

Multiprotocol Label Switching con GNS3

Diseño Básico de IP/MPLS

Lucas Aimaretto

Comunicaciones de Datos

Facultad de Ciencias Exactas, Físicas y Naturales

Universidad Nacional de Córdoba

Enero/2018

Contenido

1 -Introducción.....	4
2 -Acrónimos.....	5
3 -MPLS.....	6
3.1 -Reseña.....	6
3.2 -Pila de protocolos.....	6
3.3 -Conmutación.....	8
3.4 -Asignación de etiquetas.....	8
3.5 -Label Switched Path.....	9
4 -GNS3.....	11
4.1 -Arquitectura.....	11
4.2 -Direccionamiento IP.....	11
4.3 -Construcción de la red en GNS3.....	13
4.4 -Configuración.....	13
4.4.1 -Establecimiento del hostname.....	13
4.4.2 -Interfaces de loopback.....	13
4.4.3 -Interfaces de enlace.....	14
4.4.4 -Routing.....	15
4.4.4.1 -Routing estático.....	15
4.4.4.2 -OSPF.....	15
4.4.5 -MPLS.....	16
4.4.5.1 -MPLS.....	16
4.4.5.2 -LDP.....	16
Penultimate-hop-popping.....	17
4.5 -Análisis de protocolos.....	17
4.5.1 -Verificar OSPF.....	17
4.5.2 -Análisis de LDP.....	19
4.5.2.1 -Penultimate-hop-popping.....	20
4.5.3 -Verificar MPLS punta a punta.....	22
5 -Conclusión.....	26
6 -Apéndice A.....	27
6.1 -Instalación.....	27
6.1.1 -Desde de la página GNS3.....	27
6.2 -Inicio.....	28
6.3 -Carga de Cisco IOS 3745.....	28
6.4 -Idle-PC Number.....	29
6.5 -Creación del primer proyecto.....	30
6.5.1 -Interacción con otros routers.....	31
-Especificaciones Técnicas.....	33
-Referencias.....	34

Imágenes

Ilustración 1: Cabecera MPLS.....	7
Ilustración 2: Conmutación de etiquetas.....	8
Ilustración 3: Asignación de etiquetas.....	9
Ilustración 4: LSP.....	10
Ilustración 5: Arquitectura.....	11
Ilustración 6: Arquitectura en GNS3.....	13
Ilustración 7: Enlaces PE_A.....	14
Ilustración 8: Ruta estática CE ₁	15
Ilustración 9: Señalización LDP.....	19
Ilustración 10: LDP Mensaje Hello.....	19
Ilustración 11: LDP Label Mapping Message.....	20
Ilustración 12: PHP Implicit Null.....	21
Ilustración 13: PHP Explicit Null.....	22
Ilustración 14: IP or MPLS lookup.....	23
Ilustración 15: LSP Label 1.....	25
Ilustración 16: LSP Label 2.....	25
Ilustración 17: LSP Label 3.....	25
Ilustración 18: GNS3 Doctor.....	28
Ilustración 19: Slots.....	28
Ilustración 20: Buscando idle-pc.....	29
Ilustración 21: idle-pc encontrado.....	29
Ilustración 22: Proyecto nuevo.....	30
Ilustración 23: Primera topología.....	30

Tablas

Tabla 1: Etiquetas reservadas.....	7
Tabla 2: Direccionamiento IP.....	12

1 Introducción

GNS3 es un ambiente de simulación construido sobre la filosofía del software libre. Nació para permitir la ejecución del firmware que corren dentro de sí los equipos de Cisco Systems, más conocidos como IOS, en computadoras personales [1]. GNS3 no provee dichos sistemas operativos: sólo ofrece un medio a partir del cual pueden ser ejecutados. También es posible ejecutar sistemas operativos de otras marcas al igual que sistemas operativos Linux. En el presente trabajo sólo utilizaremos sistemas operativos de la marca Cisco Systems. Asimismo, GNS3 será instalado y ejecutado en una PC corriendo Linux Ubuntu 13.04.

MPLS, Multiprotocol Label Switching, es un protocolo que se ubica entre las capas 2 y 3 del modelo en capas TCP/IP [2]. MPLS hace uso de etiquetas para poder conmutar tráfico entre los diversos saltos, y así llevarlo desde un extremo inicial a otro extremo final, dentro de lo que puede denominarse el dominio MPLS. Dichas etiquetas no son otra cosa que la representación de circuitos virtuales [3]. MPLS no contempla la asignación de etiquetas; solamente las utiliza. Las etiquetas deben ser asignadas a través de otros mecanismos. En el presente trabajo se asume que la asignación de etiquetas estará realizada por el protocolo LDP, Label Distribution Protocol. LDP, a su vez, precisa que las tablas de enrutamiento en cada componente del dominio MPLS hayan convergido [4]. El protocolo de enrutamiento que utilizaremos será OSPF.

El presente trabajo intentará explicar el funcionamiento básico de MPLS; cómo IP utiliza a MPLS como transporte; LDP, con un somero análisis de OSPF de acuerdo a lo que se requiere y la ventaja sobre el routing tradicional. Se usará como plataforma de simulación a GNS3 0.8.3 con el firmware de Cisco IOS 12.4(25) en hardware -virtualizado- 3745.

Este trabajo es presentado como parte de los requerimientos solicitados por la Facultad de Ciencias Exactas, Físicas y Naturales de la U.N.C., a fin de cumplir con el plan de trabajo aprobado para la Adscripción a la cátedra Comunicaciones de Datos.

2 Acrónimos

CE – Customer Edge Router

eLER – Egress LER

FEC – Forwarding Equivalence Class

IP – Internet Protocol

iLER – Ingress LER

LER – Label Edge Router

LFIB – Label Forwarding Information Base

LSR – Label Switching Router

LSP – Label Switched Path

MPLS – Multiprotocol Label Switching

OSPF – Open Short Path First

P – Provider Core Router

PE – Provider Edge Router

PHP – Penultimate Hop Popping

UDP – User Datagram Protocol

TCP – Transport Control Protocol

3 MPLS

3.1 Reseña

MPLS es un protocolo definido en la RFC 3031. El término Multiprotocol refiere al hecho de poder transportar cualquier tipo de carga, ya sea IP, FR, ATM, etc. Por ser una de las formas mayormente utilizadas, en el presente trabajo, se hará hincapié en el transporte de IP sobre MPLS. Como capa de enlace de datos, se utilizará Ethernet.

Label Switching, por su parte, hace referencia a la inserción o intercambio de etiquetas en cada salto. Este proceso se inicia cuando un paquete IP ingresa al dominio MPLS, vía el router PE de ingreso; se repite cada vez que una trama MPLS atraviesa un router de tipo 'P' dentro de la nube MPLS y finaliza cuando la trama MPLS arriba al router PE de salida. La denominación PE indica que el equipo es un equipo del proveedor de servicios y se ubica en los límites del dominio MPLS, colindante con la red de los clientes. El router P es ubicado dentro del dominio y sólo tiene conexión con otros routers de tipo P o PE. Otras acepciones para PE son las de iLER o eLER, por ingress/egress Label Edge Router mientras que para los routers P, existe la denominación adicional de LSR o Label Switching Router.

Los routers PE de ingreso, de salida y los routers P, insertan, remueven o intercambian respectivamente etiquetas en función de tablas previamente definidas. Las funciones de inserción, remoción e intercambio de etiquetas son conocidas como Push, Pop y Swap, respectivamente.

Dichas tablas se construyen en base a la asignación que el protocolo -en nuestro caso LDP- hace por cada destino IP. En otras palabras: se genera un mapeo de etiquetas por destino IP. Estos destinos, son conocidos como FEC o Forwarding Equivalence Class. En un entorno de routing clásico, el camino a seguir se define en cada salto. Es decir, la FEC es encontrada en cada salto. En un entorno MPLS, la FEC se decide sólo en el PE de ingreso o iLER. Las tablas suelen denominarse LFIB o Label Forwarding Information Base.

Para los routers PE de ingreso, el mapeo está dado en función del destino IP indicado en el campo destino del paquete IP. Para los routers PE de salida, el mapeo indica en realidad que la etiqueta debe eliminarse o removerse. Para los routers P, el mapeo indica que una etiqueta debe intercambiarse por otra.

Cuando un paquete IP llega a un iLER o PE de ingreso, se debe buscar la FEC adecuada y en función de ello realizar una operación de push, insertando la etiqueta apropiada. Luego se envía el paquete etiquetado al siguiente salto, el cual hará el intercambio de etiquetas según su tabla repitiendo de esta manera el procedimiento hasta llegar al PE de salida, el cual la removerá.

La secuencia de etiquetas -o la secuencia de routers LSR- que permiten que un paquete IP atraviese todo el dominio MPLS, desde el PE de ingreso hacia el PE de salida, constituye lo que se conoce como LSP o Label Switched Path [5] .

3.2 Pila de protocolos

Como se introdujera, MPLS se ubica dentro del modelo TCP/IP entre las capas 2 y 3. La cabecera MPLS consta de 32 bits dentro de la cual se distinguen 4 campos: Label, Exp, S, TTL.

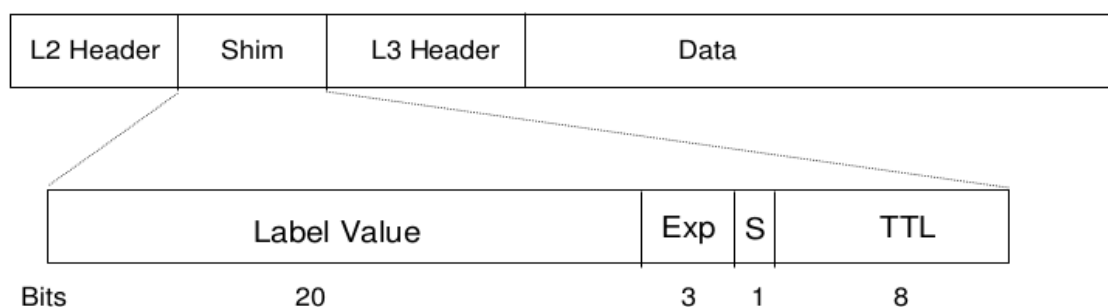


Ilustración 1: Cabecera MPLS

En el caso del presente trabajo, se entiende que tanto los campos 'L2 Header' como 'L3 Header' hacen referencia a Ethernet e IP respectivamente.

El campo Label de MPLS tiene una longitud de 20 bits y el valor que asuma permitirá diferenciar ciertas conductas a realizar sobre el paquete recibido. La primera función es la de permitir la conmutación a lo largo del domino MPLS. Por otro lado, existen asimismo ciertos valores reservados [6], a saber:

Valor	Descripción
0	IPv4 Explicit NULL Label
1	Router Alert Label
2	IPv6 Explicit NULL Label
3	Implicit NULL Label
4-6	Unassigned
7	Entropy Label Indicator (ELI)
8-12	Unassigned
13	GAL Label
14	OAM Alert Label
15	Unassigned

Tabla 1: Etiquetas reservadas

El valor 3 es un valor interesante y se utiliza cuando un router PE_n indica a su router P_{n-1} que a la hora de entregarle tráfico lo haga sin etiquetarlo. Es lo que se conoce como Penultimate-Hop-Popping (PHP). La idea detrás de esta conducta es evitar el proceso de remoción de etiquetas en el router PE y acelerar el procesamiento de la capa L3 en los routers eLER.

A diferencia de otros protocolos, MPLS no incluye un campo que permita identificar qué protocolo se transporta a nivel L3. En Ethernet el campo Type [7] permite saber qué protocolo está encapsulando. Por ejemplo, cuando Ethernet transporta MPLS Unicast, el campo Type tiene un valor de 0x8847. Por su parte, IP especifica en su campo Protocol cuál es el protocolo encapsulado en L4. Sin embargo, MPLS no lo especifica. ¿Cómo hacer entonces para identificar qué es lo que se está transportando? En el caso de no utilizar PHP, un router PE de salida elimina la etiqueta, y la etiqueta misma deberá indicar o permitir inferir qué protocolo de L3 es el que se está llevando. Es la

única manera de saber cómo se ha de interpretar la capa L3 [8]. En cambio, si PHP sí se utiliza, entonces el router PE de salida simplemente recibe una trama Ethernet encapsulando a IP, por lo que, básicamente, realiza una operación estándar de routing sobre el mismo.

El campo EXP, también denominado de Bits Experimentales, es el encargado de llevar la información de calidad de servicio o QoS. Tiene 3 bits y por lo tanto permite 8 clases de servicio. Un buen diseño de QoS permitirá el mapeo del campo TOS de IP con el campo EXP. A su vez, si a la salida del dominio MPLS se entrega el tráfico en VLANs, el campo EXP deberá mapearse correctamente con el campo 802.1p.

El campo S, de 1 bit, especifica si hay etiquetas MPLS apiladas. En caso de valer '1', se ha arribado al fondo de la pila y por lo tanto no hay más etiquetas que procesar. Es usual encontrar etiquetas apiladas a la hora de implementar VPNs en el dominio MPLS, donde la etiqueta más cercana a la capa L2 es la etiqueta de transporte, mientras que la más cercana a la capa L3 es la correspondiente a la VPN.

El campo TTL o tiempo de vida, es un contador que se decrementa cada vez que se atraviesa un router MPLS. Este campo evita que un paquete MPLS quede en el dominio por tiempo indeterminado.

3.3 Conmutación

La conmutación de etiquetas es la base de MPLS. Cuando un paquete IP llega al router PE de ingreso, en función de la dirección IP de destino de dicho paquete, se inserta una etiqueta MPLS. Dicha etiqueta es conmutada en cada salto hasta llegar al final donde es removida y el paquete es entregado al destino correspondiente.

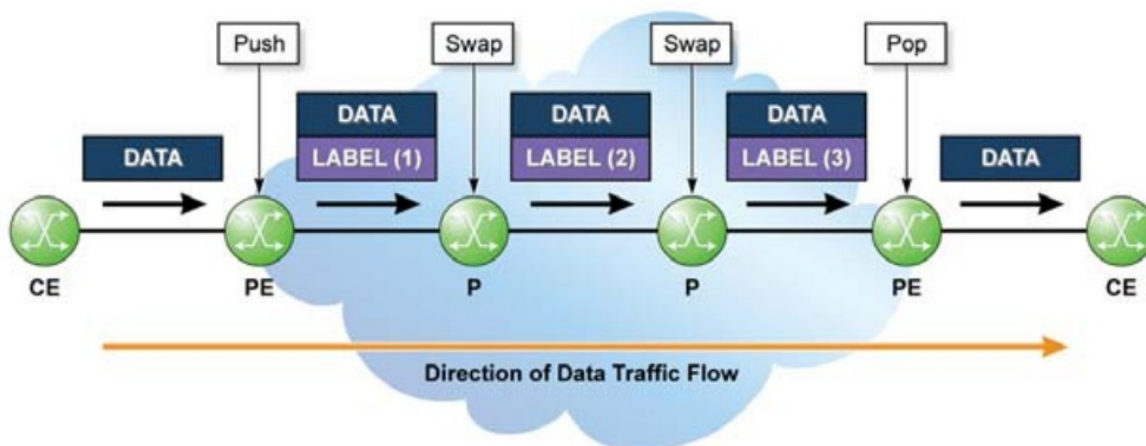


Ilustración 2: Conmutación de etiquetas

3.4 Asignación de etiquetas

Se hizo mención que MPLS hace uso de etiquetas para conmutar el tráfico, pero no es el encargado en sí mismo de asignarlas. La asignación las hace un protocolo diferente. En el presente trabajo se utilizará LDP, Label Distribution Protocol, el cual está definido en el RFC 5036.

La forma en la que LDP asigna etiquetas es en base a una FEC y, por defecto, de manera no solicitada: es decir, una vez establecida la vecindad con el router vecino, comienza a comunicar sus

bindings sin esperar solicitudes de ningún tipo. Todos los routers que hablen LDP se comportarán de la misma manera. Un binding es la asociación etiqueta:FEC[9].

Un router con LDP analiza su tabla de ruteo y por cada entrada o FEC, establece una etiqueta. Es importante entender que para que un router reciba y acepte un binding etiqueta:FEC, el FEC debe existir a priori en su tabla de ruteo. En otras palabras: el protocolo de routing utilizado en el dominio tiene que haber convergido antes de que LDP pueda asignar etiquetas, visto que la tabla de routing tiene que estar completa. Caso contrario descartará el binding recién recibido. Si existe una coincidencia entre el prefijo de red recibido por LDP y alguna entrada en la tabla de routing, entonces dicho binding se instala dentro de la LFIB.

La etiqueta elegida por el router n-ésimo tiene sentido local. Es decir, el router vecino puede utilizar la misma si así lo desea, siempre y cuando no la haya utilizado antes para un LSP distinto.

En la imagen se aprecia que el router k informa al router j acerca del binding 100:Z. Y a su vez establece una operación de POP para cada paquete que sea recibido con la etiqueta 100.

El router j crea un binding 200:Z y lo envía al router i. A su vez establece una operación de SWAP, para cada paquete recibido con la etiqueta 200 y la intercambiará por la etiqueta 100.

Finalmente, el router i, por ser el último en la cadena, sólo recibe el binding de j y establece una operación de PUSH para cada paquete IP que tenga a Z como destino.

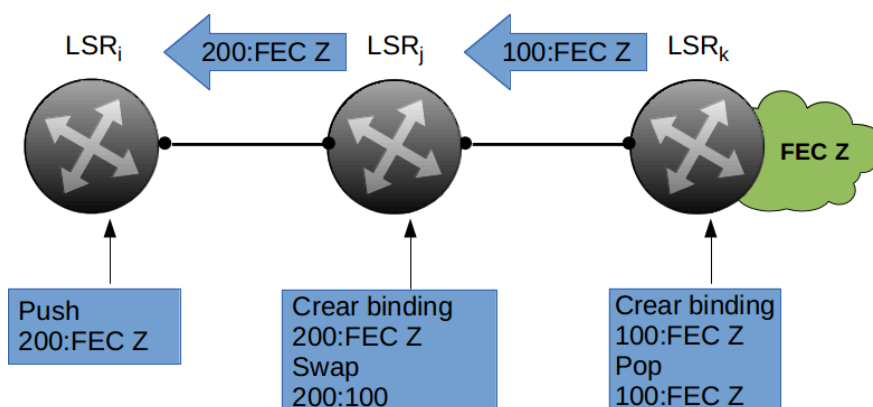


Ilustración 3: Asignación de etiquetas

Si el router j no hubiese tenido dentro de su tabla de routing el FEC Z, entonces el binding 100:Z recibido desde k no habría sido considerado y por ende habría sido descartado.

3.5 Label Switched Path

Una vez finalizada la asignación punta a punta, y de acuerdo a la imagen anterior, se aprecia que el LSP en LSR_i para el FEC Z está dado por la serie de etiquetas 200 y 100. Un LSP es unidireccional: el LSP que va desde el al LSR_k está comprendido por las etiquetas 200 y 100. El LSP que vaya en sentido contrario podrá utilizar otras etiquetas, según lo disponga la señalización de LDP en su momento [5].

En la imagen siguiente el LSP que va desde el iLER hacia el eLER comprende las etiquetas 1, 2 y 3.

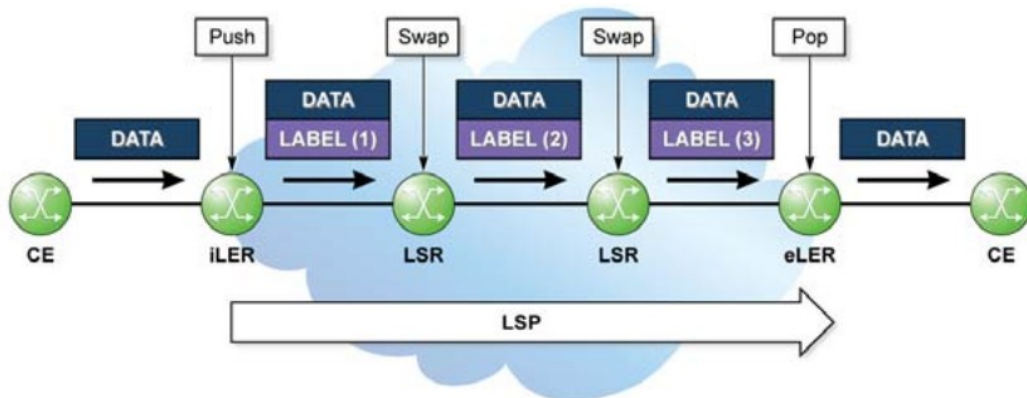


Ilustración 4: LSP

Se puede decir que la asignación de etiquetas -o bien los bindings- viajan en sentido contrario al sentido del LSP propiamente dicho. En otras palabras, se puede decir que los LSP se señalizan desde atrás hacia adelante, siendo 'atrás' el eLER y 'adelante' el iLER.

4 GNS3

En el presente trabajo se realizará la implementación y configuración de una pequeña red en GNS3 con el fin de mostrar la utilización tanto del software de simulación así como poner en práctica lo visto en materia de MPLS y la asignación de etiquetas.

La arquitectura que utilizaremos es la que se muestra a continuación. Más allá de la simpleza que a priori muestra la misma, el objetivo es analizar los protocolos involucrados.

4.1 Arquitectura

La arquitectura a utilizar constará de dos routers PE y dos routers de core P. Adicionalmente tendremos 4 routers ajenos al dominio MPLS, que utilizarán dicho servicio para poder comunicarse entre sí.

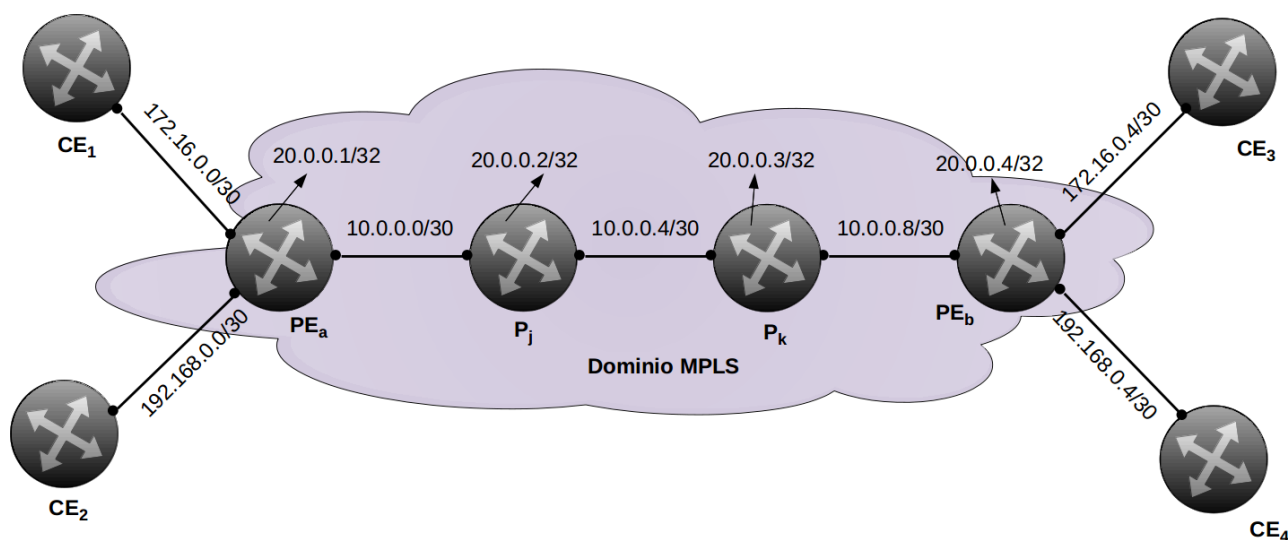


Ilustración 5: Arquitectura

4.2 Direccionamiento IP

Dentro de las buenas prácticas del networking, existe una conducta que evita dolores de cabeza a posteriori: asignar direcciones de loopback únicas a cada router participante en una red.

Las direcciones de loopback son útiles para identificar a los routers. Los distintos protocolos que corren dentro de la red utilizan a la dirección de loopback para identificar a sus vecinos con los cuales mantienen sesiones activas. ¿Por qué una IP de loopback y no una de alguna otra interfaz? La respuesta es simple: la IP de loopback está asociada a una interfaz, también, de tipo loopback y ésta siempre está activa. Una interfaz de enlace puede llegar a desactivarse en algún momento y si eso sucediera, su dirección se deshabilitaría, perdiéndose de esta manera la vecindad a nivel protocolo.

En nuestro diseño, el dominio MPLS tendrá dos protocolos que, como dicho, harán uso de

direcciones IP de loopback: OSPF y LDP. Por simplicidad, los routers ajenos a la nube MPLS no serán configurados en esta ocasión con dicho tipo de direcciones. En lo que respecta a los routers CE, cada uno de ellos tendrá una puerta de enlace por defecto a través del router PE correspondiente.

Las subredes a ser utilizadas están descriptas en la imagen pero se detallan para mayor claridad en la siguiente tabla:

Router	Interfaz	Tipo	IP
PE _a	loop0	Loopback	20.0.0.1/32
PA _b	loop0	Loopback	20.0.0.2/32
P _j	loop0	Loopback	20.0.0.3/32
P _k	loop0	Loopback	20.0.0.4/32
PE _a	to_CE1	FastEthernet	172.16.0.2/30
PE _a	to_CE2	FastEthernet	192.168.0.2/30
PE _a	to_PJ	FastEthernet	10.0.0.1/30
PA _b	to_CE3	FastEthernet	172.16.0.5/30
PA _b	to_CE4	FastEthernet	192.168.0.5/30
PA _b	to_PK	FastEthernet	10.0.0.10/30
P _j	to_PK	FastEthernet	10.0.0.5/30
P _j	to_PEA	FastEthernet	10.0.0.2/30
P _k	to_PJ	FastEthernet	10.0.0.6/30
P _k	to_PEB	FastEthernet	10.0.0.9/30
CE ₁	to_PEA	FastEthernet	172.16.0.1/30
CE ₂	to_PEA	FastEthernet	192.168.0.1/30
CE ₃	to_PEB	FastEthernet	172.16.0.6/30
CE ₄	to_PEB	FastEthernet	192.168.0.6/30

Tabla 2: Direcccionamiento IP

4.3 Construcción de la red en GNS3

Lo primero a realizar será construir la red dentro del simulador¹. Para ello se ejecutará GNS3 y se creará un proyecto nuevo. Una vez que el simulador esté en ejecución, comenzaremos a construir la red, disponiendo los routers sobre el panel principal.

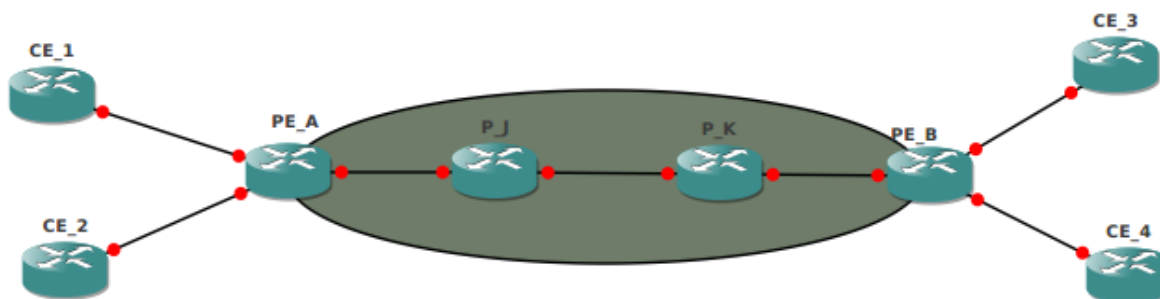


Ilustración 6: Arquitectura en GNS3

4.4 Configuración

Dentro de lo que comprende la configuración básica de los routers, se entiende que hay que establecer el hostname, sólo por una cuestión de prolijidad, además de las interfaces necesarias para que los protocolos a utilizar puedan dialogar entre sí.

4.4.1 Establecimiento del hostname

Cada equipo, para facilidad de operación, deberá disponer de un nombre apropiado. A modo de ejemplo, se muestra cómo cambiar el nombre al router PE_A.

```
router1>enable
router1#configure terminal
router1(config)#hostname PE_A
```

Observar cómo cambia el contexto conforme se van ejecutando los comandos. Este procedimiento se realizará para todos los equipos, utilizando el nombre apropiado.

4.4.2 Interfaces de loopback

Según lo visto en apartados anteriores, cada router participante en la nube MPLS, deberá disponer de una interfaz de tipo loopback con una dirección IP apropiada. A modo de ejemplo se muestra la configuración de la interfaz de loopback para el router PE_A.

```
PE_A>enable
```

¹ Para detalles técnicos respecto del uso de GNS3 y sobre cómo agregar IOS, cargar routers y construir la red, por favor ver el Apéndice al respecto. En estos apartados nos enfocaremos meramente en lo que refiere a configuración de los routers y en cómo utilizar los protocolos.

```
PE_A#configure terminal
PE_A(config)#interface loopback 0
PE_A(config-if)#ip address 20.0.0.1 255.255.255.252
PE_A(config-if)#no shutdown
```

La máscara de subred en cada caso tiene que ir en formato de octetos separados por puntos. Recordar habilitar la interfaz mediante el comando 'no shutdown'.

4.4.3 Interfaces de enlace

Las interfaces de enlace son aquellas que conectan un router con sus routers vecinos. Habrá más o menos interfaces de acuerdo a la cantidad de enlaces disponibles. En todos los casos del presente trabajo, las interfaces son de tipo FastEthernet. Siempre siguiendo los puntos anteriores, a modo de ejemplo, se configurarán las interfaces del router PE_A.

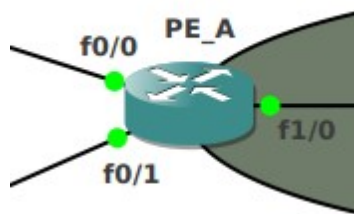


Ilustración 7: Enlaces PE_A

La configuración se realizará teniendo en cuenta las direcciones IP de la Tabla 2.

```
PE_A(config)#interface f0/0
PE_A(config-if)#ip address 172.16.0.2 255.255.255.252
PE_A(config-if)#no shutdown
```

```
PE_A(config)#interface f0/1
PE_A(config-if)#ip address 192.168.0.2 255.255.255.252
PE_A(config-if)#no shutdown
```

```
PE_A(config-if)#interface f1/0
PE_A(config-if)#ip address 10.0.0.1 255.255.255.252
PE_A(config-if)#no shutdown
```

Al ejecutar el comando 'show ip interface brief' se podrá ver entonces el estado de cada interfaz y sus direcciones asignadas:

```
PE_A#show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	172.16.0.2	YES	manual	up	up
FastEthernet0/1	192.168.0.2	YES	manual	up	up
FastEthernet1/0	10.0.0.1	YES	manual	up	up
Loopback0	20.0.0.1	YES	manual	up	up

De esta manera se observa que todas las interfaces han sido configuradas apropiadamente. El resto de los routers deberán ser configurados de acuerdo a estos puntos antes de proseguir con la configuración de OSPF, LDP y, finalmente, MPLS.

4.4.4 Routing

Una vez que las interfaces, tanto de loopback como de enlace, han sido configuradas, se procederá a la configuración del routing. Antes de continuar es recomendable verificar el estado de las interfaces mediante el comando 'show ip interface brief'.

4.4.4.1 Routing estático

Como se ha dicho, para el caso de los routers de los clientes, CE_x , el routing sólo comprenderá la instalación de una ruta estática, cuyo siguiente salto será el PE apropiado. A modo de ejemplo se muestra la configuración de CE_1 .

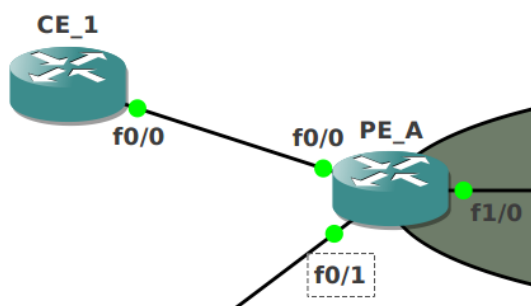


Ilustración 8: Ruta estática CE_1

```
CE_1#configure terminal
```

```
CE_1(config)#ip route 0.0.0.0 0.0.0.0 172.16.0.2
```

4.4.4.2 OSPF

Sólo los routers dentro del dominio MPLS tendrán habilitado el protocolo de enrutamiento dinámico OSPF. Una buena práctica de configuración habilitará OSPF individualmente por red, y cada una llevará como parámetro de configuración la máscara wildcard². Las redes donde OSPF

² La máscara wildcard indica qué partes de una dirección IP han de tenerse en cuenta para análisis. En este caso, no

está habilitado, son anunciadas automáticamente. Adicionalmente, cada router con OSPF, deberá tener como router-id el valor de la IP de loopback.

Por ser un dominio pequeño, sólo se utilizará el área backbone para implementar OSPF.

A modo de ejemplo, se muestra la configuración para el router PE_A.

```
PE_A(config)#router ospf 1
PE_A(config-router)#router-id 20.0.0.1
PE_A(config-router)#network 10.0.0.0 0.0.0.3 area 0
PE_A(config-router)#redistribute connected subnets
```

Como OSPF sólo se habilitará hacia el dominio MPLS mediante el comando 'network', se debe indicar además que anuncie a sus vecinos las subredes directamente conectadas mediante el comando 'redistribute connected subnets'.

4.4.5 MPLS

Habilitar MPLS en routers Cisco, comprende dos partes fundamentales: a) habilitar MPLS globalmente en el equipo indicando a su vez qué protocolo de señalización de etiquetas se utilizará, y b) habilitar MPLS por se, en cada interfaz que lo requiera [10].

Estas configuraciones sólo tienen sentido en los routers cuya función sea la de PE o P.

4.4.5.1 MPLS

En este apartado se indica cómo ha de habilitarse tanto MPLS como LDP a nivel global.

El comando 'mpls ip' habilita MPLS lo que permite el intercambio de etiquetas, en función de lo que dicte la LFIB. El comando 'mpls label protocol ldp' habla por sí solo y establece LDP como protocolo de intercambio de etiquetas de manera global. Una vez que LDP se ha habilitado de manera global el solo hecho de habilitar MPLS asume por defecto el uso de éste protocolo para intercambiar etiquetas.

```
PE_A>enable
PE_A#configure terminal
PE_A(config)#mpls ip
PE_A(config)#mpls label protocol ldp
```

4.4.5.2 LDP

Al igual que OSPF, LDP y MPLS sólo tienen sentido dentro del dominio MPLS. Es decir, se habilitará el intercambio de etiquetas sólo en las interfaces que pertenecen a la nube [10]. La configuración a continuación muestra a modo de ejemplo lo habilitado en P_j, no obstante debe repetirse lo mismo en el resto de los routers participantes.

es otra cosa que el número correspondiente a la máscara de subred, el cual ha sido invertido. La lógica de éste tipo de máscara es que aquellos valores en '0' deben coincidir exactamente.


```
P_J#configure terminal
P_J(config)#interface fastEthernet 0/0
P_J(config-if)#mpls ip
P_J(config-if)#exit
P_J(config)#interface fastEthernet 0/1
P_J(config-if)#mpls ip
```

A partir de este momento, la red empezará a distribuir etiquetas MPLS por cada FEC existente y los routers PE de entrada harán su trabajo de insertar etiquetas cada vez que exista una relación etiqueta:FEC para un destino IP conocido por cada paquete IP entrante. Los routers PE de salida, removerán las etiquetas

Penultimate-hop-popping

Como se vio en la Tabla 1, existen ciertos valores reservados a la hora de asignar etiquetas por FEC. Por defecto, los routers eLER hacen uso de la funcionalidad PHP. Sólo a modo de ilustrar el comportamiento de remoción de etiquetas en los eLER, se deshabilitará PHP. Cuando PHP no se usa, los eLER señalizan sus FEC con el valor 0 (a diferencia del valor 3, que es el valor que se utiliza cuando PHP sí está activo). De cualquier manera, siempre que se pueda, es recomendable utilizar la funcionalidad PHP.

La configuración se aplica en PE_B, pero también debe hacerse lo propio en PE_A.

```
PE_B#configure terminal
PE_B(config)#mpls ldp explicit-null
```

4.5 Análisis de protocolos

Esta sección comenzará con la verificación de lo configurado en el apartado anterior. Se iniciará con la verificación de OSPF, mas no se lo analizará en profundidad puesto que escapa al alcance de este trabajo y con sólo saber que las tablas de enrutamiento han convergido, es suficiente.

Se continuará, sí, con el análisis de LDP, para entender cómo es que se sucede la asignación de etiquetas.

Finalmente, se realizarán pruebas de extremo a extremo para ver cómo lo asignado por LDP realmente sucede a la hora de enviar tráfico de usuario.

4.5.1 Verificar OSPF

Se vio con anterioridad que LDP basa su comportamiento en lo que haga OSPF primeramente. Por eso es de vital importancia comprobar que OSPF está funcionando de manera correcta.

El comando que nos indica que OSPF ha establecido vecindad correctamente, es 'show ip ospf neighbor'. Se muestra la salida del router P_J.

```
P_J#sh ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
20.0.0.4	1	FULL/DR	00:00:38	10.0.0.6	FastEthernet0/1
20.0.0.1	1	FULL/BDR	00:00:39	10.0.0.1	FastEthernet0/0

Esta información muestra que P_J tiene dos vecindades: a) contra 20.0.0.4 vía la interfaz con IP 10.0.0.6 y b) contra 20.0.0.1 vía la interfaz con IP 10.0.0.1. A su vez, es importante notar que ambas vecindades están en estado FULL/[B]DR. Que una sea DR y la otra BDR no representa ningún problema: esto sólo indica cuál es la función del router en cada segmento en particular³.

Si la vecindad no está en estado FULL, entonces hay un problema y se debe verificar la configuración.

La configuración de OSPF incluye el comando 'redistribute connected subnets'. Esto indica a OSPF que debe anunciar hacia el dominio información de enrutamiento para las redes directamente conectadas. Para verificarlo simplemente vemos la tabla de routing en PE_B y vemos si hay información para llegar a las rutas detrás de PE_A. La recíproca debe ser válida.

```
PE_B#show ip route
```

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

```
Gateway of last resort is not set
```

```

    20.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
O E2   20.0.0.4/32 [110/20] via 10.0.0.9, 01:16:50, FastEthernet0/0
O E2   20.0.0.0/30 [110/20] via 10.0.0.9, 01:16:50, FastEthernet0/0
C       20.0.0.2/32 is directly connected, Loopback0
O E2   20.0.0.3/32 [110/20] via 10.0.0.9, 01:16:50, FastEthernet0/0
    172.16.0.0/30 is subnetted, 2 subnets
C       172.16.0.4 is directly connected, FastEthernet0/1
O E2   172.16.0.0 [110/20] via 10.0.0.9, 01:16:50, FastEthernet0/0
    10.0.0.0/30 is subnetted, 3 subnets
C       10.0.0.8 is directly connected, FastEthernet0/0
O       10.0.0.0 [110/30] via 10.0.0.9, 01:16:50, FastEthernet0/0
O       10.0.0.4 [110/20] via 10.0.0.9, 01:16:50, FastEthernet0/0
    192.168.0.0/30 is subnetted, 2 subnets
```

³ En un segmento de multiacceso de tipo broadcast, cada router participante en la señalización OSPF puede tomar una de tres funciones: DR, BDR o pasivo. En este caso, por haber solamente dos routers por segmento, uno será DR y el otro indefectiblemente BDR.

```
O E2    192.168.0.0 [110/20] via 10.0.0.9, 01:16:51, FastEthernet0/0
C       192.168.0.4 is directly connected, FastEthernet1/0
```

Observar que existen rutas de tipo E2 recibidas por OSPF hacia 172.16.0.0/30 y 192.168.0.0/30, lo cual es correcto.

4.5.2 Análisis de LDP

LDP es el protocolo encargado de armar las tablas de mapeo o bindings entre etiquetas y FECs. Cada LSR que tenga LDP habilitado se dará a conocer a la red, a modo de anuncio propio, en base a mensajes Hello. Estos mensajes Hello se envían a la dirección multicast 224.0.0.2 siendo la IP de origen la de la interfaz por la que el mensaje ha salido. Sin embargo, y no menos importante, dentro del paquete, el identificador del LSR es la IP de loopback. Una vez advertido el vecino, y luego de un intercambio de mensajes de tipo Init -ahora sí, entre las IP de loopback-, tendrá lugar el intercambio de bindings [11].

En la siguiente imagen -capturada con Wireshark [12] dentro de GNS3- se aprecian los mensajes de señalización y el anuncio de bindings entre P_J y su vecino, P_K . Cabe recordar que P_J tiene como IP de loopback a 20.0.0.3 mientras que P_K a 20.0.0.4.

Source	Destination	Protocol	Info
10.0.0.6	224.0.0.2	LDP	Hello Message
10.0.0.5	224.0.0.2	LDP	Hello Message
20.0.0.4	20.0.0.3	LDP	Initialization Message
20.0.0.3	20.0.0.4	LDP	Initialization Message Keep Alive Message
20.0.0.4	20.0.0.3	LDP	Address Message Label Mapping Message Label
20.0.0.3	20.0.0.4	LDP	Address Message Label Mapping Message Label

Ilustración 9: Señalización LDP

```
Ethernet II, Src: c4:03:1d:2f:00:01 (c4:03:1d:2f:00:01), Dst: IPv4mcast_00:00:02 (01:00:5e:00:00:02)
Internet Protocol Version 4, Src: 10.0.0.5 (10.0.0.5), Dst: 224.0.0.2 (224.0.0.2)
User Datagram Protocol, Src Port: ldp (646), Dst Port: ldp (646)
Label Distribution Protocol
  Version: 1
  PDU Length: 30
  LSR ID: 20.0.0.3 (20.0.0.3)
  Label Space ID: 0
  ► Hello Message
```

Ilustración 10: LDP Mensaje Hello

Los mensajes Label Mapping Message son aquellos que llevan dentro de sí, propiamente dicho, los bindings entre FECs y etiquetas. A continuación se aprecia el detalle del binding anunciando acerca de la FEC 192.168.0.4/30. Se puede observar que el label informado para dicho FEC es 23.

```

Label Mapping Message
  0... .... = U bit: Unknown bit not set
  Message Type: Label Mapping Message (0x400)
  Message Length: 24
  Message ID: 0x0000001c
  Forwarding Equivalence Classes TLV
    00.. .... = TLV Unknown bits: Known TLV, do not Forward (0x00)
    TLV Type: Forwarding Equivalence Classes TLV (0x100)
    TLV Length: 8
  FEC Elements
    FEC Element 1
      FEC Element Type: Prefix FEC (2)
      FEC Element Address Type: IPv4 (1)
      FEC Element Length: 30
      Prefix: 192.168.0.4
  Generic Label TLV
    00.. .... = TLV Unknown bits: Known TLV, do not Forward (0x00)
    TLV Type: Generic Label TLV (0x200)
    TLV Length: 4
    Generic Label: 23

```

Ilustración 11: LDP Label Mapping Message

4.5.2.1 Penultimate-hop-popping

En el siguiente apartado se verá cómo a partir del comando "mpls ldp explicit-null" se instruye a PE_B que señalice la red 192.168.0.4/30 con un label explícito usando la etiqueta 0 en lugar de la 3, como se indica en la Tabla 1.

Gráficamente, se aprecia en el siguiente trace realizado desde CE₁ hacia CE₃, antes y después del cambio.

```

CE_1#traceroute 192.168.0.6

Type escape sequence to abort.
Tracing the route to 192.168.0.6

 1 172.16.0.2 28 msec 20 msec 4 msec
 2 10.0.0.2 [MPLS: Label 23 Exp 0] 12 msec 28 msec 24 msec
 3 10.0.0.6 [MPLS: Label 23 Exp 0] 16 msec 12 msec 12 msec
 4 10.0.0.10 12 msec 28 msec 16 msec
 5 192.168.0.6 20 msec 24 msec *
```

Se puede apreciar que el último salto no incluye etiqueta MPLS. A continuación se observa lo mismo desde el punto de vista de la señalización LDP.

```

Label Mapping Message
  0... .... = U bit: Unknown bit not set
  Message Type: Label Mapping Message (0x400)
  Message Length: 24
  Message ID: 0x00000026
  Forwarding Equivalence Classes TLV
    00.. .... = TLV Unknown bits: Known TLV, do not Forward (0x00)
    TLV Type: Forwarding Equivalence Classes TLV (0x100)
    TLV Length: 8
  FEC Elements
    FEC Element 1
      FEC Element Type: Prefix FEC (2)
      FEC Element Address Type: IPv4 (1)
      FEC Element Length: 30
      Prefix: 192.168.0.4
  Generic Label TLV
    00.. .... = TLV Unknown bits: Known TLV, do not Forward (0x00)
    TLV Type: Generic Label TLV (0x200)
    TLV Length: 4
    Generic Label: 3

```

Ilustración 12: PHP Implicit Null

Luego de realizar el cambio, el trace muestra que el último salto antes de PE_B, lleva la etiqueta MPLS 0:

```

CE_1#traceroute 192.168.0.6

Type escape sequence to abort.
Tracing the route to 192.168.0.6

 1 172.16.0.2 20 msec 20 msec 4 msec
 2 10.0.0.2 [MPLS: Label 23 Exp 0] 16 msec 12 msec 12 msec
 3 10.0.0.6 [MPLS: Label 23 Exp 0] 12 msec 36 msec 16 msec
 4 10.0.0.10 [MPLS: Label 0 Exp 0] 12 msec 20 msec 24 msec
 5 192.168.0.6 12 msec 36 msec *
```

Finalmente, desde el punto de vista de la señalización LDP, se observa que el prefijo 192.168.0.4/30 es anunciado con la etiqueta 0.

```

Label Mapping Message
  0... .... = U bit: Unknown bit not set
  Message Type: Label Mapping Message (0x400)
  Message Length: 24
  Message ID: 0x0000002d
  Forwarding Equivalence Classes TLV
    00.. .... = TLV Unknown bits: Known TLV, do not Forward (0x00)
    TLV Type: Forwarding Equivalence Classes TLV (0x100)
    TLV Length: 8
  FEC Elements
    FEC Element 1
      FEC Element Type: Prefix FEC (2)
      FEC Element Address Type: IPv4 (1)
      FEC Element Length: 30
      Prefix: 192.168.0.4
  Generic Label TLV
    00.. .... = TLV Unknown bits: Known TLV, do not Forward (0x00)
    TLV Type: Generic Label TLV (0x200)
    TLV Length: 4
    Generic Label: 0

```

Ilustración 13: PHP Explicit Null

De esta manera es que cada FEC es anunciado a la red y asociado a una etiqueta, la cual, como se ha dicho, no es más que el inicio -en caso de ser la primera- o parte de un túnel de transporte o LSP -en caso de ser una etiqueta intermedia-.

4.5.3 Verificar MPLS punta a punta

Una vez que todos los LSP han sido establecidos, el tráfico de usuario debe poder utilizarlos. Cuando el paquete IP llega a un iLER se debe, en función de la FEC, encontrar el LSP apropiado, agregarle la etiqueta MPLS de dicho LSP y enviarlo al siguiente salto.

Siguiendo con el ejemplo anterior, se recuerda el camino desde CE₁ hasta CE₃:

```

CE_1#traceroute 192.168.0.6

Type escape sequence to abort.
Tracing the route to 192.168.0.6

 1 172.16.0.2 20 msec 20 msec 4 msec
 2 10.0.0.2 [MPLS: Label 23 Exp 0] 16 msec 12 msec 12 msec
 3 10.0.0.6 [MPLS: Label 23 Exp 0] 12 msec 36 msec 16 msec
 4 10.0.0.10 [MPLS: Label 0 Exp 0] 12 msec 20 msec 24 msec
 5 192.168.0.6 12 msec 36 msec *

```

A continuación se analizarán las tablas de conmutación de cada uno de los routers, desde PE_A hasta PE_B, teniendo presente que la IP de destino es en un primer momento 192.168.0.6 pero para la respuesta lo es 172.16.0.1. Esto quiere decir que el LSP de ida, que va desde PE_A a PE_B, puede no ser el mismo que aquél de retorno.

Cuando un router recibe un paquete, en función del campo Type de la capa L2 [7], sabe si debe

analizarlo a nivel IP o a nivel MPLS para poder conmutarlo [10][13]. Como el PE_A recibe una trama Ethernet que transporta a un paquete IP, entonces el análisis es ligeramente diferente para la entrada. Lo mismo sucede para la respuesta, en PE_B.

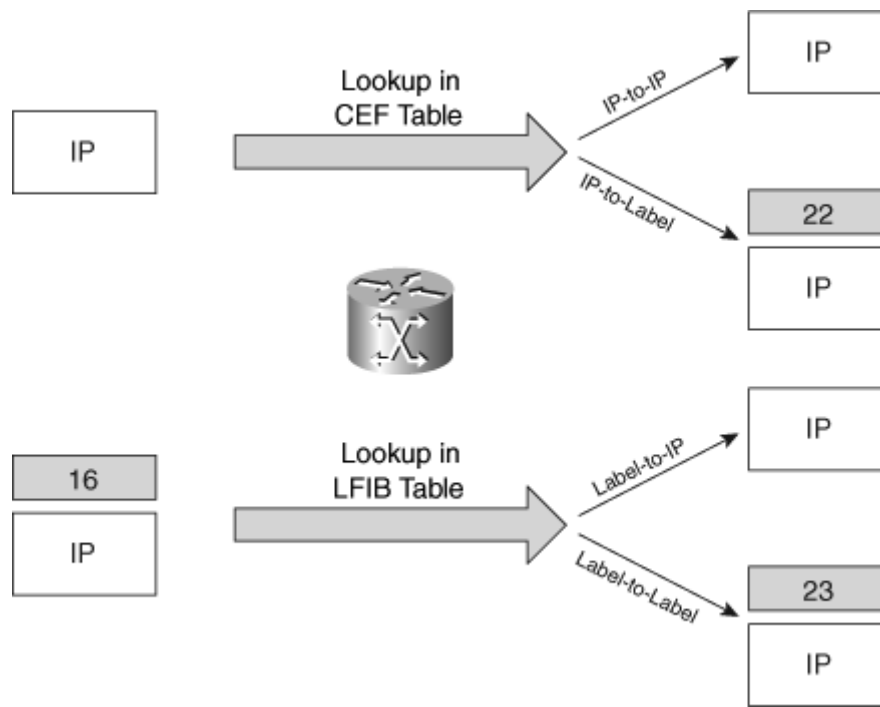


Ilustración 14: IP or MPLS lookup

En función de lo mencionado, el comando 'show ip cef 192.168.0.6' en PE_A, nos dirá qué etiqueta se utilizará para luego conmutar el paquete. En el resto de los routers el comando 'show mpls forwarding-table' hará el trabajo.

```
PE_A#show ip cef 192.168.0.6
192.168.0.4/30, version 31, epoch 0, cached adjacency 10.0.0.2
0 packets, 0 bytes
tag information set
  local tag: 21
  fast tag rewrite with Fa1/0, 10.0.0.2, tags imposed: {23}
via 10.0.0.2, FastEthernet1/0, 0 dependencies
  next hop 10.0.0.2, FastEthernet1/0
  valid cached adjacency
  tag rewrite with Fa1/0, 10.0.0.2, tags imposed: {23}
```

```
P_J#show mpls forwarding-table
```

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	Pop tag	10.0.0.8/30	0	Fa0/1	10.0.0.6
17	Pop tag	20.0.0.4/32	0	Fa0/1	10.0.0.6
18	0	20.0.0.0/30	0	Fa0/0	10.0.0.1
19	19	20.0.0.2/32	0	Fa0/1	10.0.0.6
20	0	172.16.0.0/30	1822	Fa0/0	10.0.0.1
21	21	172.16.0.4/30	0	Fa0/1	10.0.0.6
22	0	192.168.0.0/30	0	Fa0/0	10.0.0.1
23	23	192.168.0.4/30	1536	Fa0/1	10.0.0.6

```
P_K#show mpls forwarding-table
```

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	Pop tag	10.0.0.0/30	0	Fa0/0	10.0.0.5
17	Pop tag	20.0.0.3/32	0	Fa0/0	10.0.0.5
18	18	20.0.0.0/30	0	Fa0/0	10.0.0.5
19	0	20.0.0.2/32	0	Fa0/1	10.0.0.10
20	20	172.16.0.0/30	1822	Fa0/0	10.0.0.5
21	0	172.16.0.4/30	0	Fa0/1	10.0.0.10
22	22	192.168.0.0/30	0	Fa0/0	10.0.0.5
23	0	192.168.0.4/30	834	Fa0/1	10.0.0.10

```
PE_B#show ip cef 192.168.0.6
```

```
192.168.0.6/32, version 30, epoch 0, connected, cached adjacency 192.168.0.6
0 packets, 0 bytes
  via 192.168.0.6, FastEthernet1/0, 0 dependencies
    next hop 192.168.0.6, FastEthernet1/0
    valid cached adjacency
```

De esta forma se comprueba que el LSP que va desde PE_A hacia PE_B está compuesto por las etiquetas 23, 23 y 0, y no es otra cosa que lo que vimos a la salida del comando traceroute anteriormente. De la misma manera, si se consulta a PE_B por el destino 172.16.0.1, se obtendrá lo siguiente:

```
PE_B#show ip cef 172.16.0.1
```

```
172.16.0.0/30, version 28, epoch 0, cached adjacency 10.0.0.9
0 packets, 0 bytes
  tag information set
    local tag: 21
    fast tag rewrite with Fa0/0, 10.0.0.9, tags imposed: {20}
  via 10.0.0.9, FastEthernet0/0, 0 dependencies
    next hop 10.0.0.9, FastEthernet0/0
    valid cached adjacency
```



```
tag rewrite with Fa0/0, 10.0.0.9, tags imposed: {20}
```

Lo que muestra que al conmutar el paquete, lo hace con la etiqueta MPLS 20.

En las siguientes imágenes, se muestra lo capturado con Wireshark para un ping enviado desde CE₁ a CE₃, donde se comprueba una vez más, lo analizado hasta este momento.

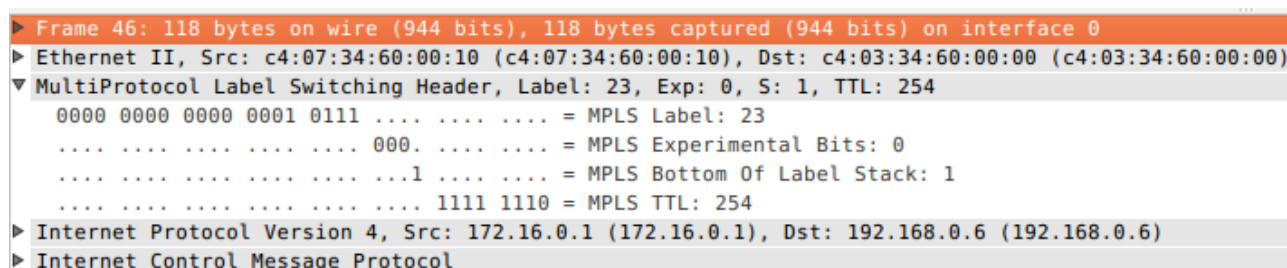


Ilustración 15: LSP Label 1

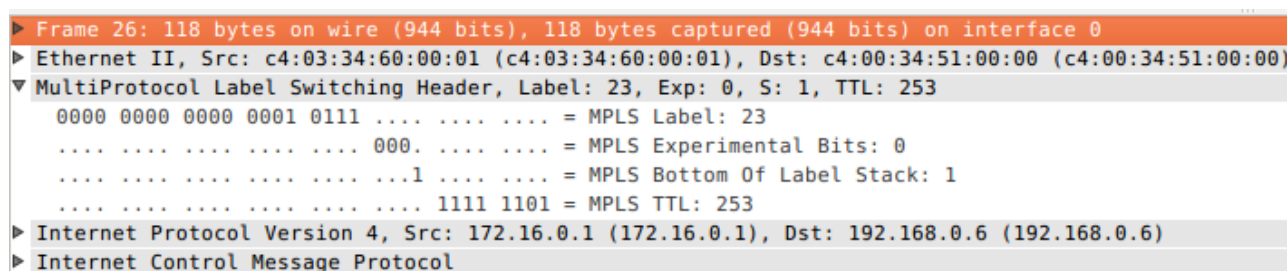


Ilustración 16: LSP Label 2

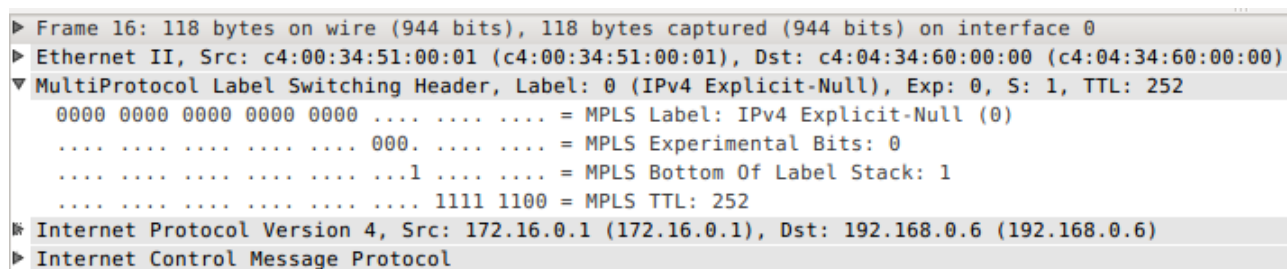


Ilustración 17: LSP Label 3

Observar que la capa L3 se mantiene inalterada a lo largo de todo el LSP. En otras palabras: las direcciones IP de origen y destino no cambian. Lo único que cambia conforme se atraviesan los distintos LSRs son las etiquetas MPLS y el campo TTL, el cual decrementa en 1 a cada salto. Observar también que en el último salto, la etiqueta lleva el valor '0', que es lo esperado, puesto que se ha instruido al router PE_B a que anuncie tal FEC como Explicit-Null.

5 Conclusión

El objeto de este trabajo era demostrar el funcionamiento de MPLS -y otros protocolos necesarios para tal fin- en un entorno virtualizado, entorno generado vía la utilización de una plataforma de simulación como es GNS3 [1].

A lo largo del trabajo se pudo apreciar cómo GNS3 es lo suficientemente flexible para la instalación de routers, los cuales ejecutan sistemas operativos de distintas marcas (aunque en este trabajo se haya utilizado solamente sistemas operativos de Cisco Systems [14]).

Se vio también que, por medio de herramientas basadas en software libre, como Wireshark [12], se pudo hacer análisis de los protocolos involucrados en el diseño realizado, lo cual resulta muy conveniente para poder entender el funcionamiento de los mismos.

Aunque la plataforma de simulación estuvo corriendo sobre hardware de una PC estándar⁴, no hubo limitaciones relevantes a la hora de probar la arquitectura y el buen funcionamiento de los protocolos. Está claro que conforme aumente la complejidad de la red simulada -ya sea por cantidad de protocolos así como por cantidad de elementos-, mayor será el consumo de recursos de hardware, por lo que sería conveniente realizar trabajos futuros con vista al estudio de escalabilidad para este tipo de tareas de simulación. Lo mismo se sugiere cuando se quiera probar el rendimiento de una red simulada, sobre todo a la hora de querer estudiar la respuesta de la red cuando es atacada por un flujo de tráfico intenso, ya que generar tráfico en el mismo equipo donde se simula la red no presenta a las claras la independencia necesaria entre los participantes de la prueba (generador de tráfico/simulador de red). Mayor será el problema cuando se quiera analizar el rendimiento de una red compleja (por tamaño y por cantidad de protocolos).

De cualquier manera, y en pocas palabras, está demostrado que para pruebas de concepto, una PC estándar con GNS3, Wireshark y sistemas operativos de alguna marca -no necesariamente Cisco Systems- son suficientes para poder generar redes simples que permitan estudiar y entender una variedad muy amplia de protocolos, siempre y cuando lo permita el hardware que aloja la plataforma y la red esté acotada en tamaño y complejidad, tal como la del presente trabajo.

4 Ver al final el apartado Especificaciones Técnicas para ver las características técnicas de los elementos usados en la simulación, ya sean de hardware o software.

6 Apéndice A

Este apéndice tiene por objeto explicar el funcionamiento de GNS3, cómo instalarlo, cómo agregar sistemas operativos, cómo agregar routers y cómo dejarlos listos para configurarlos y poder utilizarlos.

6.1 Instalación

Se puede instalar la versión por defecto desde el repositorio (más vieja) o la de la página de GNS3 (más nueva y actualizada). Se recomienda esta última opción.

6.1.1 Desde de la página GNS3

Se puede instalar GNS3 descargando el instalador desde su propia página. Para ello hace falta registrarse y proceder a la instalación: <https://www.gns3.com/software/download>

Una vez registrados, se debe agregar el repositorio de GNS3 e instalarlo como cualquier otra aplicación. De esta forma se obtendrá la última versión estable de GNS3.

```
sudo add-apt-repository ppa:gns3/ppa
sudo apt-get update
sudo apt-get install gns3-gui wireshark
```

Para poder usar Wireshark desde el contexto de GNS3, se debe agregar el propio usuario al grupo de wireshark.

```
sudo adduser $USER wireshark
```

Una vez instalado se procede a su configuración. Tener presente que luego de la instalación hay que cerrar sesión / volver a loguearse -a nivel sistema operativo-, ya que hay cambios de grupos en el usuario.

NOTA: Algunos firmwares/OS requieren que las funcionalidades de virtualización de la BIOS estén activadas.

6.2 Inicio

Lo primero a hacer, es ir al menú Help → GNS3 Doctor. Con ello se verificará que todos los componentes de GNS3 esté funcionando. Si alguno falla, no seguir avanzando hasta solucionarlo.

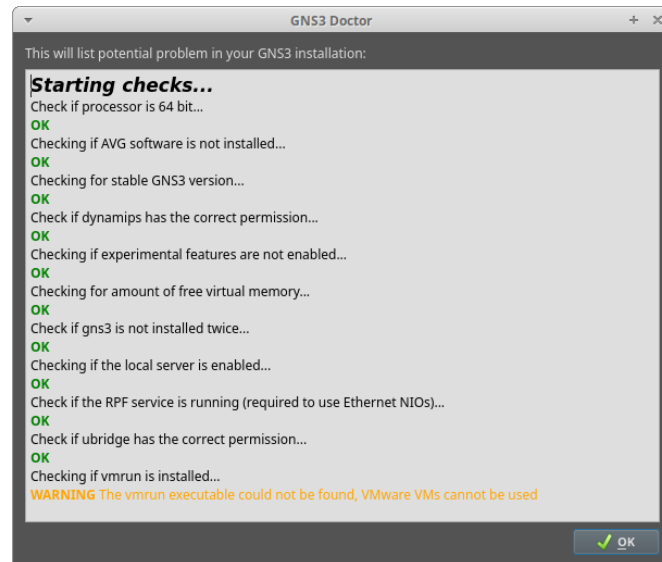


Ilustración 18: GNS3 Doctor

6.3 Carga de Cisco IOS 3745

Como ejemplo se explicará cómo cargar la IOS del router Cisco 3745.

Se procede a cargar la IOS del equipo. Para ello ir a: Edit → Preferences; Dynamips → IOS routers; New → New Image: elegir el folder donde se encuentra la imagen del OS de Cisco 3745. Elegir 256MiB de RAM. Configurar los Slots de la siguiente manera:

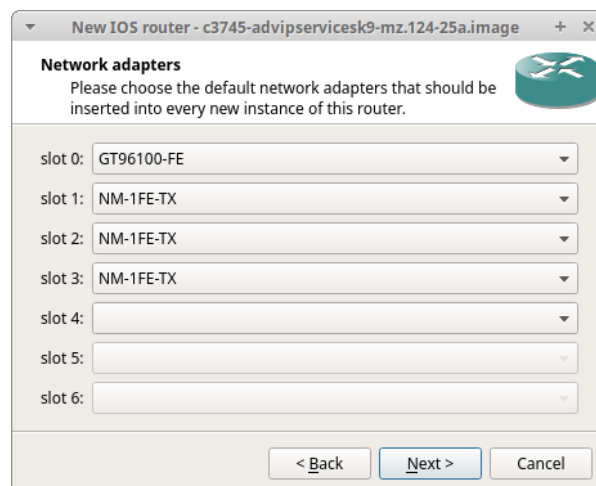


Ilustración 19: Slots

6.4 Idle-PC Number

Una vez configurado el hardware del equipo, se procede a encontrar el idle-PC number para que no consuma tantos recursos de CPU. Al apretar el botón 'Idle-PC Finder' se comenzará el proceso de búsqueda.

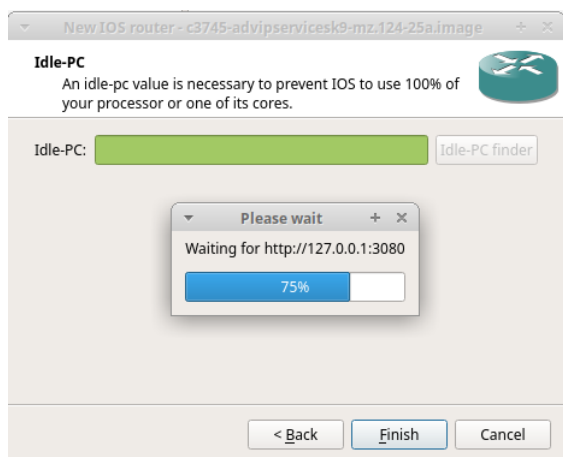


Ilustración 20: Buscando idle-pc

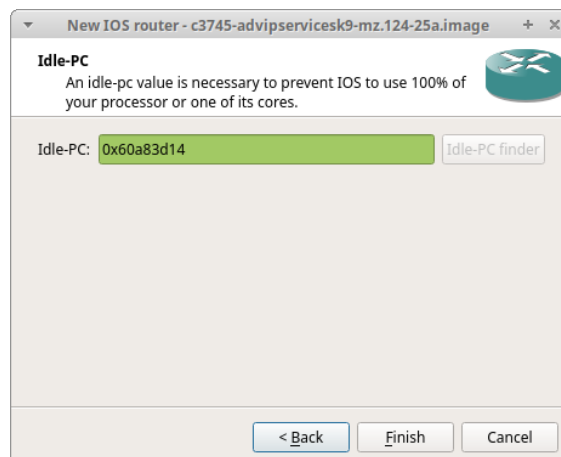
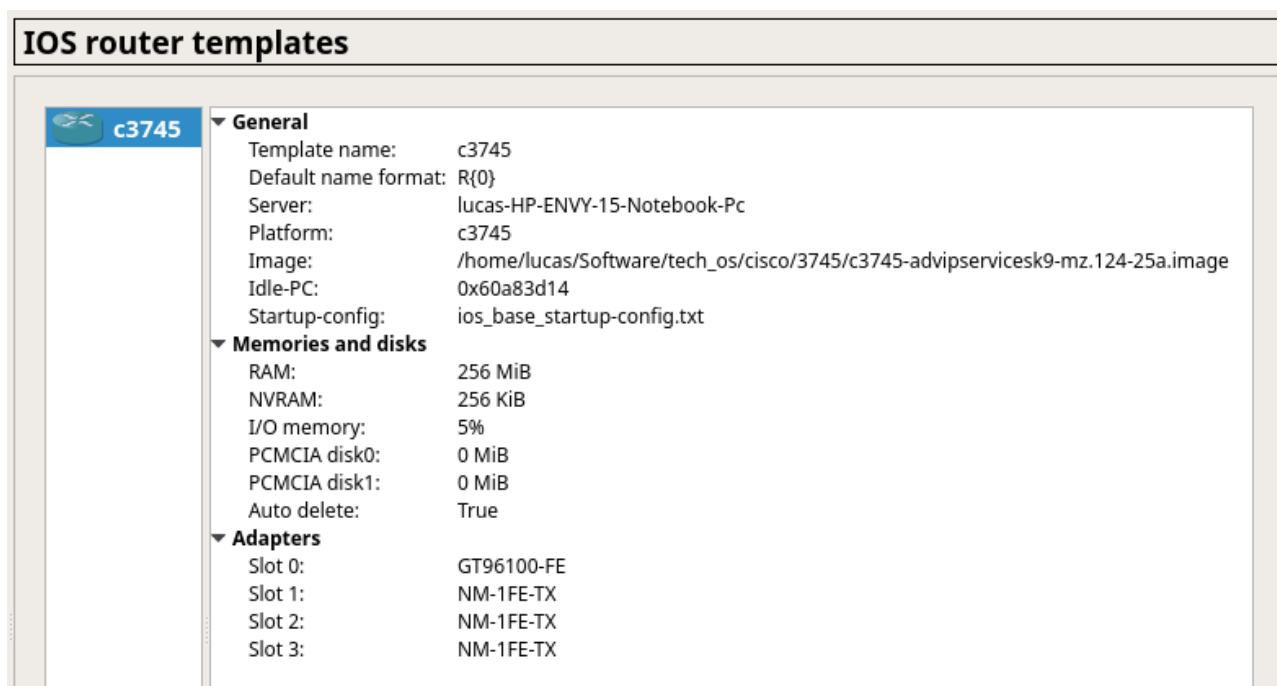


Ilustración 21: idle-pc encontrado

Con esto ya está finalizado el proceso de configuración del IOS. Se debe repetir esto para cada IOS distinta que se quiera utilizar.

En el menú Edit → Preferences; Dynamips → IOS routers podemos ver el estado de las IOS instaladas.



6.5 Creación del primer proyecto

Para empezar a usar los routers agregados, se debe crear un proyecto: File → New Blank Project.

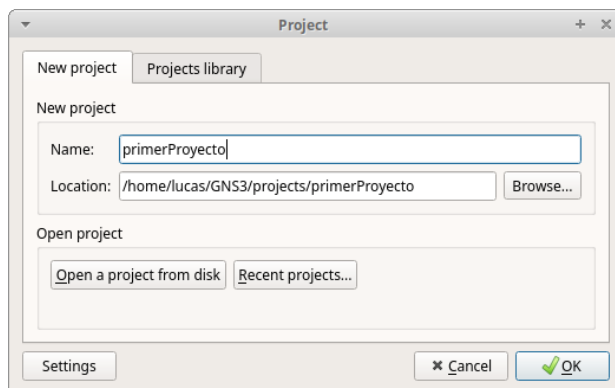


Ilustración 22: Proyecto nuevo

Una vez creado, se procede a agregar tantos routers como se desee. Recordar que sólo tendremos disponible el router C3745.

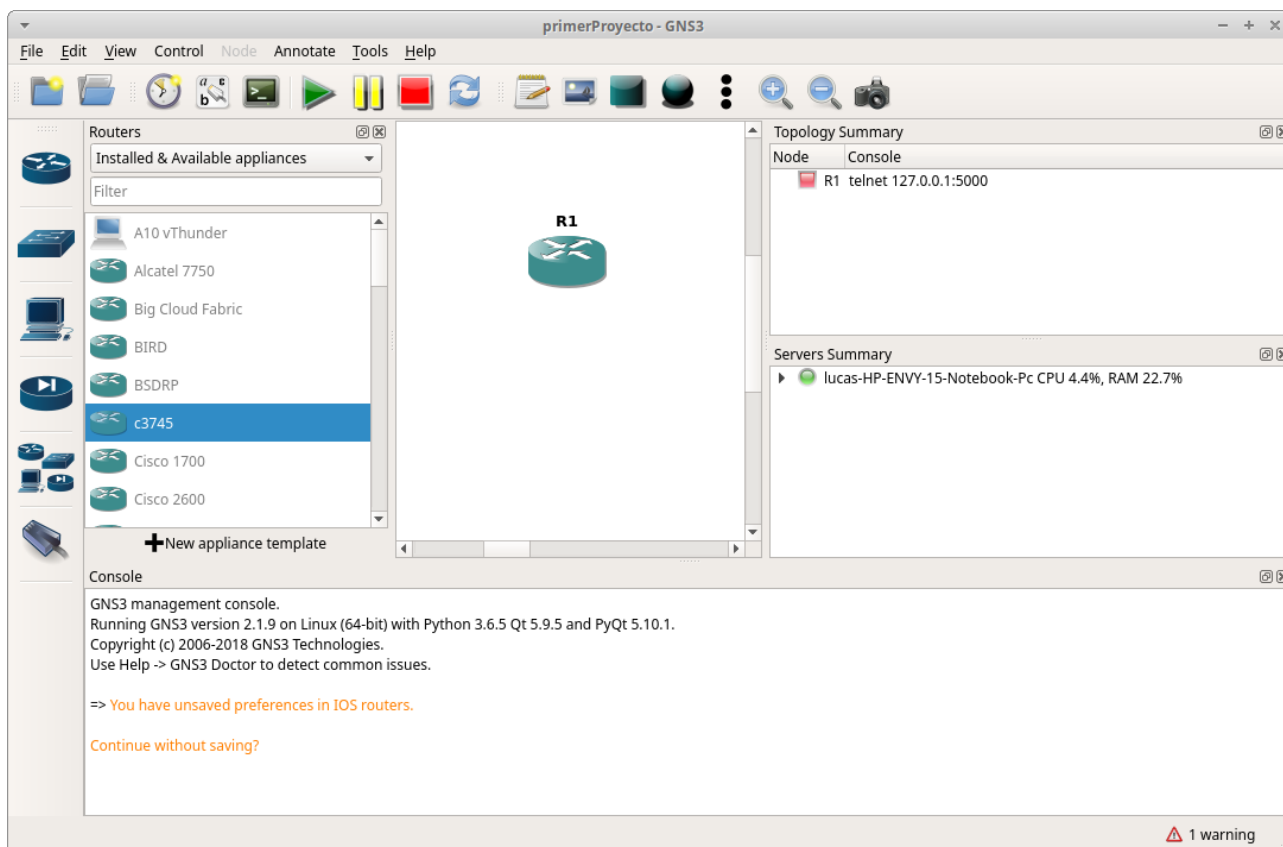


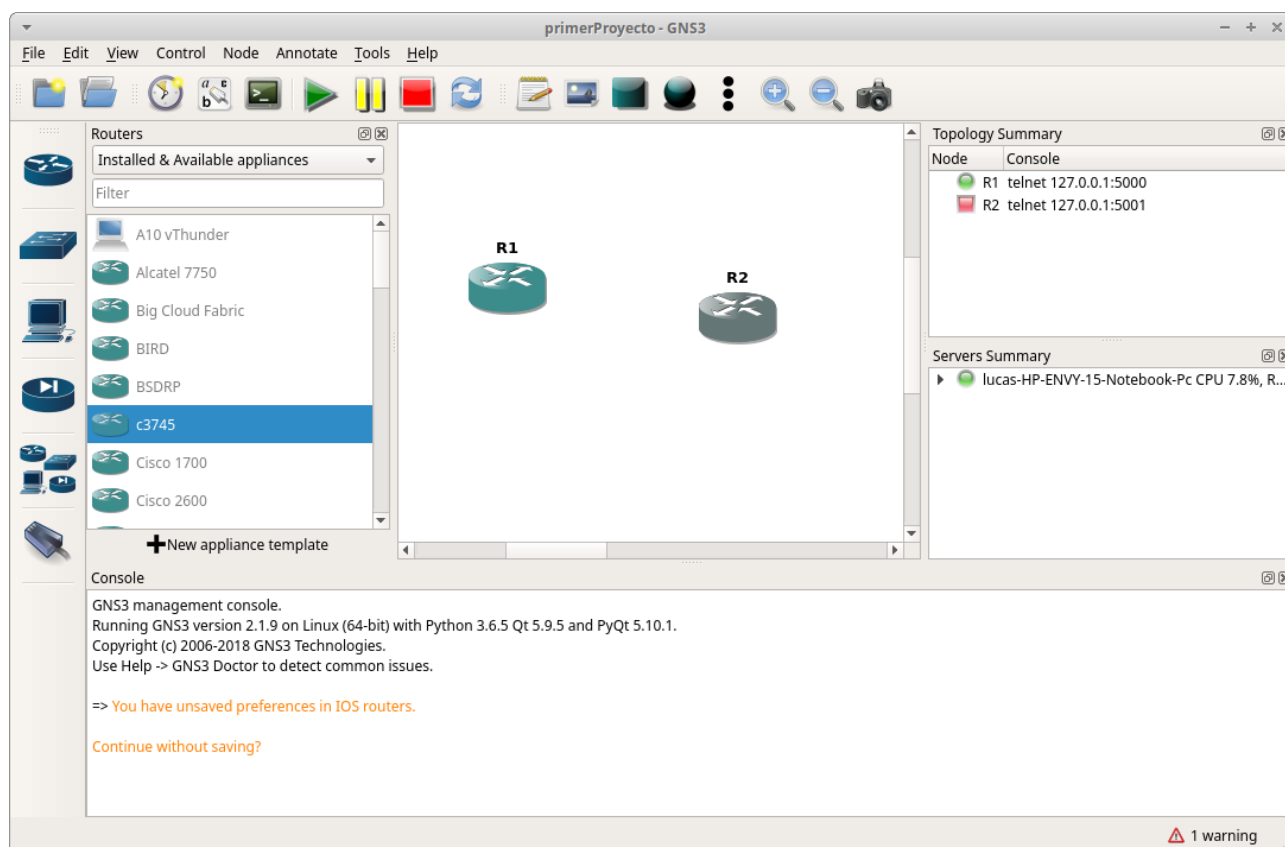
Ilustración 23: Primera topología



Notar que el router está apagado. Se aprecia esto por el indicador rojo sobre la derecha. Para iniciarlo, apretar el botón triangular verde de la barra de herramientas.

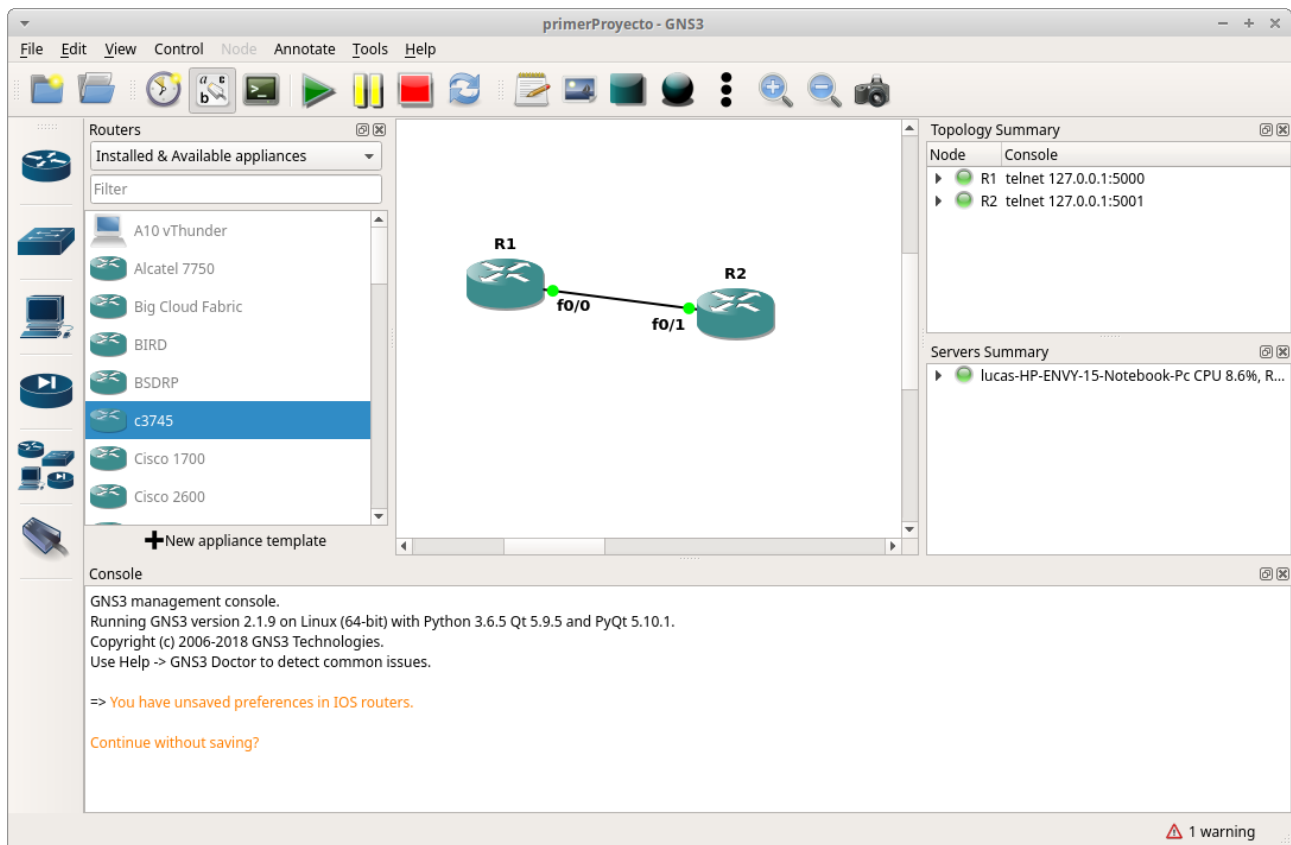
Una vez iniciado, se podrá apreciar que la PC eleva su consumo de CPU (el consumo dependerá de cuántos routers tenga el proyecto) para el proceso Dynamips. Se puede verificar esto mediante el comando 'top' en una consola de Linux. También se puede apreciar el historial de consumo de CPU mediante el programa 'System Monitor'.

6.5.1 Interacción con otros routers

El segundo router, una vez agregado, está apagado. Para iniciarlo, hacer click+derecho sobre el mismo y elegir el comando 'start'.



A esta instancia, se deben conectar ambos routers. El botón  permite conectar routers entre sí, permitiendo a su vez elegir las interfaces disponibles en cada uno. Al apretarlo, elegir luego el router1 R1, elegir la interfaz de origen y a continuación elegir el router2 R2 seleccionando entonces la interfaz de destino. Al apretar el botón observar que el cursor cambia de forma. Al finalizar la conexión, se debe deseleccionar la función .



Especificaciones Técnicas

Dentro de los elementos de hardware y software utilizados en este trabajo, se destacan los siguientes:

La PC donde se ejecutó la simulación con GNS3 estuvo conformada por lo siguiente:

- Hardware: HP Pavilion dv5
 - 4GB de RAM
 - CORE i5 de 4 núcleos @2,67GHz
- Sistema Operativo: Linux Ubuntu 13.04 de 64Bits con kernel 3.8.0-35-generic
- GNS3 versión 0.83 instalada según se informa en Apéndice A
- IOS 3700 Software (C3745-ADVIPSERVICESK9-M), Version 12.4(25a), RELEASE SOFTWARE (fc2)

Referencias

- [1] - <http://www.gns3.net/>
- [2] - <http://tools.ietf.org/html/rfc3032#section-2>
- [3] - <http://tools.ietf.org/search/rfc3031#section-2.1>
- [4] - <http://tools.ietf.org/html/rfc5036#section-3.5.7.1>
- [5] - <http://tools.ietf.org/search/rfc3031#section-3.15>
- [6] - <http://tools.ietf.org/html/rfc3032#section-2>
- [7] - <http://en.wikipedia.org/wiki/EtherType>
- [8] - <http://tools.ietf.org/html/rfc3032#section-2.2>
- [9] - <http://tools.ietf.org/search/rfc3031#section-3.4>
- [10] - <http://www.cisco.com/c/en/us/support/docs/multiprotocol-label-switching-mpls/mpls/13736-mplsospf.html>
- [11] - <http://tools.ietf.org/html/rfc5036#section-1.2>
- [12] - <http://www.wireshark.org/>
- [13] - <http://www.ciscopress.com/articles/article.asp?p=680824>
- [14] - <http://www.cisco.com/>
- [15] - <http://www.gns3.net/documentation/gns3/introduction-to-gns3/>
- [16] - <http://www.gns3.net/documentation/gns3/quick-start-guide-for-linux-users/>
- [17] - <https://help.ubuntu.com/community/Repositories/Ubuntu>
- [18] - <http://www.gns3.net/documentation/gns3/memory-and-cpu-usage/>