

26/06/2019

Redes de areas extensas

Informe sobre VPN MPLS TE



Br. Jorge Vladimir López
Br. Luis Alberto Potosme Aguilera
Br. Erwin Antonio Sandoval

VPN MPLS con ingeniería de tráfico en gns3

Introducción

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

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

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

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

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

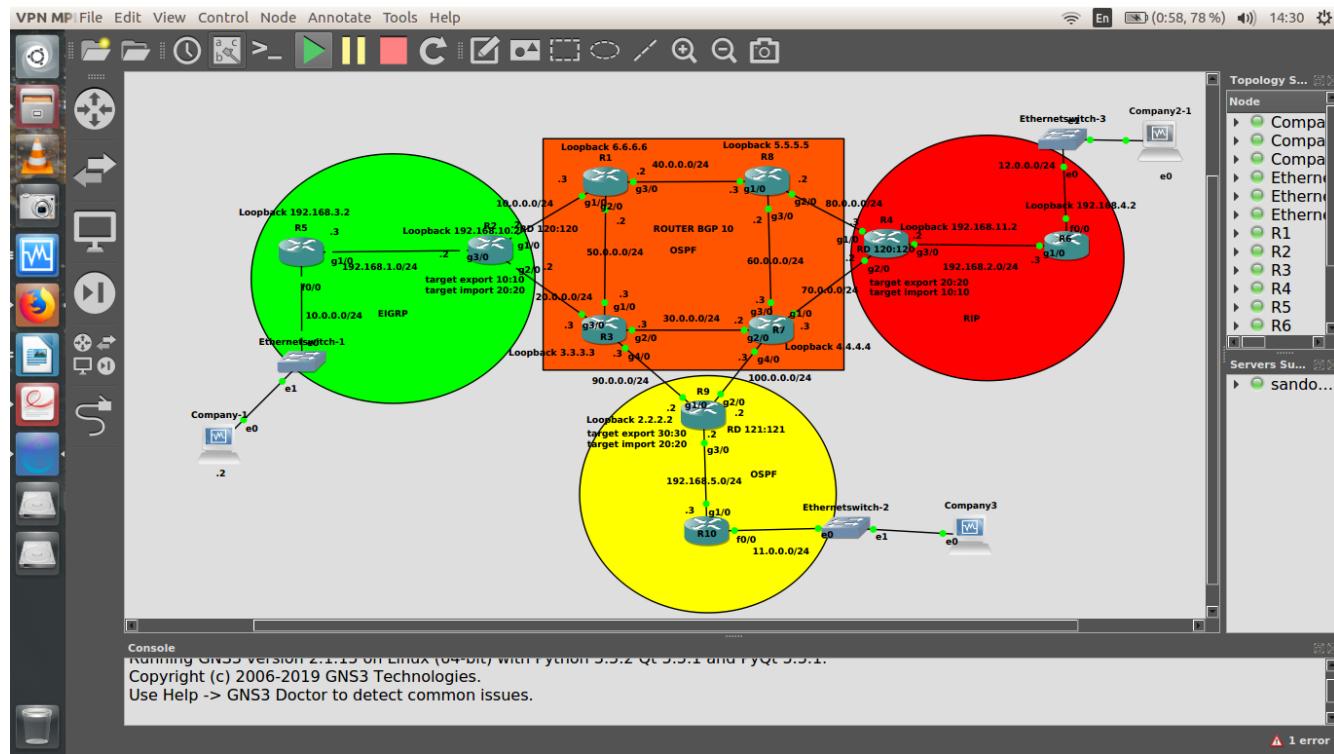
Las **VPN MPLS** son posibles porque el proveedor de servicios dispone de una red MPLS por debajo, que desvincula el plano de control del plano de tráfico lo cual es imposible con una red IP tradicional. En MPLS existen dos tipos de VPN's L2 y L3 VPN que se corresponden con los niveles 2 y 3 de la capa OSI.

Objetivo:

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

Escenario:

en la siguiente imagen se observa la topología de la red a montar.



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

Dispositivo	Interfaz	Direccion IP	Mascarad de red	Gateway
R1	G1/0	10.0.0.3	255.255.255.0	--
	G2/0	50.0.0.2	255.255.255.0	--
	G3/0	40.0.0.2	255.255.255.0	--
	Lo1	6.6.6.6	255.255.255.255	--
R2	G1/0	10.0.0.2	255.255.255.0	--
	G2/0	20.0.0.2	255.255.255.0	--
	G3/0	192.168.1.2	255.255.255.0	--
	Lo1	192.168.10.2	255.255.255.255	--
R3	G1/0	50.0.0.3	255.255.255.0	--
	G2/0	30.0.0.3	255.255.255.0	--
	G3/0	20.0.0.3	255.255.255.0	--
	G4/0	90.0.0.3	255.255.255.0	--
	Lo1	3.3.3.3	255.255.255.255	--

Dispositivo	Interfaz	Direccion IP	Mascarad de red	Gateway
R4	G1/0	80.0.0.3	255.255.255.0	--
	G2/0	70.0.0.2	255.255.255.0	--
	G3/0	192.168.2.2	255.255.255.0	--
	Lo1	192.168.11.2	255.255.255.255	--
R5	G1/0	192.168.1.3	255.255.255.0	--
	F0/0	10.0.0.1	255.255.255.0	--
	Lo1	192.168.3.2	255.255.255.0	--
R6	G1/0	192.168.2.3	255.255.255.0	--
	F0/0	12.0.0.1	255.255.255.0	--
	Lo1	192.168.4.2	255.255.255.0	--
R7	G1/0	70.0.0.3	255.255.255.0	--
	G2/0	30.0.0.2	255.255.255.0	--
	G3/0	60.0.0.3	255.255.255.0	--
	G4/0	100.0.0.3	255.255.255.0	--
	Lo1	4.4.4.4	255.255.255.255	--
R8	G1/0	40.0.0.3	255.255.255.0	--
	G2/0	80.0.0.2	255.255.255.0	--
	G3/0	60.0.0.2	255.255.255.0	--
	Lo1	5.5.5.5	255.255.255.255	--
R9	G1/0	90.0.0.2	255.255.255.0	--
	G2/0	100.0.0.2	255.255.255.0	--
	G3/0	192.168.5.2	255.255.255.0	--
	Lo1	2.2.2.2	255.255.255.255	--
R10	G1/0	192.168.5.3	255.255.255.0	--
	F0/0	11.0.0.1	255.255.255.0	--
Company-1	e0	10.0.0.2	255.255.255.0	10.0.0.1
Company2-1	e0	12.0.0.2	255.255.255.0	12.0.0.1
Company3	e0	11.0.0.3	255.255.255.0	11.0.0.1

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

Para establecer la VPN y enlazar los PE's con los CE's, se utilizará el protocolo eBGP, que nos permitirá intercambiar la información de routing entre los diferentes sistemas autónomos, en la frontera formada por las interfaces G3/0 de ambos routers PE, donde se crean las tablas VRF, que actúan como un router lógico, permitiendo definir caminos virtuales entre la sede central y la sucursal remota. Así pues, se crearán dos VRF's, a las que llamaremos Cliente1. En primer lugar, comenzaremos asignando las direcciones a los interfaces del backbone, en el caso del router PE1, los comandos a introducir serían:

```
R2#configure terminal  
R2(config)#hostname PE1  
PE1(config)#interface Loopback 1  
PE1(config-if)#ip address 192.168.10.2 255.255.255.255  
PE1(config-if)#interface g1/0  
PE1(config-if)#ip address 10.0.0.2 255.255.255.0  
PE1(config-if)#no shutdown  
PE1(config-if)#interface g2/0  
PE1(config-if)#ip address 20.0.0.2 255.255.255.0  
PE1(config-if)#no shutdown  
PE1(config-if)#interface g3/0  
PE1(config-if)#ip address 192.168.1.2 255.255.255.252  
PE1(config-if)#no shutdown  
PE1(config-if)#exit
```

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

Continuemos configurando el protocolo IGP dentro del backbone MPLS, en este caso utilizaremos OSPF, para el caso de los routers que conforman el backbone incluyendo los PE's los comandos a introducir son cambiando las direcciones IP en cada router que se configure:

```
R1(config)#router ospf 1  
R1(config-router)#network 40.0.0.0 0.0.0.255 area 0  
R1(config-router)#network 50.0.0.0 0.0.0.255 area 0  
R1(config-router)#network 10.0.0.0 0.0.0.255 area 0  
R1(config-router)#exit  
R1(config)#

```

Es importante aclarar que el loopback de todos los PE's debe estar incluido en la configuración del protocolo IGP ya que estos sirven para identificar el router en la red.

Repetir estos mismos pasos para todos los PE's de la red.

Vamos a proceder a configurar MPLS para ello es necesario habilitar ip cef, mpls ip, de forma general y en cada interfaz perteneciente al backbone MPLS, es decir, también en los routers PE, además del protocolo de distribución de etiquetas. Utilizaremos los siguientes comandos:

```
PE1#configure terminal
PE1(config)#ip cef
PE1(config)#mpls ip
PE1(config)#mpls label protocol ldp
PE1(config)#interface g1/0
PE1(config-if)#mpls ip
PE1(config-if)#exit
PE1(config)#interface g2/0
PE1(config-if)#mpls ip
PE1(config-if)#exit
PE1(config)#

```

Repetiremos los anteriores pasos en las interfaces pertenecientes a la red MPLS en los demás routers PE y los que conformen nuestra red MPLS.

Seguiremos con la VPN propiamente dicha, vamos a habilitar el routing y el forwarding en la misma, para ello crearemos una tabla VRF a la que denominaremos VPN con la que PE1 y PE2 se van a comunicar, será necesario ejecutar los siguientes comandos en ambos routers PE:

```
PE1(config)#vrf definition VPN
PE1(config-vrf)#rd 120:120
PE1(config-vrf)#address-family ipv4
PE1(config-vrf-af)#route-target export 10:10
PE1(config-vrf-af)#route-target import 20:20
PE1(config-vrf-af)#exit
PE1(config-vrf)#exit
PE1(config)#

```

```
PE2(config)#vrf definition VPN
PE2(config-vrf)#rd 120:120
PE2(config-vrf)#address-family ipv4
PE2(config-vrf-af)#route-target import 10:10
PE2(config-vrf-af)#route-target export 20:20
PE2(config-vrf-af)#exit
PE2(config-vrf)#exit
PE2(config)#

```

En primer lugar se define el RD, como ya sabemos de la teoría, el RD identifica las rutas VPN y se representa como ASN:nn (ASN, Autonomous System Number), en este caso hemos establecido que el sistema autónomo sea el 120 y lo identificamos como 120, el RD debe de ser único y diferente para cada VPN de cliente. De esta forma podemos diferenciar entre distintos clientes, aunque los mismos utilicen rangos de direcciones IP superpuestos.

Seguidamente definimos los RT, los RT nos indican las rutas de la VRF que se distribuirán hacia el otro PE a través de la red MPLS. Dichas rutas se intercambiarán mediante MP-BGP. En este caso sólo importaremos y exportaremos la ruta hacia el único sistema autónomo que hemos definido, el 120.

Una vez definidas las VRF, es necesario asignarlas a una interfaz, en el caso de nuestra red, estas interfaces serán las g3/0 de ambos routers PE, la asignación se realiza de la siguiente manera:

```

PE1(config)#interface gigabitEthernet3/0
PE1(config-if)#ip vrf forwarding VPN
PE1(config-if)#ip address 192.168.1.2 255.255.255.0
PE1(config-if)#exit
PE1(config)#

```

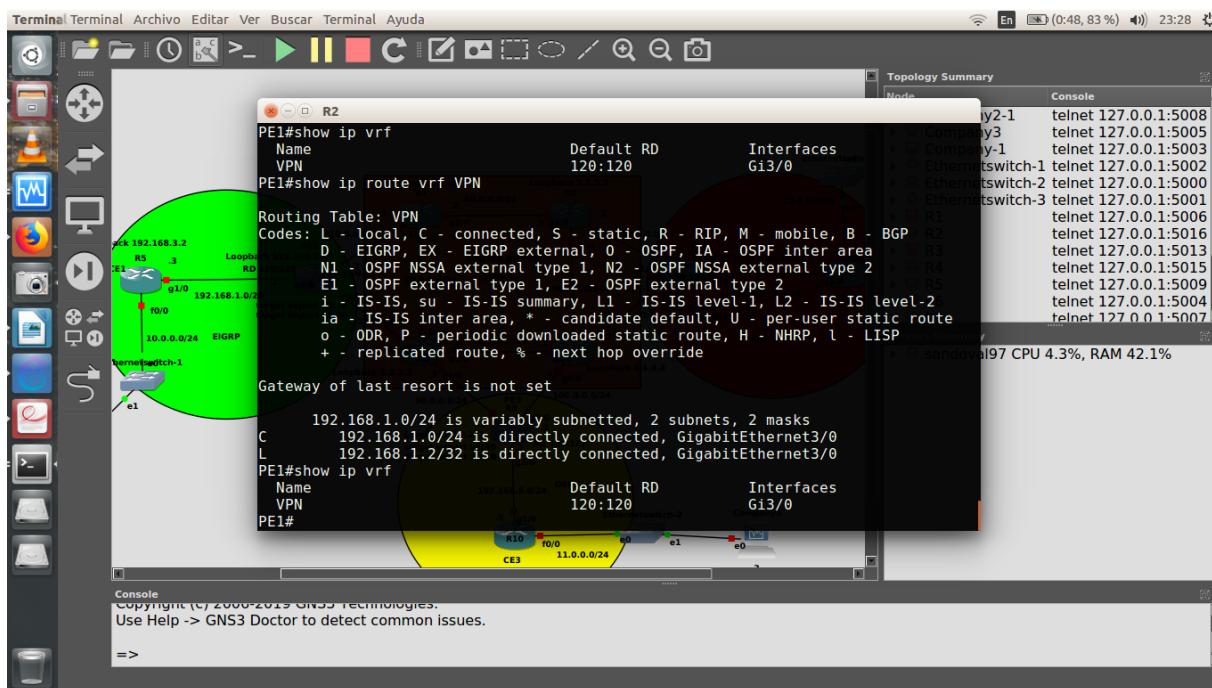
```

PE2(config)#interface gigabitEthernet3/0
PE2(config-if)#ip vrf forwarding VPN
PE2(config-if)#ip address 192.168.2.2 255.255.255.0
PE2(config-if)#exit
PE2(config)#

```

Observar que al asignar la vrf a la interfaz, la dirección ip del mismo es eliminada, por lo que es necesario volver a asignar dicha IP.

Con el comando show ip route vrf Cliente1, podemos comprobar que dicha interfaz se ha añadido a la tabla de enrutamiento de la vrf, como podemos se puede observar a continuación, desapareciendo de la tabla de routing global y podemos comprobar que la vrf se ha creado correctamente escribiendo con el comando:



Una vez definidas las VRF, estableceremos la sesión BGP entre los routers PE teniendo el mismo identificador de proceso y redistribuyendo para que los routers CE aprendan las rutas que estos a su vez aprendan por BGP esto lo realizamos con los siguientes comandos:

```

PE1(config)#router bgp 10
PE1(config-router)#neighbor 192.168.11.2 remote-as 10
PE1(config-router)#neighbor 192.168.11.2 update-source Loopback1
PE1(config-router)#address-family vpnv4
PE1(config-router-af)#neighbor 192.168.11.2 activate

```

```
PE1(config-router-af)#neighbor 192.168.11.2 send-community extended  
PE1(config-router-af)#exit  
PE1(config-router)# address-family ipv4 vrf VPN  
PE1(config-router-af)#redistribute eigrp 2  
PE1(config-router-af)#exit  
PE1(config-router-af)#exit  
PE1(config)#
```

En primer lugar, con el comando router bgp sistema-autónomo, se habilita bgp en el router, en el caso que nos ocupa hemos utilizado el AS 10 para el backbone MPLS.

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

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

Seguidamente para habilitar MP-BGP tendremos que activar la familia v4 dentro del proceso bgp y para ello accedemos a la familia de direcciones ipv4, con esto también le decimos que envíe el atributo de comunidad a al vecino BGP. Una comunidad es un grupo de prefijos que comparten una cierta propiedad en común y se puedan configurar con el atributo de la comunidad BGP.

El siguiente paso es definir la familia ipv4, en dicha familia se declaran los vecinos con los que se van a intercambiar rutas ipv4, en nuestro caso estos serán los routers CE1 y CE2, dependiendo de que router PE vayamos a configurar. Especificando la vrf de donde se van a intercambiar los routers decimos que vamos a redistribuir con eigrp 2 (siendo este protocolo el implementado para que PE1 se comunique y aprenda rutas de CE1 y viceversa).

Ahora solo nos quedaría configurar EIGRP en PE1 para que este redistribuya sus rutas por BGP al PE2 y también aprenda las rutas que le intercambia BGP para ello utilizaremos los siguientes comandos:

```
PE1(config)#router eigrp 100  
PE1(config-router)#address-family ipv4 vrf VPN  
PE1(config-router-af)#redistribute bgp 10 metric 100 1 255 1 1500  
PE1(config-router-af)#network 192.168.1.0  
PE1(config-router-af)#autonomous-system 2  
PE1(config-router-af)#exit  
PE1(config-router)#exit  
PE1(config)#
```

Activamos la familia de direcciones ipv4 indicando la vrf antes configurada dentro del proceso eigrp y con esto también especificamos la métrica para que redistribuya sus rutas a BGP y este a su vez las hará llegar hasta PE2, además especificamos las redes con las que vamos a estar comunicándonos con los demás vecinos e indicamos el **autonomous-system**.

El comando del sistema autónomo hace que el enrutador envíe paquetes EIGRP con EIGRP AS = 2 en las interfaces asociadas a vrf VPN, y que acepte paquetes para EIGRP con AS = 2 en las mismas interfaces de vrf VPN. De esta manera el router puede hablar con un enrutador CE que ejecuta EIGRP con AS=220.

El número de AS de EIGRP debe coincidir para formar una adyacencia de EIGRP válida, el comando del sistema autónomo permite usar un AS diferente a cada VRF según las necesidades.

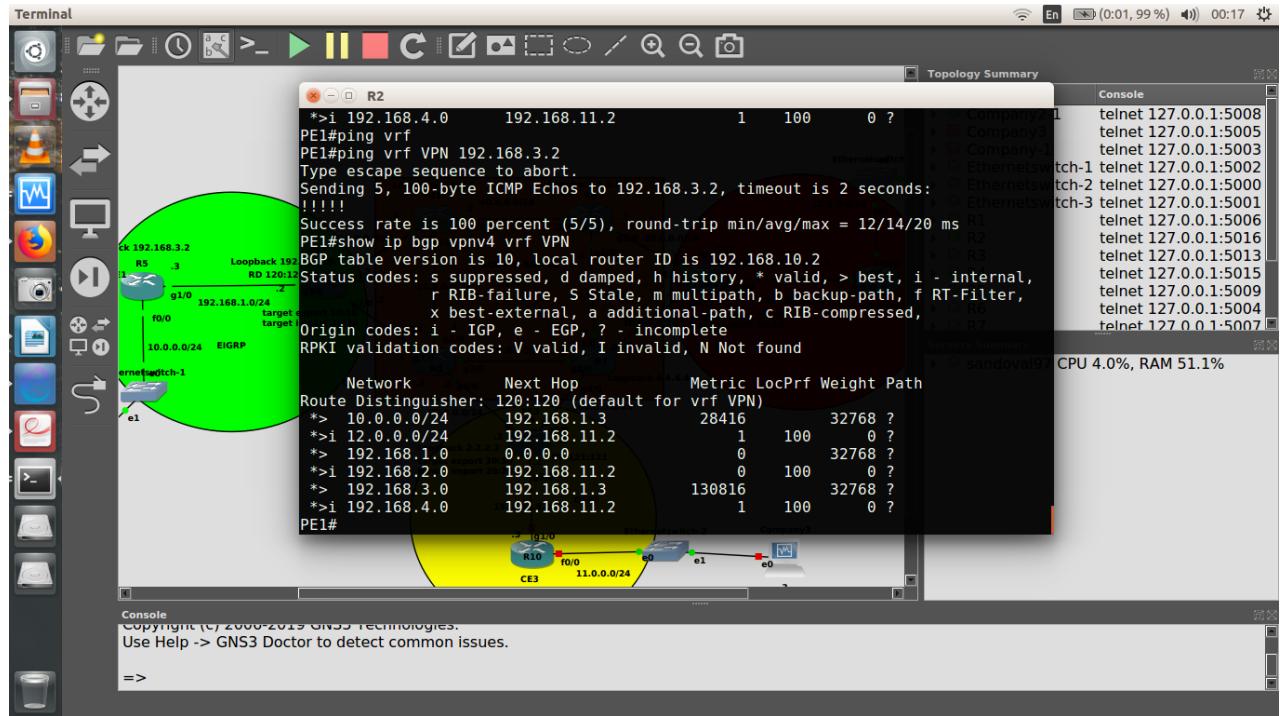
Con todo esto ya tenemos configurado nuestro PE1 ahora vamos a configurar PE2 basicamente son los mismo comando exceptuando en ciertos aspectos que vamos a recalcar acontinuacion.

```
PE2(config)#router bgp 10
PE2(config-router)#neighbor 192.168.10.2 remote-as 10
PE2(config-router)#neighbor 192.168.10.2 update-source Loopback1
PE2(config-router)#address-family vpnv4
PE2(config-router-af)#neighbor 192.168.10.2 activate
PE2(config-router-af)#neighbor 192.168.10.2 send-community extended
PE2(config-router-af)#exit
PE2(config-router)# address-family ipv4 vrf VPN
PE2(config-router-af)#redistribute rip
PE2(config-router-af)#exit
PE2(config-router-af)#exit
PE2(config)#
```

Esta vez vamos a redistribuir las rutas aprendidas por BGP a RIP ya que este es el protocolo que estamos utilizando para comunicarnos con el CE2. Esto nos lleva a configurar rip para que trabaje con la vrf (con nombre VPN) que hemos creado anteriormente con los siguientes comandos:

```
PE2(config)#router rip
PE2(config-router)#address-family ipv4 vrf VPN
PE2(config-router-af)#redistribute bgp 10 metric 2
PE2(config-router-af)#network 192.168.2.0
PE2(config-router-af)#no auto-summary
PE2(config-router-af)#version 2
PE2(config-router-af)#exit
PE2(config-router-af)#exit
PE2(config)#
```

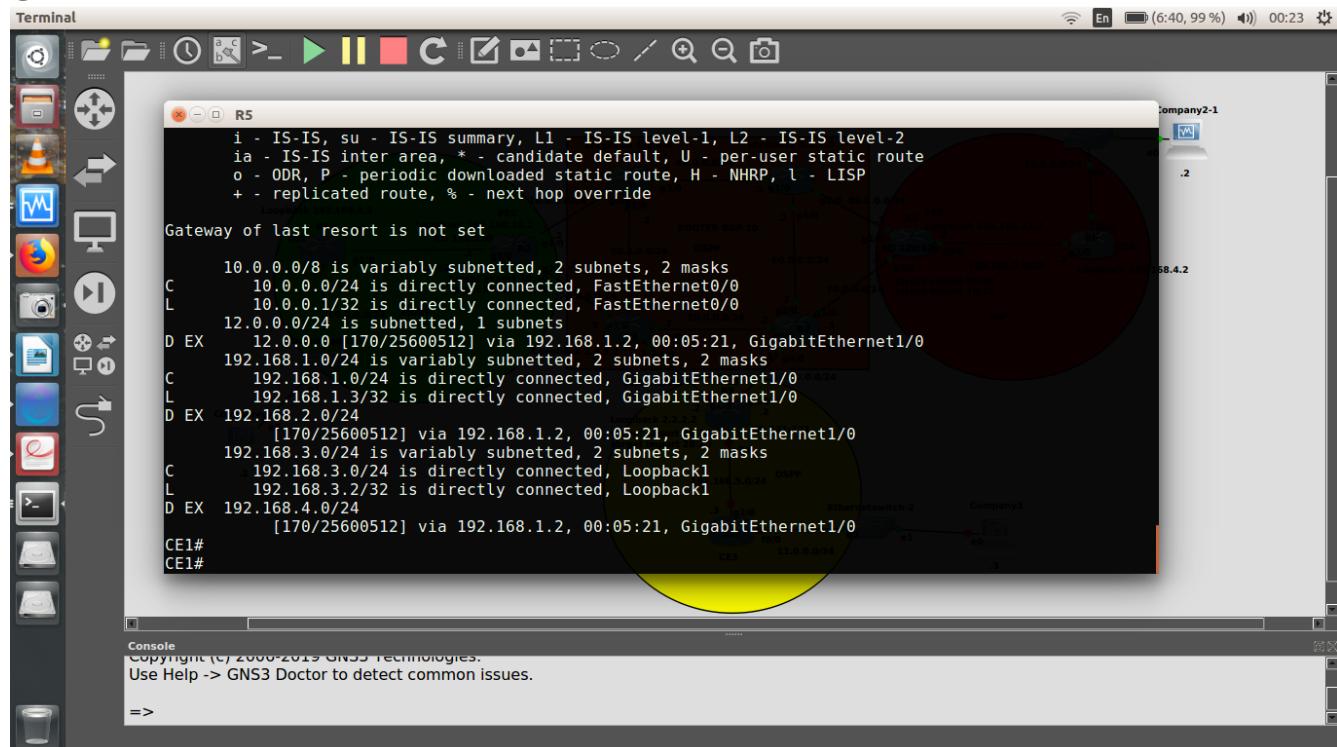
Para comprobar que la anterior configuración es correcta y las rutas han sido aprendidas por los router PE mediante bgp, utilizamos el comando **show ip bgp vpnv4 vrf VPN**, si se ha configurado correctamente, el resultado obtenido debería ser similar al siguiente:



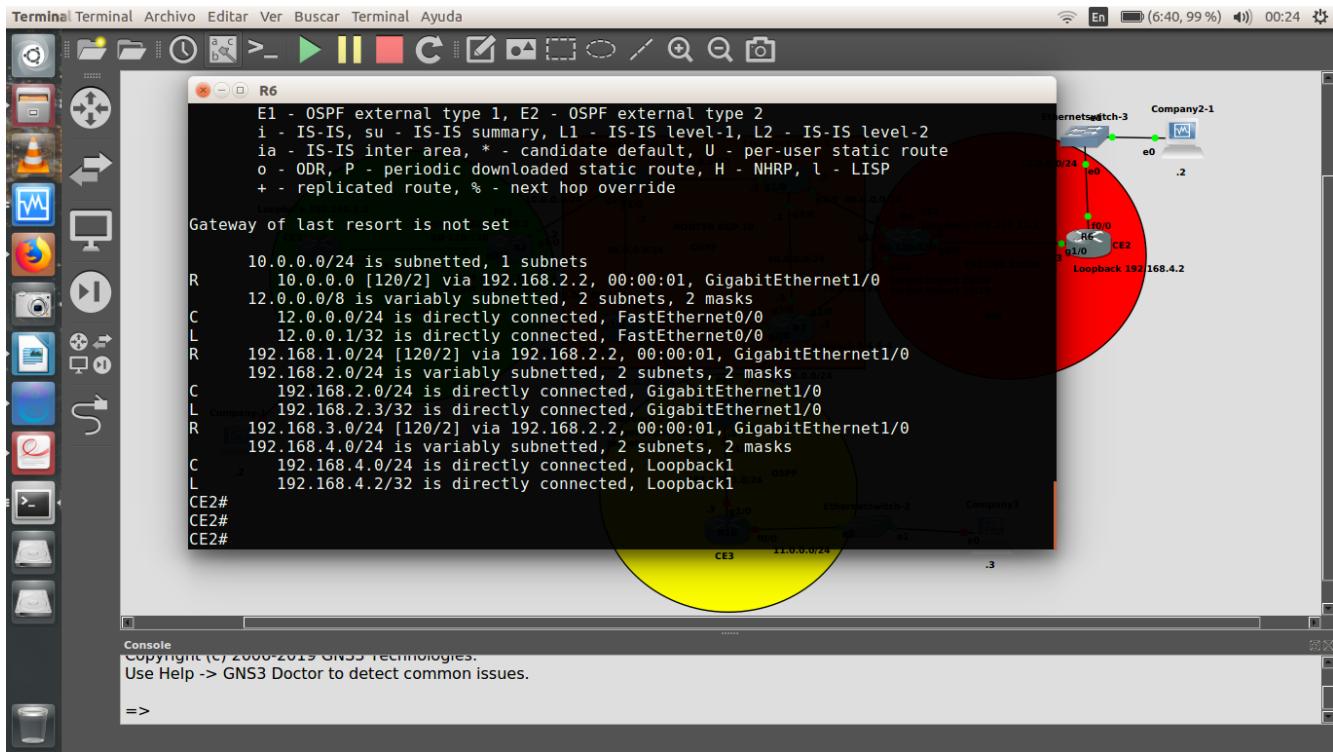
La i que podemos ver delante de la ruta hacia la sucursal, significa que esta ha sido aprendida mediante iBGP (internal BGP).

De la misma manera observamos como los CE's aprenden rutas de sus extremos opuestos, esto ocurre por la redistribucion de rutas que hacemos en los protocolos de enrutamiento.

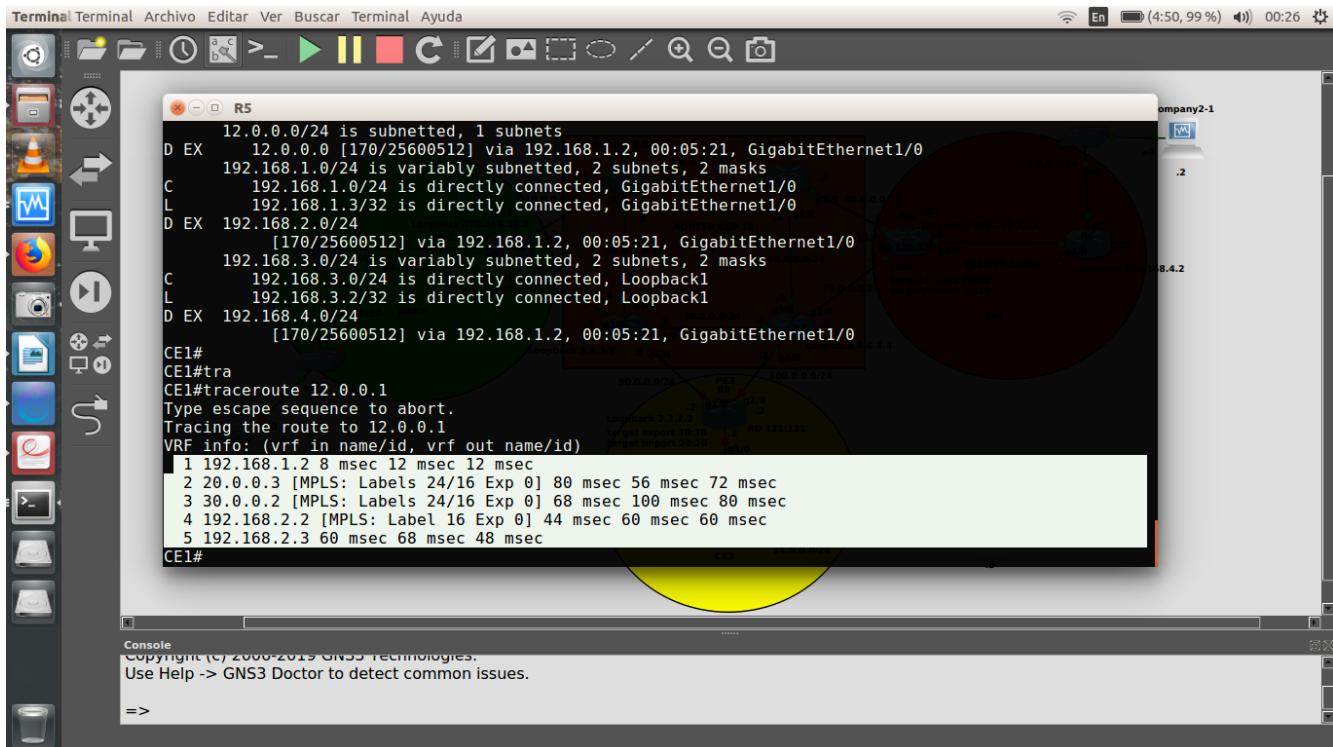
CE1



CE2



En estos instantes hemos finalizado la configuración de la VPN, con lo que los equipos de la Central deberían de poder comunicarse con los de la sucursal remota y viceversa a través de la misma comprobamos realizando un traceroute y observamos que la red MPLS interactua bien con la VPN.



En la pasada imagen observamos que los paquetes estan viajando por el backbone utilizando las etiquetas MPLS estas colocadas por los router's y como podemos observar hay conexión con la red 12.0.0.0 siendo esta ubicada en el CE2 del otro lado de la topología.

Anexo de conexión con una VPN diferente

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

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

Lo que vamos a realizar ahora es configurar el acceso entre dos VPN diferentes para que estas aprendan las rutas una de la otra para que pueda surgir la comunicación entre los CE's para ello vamos a configurar el router PE3 que este forma parte de la red que implementa MPLS y por la otra red usa ospf para comunicarse con su CE para ello vamos a realizar la configuración respectiva en el PE3 para la su tabla vrf:

```
PE3(config)#vrf definition VPN
PE3(config-vrf)#rd 121:121
PE3(config-vrf)#address-family ipv4
PE3(config-vrf-af)#route-target import 20:20
PE3(config-vrf-af)#route-target export 30:30
PE3(config-vrf-af)#exit
PE3(config-vrf)#exit
PE3(config)#exit
```

Como podemos ver el Route Distinguisher es diferente a la VPN anterior ademas que el nombre que damos a nuestra vrf es VPN2 , así mismo le decimos que importe las rutas que serán exportadas por el router PE2 exportar un RT significa que a cada ruta VPNv4 exportada se le añade una comunidad BGP extendida (esto es el RT), cuando esta ruta se redistribuye de la tabla de rutas VRF al MP-BGP.

Importar un RT significa que para cada ruta VPNv4 recibida de MP-BGP se comprueba si su comunidad extendida (RT) coincide con alguna de las asociadas a alguna VRF. Si coincide el prefijo se incluye en la tabla de rutas VRF como una ruta IP. Si no coincide el prefijo es rechazado.

Seguidamente configuraremos el protocolo BGP para que este se comunique con PE2 y puedan intercambiar sus prefijos.

```
PE3(config)#router bgp 10
PE3(config-router)#neighbor 192.168.11.2 remote-as 10
PE3(config-router)#neighbor 192.168.11.2 update-source Loopback1
PE3(config-router)#address-family vpnv4
PE3(config-router-af)#neighbor 192.168.11.2 activate
```

```
PE3(config-router-af)#neighbor 192.168.11.2 send-community extended  
PE3(config-router-af)#exit  
PE3(config-router)# address-family ipv4 vrf VPN  
PE3(config-router-af)#redistribute ospf 2  
PE3(config-router-af)#exit  
PE3(config-router-af)#exit  
PE3(config)#
```

De igual manera especificamos el AS 10 para el backbone MPLS con el que trabajan los demás routers, ademas de eso realizamos la misma configuracion que en los demás router's que sirven como PE solo que esta vez le decimos que redistribuya sus rutas a traves del protocolo ospf que utiliza el identificador 2, con esto dejamos en claro que este router ejecuta dos veces el protocolo ospf solo que es usado para diferentes propósitos ya que uno lo utilizamos para comunicarnos con su CE (2) y el otro se emplea para la comunicación con el backbone (1).

Seguidamente configuramos ospf con el que se comunica con su CE (CH3):

```
PE3(config)#router ospf 2 vrf VPN2  
PE3(config-router)#redistribute bgp 10 subnets  
PE3(config-router)#network 192.168.5.0 0.0.0.255 area 1  
PE3(config-router)#exit  
PE3(config)#
```

Esta vez la configuracion de ospf varia esto debido a que vamos a trabajar con la vrf antes creada, ademas de decirle al protocolo que redistribuya sus rutas al protocolo BGP tambien especificamos la red directamente conectada a el con la que se comunica con el CE3 y especificamos su area.

De esta forma hemos concluido con la configuracion de CE3 ahora vamos a proceder a configurar PE2 para que importe las rutas exportadas por el router PE3 editamos la vrf y especificamos el RT con el que son exportadas esto lo hacemos de la siguiente manera:

```
PE2(config)#vrf definition VPN  
PE2(config-vrf)#rd 120:120  
PE2(config-vrf)#address-family ipv4  
PE2(config-vrf-af)#route-target import 30:30  
PE2(config-vrf-af)#exit  
PE2(config-vrf)#exit  
PE2(config)#exit
```

Tambien editamos la configuracion de bgp para activar la comunidad extendida para la comunicación con el PE3 y especificamos que es uno de los vecinos con los que se va a comunicar:

```
PE2(config)#router bgp 10  
PE2(config-router)#neighbor 2.2.2.2 remote-as 10  
PE2(config-router)#neighbor 2.2.2.2 update-source Loopback1  
PE2(config-router)#address-family vpnv4  
PE2(config-router-af)#neighbor 2.2.2.2 activate  
PE2(config-router-af)#neighbor 2.2.2.2 send-community extended  
PE2(config-router-af)#exit
```

```

PE2(config-router)# address-family ipv4 vrf VPN
PE2(config-router-af)#redistribute rip
PE2(config-router-af)#exit
PE2(config-router-)#exit
PE2(config)#

```

La configuración nos quedaría de la siguiente manera:

```

R4
Current configuration : 2796 bytes
!
upgrade fpd auto
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE2
!
boot-start-marker
boot-end-marker
!
vrf definition VPN
  rd 120:120
!
address-family ipv4
  route-target export 20:20
  route-target import 10:10
  route-target import 30:30
exit-address-family
!
no aaa new-model
no ip icmp rate-limit unreachable
!
router bgp 10
  redistribute rip
  !
  address-family ipv4 vrf VPN
    redistribute rip
  !
  exit-address-family
!
no ip domain lookup
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
mpls label protocol ldp
--More--

```

La configuración nos quedaría de la siguiente manera:

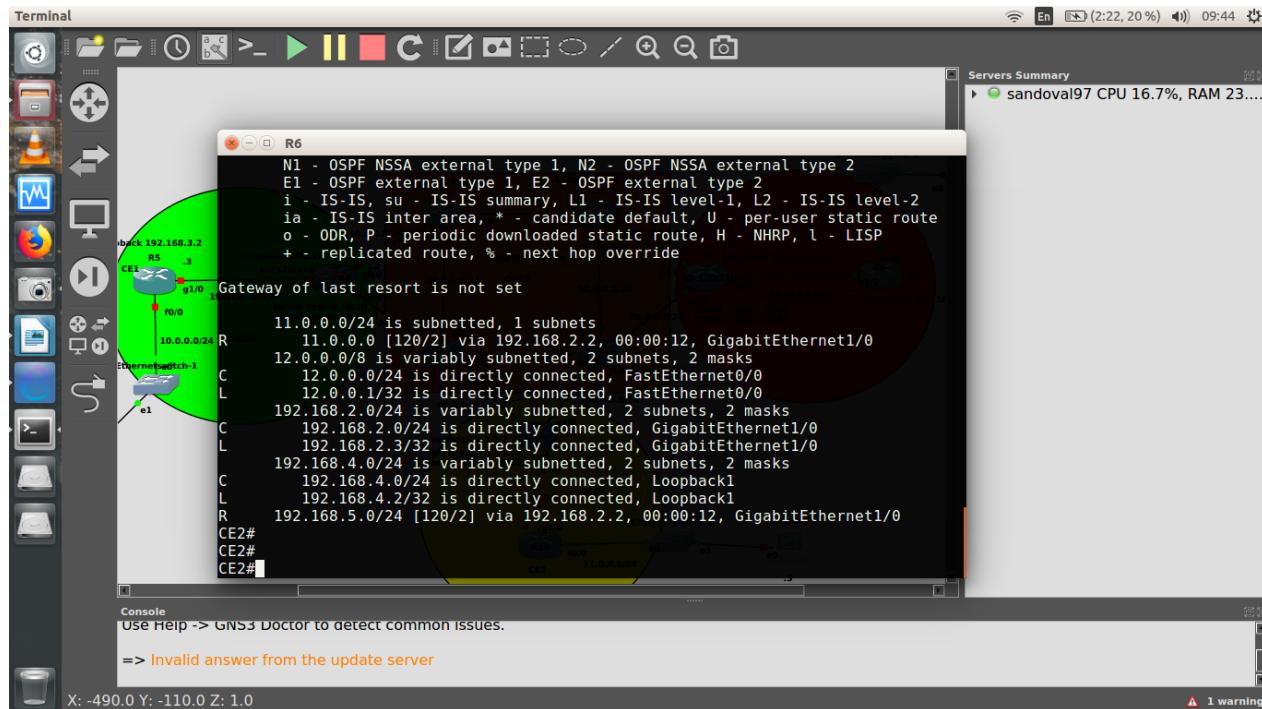
```

Terminal Archivo Editar Ver Buscar Terminal Ayuda
router rip
!
address-family ipv4 vrf VPN
  redistribute bgp 10 metric 2
  network 192.168.2.0
  no auto-summary
  version 2
exit-address-family
!
router bgp 10
  bgp log-neighbor-changes
  neighbor 2.2.2.2 remote-as 10
  neighbor 2.2.2.2 update-source Loopback1
  neighbor 192.168.10.2 remote-as 10
  neighbor 192.168.10.2 update-source Loopback1
!
address-family vpnv4
  neighbor 2.2.2.2 activate
  neighbor 2.2.2.2 send-community extended
  neighbor 192.168.10.2 activate
  neighbor 192.168.10.2 send-community extended
exit-address-family
!
address-family ipv4 vrf VPN
  redistribute rip
exit-address-family
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
ip explicit-path name tunel2 enable
next-address 4.4.4.4
next-address 5.5.5.5
next-address 6.6.6.6
--More--

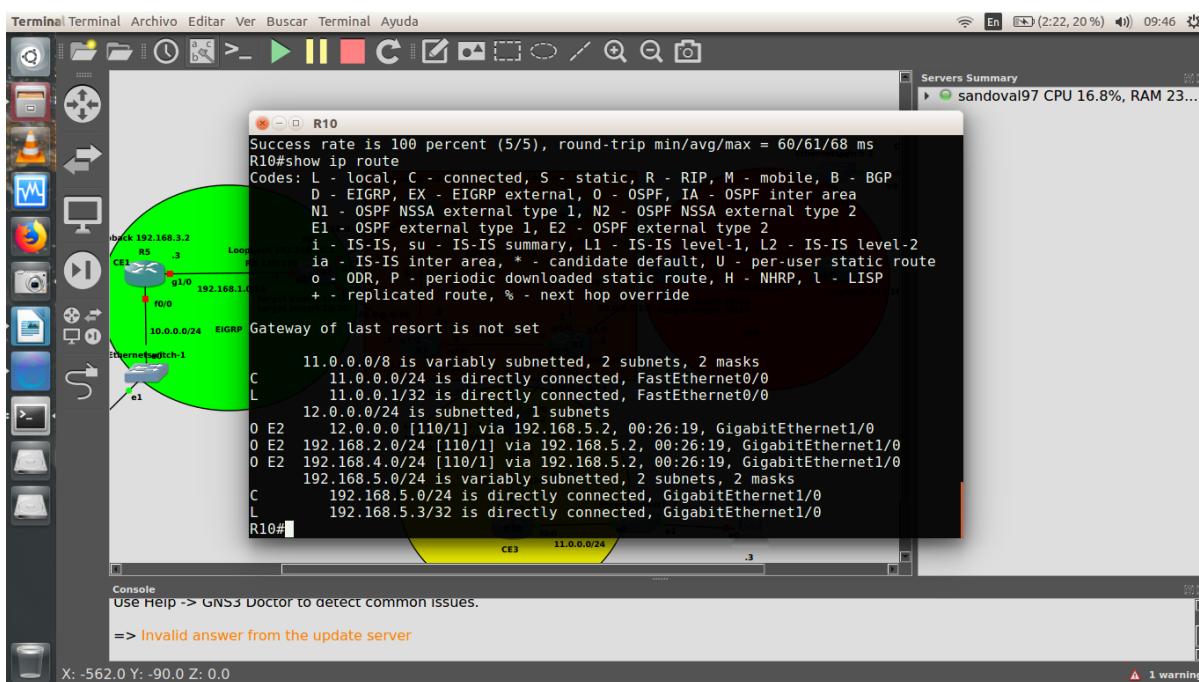
```

Como podemos observar en las siguientes imágenes tanto el CE2 como el CE3 (estando este último conectado al router con la nueva vpn que hemos configurado) han aprendido las sus rutas respectivas tanto el CE2 aprendió que para llegar a la red 11.0.0.0 (Conectada al CE3) debe enviarla a la interfaz g3/0 del router PE2 ya que es de donde se puede comunicarse con el PE3 a través de su loopback 2.2.2.2 con la que se identifica el PE3 así como CE3 aprendió que para llegar a la red 12.0.0.0 (Conectada al CE2) debe enviarla a la interfaz g3/0 del router PE3 ya que es de donde se puede comunicarse con el PE2 a través de su loopback 192.168.11.2 con la que se identifica el PE2 en la red MPLS De esta manera ya tendremos comunicación entre las dos sedes que utilizan VPN diferentes.:

CE2



CE3



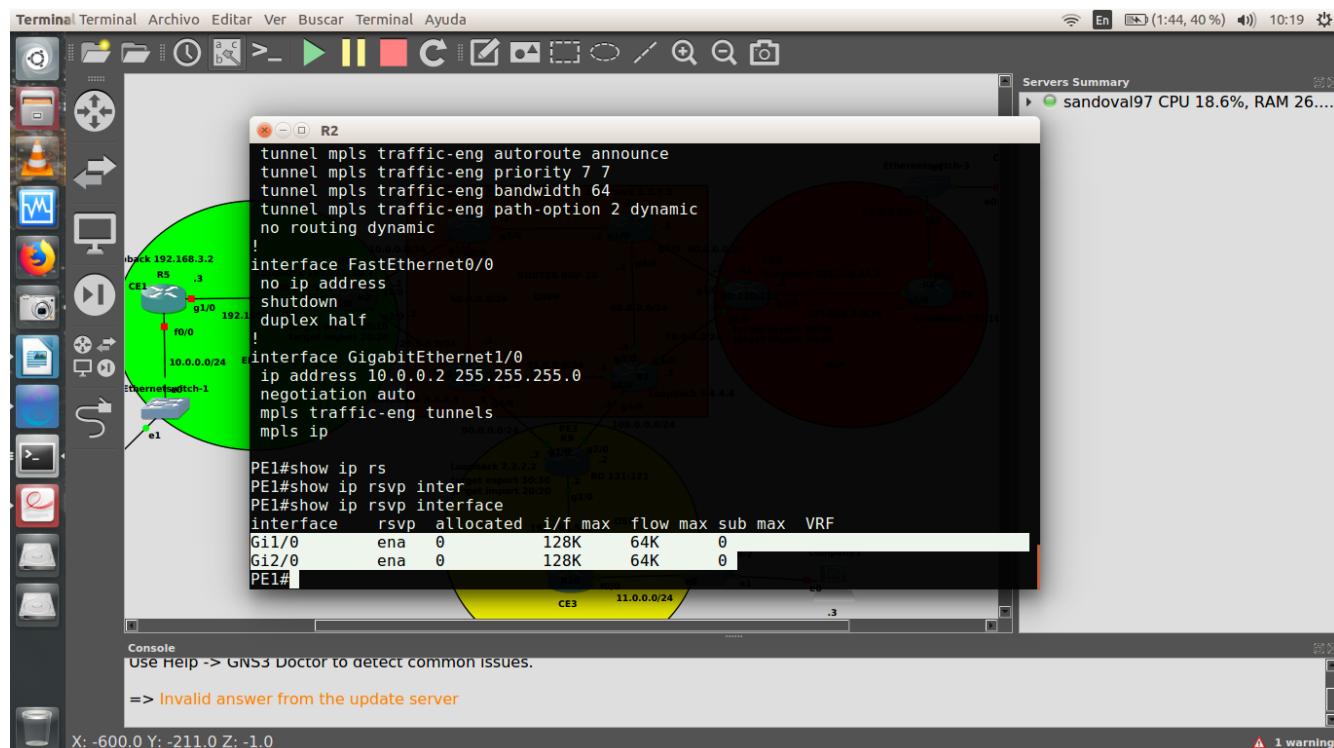
Implementación de ingeniería de tráfico en MPLS

Ahora comenzaremos propiamente con la configuración de TE, en primer lugar es necesario habilitar la ingeniería de tráfico de modo general y posteriormente en cada uno de los interfaces exceptuando las interfaces que no estén conectadas a la red MPLS, para ello utilizaremos el comando mpls traffic-eng tunnels. En el caso del PE1 deberemos emplear los siguientes comandos:

```
PE1#configure terminal
PE1(config)#mpls traffic-eng tunnels
PE1(config)#interface g1/0
PE1(config-if)#mpls traffic-eng tunnels
PE1(config-if)#ip rsvp bandwidth 128 64
PE1(config-if)#interface g2/0
PE1(config-if)#mpls traffic-eng tunnels
PE1(config-if)#ip rsvp bandwidth 128 64
PE1(config-if)#exit
PE1(config)#exit
```

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

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



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

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

```
PE1(config)#router ospf 1
PE1(config-router)#mpls traffic-eng router-id Loopback 1
PE1(config-router)#mpls traffic-eng area 0
PE1(config-router)#exit
PE1(config)#exit
```

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

Una vez habilitada la TE en toda la red, procederemos a establecer los túneles, empezaremos por el túnel dinámico de PE1 a PE2.

Un túnel se trata igual que una interfaz en CISCO, el primer paso es declarara la interfaz. Al primer túnel lo identificaremos como tunnel 1. El túnel ha de establecerse desde su inicio (Headend), en este caso desde PE1. Introducir los siguientes comandos para establecer el túnel dinámico:

```
PE1(config)#interface tunnel 1
PE1(config-if)#ip unnumbered loopback 1
PE1(config-if)#tunnel mode mpls traffic-eng
PE1(config-if)#tunnel destination 192.168.11.2
PE1(config-if)#tunnel mpls traffic-eng autoroute announce
PE1(config-if)#tunnel mpls traffic-eng path-option 2 dynamic
PE1(config-if)#tunnel mpls traffic-eng bandwith 64
PE1(config-if)#tunnel mpls traffic-eng priority 7 7
PE1(config-if)#exit
PE1(config)#
```

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

- **ip unnumbered loopback 1**, asignamos la ip de la interfaz de loopback al túnel.
- **tunnel mode mpls traffic-eng**, habilita el modo MPLS-TE en el túnel.
- **tunnel destination 192.168.11.2**, especifica el final del túnel.
- **tunnel mpls traffic-eng autoroute announce**, anuncia el túnel a través de OSPF, de esta manera todo el tráfico dirigido hacia el Tailend circulará a través del túnel.
- **tunnel mpls traffic-eng path-option 2 dynamic**, con path option indicamos el orden con el que se intenta establecer el túnel, un túnel con path-option 1 es prioritario frente a uno con 2. Si

la interfaz no dispusiera de recursos suficientes para los dos túneles, únicamente establecería el primero. Mientras que con dynamic indicamos que el protocolo CBR se encargue de calcular el LSP del túnel.

- **tunnel mpls traffic-eng bandwidth 64**, establece el ancho de banda reservado del túnel, en este caso 64 kbps.
- **tunnel mpls traffic-eng priority 7 7**, indica la prioridad del túnel, un valor menor indica mayor prioridad.

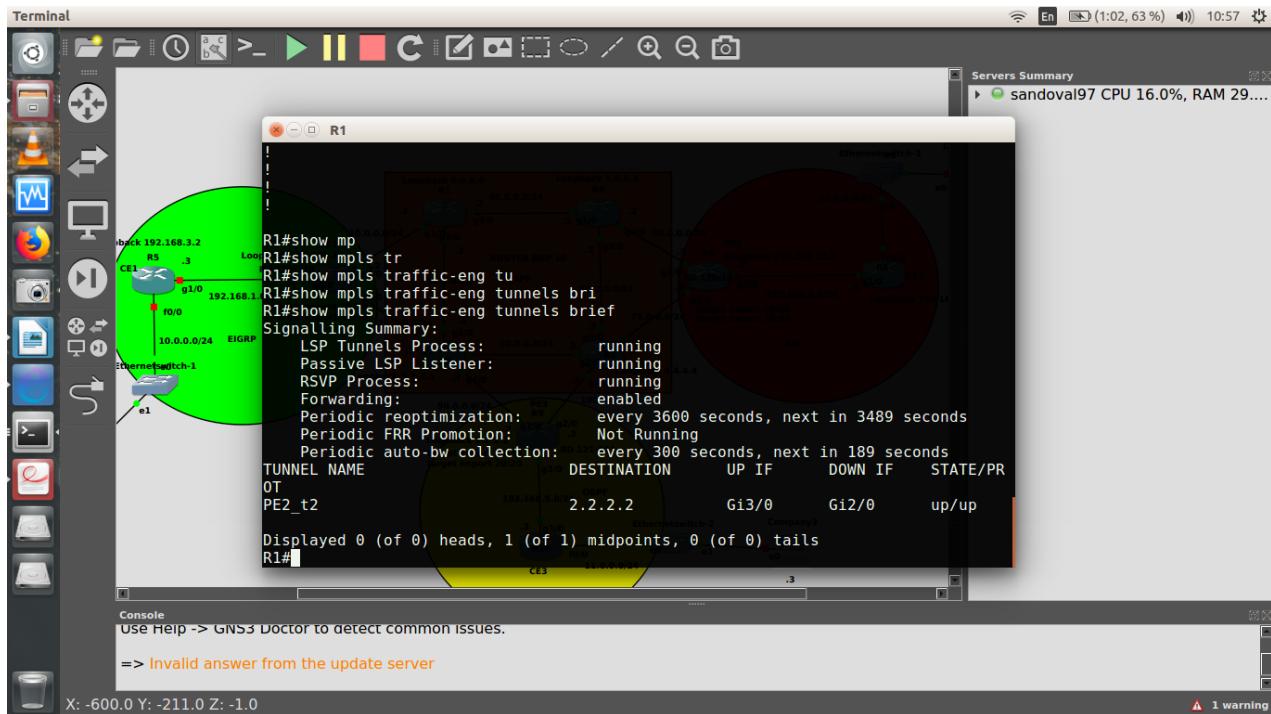
En este punto ya tenemos un túnel dinámico establecido entre PE1 y PE2. Seguidamente configuraremos el túnel explícito entre PE2 y PE3. Para ello utilizaremos los siguientes comandos:

```
PE2(config)#interface tunnel 2
PE2(config-if)#ip unnumbered loopback 1
PE2(config-if)#tunnel mode mpls traffic-eng
PE2(config-if)#tunnel destination 2.2.2.2
PE2(config-if)#tunnel mpls traffic-eng autoroute announce
PE2(config-if)#tunnel mpls traffic-eng path-option 2 explicit name tunel2
PE2(config-if)#tunnel mpls traffic-eng bandwidth 64
PE2(config-if)#tunnel mpls traffic-eng priority 6 6
PE2(config-if)#exit
PE2(config)# ip explicit-path name tunel2
PE2(cfg-ip-expl-path)#next-address 4.4.4.4
PE2(cfg-ip-expl-path)#next-address 5.5.5.5
PE2(cfg-ip-expl-path)#next-address 6.6.6.6
PE2(cfg-ip-expl-path)#next-address 3.3.3.3
PE2(cfg-ip-expl-path)#next-address 2.2.2.2
PE2(cfg-ip-expl-path)#exit
PE2(config)#

```

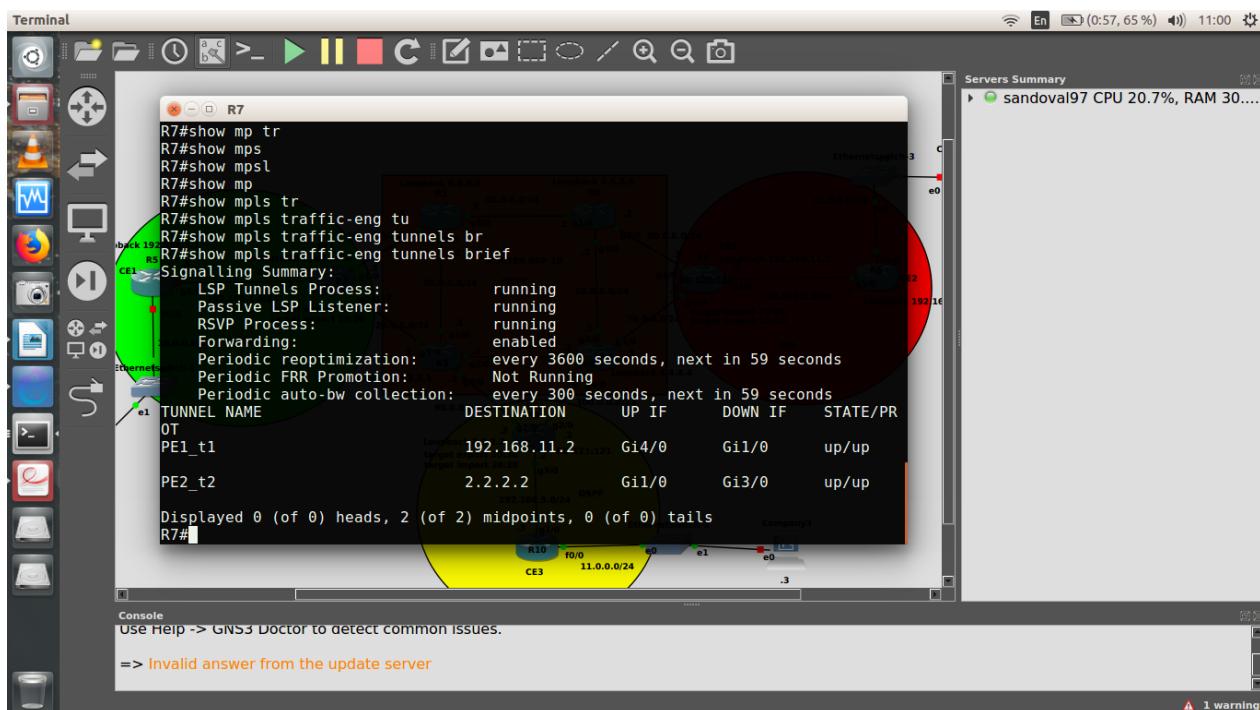
Como podemos ver la única diferencia frente a un túnel dinámico, es que se especifican los saltos por los que discurre el túnel de manera manual.

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



En este caso en R1 sólo visualizamos el túnel 2, pero es posible en determinadas ocasiones, que ambos túneles discutan a través de él.

Como podemos observar de la ejecución del anterior comando, en el caso del R1 observamos que está establecido uno de los dos túneles que habíamos definido, el otro túnel debe de haberse establecido a través de R7, R8 y R3 que fue la ruta que le indicamos en la configuración. Comprobamos con el mismo comando en el R7 que el túnel se haya establecido correctamente. Existe también la posibilidad de que ambos túneles se establezcan atravesando el R7.



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

```
R2
PE1#show mpls traffic-eng tunnels tunnel
% Incomplete command.

PE1#show mpls traffic-eng tunnels tunnell
Name: PE1_t1                               (Tunnell) Destination: 192.168.11.2
Status:
  Admin: up      Oper: up     Path: valid      Signalling: connected
  path option 2, type dynamic (Basis for Setup, path weight 4)

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

InLabel : -
OutLabel : GigabitEthernet2/0, 27
RSVP Signalling Info:
  Src 192.168.10.2, Dst 192.168.11.2, Tun_Id 1, Tun_Instance 95
  RSVP Path Info:
    My Address: 20.0.0.2
    Explicit Route: 20.0.0.3 90.0.0.3 90.0.0.2 100.0.0.2
      100.0.0.3 70.0.0.3 70.0.0.2 192.168.11.2
    Record Route: NONE
    Tspec: ave rate=64 kbytes, burst=1000 bytes, peak rate=64 kbytes
  RSVP Resv Info:
    Record Route: NONE
    Fspec: ave rate=64 kbytes, burst=1000 bytes, peak rate=64 kbytes
Shortest Unconstrained Path Info:
  Path Weight: 3 (TE) > GNS3 Doctor to detect common issues.
  Explicit Route: 20.0.0.2 20.0.0.3 30.0.0.3 30.0.0.2
    70.0.0.3 70.0.0.2 192.168.11.2

History:
--More--
```

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

Por otra parte, si volvemos a ejecutar el comando **show mpls forwarding-table destination-prefix detail** en alguno de los dos routers, observaremos que el túnel para alcanzar su destino utiliza diferente etiquetado, al ser este asignado por el protocolo RSVP.

```
Terminal
R1
Signalling Summary:
  LSP Tunnels Process:          running
  Passive LSP Listener:        running
  RSVP Process:                running
  Forwarding:                 enabled
  Periodic reoptimization:    every 3600 seconds, next in 3489 seconds
  Periodic FRR Promotion:     Not Running
  Periodic auto-bw collection: every 300 seconds, next in 189 seconds
TUNNEL NAME           DESTINATION   UP IF    DOWN IF   STATE/PR
PE2_t2               2.2.2.2       Gi3/0    Gi2/0     up/up

Displayed 0 (of 0) heads, 1 (of 1) midpoints, 0 (of 0) tails
R1#show mp
R1#show mpls for
R1#show mpls forwarding-table 2.2.2.2 data
R1#show mpls forwarding-table 2.2.2.2 detail
Local   Outgoing Prefix   Bytes Label  Outgoing   Next Hop
Label   Label or Tunnel Id Switched interface
24      20      2.2.2.2/32 0          Gi3/0     50.0.0.3
MAC/Encaps=14/18, MRU=1500, Label Stack{20}
CA03123C001CCA01121D00388847 00014000
No output feature configured
R1#>
R1#> Invalid answer from the update server
R1#>
```