Sirve como un identificador de conexión, con significado local, que establece una correspondencia entre el tráfico y un FEC específico. Se puede asignar una etiqueta al paquete basándose en la dirección de destino, los parámetros de tipo de servicio, la VPN a la que pertenece o cualquier otro criterio.

Los campos de la cabecera MPLS son:

- Label (20 bits): Indica el valor actual de la cabecera MPLS. Este valor determinará el próximo salto del paquete.
- QoS (3 bits): Este campo indica la calidad de servicio del paquete. Permite diferenciar entre distintos tipos de tráfico y mejorar el rendimiento de un tipo determinado respecto a otros.
- Stack (1 bit): Este bit soporta una pila de etiquetas jerárquicas, esto es, nos indica si existen más etiquetas MPLS. Esta posibilidad de encapsular cabeceras MPLS en otras, tiene sentido si se da el caso de que el paquete tenga que atravesar una red MPLS perteneciente a un proveedor de servicio u organismo distinto, de tal manera que al terminar de atravesar dicha red, se continúe trabajando con MPLS.

2.2. La arquitectura MPLS

2.2.1. Elementos de una red MPLS

En general, la diferencia principal entre MPLS y las tecnologías WAN **[2]** utilizadas tradicionalmente es la forma en la que se asignan las etiquetas y la capacidad de cargar una pila de etiquetas asociada a un paquete.
El envío de paquetes en MPLS está en un fuerte contraste con los entornos de redes sin conexión actuales, donde se analiza el paquete en cada salto.

Un concepto importante dentro de MPLS es el de LSP (Label Switch Path) que es un camino específico de tráfico a través de la red MPLS, el cual se crea utilizando protocolos de distribución de etiquetas, tales como RSVP-TE o CR-LDP, si bien el más comúnmente utilizado es el primero de ellos.

El LDP posibilita que los nodos MPLS se descubran y establezcan comunicación entre ellos, a fin de informarse del valor y significado de las etiquetas que serán utilizadas en los enlaces contiguos.

Básicamente, la arquitectura del protocolo puede dividirse en dos elementos fundamentales: los componentes de envío, y los componentes de control. Los primeros utilizan una base de datos de etiquetas de envío mantenida por un conmutador de etiquetas para poner en marcha el envío de paquetes de datos basados en etiquetas llevadas por paquetes. Los componentes de control, por su parte, son responsables de crear y mantener la información de envío de etiquetas entre un grupo de switches interconectados.

La diferencia entre un router que no implemente MPLS y uno que sí, es que en el primero se intercambia la información de routing con otros routers y se almacena en una tabla de almacenamiento IP, y los paquetes se reenvían consultando esta tabla. Uno que implemente MPLS, posteriormente utiliza la tabla de routing para establecer e intercambiar etiquetas y almacena esa información en la tabla *Label Forwarding*. Los paquetes que entren y salgan del router, se etiquetarán entonces según la información de esta tabla

Finalmente, una red MPLS va a estar compuesta por dos tipos de nodos, como se puede apreciar en la figura 1. Estos nodos son denominados LER (Label Edge Routers) y LSR (Label Switching Routers). Cada uno de ellos se detallará en profundidad en los siguientes epígrafes.

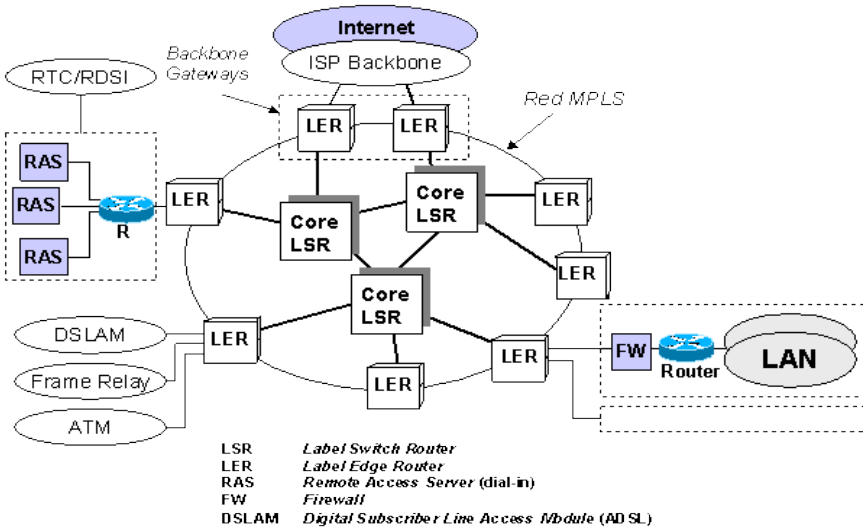


Figura 1. Ejemplo de una red MPLS. Se pueden observar como los principales componentes son LER y LSR. (www.etutorials.org)

2.2.2. Bloques constituyentes de la arquitectura MPLS (I) : Label Switch Routers

El elemento primordial en cualquier red MPLS es el Label Switch Router (LSR). **[2]** En esta categoría se engloban todos aquellos dispositivos que puedan implementar procedimientos de distribución de etiquetas y puedan enviar paquetes.

Dependiendo de la localización dentro de la red, podemos diferenciar hasta 4 tipos diferentes de LSR:

a) Router de ingreso: Se sitúa al comienzo del LSP (punto de entrada), siendo el único router por donde puede entrar tráfico IP a la red MPLS. Sirven como routers entrantes, ya que reciben la información del tráfico de red que circulará por el LSP hasta alcanzar su destino. El router entrante encapsula el tráfico usando una cabecera MPLS.
b) Router de tránsito: Se sitúa en el medio del LSP. Estos routers solo envían el tráfico recibido al siguiente punto del LSP, usando la interfaz desde la cual ha venido el paquete así como la cabecera MPLS para obtener la información de destino.
c) Penúltimo router: Se sitúa antes del router de salida, y se usa para eliminar la cabecera MPLS antes de enviarle el tráfico. Al ser el último salto el router de salida, la cabecera MPLS ya no es necesaria.
d) Router de salida: Se sitúa al final del LSP (punto de salida). Recibe el tráfico IP proveniente del penúltimo router y lo envía usando un enrutamiento IP normal.

2.2.3. Bloques constituyentes de la arquitectura MPLS (II): Label Edge Routers (LER)

Un LER es un tipo de router que también implementa el mecanismo de imposición de etiquetas, así como el de disposición.
Estos routers asignan (o eliminan) etiquetas de los paquetes de datos en función del tipo de información que estos lleven.
El uso principal de este tipo de routers se da en redes MPLS grandes, especialmente en aquellas orientadas a altas prestaciones tecnológicas.

En las figuras 2 y 3, se muestran diagramas de cómo son las arquitecturas de un LSR y un LER. Los mecanismos de imposición de etiquetas se describen en el siguiente epígrafe.

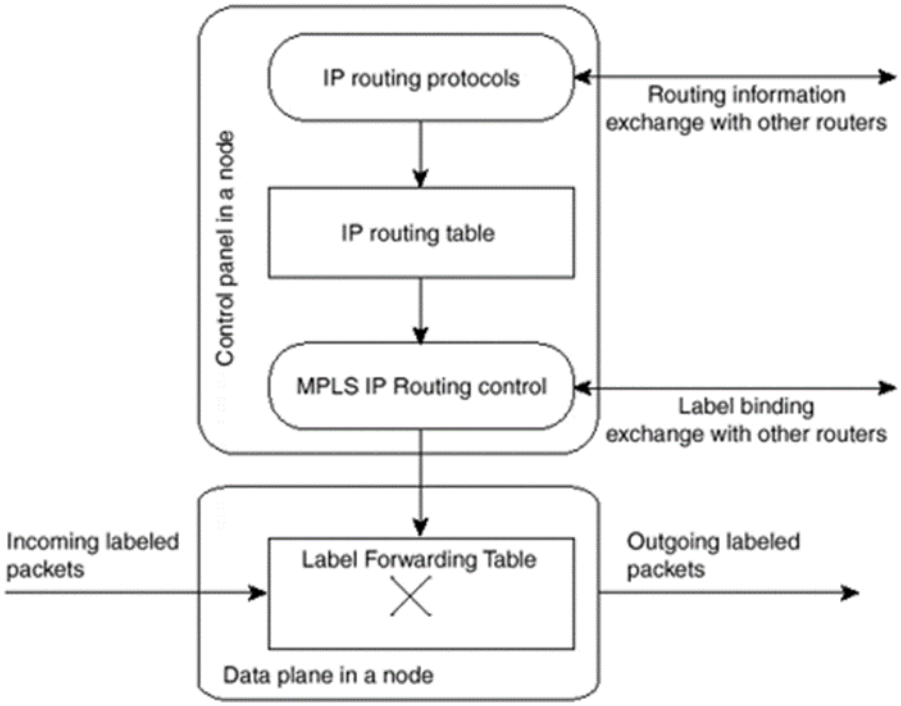


Figura 2. Ejemplo de arquitectura de un router LSR. (www.etutorials.com)

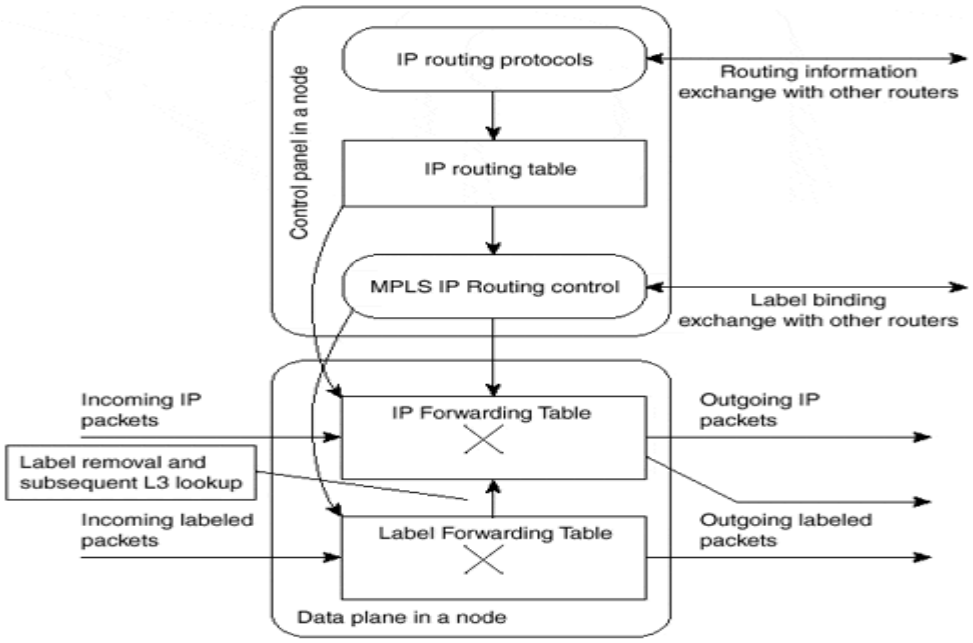


Figura 3. Ejemplo de arquitectura de un router LER. (www.etutorials.org)

2.2.4. El mecanismo de imposición (y disposición) de etiquetas en un LER.

La imposición de etiquetas se define como el acto de anteponer una etiqueta, o una pila de etiquetas en un paquete de datos, dentro del punto de ingreso en el dominio MPLS. A su vez, su contraparte es la disposición de estas mismas etiquetas en el punto de salida antes de reenviarlo a un vecino fuera del dominio MPLS.

Para implementar estas funciones, el LER necesita comprender dónde está la cabecera del paquete y qué etiqueta se le debe asignar al paquete. En el encaminamiento tradicional de la capa IP, cada salto en la red efectúa una búsqueda en la tabla de encaminamiento IP para la dirección IP de la cabecera. Se selecciona la siguiente dirección IP para el paquete en cada iteración de la actualización de la tabla, y se envía el paquete hasta su destino final.

Con la arquitectura MPLS, escoger el siguiente salto del paquete IP combina dos funciones. La primera de ellas es repartir el conjunto de posibles paquetes dentro de un conjunto de IPs prefijadas. El resultado de esta función es conocido como FEC. (Forwarding Equivalence Classes) La segunda consiste en relacionar cada IP destino prefijada al siguiente salto. Esto significa que cada destino en la red es alcanzable por un camino determinado por el flujo de tráfico desde un dispositivo de ingreso hasta el dispositivo de salida.

Cuando se utiliza el protocolo MPLS, un paquete en particular se asigna a una FEC particular una sola vez, y esto ocurre en el dispositivo frontera al entrar el paquete en la

red. La FEC a la que se asigna el paquete se codifica como un identificador fijo de corta longitud, conocido como una etiqueta.

En la figura 4 se ilustra el proceso de imposición de etiqueta.

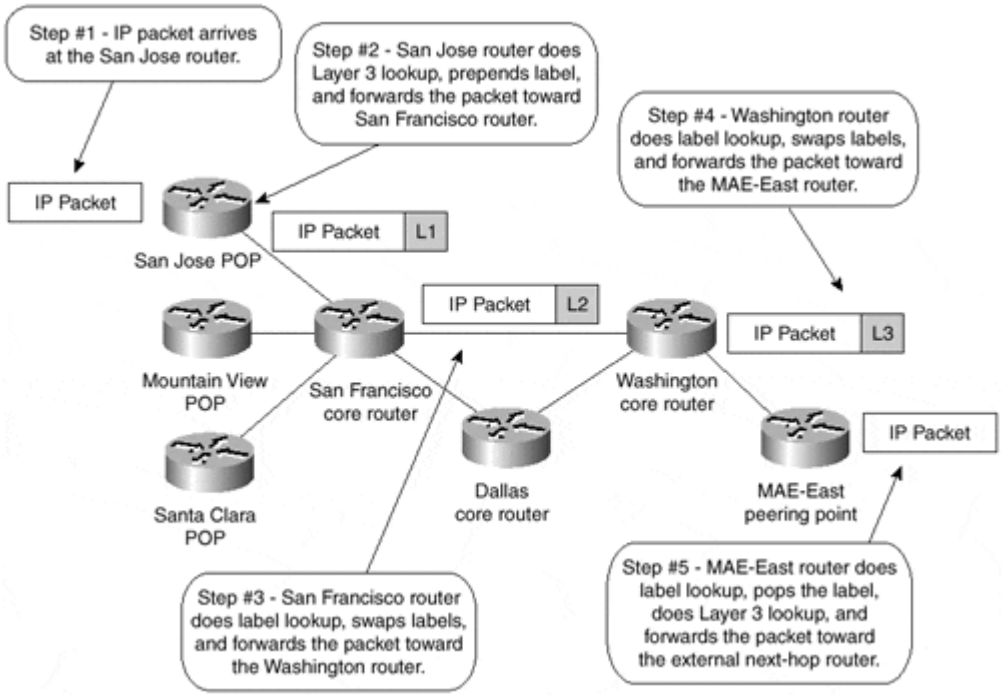


Figura 4. Proceso de imposición de la etiqueta en el protocolo MPLS (www.etutorials.org)

2.3. Conceptos generales de CISCO para MPLS.

Dado que en este proyecto vamos a utilizar dispositivos CISCO para desarrollar este proyecto, conviene reseñar algunos conceptos específicos de la marca para este protocolo. En particular, se va a hablar del sistema CISCO IOS, y de las características que oferta este sistema a la hora de trabajar con el protocolo MPLS.

2.3.1. El sistema CISCO IOS

El sistema CISCO IOS (InterNetwork Operating System) es el sistema operativo utilizado en los dispositivos de red de CISCO (routers y switches), para la implementación de redes. A este sistema se puede acceder mediante la CLI (Command Line Interface), que como su nombre indica se basa en un conjunto de comandos, ubicados en los modos de operación definidos en cada dispositivo de red.

Elegir una versión adecuada de CISCO IOS va a depender de las características técnicas que puedan ofrecer los dispositivos de red que necesitemos según lo que queramos implementar en nuestra red (IPv6, MPLS, DiffServ, Seguridad…).

Además, los routers van a tener diversos tipos de interfaces, comúnmente Fast-Ethernet y Serial, por lo que es necesario que el IOS contenga los drivers adecuados para simular correctamente el funcionamiento de estas interfaces.

En este proyecto, los dispositivos disponibles serán routers de la familia c2600, por implementar ya el sistema MPLS y por haber trabajado anteriormente con ellos en otras asignaturas de la carrera, lo cual permite una mayor soltura en el manejo de los mismos.

Cabe destacar además, que a las imágenes de CISCO IOS les son asignados nombres específicos y estandarizados por la empresa que reflejan sus principales características (plataforma soportada, funciones que realiza, número de versión…).

2.3.2. MPLS en CISCO IOS

El software CISCO IOS que implementa MPLS habilita a las empresas y a los proveedores de servicio para construir redes que entreguen servicios añadidos de alto valor sobre una infraestructura simple. **[3]** Otra gran ventaja es que esta solución se puede implementar sobre cualquier infraestructura ya existente. Por último, esta tecnología permite agregar a los suscriptores a un router frontera sin cambiar sus entornos actuales, ya que MPLS permite independencia de la tecnología de acceso.

Todo ello redunda en que, al aplicar los componentes de esta tecnología (VPNs Layer 2 y 3, Traffic Engineering, QoS…) permiten el desarrollo de redes altamente eficientes, escalables y seguras que cumplen con los requerimientos de nivel de servicio mínimos que deben ser garantizados.

En la bibliografía se incluye la página web de CISCO donde pueden verse las características de la familia de dispositivos. **[10]**

CAPÍTULO 3: EL SIMULADOR GNS3

3.1. Introducción

Graphical Network Simulator o GNS3 es una herramienta de simulación de redes potente y muy útil para el estudio y emulación de topologías de red complejas. Se basa en la emulación de plataformas hardware de CISCO, y además permite la ejecución del CISCO IOS, del que ya se habló en el capítulo anterior. **[5]** Consiste en un entorno gráfico amigable al usuario para permitir una creación sencilla de las topologías deseadas, apoyándose en simuladores de bajo nivel. El más importante es Dynamips **[6] [7]** ya que permite la emulación de los dispositivos CISCO. Mediante Virtualbox también se hace posible la integración de hosts en la topología.

GNS3 se distribuye bajo licencia OpenSource y su uso es totalmente gratuito. Puede ejecutarse tanto en Linux como en Windows, aunque se obtienen mayores rendimientos si se corre en Linux. Esta herramienta es ampliamente usada en la formación y aprendizaje de CISCO Academy.

En este proyecto, se va a trabajar con GNS3 ejecutado sobre Windows 7, y la versión que se va a utilizar es la 1.2.1., si bien se puede trabajar con cualquier versión desde la 0.8.7.en adelante, que es la que incluyen los ordenadores de la ETSIT.

3.2. Instalación de GNS3 en Windows

La instalación de la herramienta en Windows es sencilla. Para ello, seguimos los siguientes pasos:

Paso 1: Descargar el paquete GNS3.exe desde la página oficial (recomendado especialmente para evitar virus, troyanos, etc…): www.gns3.net . Hacer clic en la casilla Free Download. Se redirigirá a una página nueva en la que habrá que crearse una cuenta para entrar en la Comunidad GNS3 (totalmente gratis). Una vez creada y confirmada la cuenta, se abrirá la página con los selectores para elegir versión de Windows y versión del programa. Hacer clic en "Aceptar" y dejar que empiece la descarga. El paquete incluye, como componentes fundamentales, Dynamips (el software para la ejecución de GNS3), Putty (cliente para conexiones SSH y Telnet) y WinPcap (conjunto de librerías que permite trabajar con protocolos de red y analizadores de red, como Wireshark).

Paso 2: Después de cargar el .exe y seguir los pasos haciendo clic en "Next" (previo cuidado de no dejar que se nos instalen programas raros), llegamos a la siguiente ventana:

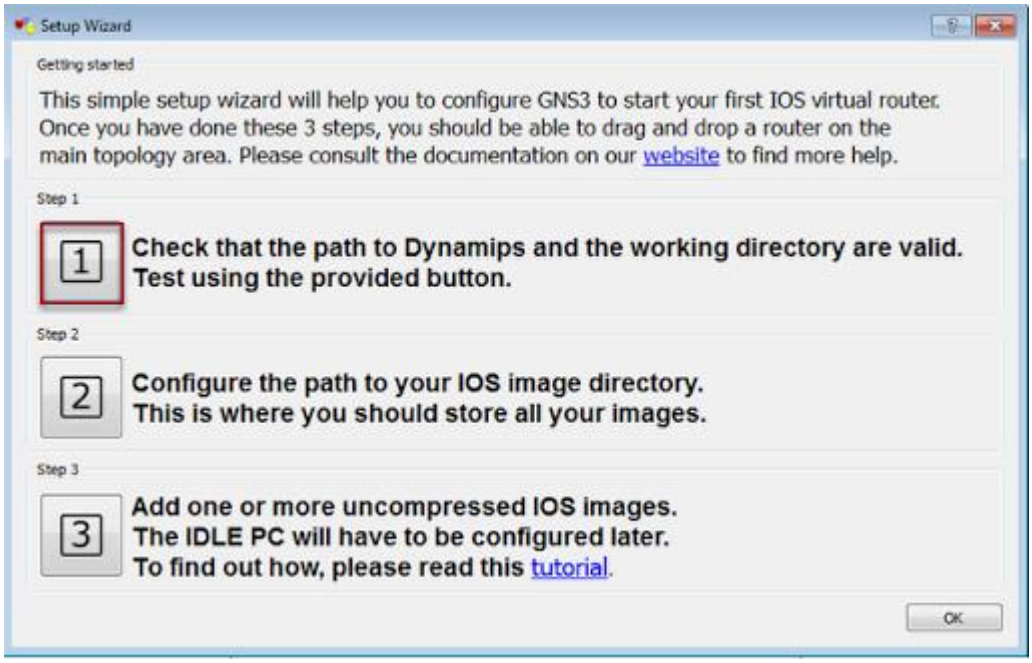


Figura 5. Ventana de opciones del instalador de GNS3

La opción 1 de esta imagen indica que tenemos que elegir, configurar y comprobar que es válida una ruta donde hayamos instalado el software de Dynamips.

La opción 2 nos permite ir a la configuración de la ruta de nuestro directorio de imágenes IOS.

La opción 3 nos pide seleccionar una o varias imágenes de dispositivos que implementen CISCO IOS para empezar a trabajar.

Paso 3: Ahora tenemos que seleccionar las imágenes IOS de la carpeta elegida anteriormente. Para ello, tenemos que ir a la pestaña Editar -> Imágenes IOS e Hypervisors. (Véase Figura 7). *En la pestaña de Imágenes seleccionaremos los IOS que vamos a usar, así como también podremos configurar datos como el modelo del IOS, la RAM y algunas otras opciones.*

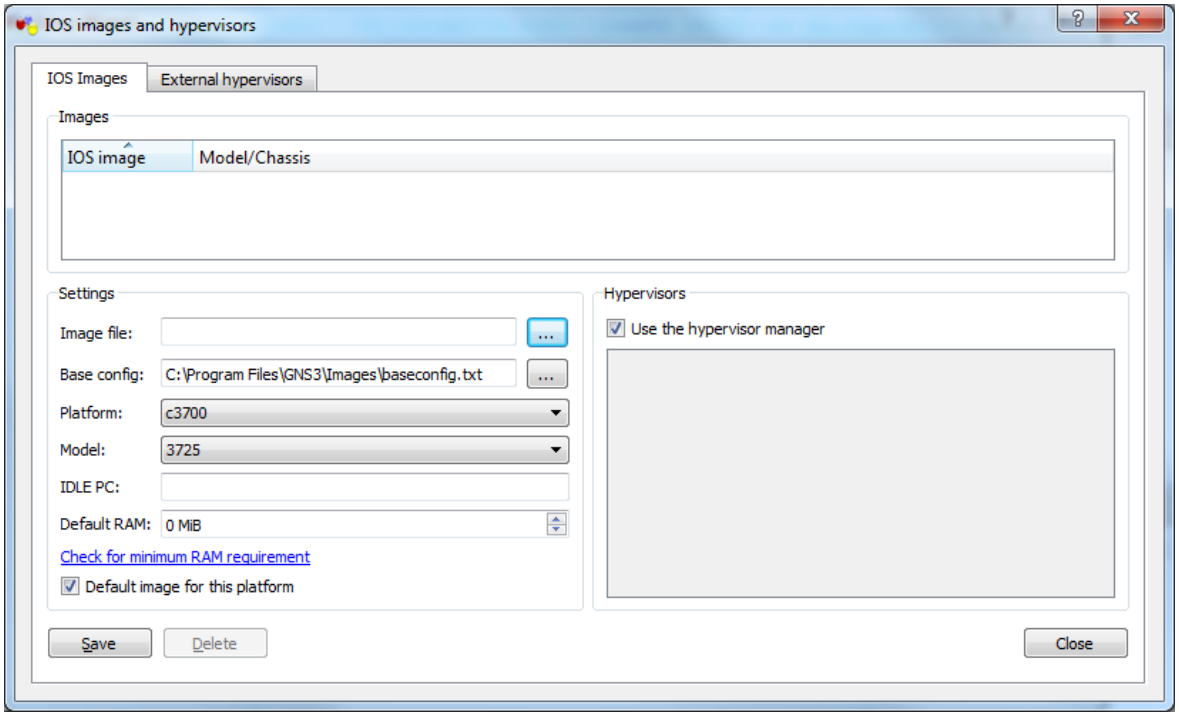


Figura 6. Pantalla de selección de imágenes IOS

3.3. El entorno gráfico de GNS3.

Como se puede apreciar en la Figura 8, la ventana principal de GNS3 se divide en cuatro subventanas. A continuación se describen sus características.

1) En la ventana izquierda, denominada Node Types, se van a mostrar los tipos de nodos disponibles para las diversas plataformas: firewalls PIX y ASA, conmutadores Ethernet, switches ATM y Frame Relay, y por supuesto, los diversos routers de la familia de CISCO. Se puede cambiar fácilmente de tipo de nodo haciendo clic en los iconos del margen izquierdo.

2) La ventana central es el área de trabajo. Aquí es donde se crean las topologías de forma gráfica, mediante el arrastre de los diversos nodos de la ventana anterior.

3) La ventana de la derecha muestra un resumen de la topología creada, detallando tanto el nodo como el estado en el que se encuentra. También aquí se muestran las capturas activas, ya que GNS3 permite capturar el tráfico de la topología simulada utilizando Wireshark.

4) Finalmente, en el panel de la consola, se muestran los mensajes y errores del simulador Dynamips (se describirá más adelante).

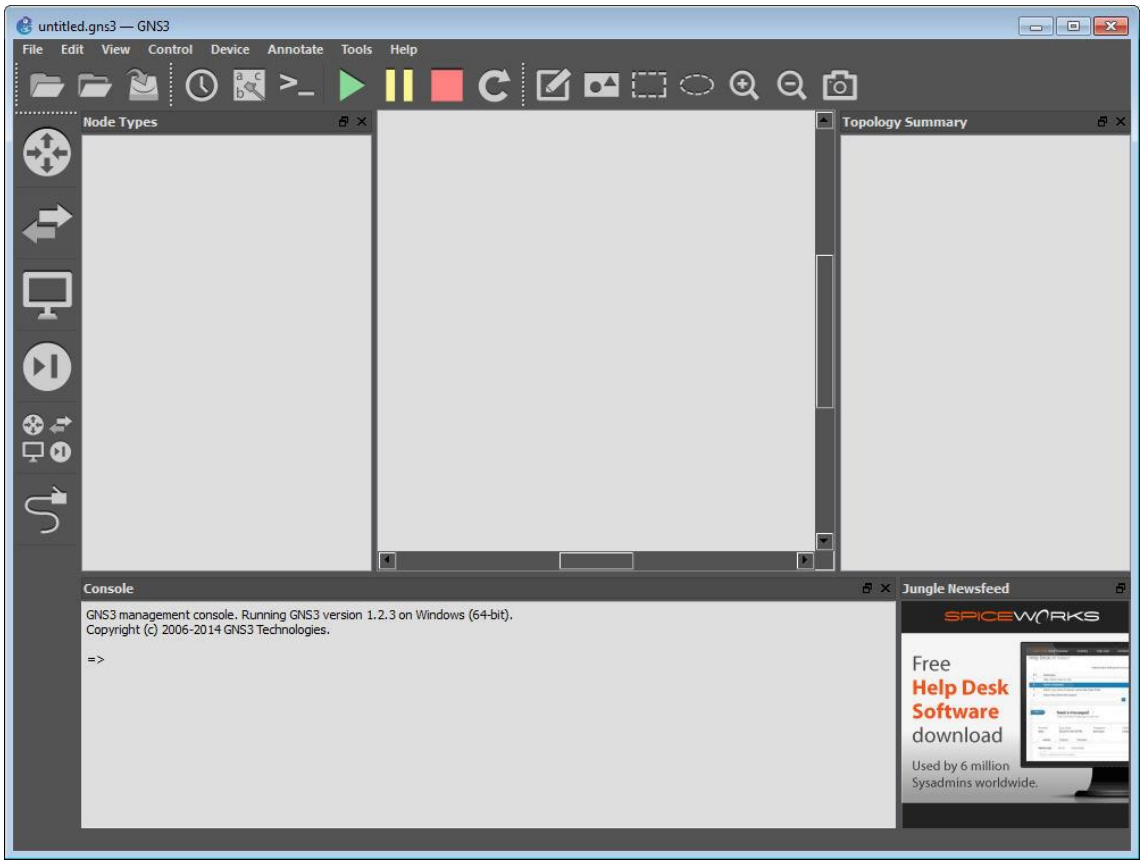


Figura 7. Pantalla principal del simulador GNS3 en Windows

3.4. Configuración de GNS3

3.4.1. Preferencias generales

Para editar las preferencias generales (ver Figura 9), seguimos la ruta: *Edit -> Preferences -> General.*

En la pestaña de "General Settings", se configuran las siguientes rutas:

- *My projects:* En este directorio se irán almacenando los proyectos que se vayan creando.
- *My binary images:* En este directorio se almacenan las imágenes de los nodos.
- *Temporary files:* Localización de los archivos temporales.

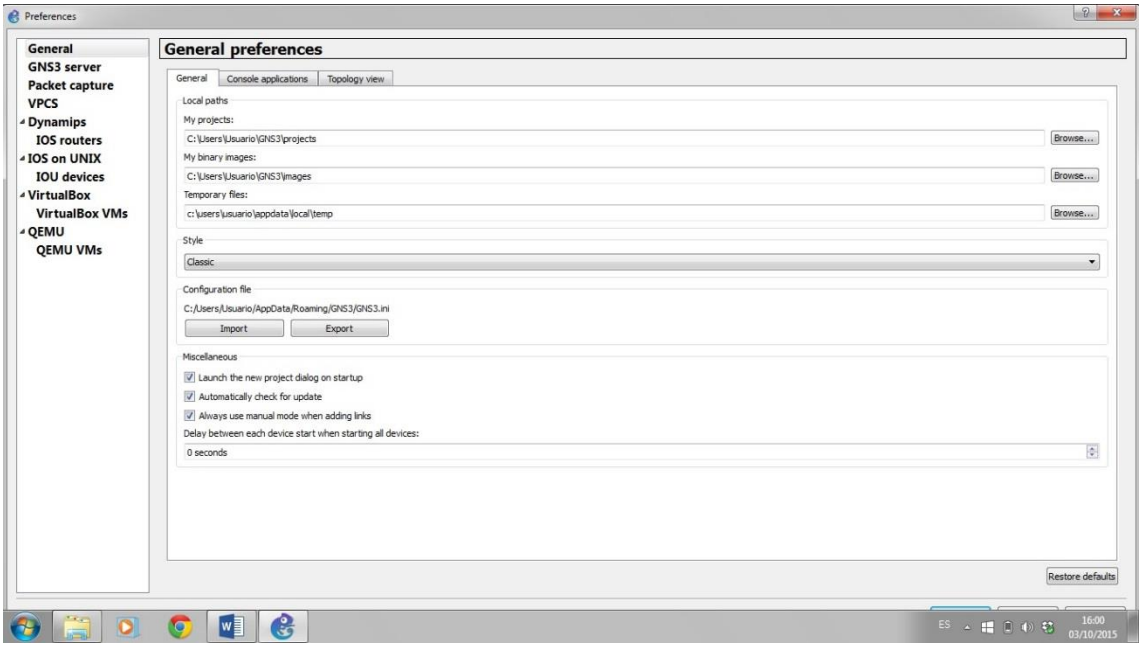


Figura 8. Configuración general de GNS3.

Si se marca la opción "*Always use manual mode when adding links*" en la pestaña *Miscellaneous* del apartado *General*, podremos asignar de forma manual las interfaces de los routers cuando creemos un nuevo enlace. Esto nos va a permitir mayor control a la hora de crear topologías complejas.

3.4.2 Preferencias de Dynamips.

Para establecer las preferencias de Dynamips, tenemos que pulsar la pestaña correspondiente en el panel de la izquierda y configurar las siguientes opciones:

- *Executable path to Dynamips:* En esta ruta está la localización del archivo binario de Dynamips. Se tiene que pulsar "*Apply*".
- Pulsar "*Test settings*" para comprobar que Dynamips funciona. Tras unos segundos debe aparecer un mensaje indicando el resultado del test. La Figura 13 muestra un ejemplo.

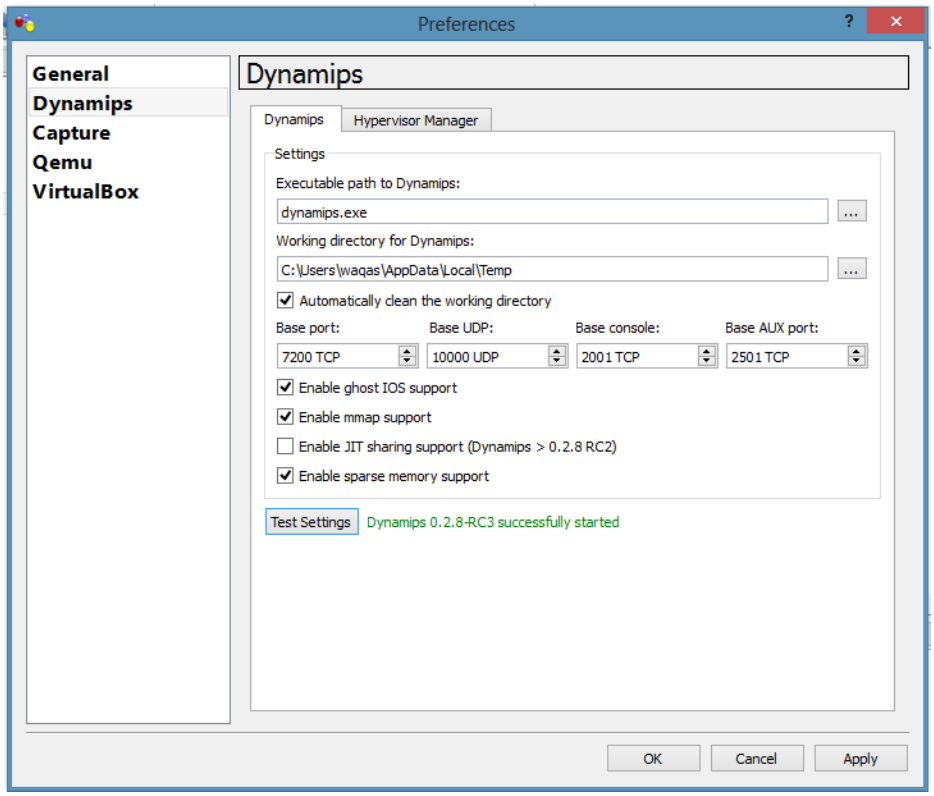


Figura 9. Configuración de los parámetros de Dynamips.

3.5. Dynamips y Dynagen

3.5.1. Dynamips

Dynamips **[6]** es el motor de emulación utilizado en GNS3 para la simulación de dispositivos CISCO. Entre las plataformas hardware que puede emular se encuentran las familias de routers 1700, 2600, 2691, 3620, 3640, 3660 y 7200.

Para efectuar las emulaciones, Dynamips utiliza una gran cantidad de RAM y de CPU. En otras palabras, si se intenta emular una imagen IOS de un router que requiere 256 MB de RAM, el set de memoria colocará 256 MB de RAM en la instancia virtual del router (Lo hará trabajar como un router real a 256 MB). Por defecto, también asigna 16 MB en sistemas Windows para manejar la memoria caché y las transacciones JIT. Para asignar la memoria virtual de los routers, Dynamips utiliza archivos mapeados de memoria. Esto hará que el sistema operativo almacene en caché en la RAM las secciones de los archivos .mmap que se estén utilizando.

El hecho de que además, Dynamips tenga un consumo tan alto de CPU, es porque simula la CPU del router "instrucción-a-instrucción". Para tratar de paliar esto, existen 2 métodos de optimización de recursos: el Idle-PC y las opciones de memoria.

3.5.2. Optimización de recursos. El Idle-PC

Para optimizar la CPU, GNS3 y Dynamips utilizan el parámetro Idle-PC, que consiste en un valor que permite disminuir la ocupación de la CPU, durmiendo los procesos dynamips mientras que no se requiera que estén activos, y despertándolos y pasándolos a la CPU cuando la ejecución de la imagen IOS así lo requiere.

Los valores son calculados por GNS3, si bien existe un valor por defecto que depende de cada versión del programa utilizado.

Para calcular un valor de Idle-PC, se debe pinchar con el botón derecho del ratón sobre el router en cuestión, y seleccionar la opción "*Calculate Idle-PC*". Después de unos instantes, se mostrará una ventana con los valores calculados. Los mejores valores potenciales se señalan con un asterisco. Seleccionar uno y pulsar "*Apply*".

Posteriormente, comprobaremos el consumo de CPU mediante el panel de control o el administrador de tareas de Windows.

Los valores que se calculan son particulares de la imagen IOS, sistema operativo, PC o hasta la versión de Dynamips o GNS3. Puede ocurrir que GNS3 no encuentre un valor que reduzca el consumo de CPU. Si esto ocurre, se puede volver a repetir el proceso.

3.5.3. Optimización de recursos. Memoria.

En topologías complejas, también la asignación de memoria puede llegar a suponer un quebradero de cabeza bastante serio. Para reducir el consumo de memoria, Dynamips ofrece las opciones *ghostIOS* y *Sparemem*. La primera opción permite utilizar una memoria compartida por todas las IOS en vez de asignar una individual a cada imagen. La segunda opción permite reservar la memoria que realmente utiliza la IOS en vez de la memoria que se ha configurado.

3.5.4. Dynagen

Finalmente, Dynagen es el cliente basado en texto de Dynamips, y se comunica con él a través del modo "*Hypervisor*". Este cliente permite construir y trabajar con redes más simples, ya que utiliza una configuración fácil de comprender para obtener los archivos de configuraciones específicas del router. Posee además una sintaxis simple que permite interconectar prácticamente cualquier tipo de nodo, sin tener que tratar con NetIOS. Otra de sus ventajosas características es que permite trabajar en modo cliente/servidor, pudiendo controlar aparte múltiples servidores Dynamips de manera simultánea. Por último, también provee una interfaz CLI para la gestión de dispositivos.

3.6. Creación de topologías en GNS3

3.6.1. Elementos de un router

Por norma general, los routers son dispositivos fijos en su configuración hardware en su gama más baja, y su flexibilidad y modularidad en las tarjetas que pueden soportar aumentan conforme vamos subiendo en la gama alta. Los routers que vamos a usar en este proyecto son routers c2600 de gama alta, dado la intensa variedad de servicios a los que son capaces de dar soporte (según la página oficial de CISCO). Cada router dispone además de una procesadora, de la cual se encuentran varios modelos en función de la cantidad de paquetes por segundo que son capaces de procesar, desde la más lenta NPE-100 hasta el ultimísimo modelo NPE-G2 (capaz de conmutar 2 millones de paquetes por segundo). La Figura 10 muestra un ejemplo sencillo de una topología.

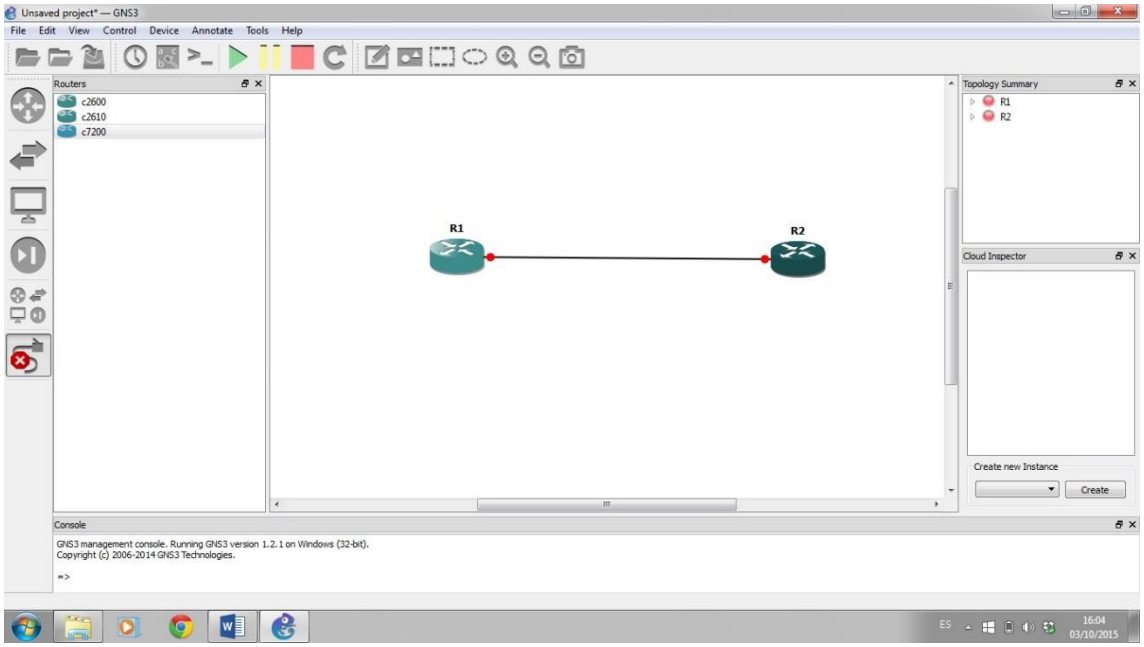


Figura 10. Ejemplo de una topología en GNS3.

A la hora de simular, en nuestros routers utilizaremos la procesadora NPE-400, ya que es la más estable al tener menos bugs reconocidos que otras, y para permitir el uso de adaptadores de tipo Ethernet. Así dispondremos de los siguientes puertos:

- En el slot 0: c2600-IO-FE -> 1 interfaz FastEthernet que vendría referenciada como FE0/0. También hay 2 interfaces del tipo c2600-IO-2FE, que en este caso se referencian como FE0/0 y FE0/1
- En el slot 1: PA-GE -> 1 interfaz GigaEthernet, que se referencia como GE1/0. También se encuentran 2 interfaces FastEthernet del tipo PA-2FE-TX, que se referencian como FE1/0 y FE1/1.

3.6.2. Creación de nodos

Para crear un nodo, simplemente basta con elegir el tipo c2600 de la ventana de *Node Types* y arrastrarlo al área de trabajo, con lo que se mostrará en pantalla. Para acceder a la configuración basta con hacer clic con el botón derecho sobre el nodo.

En la pantalla que aparece se elige la procesadora NPE-400. Por último se pincha en la pestaña *Slots* y configurar el nodo según los puertos requeridos. Como mínimo, se recomienda incluir 4 puertos Ethernet.

3.6.3. Creación de enlaces

Para crear un enlace, hay que pulsar sobre el icono del cable RJ-45 que aparece a la izquierda de la ventana de *Node Types*. El puntero del ratón cambiará a una cruz. Seleccionamos el primer router que formará parte del enlace, y aparecerá un menú donde tenemos que elegir la interfaz que queremos utilizar. Hacemos lo propio con el segundo router y la interfaz correspondiente. Para salir del modo enlace, basta con volver a pulsar el icono del RJ-45.

3.6.4. Arranque y parada del router

Si se quiere arrancar o parar la ejecución de la topología al completo, basta con utilizar los botones de *Play, Pause o Stop* que aparecen en la parte superior de la ventana. Si se quiere actuar sobre un router concreto, basta con abrir el menú de configuración haciendo clic con el botón derecho y pulsar sobre la acción deseada.

3.6.5. Conexión con el router

Para abrir la Command Line Interface (CLI) de un router sólo hay que pulsar sobre él una vez se haya arrancado. La Command Line Interface es básicamente, el terminal de consola del router desde el cual introducimos los comandos para configurarlo. Este es un procedimiento rápido, pero a veces es recomendable hacer una conexión manual para controlar mejor la interfaz de ventanas del terminal desde el que se efectúa la conexión. El comando correspondiente es:

telnet 127.0.0.1 <puerto de consola>

El puerto de consola se sabe con pasar lentamente el ratón por encima del router, tras lo que aparecerá una etiqueta con las características del router, entre ellas el puerto de consola correspondiente. No es posible establecer dos accesos simultáneos en una consola.

3.7. Captura de tráfico con Wireshark

Mediante GNS3 se puede capturar tráfico circulante en los enlaces virtuales de la topología utilizando el programa Wireshark. Para ello, basta con hacer clic con el botón derecho sobre el enlace para iniciar la captura, generar el tráfico que resulte de interés y en la ventana de capturas de la interfaz gráfica hacer clic con el botón derecho para arrancar el programa y mostrar los paquetes capturados.

Puede encontrarse más información adicional en el enlace de CISCO disponible en la bibliografía.

CAPÍTULO 4: SIMULACIÓN DE LA RED EN LA HERRAMIENTA GNS3

Una vez explicado en qué consiste el protocolo MPLS y descrita la herramienta GNS3, toca entrar en materia y comenzar con la simulación.

Utilizaremos la siguiente topología, similar a la desarrollada en **[8]**:

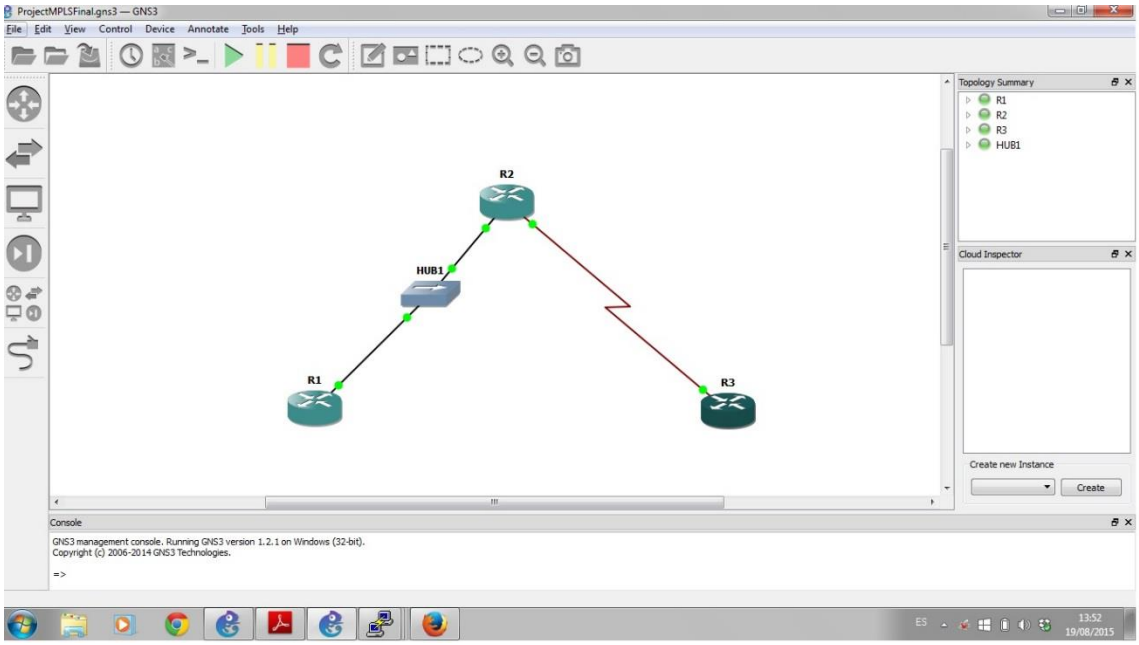


Figura 11. Maqueta de la red que se simulará en la herramienta GNS3 y de la que se hará montaje posterior.

4.1. Actividades a realizar en el montaje de la maqueta.

Para la simulación de esta maqueta, se realizarán las siguientes actividades:

a) Configurar las IP en las interfaces de los routers mediante Routing OSPF.
b) Configuración del protocolo de distribución de etiquetas.
c) Adaptar el tamaño del MTU a los requisitos que especifica MPLS.
d) Verificar el comportamiento esperado de MPLS en la red, así como las tablas especificadas en su funcionamiento.

En la simulación en la herramienta, los routers que vamos a utilizar son de la clase c2600 que implementan el software IOS **"c2600-telco-mz.123-26.bin".**

Realizando una búsqueda por internet es posible descargar casi cualquier IOS que necesitemos, e incluso, poder utilizarla (más adelante veremos cómo) en los equipos reales.

4.2. Paso 1: Inicio de la herramienta GNS3 en Windows 7.

Estamos utilizando Windows como SO, por lo que es importante reseñar dos cosas al arrancar GNS3:

- Se deberá abrir el programa haciendo clic con el botón derecho en el icono del mismo, y eligiendo la opción "Ejecutar como administrador".
- Antes de iniciar al programa, conviene ir a la carpeta C:/Archivos de programa/GNS3 (o la ruta donde se encuentre la misma), y abrir, ejecutando como administrador, el ejecutable GNS3server.exe.

Siguiendo estos dos pasos, evitaremos un error que denomino "La S Roja de la muerte" **[9]**, que es exclusivo del programa en Windows y que impide que se pueda establecer conexión con los routers para configurarlos o incluso encenderlos. Las experiencias que tuve con el programa en Linux en asignaturas de la carrera, no demostraban este error. Otro error reseñable en Windows (y para el que de momento no existe, o no he encontrado solución), es la imposibilidad de poder configurar Hosts, por lo que, hasta que exista solución, sólo se pueden efectuar montajes y simulaciones que sólo involucren Routers y Switches.

Como último dato, la versión de GNS3 que estamos utilizando es la 1.2.1, si bien actualmente puede descargarse la 1.3.9. Para el montaje y cableado de la maqueta en el simulador, basta proceder como se explican en el apartado 6 del capítulo 2 "Creación de topologías".

4.3. Paso 2: Configuración del direccionamiento IP

Una vez tenemos montada y cableada la maqueta, el siguiente paso es la configuración del direccionamiento IP en las interfaces. Para ello, pinchamos con el botón derecho sobre el router que queremos configurar, y en el menú desplegable que aparece elegimos la opción "Console". Aparecerá un terminal de consola con el que podremos trabajar para establecer la configuración del router.

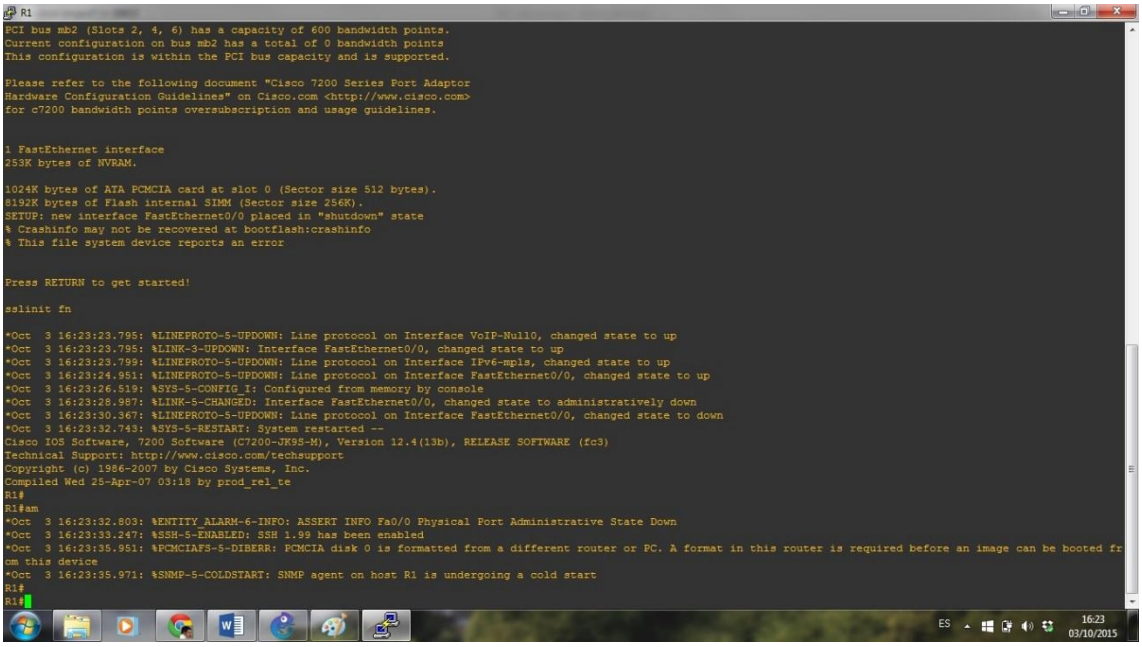


Figura 12. Ejemplo de terminal de consola del router en GNS3

Antes de introducir cualquier comando, es útil introducir lo siguiente:

```
Router>enable
Router#configure terminal
Router(config)#hostname R1
R1(config)#no ip domain-lookup
R1(config)#exit
```

Con el comando resaltado en negrita, evitaremos que al introducir cualquier comando erróneo, se consulte al DNS ya que lo interpreta como un nombre a buscar. Con ello, ahorraremos el tiempo de espera que supone el que la operación finalice.

Los tres routers se configuran como sigue a continuación:

a) Router R1:

```
R1(config)# interface loopback 0
R1(config-if)# ip address 172.16.1.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# interface ethernet 0/0
R1(config-if)# ip address 172.16.12.1 255.255.255.0
R1(config-if)# no shutdown
```

b) Router R2:

```
R2(config)# interface loopback 0
R2(config-if)# ip address 172.16.2.1 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# interface ethernet 0/0
R2(config-if)# ip address 172.16.12.2 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# interface serial 0/0
R2(config-if)# ip address 172.16.23.2 255.255.255.0
R2(config-if)# clockrate 64000
R2(config-if)# no shutdown
```

c) Router R3:

```
R3(config)# interface loopback 0
R3(config-if)# ip address 172.16.3.1 255.255.255.0
R3(config-if)# no shutdown
R3(config-if)# interface serial 0/0
R3(config-if)# ip address 172.16.23.3 255.255.255.0
R3(config-if)# clockrate 64000
R3(config-if)# no shutdown
```

4.4. Paso 3: Configuración de OSPF en los routers

Para la configuración de OSPF en los routers, nos basta simplemente con configurar la clase mayor de las subredes que estamos utilizando, en el área 0. En nuestra maqueta, estamos utilizando subredes del tipo 172.16.0.0/24, con lo que los comandos que hemos de introducir en cada router son:

```
R1(config)# router ospf 1
R1(config-router)# network 172.16.0.0 0.0.255.255 area 0
R2(config)# router ospf 1
R2(config-router)# network 172.16.0.0 0.0.255.255 area 0
R3(config)# router ospf 1
R3(config-router)# network 172.16.0.0 0.0.255.255 area 0
```

Como se ve en las siguientes imágenes, se establecen las adyacencias correspondientes y las redes se anuncian. Este punto es de vital importancia, ya que sin una adecuada conectividad IP, MPLS **NO** puede funcionar.

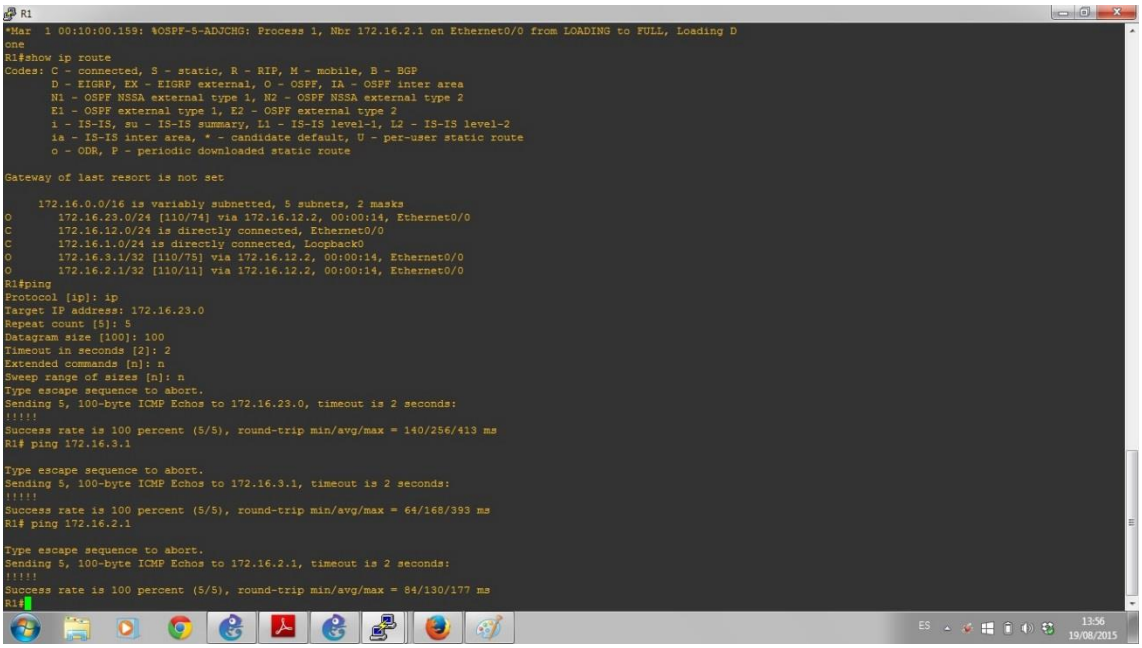


Figura 13. Adyacencias de red del router R1 tras implementar OSPF.

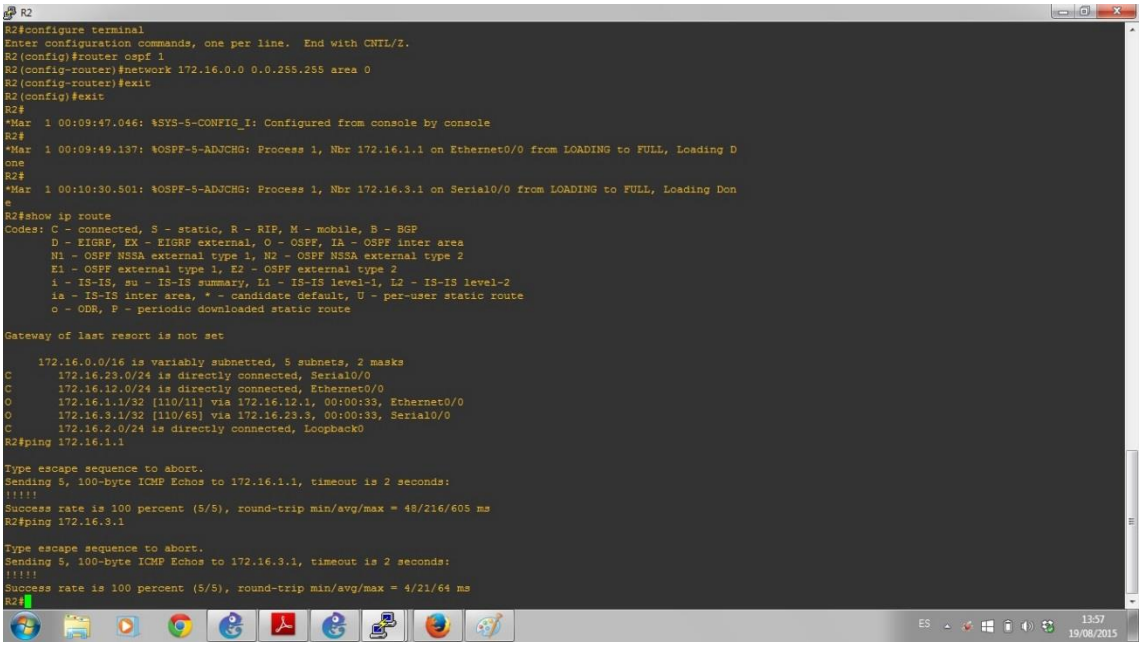


Figura 14. Adyacencias de red del router R2 tras implementar OSPF

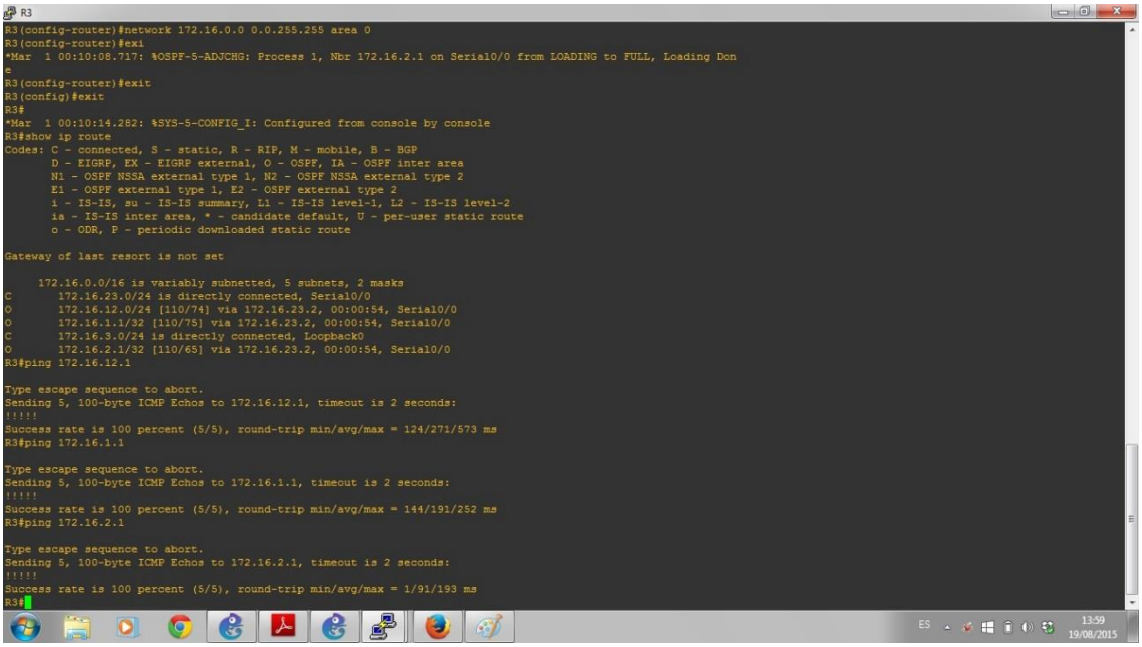


Figura 15. Adyacencias de red del router R3 tras implementar OSPF

4.5. Paso 4: Comprobación del funcionamiento de CEF (Cisco Express Forwarding)

En este momento, si hiciéramos un ping extendido en todos los routers, quedaría comprobado que existe conectividad total, pues los routers sabrían como encaminar el tráfico según sus tablas de encaminamiento, y no habría ninguna interfaz que no respondiera el ping.

Podemos, por ejemplo, observar el camino que seguiría el tráfico entre los routers R1 y R3 mediante el comando **"traceroute"**

```
R1#traceroute 172.16.3.1

Type escape sequence to abort.
Tracing the route to 172.16.3.1

 172.16.12.2 208 msec 113 msec 384 msec
 172.16.23.3 136 msec 125 msec 128 msec


R3#traceroute 172.16.1.1

Type escape sequence to abort.
Tracing the route to 172.16.1.1

172.16.23.2 213 msec 252 msec 148 msec
172.16.12.1 301 msec 328 msec 253 msec
```

Y mediante el siguiente comando, vemos si CEF está o no activado: **"show ip cef"**. Un ejemplo de lo que se ve en el Router R1:

```
Prefix              Next Hop            Interface
0.0.0.0/0           drop                Null0 (handler entry)
0.0.0.0/32          receive
172.16.1.0/24       attached            Loopback0
172.16.1.0/32       receive
172.16.1.1/32       receive
172.16.1.255/32     receive
172.16.2.1/32       172.16.12.2         Ethernet0/0
172.16.3.1/32       172.16.12.2         Ethernet0/0
172.16.12.0/24      attached            Ethernet0/0
172.16.12.0/32      receive
172.16.12.1/32      receive
172.16.12.2/32      172.16.12.2         Ethernet0/0
172.16.12.255/32    receive
172.16.23.0/24      172.16.12.2         Ethernet0/0
224.0.0.0/4         drop
224.0.0.0/24        receive
255.255.255.255/32  receive
```

La función que tiene CEF **[11]** es el poder asociar etiquetas de una dirección IP, a una interfaz física. En esta etiqueta se incluye información de la capa 2 para permitir el reenvío del paquete según indique la tabla de encaminamiento del router. Por consiguiente, esta tabla es la que utiliza el software de CISCO para poder implementar el protocolo MPLS, utilizando la información que le da.

4.6. Paso 5: Configuración de MPLS

Una vez tenemos habilitado y funcionando el direccionamiento IP, toca configurar el protocolo MPLS. Para ello, introduciremos el comando **"mpls ip"** sólo en aquellas interfaces físicas (no en las loopback).

De esta manera, se le indica al router que conmute en entrada y salida las tramas de tráfico MPLS que reciba o envíe, así como la detección de routers vecinos que también tengan el protocolo de distribución de etiquetas.

Tras configurarlo, en los terminales de los routers deberá aparecer lo siguiente:

a) Router R1:

```
R1(config)#interface ethernet 0/0
R1(config-if)#mpls ip
R1(config-if)#exit
```

b) Router R2:

```
R2(config)#interface ethernet 0/0
R2(config-if)#mpls ip
R2(config-if)#inter
*Mar    1  01:37:16.451:  %LDP-5-NBRCHG:  TDP  Neighbor
172.16.1.1:0 is UP
R2(config-if)#interface serial 0/0
R2(config-if)#mpls ip
```

c) Router R3:

```
R3(config)#interface serial 0/0
R3(config-if)#mpls ip
R3(config-if)#
*Mar    1  01:42:50.795:  %LDP-5-NBRCHG:  TDP  Neighbor
172.16.2.1:0 is UP
```

Como se puede ver, cuando en los dos extremos de una conexión se activa MPLS, aparece un mensaje indicándonos que se ha establecido vecindad, y sirve como comprobación de que la red está funcionando correctamente.

4.7. Paso 6: Verificación del funcionamiento de MPLS

Para verificar que MPLS funciona correctamente, vamos ahora a utilizar los comandos **"show"** disponibles para MPLS. Para saber cuáles son, introducimos en el terminal **"show mpls ?".** Destacar además, que este comando sirve como comprobación acerca de si el router puede o no implementar MPLS. Si aparece en la pantalla **"unrecognized command"**, el software que tiene ese router no implementa MPLS y siempre que se pueda, deberemos cambiarlo por otro (En el capítulo 4, se explica cómo).

Los comandos que aparecen son los siguientes:

```
R1#show mpls?
atm-ldp                      ATM LDP Protocol information
forwarding-table  Show the Label Forwarding Information Base(LFIB)
interfaces              Per-interface MPLS forwarding information
ip                                MPLS IP information
label                             Label information
ldp                     Label Distribution Protocol information
traffic-eng                Traffic engineering information
```

Vamos a verificar qué interfaces tienen implementado MPLS con **"show MPLS interfaces"**

```
R1#show mpls interfaces
Interface              IP            Tunnel   Operational
Ethernet0/0         Yes (tdp)          No         Yes


R2#show mpls interfaces
Interface              IP            Tunnel   Operational
Ethernet0/0          Yes (tdp)          No        Yes
Serial0/0            Yes (tdp)          No        Yes


R3#show mpls interfaces
Interface              IP            Tunnel   Operational
Serial0/0            Yes (tdp)          No         Yes
```

Según se observa de estos resultados, las interfaces que implementan MPLS están operativas e implementan el protocolo TDP para intercambiar etiquetas. Este TDP (Tag Distribution Protocol) no es más que una versión precursora del protocolo LDP (Label Distribution Protocol) que implementan actualmente todos los dispositivos MPLS. Pero en la práctica, no hay ninguna diferencia entre usar uno o usar otro, más allá de usar LDP para buscar compatibilidad con otros fabricantes. Si quisiéramos cambiar a LDP, eso sí, tendríamos que hacerlo en todos los routers de la red, o se perdería la conectividad MPLS.

Otros comandos con los que vamos a verificar el funcionamiento de MPLS son **"show mpls ldp Discovery"** y **"show mpls ldp neighbor".** Con el primer comando, se obtiene toda la información relativa de TDP (LDP), como los identificativos del router MPLS y sus vecinos, y con el segundo, se detectan las adyacencias de la red.

Éste es el resultado obtenido en la maqueta:

a) **Router R1:**

```
R1#show mpls ldp discovery
 Local LDP Identifier:
    172.16.1.1:0
    Discovery Sources:
    Interfaces:
        Ethernet0/0 (tdp): xmit/recv
            TDP Id: 172.16.2.1:0


R1#show mpls ldp neighbor
    Peer TDP Ident: 172.16.2.1:0; Local TDP Ident
172.16.1.1:0
        TCP connection: 172.16.2.1.51337 -
172.16.1.1.711
```

```
  State: Oper; PIEs sent/rcvd: 0/13; Downstream
  Up time: 00:08:57
  TDP discovery sources:
    Ethernet0/0, Src IP addr: 172.16.12.2
  Addresses bound to peer TDP Ident:
    172.16.12.2     172.16.23.2     172.16.2.1
```

b) Router R2:

```
R2#show mpls ldp discovery
 Local LDP Identifier:
    172.16.2.1:0
    Discovery Sources:
    Interfaces:
        Ethernet0/0 (tdp): xmit/recv
            TDP Id: 172.16.1.1:0
        Serial0/0 (tdp): xmit/recv
            TDP Id: 172.16.3.1:0


R2#show mpls ldp neighbor
    Peer TDP Ident: 172.16.3.1:0; Local TDP Ident
172.16.2.1:0
        TCP connection: 172.16.3.1.47531 -
172.16.2.1.711
        State: Oper; PIEs sent/rcvd: 0/47; Downstream
        Up time: 00:40:05
        TDP discovery sources:
          Serial0/0, Src IP addr: 172.16.23.3
        Addresses bound to peer TDP Ident:
          172.16.23.3     172.16.3.1
    Peer TDP Ident: 172.16.1.1:0; Local TDP Ident
172.16.2.1:0
        TCP connection: 172.16.1.1.711 -
172.16.2.1.51337
        State: Oper; PIEs sent/rcvd: 0/22; Downstream
        Up time: 00:16:05
        TDP discovery sources:
          Ethernet0/0, Src IP addr: 172.16.12.1
        Addresses bound to peer TDP Ident:
          172.16.12.1    172.16.1.1
```

c) **Router R3:**

```
Local LDP Identifier:
    172.16.3.1:0
    Discovery Sources:
    Interfaces:
        Serial0/0 (tdp): xmit/recv
            TDP Id: 172.16.2.1:0

R3#show mpls ldp neighbor
    Peer TDP Ident: 172.16.2.1:0; Local TDP Ident
172.16.3.1:0
        TCP connection: 172.16.2.1.711 -
172.16.3.1.47531
        State: Oper; PIEs sent/rcvd: 0/52; Downstream
        Up time: 00:39:30
        TDP discovery sources:
          Serial0/0, Src IP addr: 172.16.23.2
        Addresses bound to peer TDP Ident:
          172.16.12.2     172.16.23.2     172.16.2.1
```

Como puede observarse, los router utilizan como identificador TDP la dirección IP de la interfaz de Loopback. Si no existiera, se utilizaría la IP más alta entre las de las interfaces físicas. Para establecer comunicación con los vecinos, TDP(LDP) utiliza el protocolo TCP, pero si es un LSR, mandará los mensajes "Hello" como parte de un paquete UDP. Con esta configuración, todos los routers quedarán establecidos como LSR, por lo que de forma definitiva, se establecerá conexión mediante el protocolo TCP.

4.8. Paso 7: Estudio de las tablas LIB y LFIB

Ahora que todos los routers han quedado configurados como LSR, va a ocurrir lo siguiente: A cada entrada de la tabla de rutas, se le va a asignar una etiqueta MPLS, y éstas se registrarán en la tabla LIB. Es importante reseñar que estas etiquetas pueden variar cada vez que el router se reinicia.

Lo que hace el protocolo TDP (LDP) es distribuir las etiquetas locales entre sus vecinos, para que éstos las utilicen cada vez que mandan tráfico a un destino específico mediante el LSR que asigna las etiquetas. Con las etiquetas distribuidas, la conmutación se realiza mirando la tabla LFIB, que almacena la etiqueta asignada por el vecino, la interfaz por donde enviar la trama MPLS, y la acción que debe realizar con la etiqueta añadida.

Que una etiqueta se asocie a un destino en la tabla local de rutas, sólo tiene significado para el router. Esto es, las etiquetas asignadas por un LSR a un destino, no tienen nada que ver con que otro LSR asigne etiquetas al mismo destino. Un LSR

puede asignar a un destino la etiqueta 16 y otro LSR asignar también la etiqueta 16 a ese mismo destino.

En las siguientes figuras se ilustra el funcionamiento de estas tablas

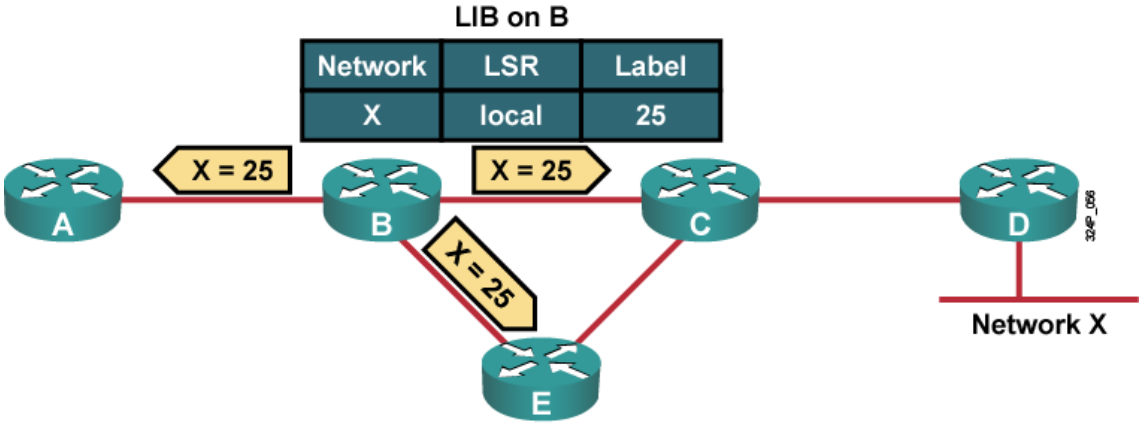


Figura 16: *En esta figura se observa que la red X (network X) es anunciada por el router D y el router B la tiene en su tabla de rutas. Para ese destino (red X), el router B elige una etiqueta, en concreto la 25 y envía su decisión a los routers vecinos A, E y C respectivamente por LDP. La asociación realizada queda registrada en la tabla LIB del router B. Ahora, cuando el router A tenga que enviar a la red X, el router A encapsulará el paquete dirigido a X en una trama MPLS con etiqueta 25, dado que B sabe qué hacer con dicha etiqueta.* **[8]**

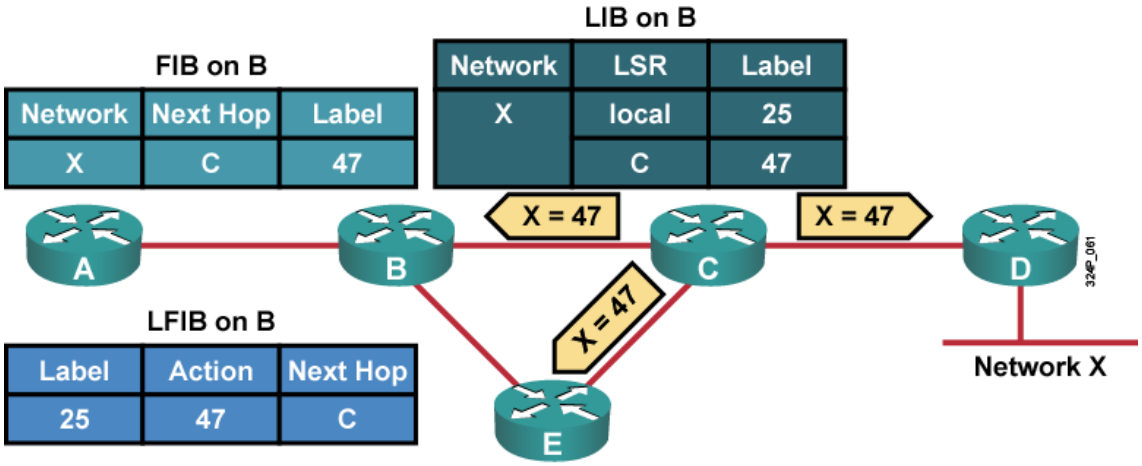


Figura 17: *Una vez tenemos las tablas inicializadas, FIB, LIB y LFIB en el router B, cuando llegue un paquete del router A con etiqueta 25, el router B sabe que tiene que cambiar la etiqueta a 47 consultando la tabla LFIB y conmutar, es decir sacarla por la interfaz que conecta con el router C.* **[8]**

Para visualizar los datos de la tabla LIB, se utiliza el siguiente comando: **"show mpls ldp bindings"**. Las tablas obtenidas en cada router se listan a continuación

a) **Router R1:**

```
R1#show mpls ldp bindings
  tib entry: 172.16.1.0/24, rev 6
        local binding:  tag: imp-null
  tib entry: 172.16.1.1/32, rev 13
        remote binding: tsr: 172.16.2.1:0, tag: 16
  tib entry: 172.16.2.0/24, rev 14
        remote binding: tsr: 172.16.2.1:0, tag: imp-null
  tib entry: 172.16.2.1/32, rev 10
        local binding:  tag: 18
  tib entry: 172.16.3.1/32, rev 8
        local binding:  tag: 17
        remote binding: tsr: 172.16.2.1:0, tag: 17
  tib entry: 172.16.12.0/24, rev 4
        local binding:  tag: imp-null
        remote binding: tsr: 172.16.2.1:0, tag: imp-null
  tib entry: 172.16.23.0/24, rev 2
        local binding:  tag: 16
        remote binding: tsr: 172.16.2.1:0, tag: imp-null
```

b) **Router R2:**

```
R2#show mpls ldp bindings
  tib entry: 172.16.1.0/24, rev 14
        remote binding: tsr: 172.16.1.1:0, tag: imp-null
  tib entry: 172.16.1.1/32, rev 6
        local binding:  tag: 16
        remote binding: tsr: 172.16.3.1:0, tag: 17
  tib entry: 172.16.2.0/24, rev 10
        local binding:  tag: imp-null
  tib entry: 172.16.2.1/32, rev 12
        remote binding: tsr: 172.16.3.1:0, tag: 18
        remote binding: tsr: 172.16.1.1:0, tag: 18
  tib entry: 172.16.3.0/24, rev 13
        remote binding: tsr: 172.16.3.1:0, tag: imp-null
  tib entry: 172.16.3.1/32, rev 8
        local binding:  tag: 17
        remote binding: tsr: 172.16.1.1:0, tag: 17
  tib entry: 172.16.12.0/24, rev 4
        local binding:  tag: imp-null
        remote binding: tsr: 172.16.3.1:0, tag: 16
        remote binding: tsr: 172.16.1.1:0, tag: imp-null
  tib entry: 172.16.23.0/24, rev 2
        local binding:  tag: imp-null
        remote binding: tsr: 172.16.3.1:0, tag: imp-null
        remote binding: tsr: 172.16.1.1:0, tag: 16
```

c) **Router R3:**

```
R3#show mpls ldp bindings
  tib entry: 172.16.1.1/32, rev 6
        local binding:  tag: 17
        remote binding: tsr: 172.16.2.1:0, tag: 16
  tib entry: 172.16.2.0/24, rev 12
        remote binding: tsr: 172.16.2.1:0, tag: imp-null
  tib entry: 172.16.2.1/32, rev 10
        local binding:  tag: 18
  tib entry: 172.16.3.0/24, rev 8
        local binding:  tag: imp-null
  tib entry: 172.16.3.1/32, rev 11
        remote binding: tsr: 172.16.2.1:0, tag: 17
  tib entry: 172.16.12.0/24, rev 4
        local binding:  tag: 16
        remote binding: tsr: 172.16.2.1:0, tag: imp-null
  tib entry: 172.16.23.0/24, rev 2
        local binding:  tag: imp-null
        remote binding: tsr: 172.16.2.1:0, tag: imp-null
```

En estas tablas, se puede apreciar que hay interfaces a las que se le asigna una etiqueta "imp-null" en vez de un número. Lo que hace esta etiqueta es enviar el paquete directamente con prefijo de red IP y no con etiqueta MPLS. Esto ocurre cuando las redes están directamente conectadas.

Por otra parte, las tramas MPLS se entregan en los routers Cisco mediante el sistema PHP, que consiste en que cuando el LSR tiene el destino directamente conectado, entrega sin más el paquete IP. Así se evitan consultas innecesarias en la tabla.

El siguiente ejemplo es ilustrativo de cómo funcionan estos dos sistemas. Si asumimos que todos los routers han realizado adyacencia con TDP (o LDP), entonces, sucede lo siguiente al ejecutar MPLS:

1) R2 asocia etiquetas localmente, por ejemplo la 17, para el prefijo 172.16.3.0/24 de su tabla de rutas

2) R2 anuncia por LDP (o TDP) la asociación local a su vecino R1.

3) R1 introduce la asociación de R2 para la red 172.16.3.0/24, clasificándola como asignación remota en su LIB, independientemente de si la utiliza para alcanzar dicha red. La asignación remota para dicha red a través de R2 es la etiqueta 17.

4) Basándose en la tabla de rutas, R2 utilizará R3 como siguiente salto para la red 172.16.3.0/24. R2 no reenviará los paquetes IP en MPLS porque R3 ha anunciado la red con la etiqueta implicit-NULL a R2. Este modo de operar se llama PHP.

En resumidas cuentas, lo que aquí se viene a decir es que MPLS asignará más de una etiqueta a un mismo destino ya que cada router asocia de forma local una etiqueta a un destino, y alerta a todos los vecinos de la etiqueta que ha enviado.

Por otra parte, la tabla LFIB se puede consultar con el comando: **"show mpls forwarding-table"**

R1#show mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	Pop tag	172.16.23.0/24	0	Et0/0	172.16.12.2
17	17	172.16.3.1/32	0	Et0/0	172.16.12.2
18	Untagged	172.16.2.1/32	0	Et0/0	172.16.12.2

R2#show mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	Untagged	172.16.1.1/32	0	Et0/0	172.16.12.1
17	Untagged	172.16.3.1/32	0	Se0/0	point2point

R3#show mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	Pop tag	172.16.12.0/24	0	Se0/0	point2point
17	16	172.16.1.1/32	0	Se0/0	point2point
18	Untagged	172.16.2.1/32	0	Se0/0	point2point

Con el comando **"traceroute"**, si ahora realizáramos un ping en R1 con destino a R3, veríamos lo siguiente:

```
R1#traceroute 172.16.3.1

Type escape sequence to abort.
Tracing the route to 172.16.3.1

172.16.12.2 [MPLS: Label 17 Exp 0] 356 msec 245 msec 352 msec
172.16.23.3 153 msec 100 msec 124 msec
```

Y si lo hacemos de R3 a R1, vemos esto:

```
R3#traceroute 172.16.1.1

Type escape sequence to abort.
Tracing the route to 172.16.1.1

172.16.23.2 [MPLS: Label 16 Exp 0] 461 msec 228 msec 80 msec
172.16.12.1 197 msec 144 msec 80 msec
```

Se observa claramente que al efectuar el ping en sentido R1-R3, el paquete sale por la interfaz cuya IP es 12.2 y le es asignada, tras mirar en la tabla LFIB la etiqueta número 17. Posteriormente, como R2 está directamente conectado con R3, el paquete es entregado sin necesidad de imponerle etiqueta. Lo mismo, pero con la etiqueta 16, se ve si efectuamos el ping en sentido contrario. Lo aquí mostrado es pues, coherente con el funcionamiento que hemos reseñado del protocolo MPLS. Esto sería extrapolable a cualquier red MPLS, pero debido al pequeño tamaño de esta maqueta, sólo es posible visualizar una etiqueta.

4.9. Paso 8: Modificación del tamaño de MTU para MPLS

Una de las características principales de MPLS es que permite añadir etiquetas MPLS en función de la aplicación a ejecutar. Esto provoca que el número de cabeceras aumente, y que haya que avisar a las interfaces de este problema para que no descarten los paquetes que excedan el MTU (En una interfaz Ethernet, dicho parámetro es de 1500 bytes). Esto es bastante útil si pensamos que una cabecera MPLS tiene 4 bytes.

Se puede comprobar el tamaño de la MTU en una interfaz mediante el comando **"show mpls interface [num int] [type int] detail"**, y los resultados que se obtienen son:

```
R1#show mpls int eth0/0 detail
Interface Ethernet0/0:
        IP labeling enabled (ldp)
        LSP Tunnel labeling not enabled
        BGP tagging not enabled
        Tagging operational
        Fast Switching Vectors:
          IP to MPLS Fast Switching Vector
          MPLS Turbo Vector
        MTU = 1500
```

```
R2#show mpls int detail
Interface Ethernet0/0:
        IP labeling enabled (ldp)
        LSP Tunnel labeling not enabled
        BGP tagging not enabled
        Tagging operational
        Fast Switching Vectors:
          IP to MPLS Fast Switching Vector
          MPLS Turbo Vector
        MTU = 1500

Interface Serial0/0:
        IP labeling enabled (ldp)
        LSP Tunnel labeling not enabled
        BGP tagging not enabled
        Tagging operational
        Fast Switching Vectors:
          IP to MPLS Fast Switching Vector
          MPLS Turbo Vector
        MTU = 1500
```

```
R3#show mpls int detail
Interface Serial0/0:
        IP labeling enabled (ldp)
        LSP Tunnel labeling not enabled
        BGP tagging not enabled
        Tagging operational
        Fast Switching Vectors:
          IP to MPLS Fast Switching Vector
          MPLS Turbo Vector
        MTU = 1500
```

Como se observa, el tamaño máximo de un paquete en un interfaz Ethernet o Serial es de 1500 bytes. Pero este parámetro puede cambiarse mediante el comando **"MPLS mtu [num bytes]"** En nuestra maqueta, bastará con establecer el tamaño a 1508 bytes. Aquí se pueden ver los resultados.

a) **Router R1:**

```
R1(config)#interface ethernet 0/0
R1(config-if)#mpls mtu 1508
R1(config-if)#exit
R1(config)#exit
R1#
*Mar  1 03:15:28.129: %SYS-5-CONFIG_I: Configured from
console by console
R1#show mpls int detail
Interface Ethernet0/0:
        IP labeling enabled (ldp)
        LSP Tunnel labeling not enabled
        BGP tagging not enabled
        Tagging operational
        Fast Switching Vectors:
          IP to MPLS Fast Switching Vector
          MPLS Turbo Vector
        MTU = 1508
```

b) **Router R2:**

```
R2#configure terminal
Enter configuration commands, one per line.  End with
CNTL/Z.
R2(config)#interface ethernet 0/0
R2(config-if)#mpls mtu 1508
R2(config-if)#exit
R2(config)#exit
R2#show
```

```
*Mar  1 03:16:23.913: %SYS-5-CONFIG_I: Configured from
console by consolem

R2#show mpls int detail
Interface Ethernet0/0:
        IP labeling enabled (ldp)
        LSP Tunnel labeling not enabled
        BGP tagging not enabled
        Tagging operational
        Fast Switching Vectors:
          IP to MPLS Fast Switching Vector
          MPLS Turbo Vector
        MTU = 1508
```

4.10. Análisis de las tramas MPLS

Finalmente, ya está todo preparado y configurado para el análisis de las tramas MPLS en nuestra red. Para ello, haremos uso de la herramienta analizadora de protocolos Wireshark, que se instalará por defecto cuando hagamos la instalación de GNS3.

En este proyecto, no hablaremos sobre el Wireshark, puesto que es de sobra conocido por todos.

Para iniciar la herramienta Wireshark, tenemos que situar el ratón sobre el cable del enlace que queramos analizar, pinchar sobre el mismo y seleccionar la opción *Start Capturing*. Una vez lo hayamos hecho, se deberá abrir la siguiente pantalla:

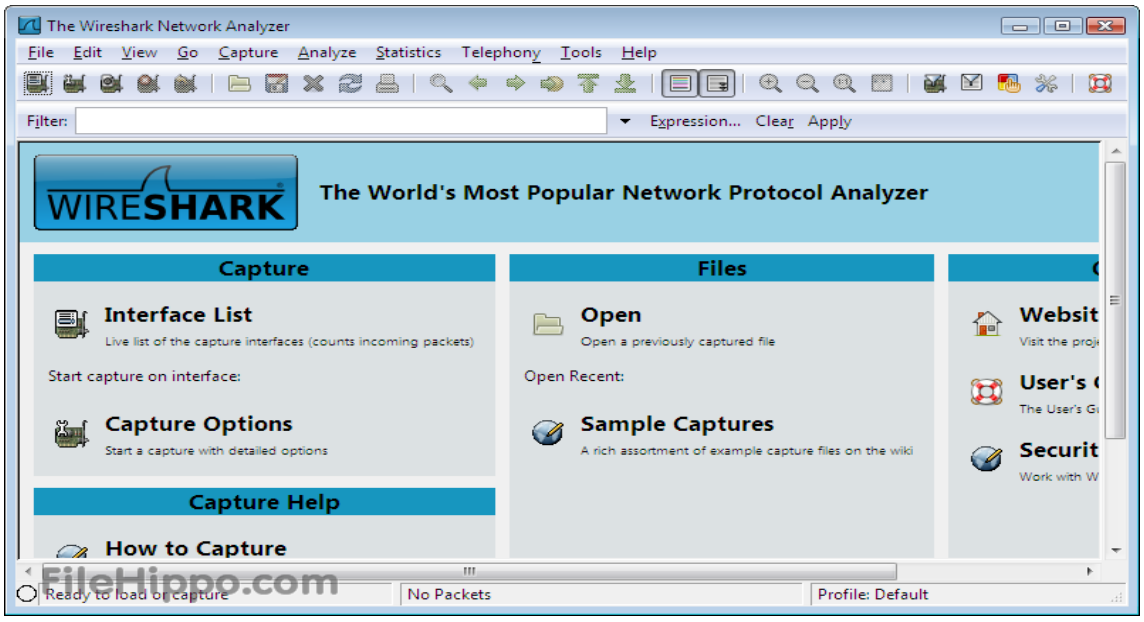


Figura 18. Pantalla de inicio de Wireshark en Windows 7.

Pinchando en la pestaña Interfaces, podremos elegir la interfaz que nos interese para analizar el tráfico. Aquí será la que vaya del Hub a R2 o del Hub a R1 (El Hub no interfiere en el tráfico).

Una vez seleccionada la interfaz en la que analizaremos el tráfico, volvemos a la consola del router y efectuamos un Ping a R3. En la pantalla deberá salir algo similar a esto :

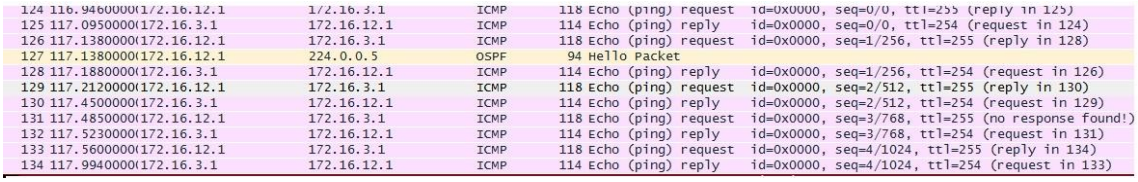


Figura 19. Captura del Ping R1-R3 en Wireshark.

La trama que vamos a analizar es la número 129, uno de los paquetes Request que envía el Router R1 para conocer la dirección. Esto es así, porque la cabecera MPLS sólo se puede ver en los paquetes Request, no en los Reply. Si extendemos el paquete, tendremos toda la información:

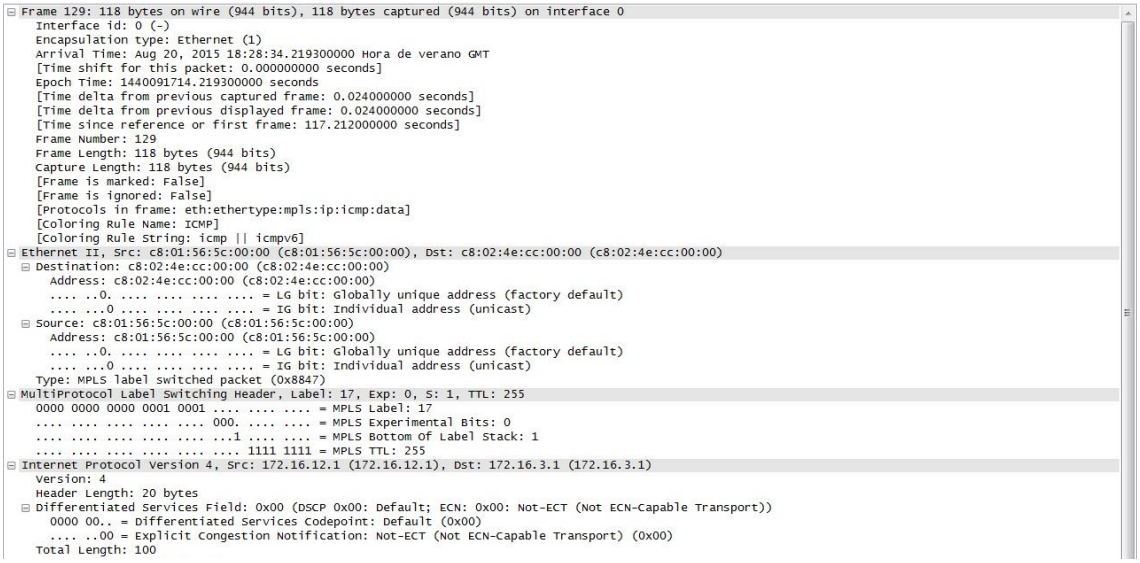


Figura 20. Información detallada del paquete Request. Se puede observar con detalle la cabecera MPLS.

Si nos fijamos en la información de la cabecera MPLS, encontramos 4 parámetros fundamentales. El primero de ellos es el campo *MPLS Label*, que indica la etiqueta que el router le ha asignado al paquete (El valor es 17). Estamos haciendo un ping del Router R1 al Router 3, y la tabla LFIB del Router 1 indica que el tráfico que vaya al Router 3 desde el Router 1 debe llevar la etiqueta… 17. Con lo que vemos que hemos configurado bien el protocolo MPLS, ya que coincide la etiqueta del paquete con la que debería llevar.

El segundo campo de la cabecera MPLS es el *Experimental Bits*, cuyo valor es 0. Es un campo de 3 bits que se utiliza para definir el grado de Calidad de Servicio que el Router debe dar al paquete. Esto es de muy alta utilidad para el caso de aplicaciones en las que se quiera implementar Servicios Diferenciados. Si, como es el caso, este campo tiene un valor 0, significa básicamente que el paquete no tiene ninguna prioridad y que tampoco ofrece QoS.

El tercer campo que vemos es el *Bottom Label Stack*, de 1 sólo bit, y que tiene valor 1. Esto significa que el paquete ha sido el último en entrar en la pila de etiquetas.

El último campo de la cabecera es el TTL o *Time To Live*, que consta de 8 bits, por lo que su valor oscila entre 0 y 255. En este caso, 255 es el TTL que tiene el paquete, que (aunque aquí no se ve) coincide con el TTL del propio paquete IP. Esto es debido a que prácticamente no existen saltos entre los routers y por tanto, el paquete llega sin problemas. Indica el tiempo que tiene el paquete para ser reenviado antes de ser descartado.

Con esto, queda finalizada la simulación en GNS3 de la maqueta, y es el momento de comprobar si todos estos valores y datos, se mantienen en una red con equipos reales.

CAPÍTULO 5: MONTAJE DE LA RED CON EQUIPOS REALES

En este capítulo se va a describir el montaje de la red con los routers reales, y algunos pasos previos, como la forma de copiar imágenes de un router a otro, que puede resultar fundamental si los equipos no incluyen el software necesario. También se responderán algunas cuestiones obviadas en el capítulo 3. No se repetirán comandos ya explicados en el capítulo anterior para evitar redundar. Sí en cambio, aparecerán imágenes que corroboren los datos más importantes.

A la hora de montar la maqueta real, debido a las limitaciones de los equipos disponibles en el laboratorio IT-3, se han hecho algunos cambios en cuanto al material empleado. Se describen en este capítulo.

5.1. Material utilizado en el montaje

. 1 Router Cisco 2611XM y 2 routers Cisco 2620XM, equipados con la imagen "**c2600-advipservicesk9-mz.123-4.T4.bin**"

. 1 cable Ethernet normal y 1 cable serie DTE-DCE.

. 3 ordenadores con el analizador de protocolos Wireshark y el programa Hyperterminal

.3 cables Serial-RJ45 para poder configurar los routers desde la consola del Hyperterminal.

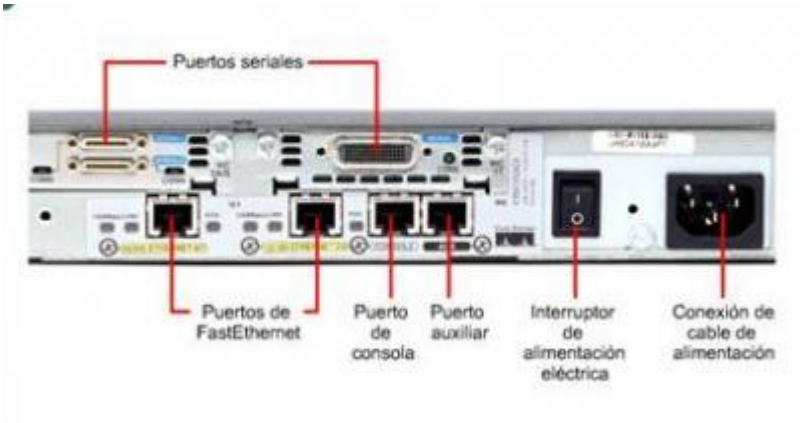


Figura 21.a. Router CISCO 2611XM con descripción de los puertos. El usado en el laboratorio sólo disponía de una tarjeta serial de 2 puertos, como la que se ve a la izquierda de la imagen.

Figura 21.b. Router CISCO2620XM. Los empleados en el laboratorio poseían una tarjeta serial de un único puerto.

Se prescindió del hub utilizado en la simulación en GNS3, en primer lugar por no encontrar ninguno en el instrumental del laboratorio, y en segundo porque realmente, el hub no introducía nada que pudiera modificar de forma sustancial la red. Las siguientes imágenes ilustran el resto del material utilizado.

Figuras 21.c, 21.d y 21.e. Ilustran los cables utilizados en el montaje. 21.c es un cable conector serie-Rj45, 21.d es un cable Ethernet y 21.e es el cable DTE-DCE que conecta los puertos serie.

5.2. Configurando los routers

Antes de empezar el montaje de la red, se debía comprobar que los routers tuvieran una imagen compatible con el protocolo MPLS, o todo el trabajo no serviría para nada. Se introdujo el comando **"show mpls ?"** en los tres routers, y se descubrió que si bien en el router 2611 funcionaba, los 2620 daban como salida **"unrecognized command"**, lo cual demostraba que su imagen no era compatible con el protocolo MPLS.

Esto llevó a tomar la decisión de cambiar el software de los mismos, puesto que no existían en el laboratorio más routers que implementaran una imagen que tuviera este protocolo.

Para cambiar la imagen IOS de un router, se siguen los siguientes pasos:

1) Comprobar la imagen y el tamaño de la memoria flash del router mediante el comando "**show flash"**. En nuestro caso, se vio que en los Routers 2620 la RAM tenía un tamaño de RAM de 128 MBytes y una memoria flash de 32 MBytes. La imagen que queríamos introducir tenía un tamaño de aproximadamente unos 24 MBytes, por lo que podía copiarse al 2620, pero antes debía de borrarse la imagen ya existente, o no habría espacio suficiente. Es importante reseñar que la imagen que queremos copiar **NO** debe exceder el tamaño de la memoria Flash del dispositivo, o no podremos copiarla ni borrando la anterior (No podemos introducir esta imagen en un dispositivo de 48 MB de RAM, y 16 MB de flash, por ejemplo).

2) Una vez comprobado que todo está en orden, procedemos a configurar el router donde está la imagen que queremos copiar como un servidor TFTP. Para ello, tendremos que introducir los siguientes comandos:

```
R1(config)# tftp server-flash: /[nombre de la imagen]
R1(config)#end
```

3) Procedemos a configurar una interfaz de red del router que actúa como servidor y una del router que actuará como clientes, asignándoles una dirección IP que estén dentro del mismo rango de red. Luego, se conectarán, bien mediante el cable Ethernet o el DTE-DCE, según la interfaz que hayamos elegido. No hay que olvidar comprobar la conectividad con un ping.

4) Cuando haya conectividad, configuramos el server cliente con el siguiente comando:

```
R2#copy tftp flash
```

Y deberá aparecer lo siguiente:

```
                        ****  NOTICE  ****
Flash load helper v1.0
This process will accept the copy options and then terminate
the current system image to use the ROM based image for the copy.
Routing functionality will not be available during that time.
If you are logged in via telnet, this connection will terminate.
Users with console access can see the results of the copy
operation.
                        ---- ******** ----
Proceed? [confirm]
Address or name of remote host [Server IP]?
Source filename [nombre imagen]?
Destination filename [nombre imagen]?
```

Pulsamos *Enter*, y entonces empezará a conectarse al Server para descargar la imagen. Si no hay suficiente espacio en la memoria Flash, nos preguntará si queremos borrar la imagen que existía. Confirmamos y entonces empezará a borrar la imagen (Lo que conllevará cierto tiempo) y a cargar la imagen que ha descargado del servidor (también llevará cierto tiempo).

5) Cuando todo el proceso haya terminado, volvemos a ejecutar el comando **"show flash"** y ahora deberá mostrar la nueva imagen que hemos copiado. Para que el router sea plenamente funcional con la nueva imagen, tendremos que reiniciarlo apagándolo y encendiéndolo.

Una vez efectuado todos estos pasos, el router ya estará totalmente operativo para nuestro propósito.

5.2.1. Conectarse al Router mediante Hyperterminal

Para poder conectarse al Router por consola, necesitaremos tener el router conectado por consola al ordenador mediante el cable Serie-RJ45, y acceder al programa HyperTerminal. Al ejecutar el programa deberíamos ver esta pantalla:

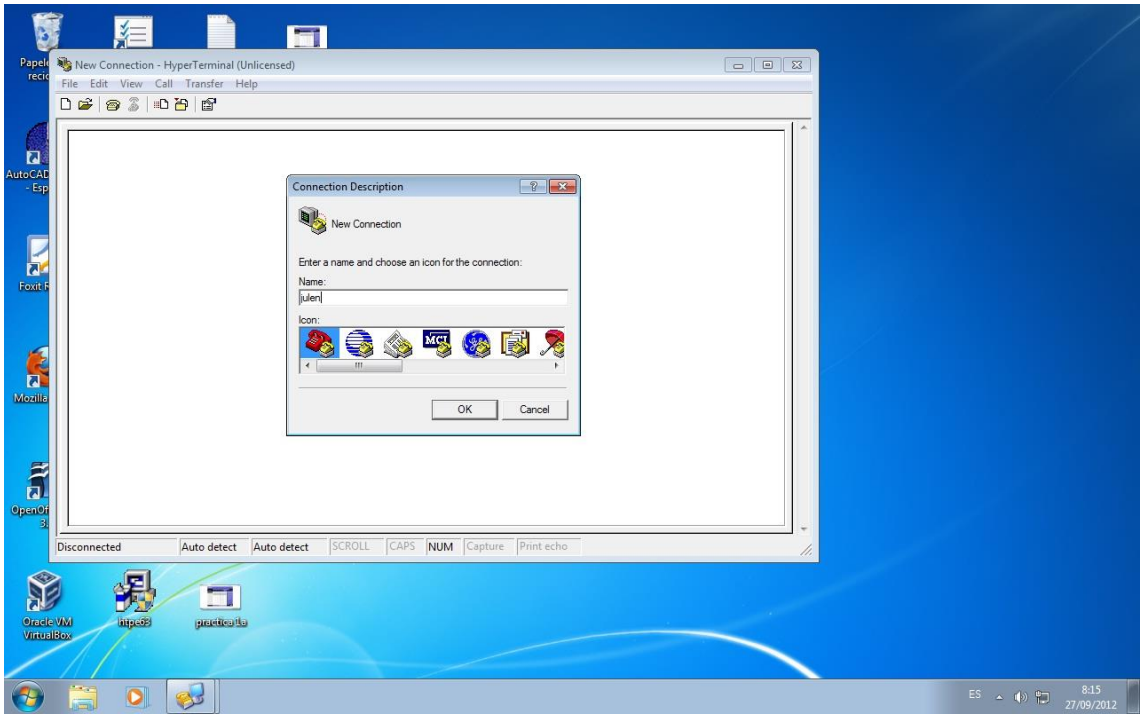


Figura 22. Pantalla de Inicio de Hyperterminal en Windows 7.

Le daremos nombre a la conexión, y nos aparecerá un menú desplegable donde deberemos elegir el puerto por el que nos estamos conectando. En el caso de estos routers, debemos elegir el puerto COM1 para efectuar la conexión. Posteriormente aparecerá otro menú desplegable con la configuración, y la que debemos escoger es la siguiente:

.9600 baudios

.8 bits de datos

.1 bit de parada

.Sin paridad

.Sin control de flujo

Si configuramos la conexión con estos datos, aparecerá la siguiente pantalla:

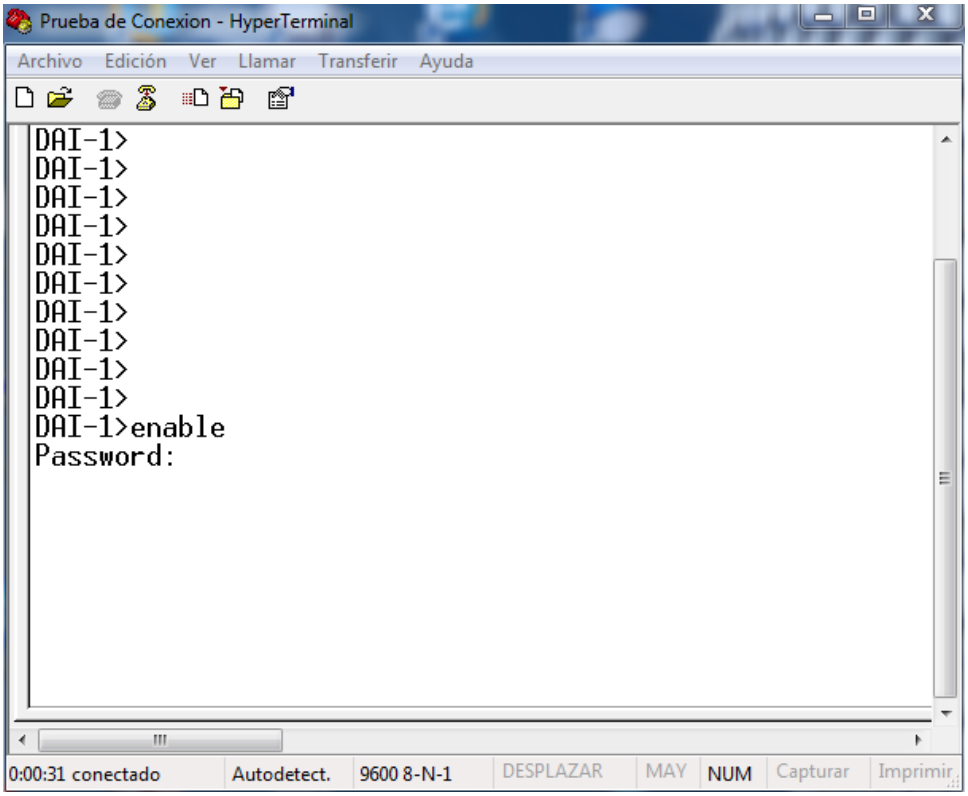
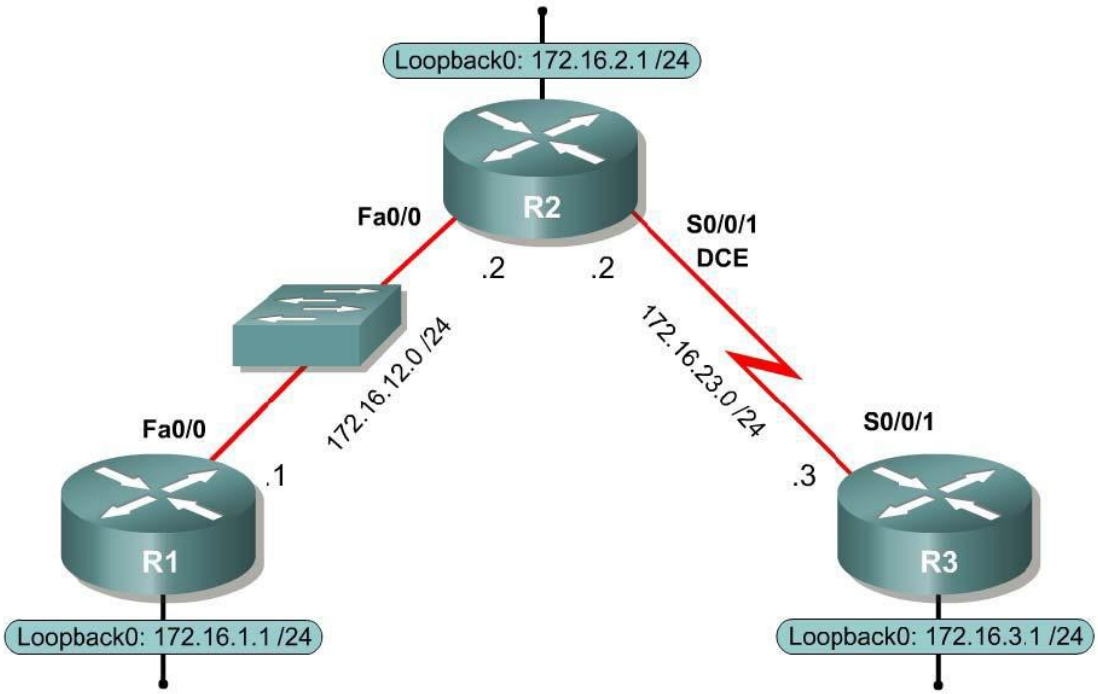


Figura 23. Conexión con el router establecida mediante HyperTerminal.

5.3. Cableado de la red.

Ahora que ya tenemos la conexión con HyperTerminal establecida y los routers con la imagen IOS que implementa MPLS, ya podemos empezar a trabajar con los equipos. Como siempre, el primer paso es cablear la red de acuerdo con nuestra maqueta. En la simulación con GNS3, lo hicimos de este modo:



Modificamos la topología y eliminamos el dispositivo hub que hay entre R1 y R2. Entonces, el cable Ethernet irá directamente conectado de la interfaz FastEthernet 0/1 del R1 a la interfaz FastEthernet 0/0 del R2. Y del R2 a R3 no tenemos más que conectar el cable DCE-DTE entre las interfaces Serial. A continuación deberían aparecer las imágenes de los equipos reales cableados, pero debido a la baja calidad de la cámara, las imágenes no pueden mostrarse con la suficiente calidad, por lo que, desgraciadamente, no pueden incluirse.

5.4. Configuración del direccionamiento IP y de OSPF

La configuración del direccionamiento IP y de OSPF ya ha quedado reseñada en los apartados 3.3 y 3.4 de esta memoria. Dado que usamos las mismas direccionas IP que en la simulación, no debería extrañar que la tabla de encaminamiento quede exactamente igual a la que obtuvimos simulando:

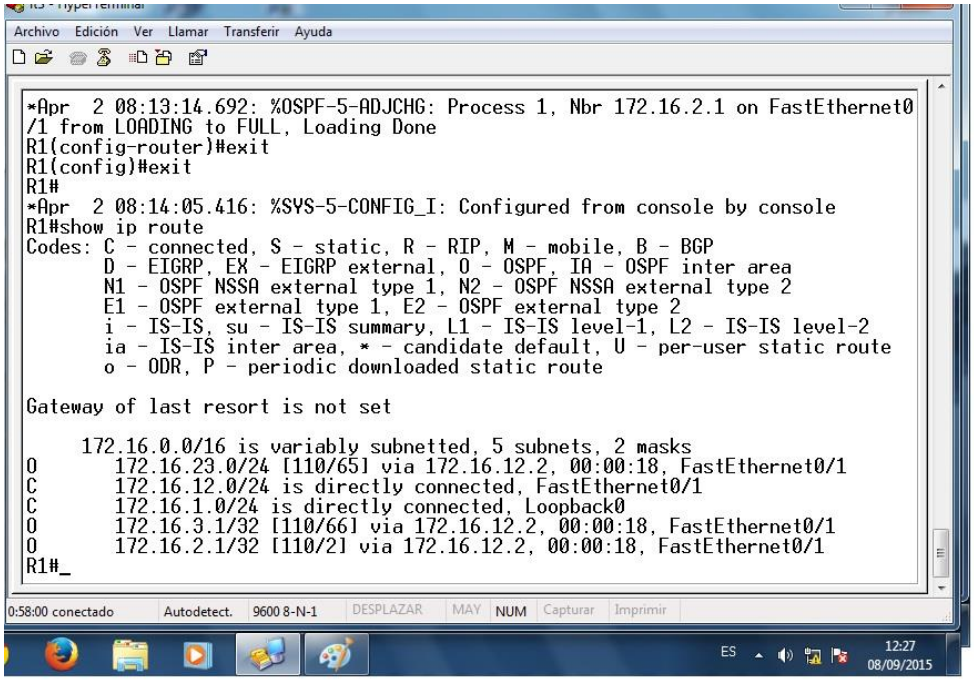


Figura 24. Tabla de encaminamiento del Router R1.

Y si intentáramos comprobar la conectividad, el ping tendría éxito, como demuestra este ping a R2:

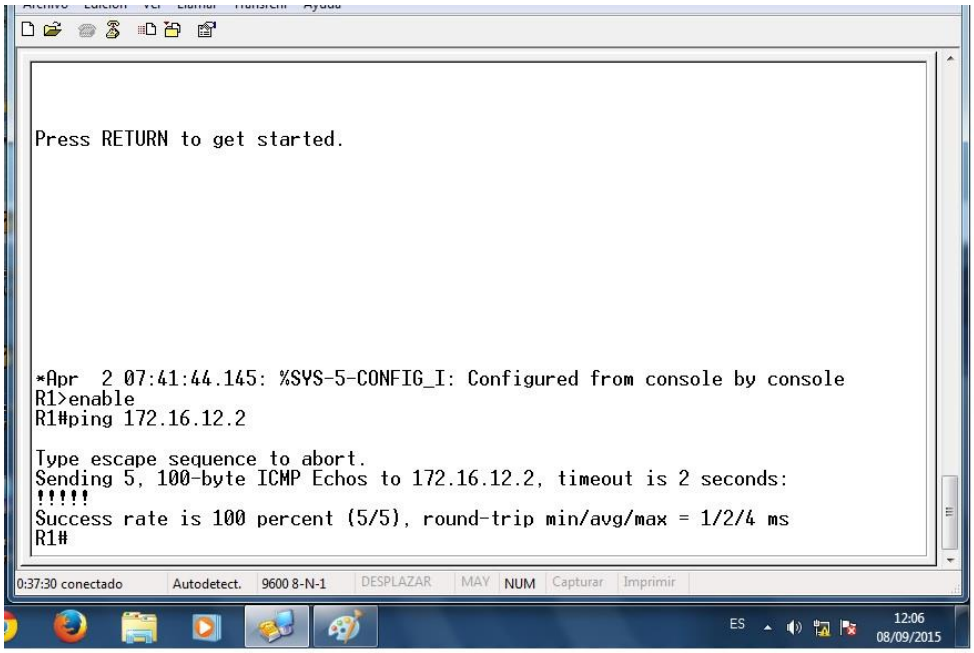


Figura 25. Prueba de conectividad del Router 1 con el Router 2

Obviaremos aquí poner imágenes de R2 y R3, pues las tablas de encaminamiento son exactamente las mismas que pueden verse en el capítulo 3. Hasta el momento no hemos variado nada reseñable.

5.5. Configuración y verificación de CEF y MPLS

Si tratáramos de comprobar la tabla CEF del Router 1, observaríamos lo siguiente:

```
R1#show ip cef
Prefix              Next Hop            Interface
0.0.0.0/0           drop                Null0 (handler entry)
0.0.0.0/32          receive
172.16.1.0/24       attached            Loopback0
172.16.1.0/32       receive
172.16.1.1/32       receive
172.16.1.255/32     receive
172.16.2.1/32       172.16.12.2         FastEthernet0/1
172.16.3.1/32       172.16.12.2         FastEthernet0/1
172.16.12.0/24      attached            FastEthernet0/1
172.16.12.0/32      receive
172.16.12.1/32      receive
172.16.12.2/32      172.16.12.2         FastEthernet0/1
172.16.12.255/32    receive
172.16.23.0/24      172.16.12.2         FastEthernet0/1
224.0.0.0/4         drop
224.0.0.0/24        receive
255.255.255.255/32  receive
```

Como se puede ver, se vuelve a obtener exactamente el mismo resultado que en la simulación, con la diferencia de la interfaz FastEthernet 0/1 por 0/0, pero que en realidad es porque hemos conectado el cable en esa interfaz. Hasta el momento, la simulación ha previsto hasta el más mínimo detalle de la red, lo que empieza a darnos una idea de su alta fiabilidad.

Para habilitar MPLS, no tenemos más que seguir los pasos ya reseñados en el apartado correspondiente del capítulo 3.

5.6. Verificación del funcionamiento de MPLS

Pasamos a comprobar en cada router si MPLS funciona correctamente. Esto lo comprobamos con los comandos **"show mpls ldp Discovery"** y **"show mpls ldp neighbor".**

a) Router 1:

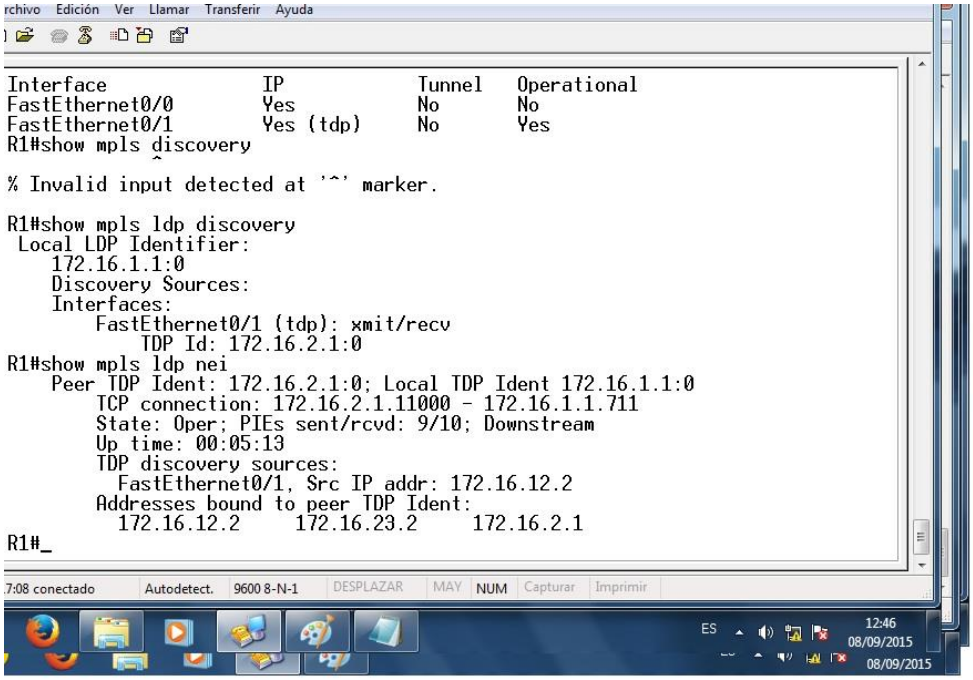


Figura 26. Verificación del funcionamiento de MPLS en Router 1

b) Router 2:

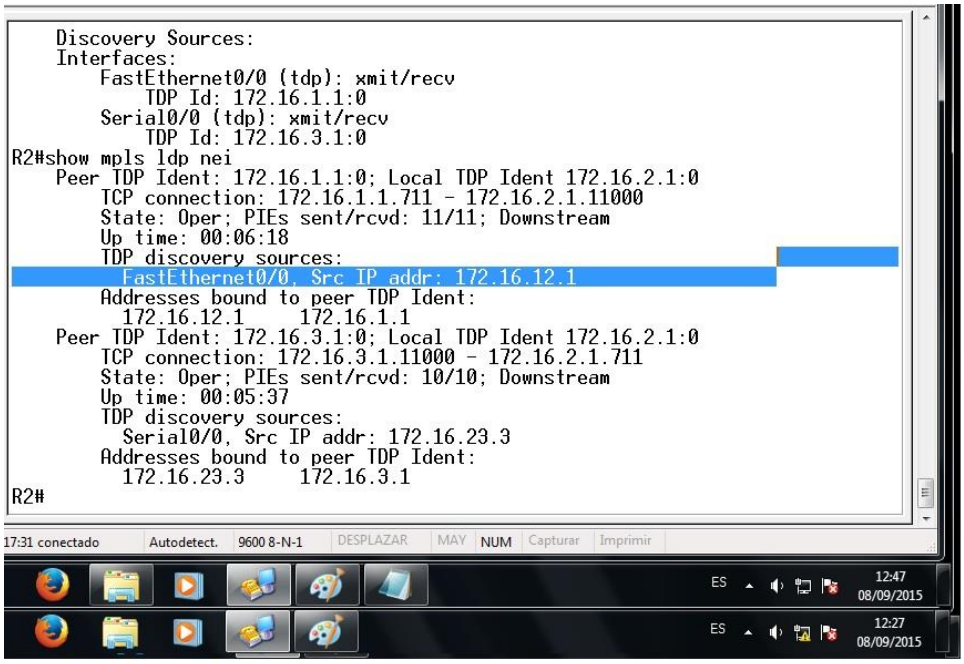


Figura 27. Verificación del funcionamiento de MPLS en Router 2.

c) Router R3:

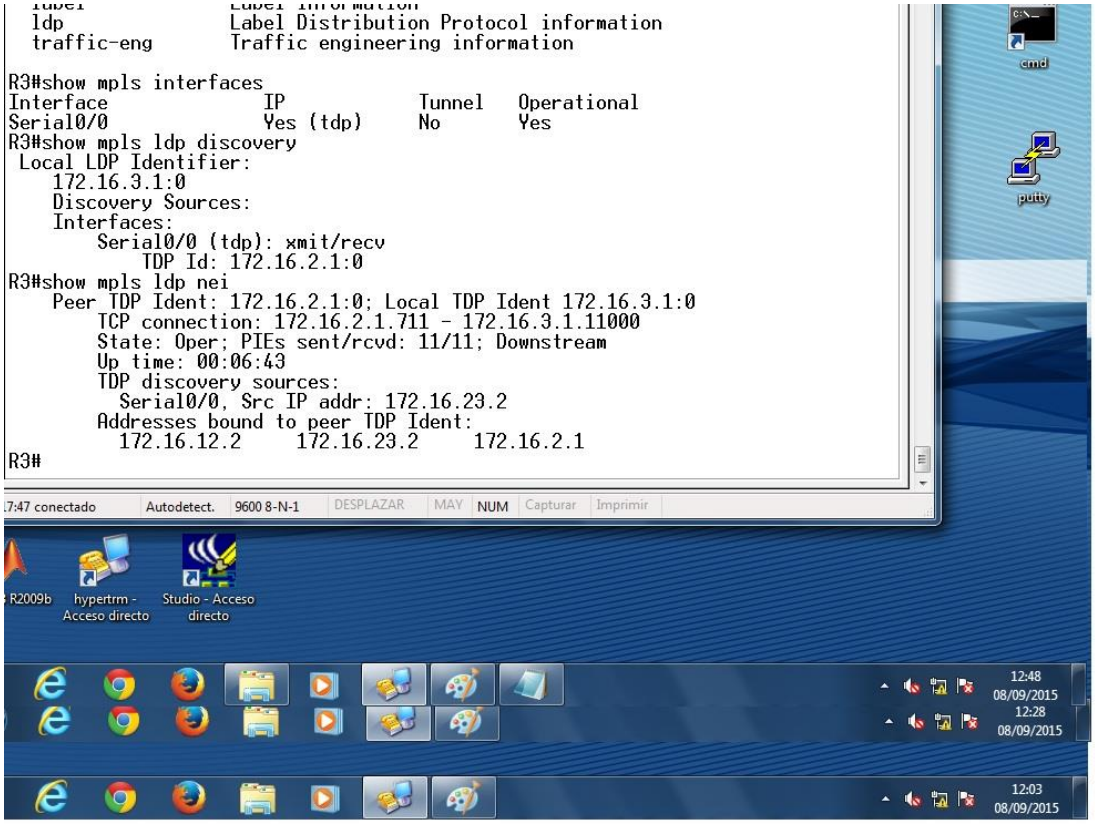


Figura 28. Verificación del funcionamiento de MPLS en el Router 3.

Tal y como salió en la simulación con GNS3, se han obtenido los mismos resultados, y como dijimos en su momento, se ve claramente que los routers han establecido comunicación entre sí utilizando el protocolo TCP. Además, dado que el router 2 es quien tiene las IPs más altas, será quien inicie la comunicación TCP.

Ahora comprobamos si las tablas LIB y LFIB también coinciden con lo previsto en la simulación.

a) Router R1:

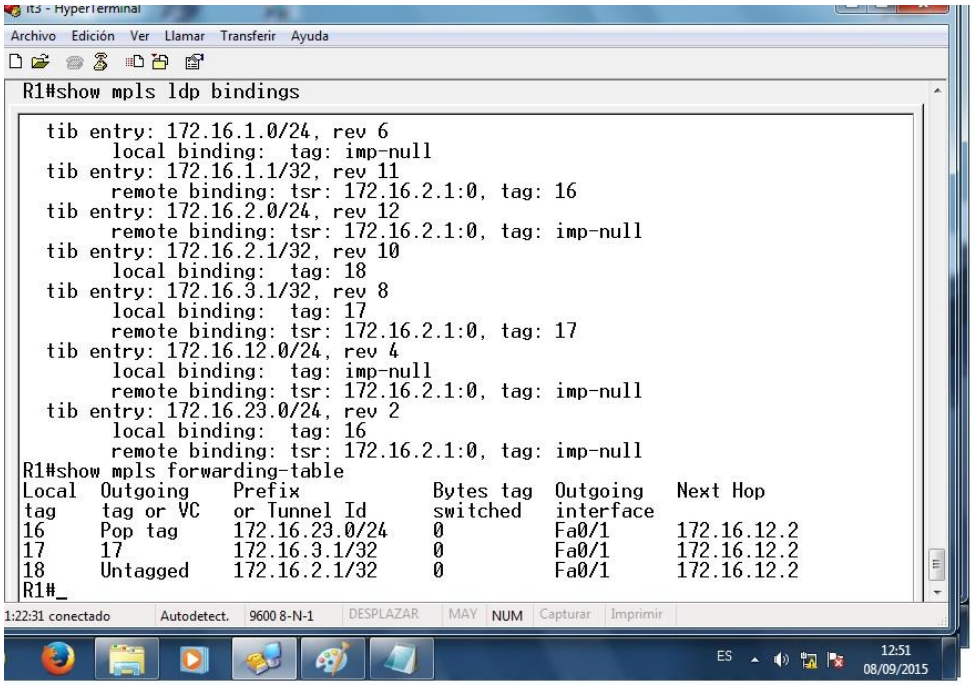


Figura 29. Tablas LIB y LFIB del Router 1.

b) Router R2:

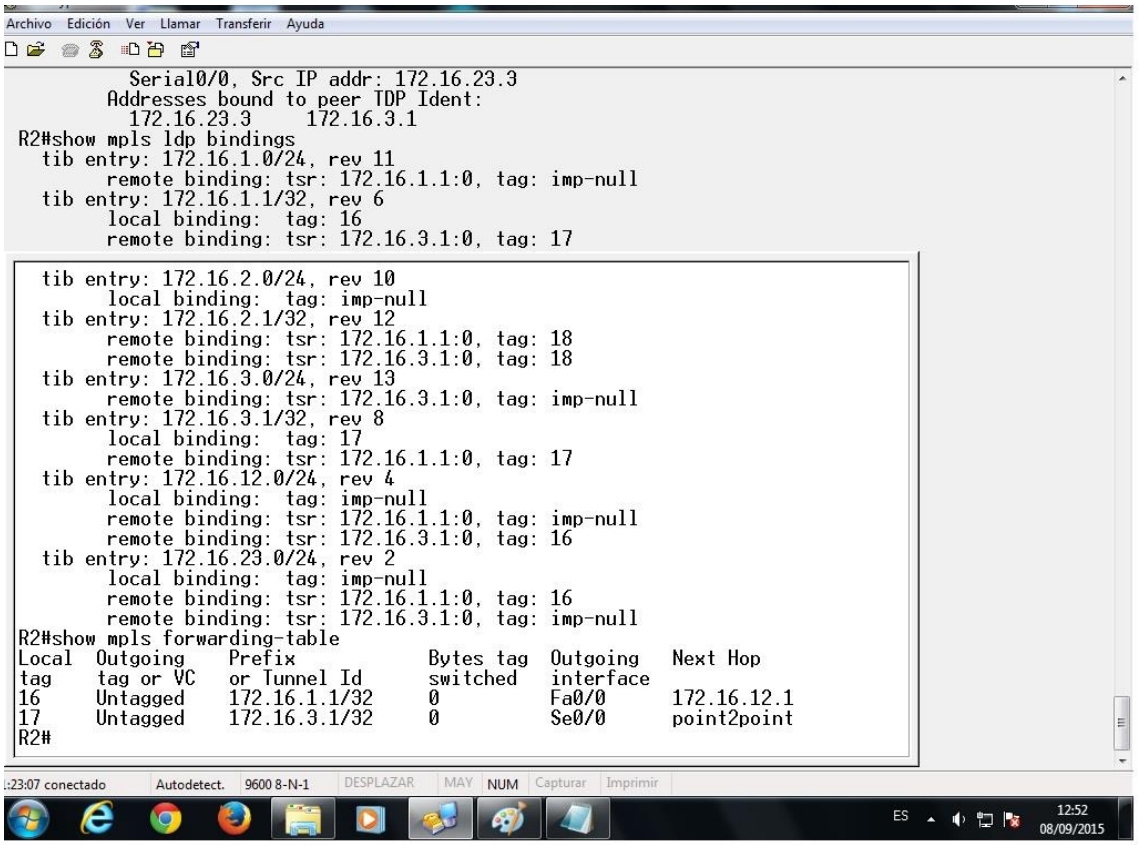


Figura 30. Tablas LIB y LFIB del Router 2

c) Router R3:

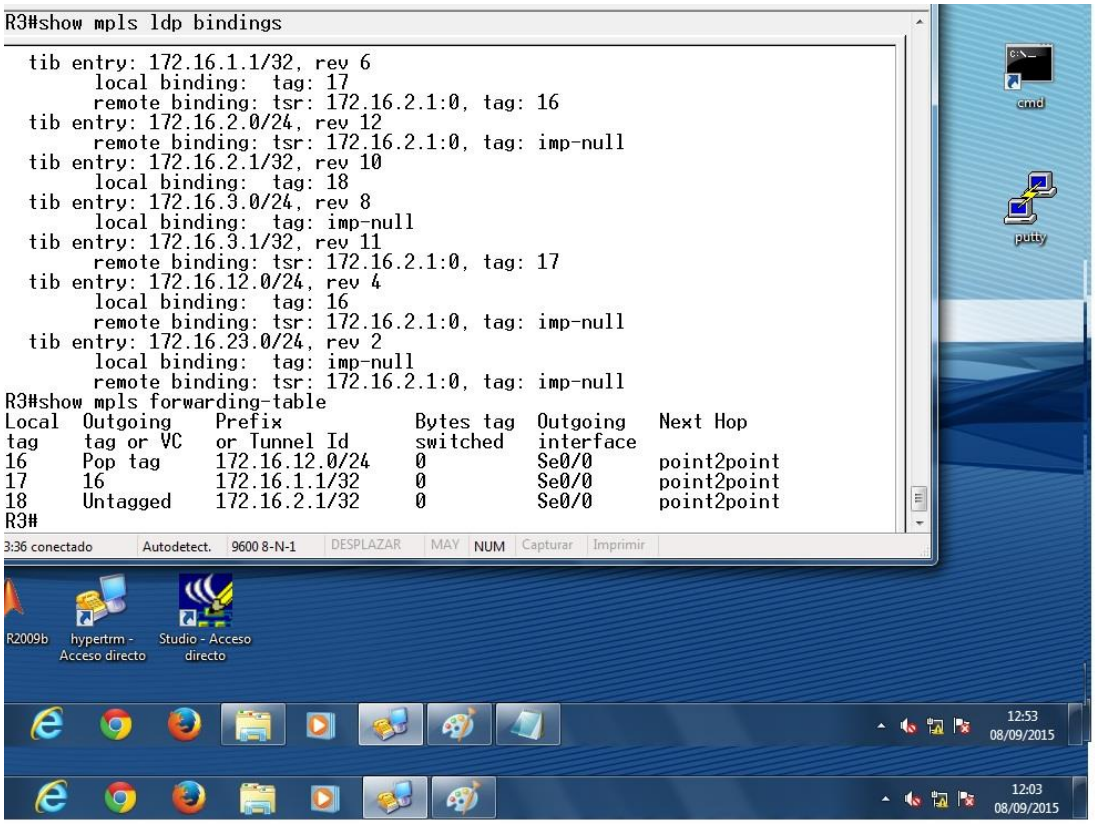


Figura 31. Tablas LIB y LFIB del Router 3.

Observamos que tenemos la misma LIB y la misma LFIB que en la simulación. Inclusive con las mismas etiquetas (siendo éste el factor que podría variar, pues las etiquetas pueden cambiar al reiniciar el router).

Podemos entonces garantizar al 100% que si se hiciera un ping de 172.16.1.1 a 172.16.3.1, la etiqueta que va a llevar el paquete es la número 17 de R1 a R2, sin necesidad ninguna de utilizar el comando traceroute. Cuando vaya de R2 a R3, el paquete llegará con la etiqueta 17, y saldrá sin etiqueta, por lo que ya se comentó de que si la red está directamente conectado, la etiqueta será "imp-null".

El campo "local binding" indica la etiqueta que el router pondrá a todo paquete que llegue a él desde una determinada dirección IP, mientras que el campo "remote binding" lo que indica es la etiqueta asignada a la subred del vecino LDP. El hecho de que en el router R2 haya para una misma subred más de un "remote binding", se debe a que tiene dos routers conectados como vecinos, que son R1 y R3, cuyas interfaces loopback son 172.16.1.1 y 172.16.3.1. Si tuviéramos más routers conectados con los cuales compartimos subred, se mostrarían los "remote binding correspondientes".

Por último, la etiqueta "imp-null" se asigna a un paquete o se anuncia a un vecino cuando se es el último salto de la red. En el caso de nuestro ping de R1 a R3, el router R2 lo marcará como "imp-null", ya que es el último salto antes de alcanzar el destino.

El análisis de la trama MPLS dentro de la red real y de la modificación del tamaño de paquete, no se han abordado y quedan pendientes de futuros trabajos. Esto se ha debido a la imposibilidad de poder conectar la interfaz Fa0/1 del router con el ordenador de forma que Wireshark pudiera capturar el tráfico.

Pero debido a la gran fiabilidad del simulador demostrada hasta el momento en esta memoria, podemos conjeturar que si se hubiera analizado una trama real de MPLS, los valores de los campos hubieran sido los mismos. Sólo hubiera podido variar el tiempo de TTL, debido al factor hardware, aunque es sumamente improbable que este hecho se hubiera dado.

CAPÍTULO 6: ANÁLISIS DE RESULTADOS Y CONCLUSIONES

Finalmente, en este capítulo analizamos los resultados obtenidos en la simulación y en el montaje real, y presentamos las conclusiones a las que hemos llegado.

6.1. ¿Qué capacidades presenta GNS3 a la hora de simular una red o hacer el montaje con equipos reales?

El primer objetivo propuesto ha sido: analizar las prestaciones de GNS3, con el fin de comprobar si sirve para preparar una maqueta de diseño de una red real.

En el capítulo 5 se ha comprobado el rigor con el que los datos obtenidos en la red real se asemejan hasta ser idénticos a los obtenidos en la simulación de la herramienta. Esto demuestra una gran capacidad de predicción de la red, ya que al obtener unos resultados idénticos, las simulaciones que se efectúen en las maquetas con esta herramienta serán prácticamente las mismas que los datos obtenidos en una red real (obviando factores de hardware/software que puedan ser despreciables o escapen a nuestro control). Es decir, no necesitamos estar montando, desmontando y configurando una red sobre el terreno, sino que en la herramienta podemos ver todos los fallos e imponer todas las condiciones que requiera nuestro diseño en la vida real.

En la consecución de estos resultados participan:

a) **El doble procesamiento del simulador:** El uso del IDLE-PC y que el ordenador busque el valor más adecuado en conjunción con el simulador, provocan que se pueda conseguir emulaciones con retardos muy bajos (que resultan ser iguales en un 99% a los de un equipo real), y que garantizan la estabilidad de la red antes cambios en el tamaño de los paquetes que circulan por la misma. Como bien se ve en los apartados 4.9 y 5.6, donde el hecho de cambiar los tamaños de paquete, no ha afectado para nada a los retardos obtenidos ni a la red en sí.

b) **Dynamips y Dynagen:** Estos dos programas permiten que en nuestra herramienta se puedan hacer conexiones vía Telnet a las consolas de los routers. Además, permiten capturar tramas y paquetes para analizarlos en Wireshark, o incluso (aunque éste es un punto a probar en futuros trabajos), la posibilidad de conectar y comunicarse con equipos reales externos a la red que estamos simulando.

Gracias a estas características especiales, podemos simular con bastante detalle cualquier red que no tenga una topología excesivamente compleja.
A mayor complejidad de la red, más retardos se introducirán y más difícil será tener unos resultados óptimos.

Aunque en este caso no ha influido demasiado, es interesante reseñar además que el sistema operativo que usemos puede afectar al rendimiento. Además, el IOS es distinto de un router a otro, y los routers que sean más complejos pueden complicar el tiempo de carga CPU para la computadora. Como se describió antes, se tuvieron que descartar algunas topologías de red porque en Windows, actualmente, la versión 1.2.1, no permite la simulación de hosts. También se tuvieron que realizar algunos pasos previos al arranque de la herramienta en Windows para dotar a la herramienta de estabilidad y que no hubiera fallos que imposibilitaran el trabajo. El rendimiento de GNS3 es peor en Windows que en Linux, pero el uso de Windows en el proyecto se justifica debido a que la gran mayoría de los equipos utilizados tenían Windows como principal sistema operativo, y ni se podía ni había tiempo de instalar Linux en los mismos.

En resumen, GNS3 presenta alta fiabilidad y una obtención de resultados óptimos siempre que la topología no sea excesivamente compleja, independientemente del sistema operativo que utilicemos. Y esto es debido a su característica de doble procesamiento, y al uso de las capacidades que ofrecen Dynamips y Dynagen. No obstante, se recomienda Linux si se quiere la máxima estabilidad y evitar tener que dar pasos complejos para evitar fallos.

6.2. ¿Qué características de MPLS hemos podido comprobar?

Entre el entorno de red físicamente implementado y el desarrollado con GNS3 hemos comparado características de MPLS.

En este proyecto, se ha estudiado cómo funciona el protocolo MPLS dentro de una trama de datos. Se han observado los 4 campos de los que consta una etiqueta MPLS, y se ha podido ver cómo al trazar el camino que debe seguir un paquete, se le van imponiendo o quitando las correspondientes etiquetas.

También se observan que los retardos difieren bastante, tanto en la red real como en la simulación, según esté o no MPLS activado. Sin MPLS activado, los retardos son mayores debido a la búsqueda que se ha de hacer en todos los routers en la tabla de encaminamiento correspondiente.

Con MPLS activado, por el contrario, existe un retardo mayor en el router emisor debido a la búsqueda en la tabla LFIB y a la imposición y creación de la etiqueta. Sin embargo, los retardos se van reduciendo conforme el paquete atraviesa routers hasta llegar a destino. Y esto es debido a que la información de la etiqueta ya indica por dónde se debe reenviar el paquete, y la búsqueda en la tabla se hace innecesaria.

En suma, pese a un mayor retardo inicial, una red con MPLS va a terminar siendo más rápida que una red sin MPLS. Esto es debido a que se evitan las búsquedas en la tabla de encaminamiento y el procesado del tráfico es más rápido. Y esto es una característica muy deseable en las redes en las que este protocolo se utiliza, o sea, aquellas que implementan Servicios Diferenciados, ya que para tener una alta calidad de servicio se requiere rapidez en el transporte.

6.3. ¿Cómo resuelve CISCO las necesidades de comunicación?

Se ha demostrado viendo los capítulos 3 y 4, que los datos obtenidos han sido iguales. Debemos destacar que el hardware y el software han sido distintos en los dos sistemas (GNS3 y red física). En GNS3 se utilizó el hardware C2600 y el software "c2600-telco-mz.bin" y en la red física se utilizaron routers 2611XM y 2620XM con un software "c2600-advipservicesk9-mz.123-4.T4.bin". La única coincidencia es que ambos podían implementar el protocolo MPLS, y hubo que ceñirse, en la red real, a los equipos que ofrecía el laboratorio IT-3.

Esto es debido a que las funcionalidades MPLS estudiadas funcionan igual en distintos hardwares, siempre que tengan unas características comunes a una determinada familia. Esto ha posibilitado que no haya influido el tipo de hardware y software utilizado en nuestro estudio.

6.4. Líneas futuras de investigación

Para concluir esta memoria, quiero reseñar aquí algunas de las posibles líneas futuras de investigación a las que este proyecto pueda dar pie:

- Estudiar otras topologías más complejas para comprobar qué redes físicas MPLS pueden estudiarse desde GNS3.
- Probar GNS3 en otros sistemas operativos (Windows y Linux).
- Trabajar con otras interfaces en el hardware emulado por GNS3, como son las interfaces de Fibra Óptica.
- Implementar VPN de nivel 2 y estudiarlo con GNS3.

BIBLIOGRAFÍA

Dado que éste trabajo ha sido eminentemente práctico, realmente no ha sido necesario emplear una gran cantidad de bibliografía. Se referencian aquí las fuentes utilizadas para su realización o extracción de información, así como el libro propiamente empleado, una lista de información Web para la información sobre CISCO, MPLS en Cisco, Dynamips, Dynagen y por último, un tutorial web sobre GNS3.

Los capítulos 5 y 6 no contienen referencia ninguna, puesto que son la demostración y las conclusiones de lo que venimos planteando, y ésas han sido realizadas por mí. También se citan aquí las referencias utilizadas para los anexos. En cuanto a las figuras empleadas, sólo de la figura 1 a la 4 se han extraído de internet, mientras que el resto (de la figura 5 a la 31), se han extraído de los diversos equipos empleados.

La lista completa de la bibliografía es la siguiente:

[1] "MPLS and VPN architectures" – Jim Guichard. 1° edición (2001) Ed.Cisco Press ISBN: 1587050021.

[2]http://searchenterprisewan.techtarget.com/definition/Multiprotocol-Label-Switching

[3] http://www.cisco.com/c/en/us/products/ios-nx-os-software/multiprotocol-label-switching-mpls/index.html

[4] http://repositorio.utp.edu.co/dspace/bitstream/11059/1311/1/0046T172.pdf

[5] http://blog.ipexpert.com/wp-content/uploads/2010/03/GNS3-on-Windows-71.pdf

[6] http://www.iteasypass.com/Dynamips.htm

[7] http://www.dynagen.org/tutorial.htm

[8] "Configuración de MPLS en modo trama" – Santiago Felici. Universidad Politécnica de Valencia.

[9] https://community.gns3.com/thread/4343 **(Sobre la S roja)**

[10]http://www.cisco.com/c/en/us/products/collateral/routers/2600-series-multiservice-platforms/product_data_sheet0900aecd800fa5be.html

[11]http://www.cisco.com/c/en/us/td/docs/ios/12_2/switch/configuration/guide/fswtch_c/xcfcef.html

ANEXO: Abreviaturas utilizadas en la memoria

Se listan a continuación las abreviaturas utilizadas en la memoria del proyecto. Son las siguientes: **[8]**

FEC (Forwarding Equivalence Class): Conjunto de paquetes que entran en la red MPLS por la misma interfaz, que reciben la misma etiqueta y por tanto circulan por un mismo trayecto. Normalmente se trata de paquetes que pertenecen a un mismo flujo.

LSP (Label Switched Path): Camino que siguen los paquetes que pertenecen a la misma FEC, es equivalente a un circuito virtual.

LSR (Label Switching Router): Router que puede encaminar paquetes en función del valor de la etiqueta MPLS.

LER (Label Edge Router): Router que opera en la frontera de una red con MPLS y actúa como punto de entrada y salida de la misma.

LIB (Label Information Base) o TIB (Tag Information Base): La tabla de etiquetas que manejan los LSR. Relaciona la pareja (interfaz de entrada - etiqueta de entrada) con (interfaz de salida - etiqueta de salida). En versiones de IOS antiguas, en lugar de Label se utilizaba Tag y de ahí, que hay algunos comandos que utilizan "tag" en lugar de "label".

LDP o TDP (Label o Tag Distribution Protocol): Protocolo utilizado para distribución de etiquetas MPLS. LDP es la versión estandarizada e integrada en las IOS con versiones 12.4(3) o superior y TDP es una versión precursora propietaria definida por Cisco Systems que ha sido reemplazada por LDP. Podríamos decir que TDP está incluido en LDP.

FIB (Forwarding Information Base): En pocas palabras es la tabla de rutas del router, pero con soporte hardware, basado en CEF. Esta tabla se actualiza automáticamente a petición de los protocolos de routing.

LFIB (Label Forwarding Information Base): Es la tabla que asocia las etiquetas con los destinos o rutas de capa 3 y la interfaz de salida en el router, indicándole al router lo que tiene que hacer: poner o quitar etiqueta.

LIB (Label Information Base): Es la tabla que contiene sólo información de etiquetas MPLS y es utilizada por LDP (o TDP) para la gestión y envío de las etiquetas.

PHP (Penultimate Hop Popping): es una alternativa de entrega de trama MPLS al final del circuito virtual, para mejorar las prestaciones y el consumo de CPU. Consiste en quitar la etiqueta MPLS cuando se sabe que el siguiente router no necesita la etiqueta MPLS por estar la red directamente conectada a él o ser el final del circuito virtual. De esta forma, se evita hacer una doble búsqueda en dicho router, tanto en la tabla de LFIB y en la tabla de rutas. Es el modo de funcionamiento por defecto en los routers de Cisco Systems.