

Facultat Informàtica de Barcelona FIB

Grau en Enginyeria Informàtica

Laboratori de Xarxes de Computadors 2

Davide Careglio, Sergio Ricciardi, Jose M. Barceló



Índex

| | |
|--|-----------|
| Entorno del laboratorio..... | 5 |
| Configuració bàsica | 5 |
| Interfícies dels PCs..... | 5 |
| Recordatorio de configuración de IP en Linux y CISCO IOS..... | 6 |
| Lab. 1 - Configuración de IPv6 y encaminamiento RIPng..... | 7 |
| 1.1. Introducción a IPv6 | 7 |
| 1.2. Configuración en Linux..... | 8 |
| 1.3. Configuración en routers CISCO | 8 |
| 1.4. Configuració de RIPng en routers CISCO..... | 9 |
| 1.5. Ejemplo | 10 |
| 1.6. Realizació de la práctica..... | 11 |
| Lab. 2 - Intra-Domain Routing: OSPF | 13 |
| 2.1. Introducción a OSPF (RFC 2328) | 13 |
| 2.2. Configuración de OSPF en un router CISCO..... | 13 |
| 2.3. Modificación del comportamiento de OSPF..... | 15 |
| 2.4. Verificación..... | 15 |
| 2.5. Realizació de la práctica..... | 17 |
| Lab. 3 - MultiProtocol Label Switching (MPLS) | 19 |
| 3.1 Objetivos de la practica | 19 |
| 3.2 Introducción a MPLS..... | 19 |
| 3.3 MPLS Traffic Engineering (MPLS-TE)..... | 20 |
| 3.2 Realización de la práctica..... | 22 |
| Lab. 4 - Inter-Domain Routing: BGPv4..... | 25 |
| 4.1. Introducción a BGPv4 (RFC 4271) | 25 |
| 4.2. Configuración de BGP en un router CISCO..... | 25 |
| 4.3. Verificación..... | 27 |
| 4.4. Realización de la práctica..... | 27 |
| Lab. 5 - BGPv4 manipulación de atributos | 29 |
| 5.1. Objectivo de la práctica | 29 |
| 5.2. Filtrado de rutas y manipulación de atributos..... | 29 |
| 5.3. Ejemplo | 30 |
| 5.4. Eliminación de route-map | 31 |
| 5.5. Realización de la práctica..... | 31 |
| Lab. 6 - BGPv4 manipulación de atributos con comunidades..... | 33 |
| 6.1. Objectivo de la práctica | 33 |
| 6.2. Uso de comunidades (community) | 33 |
| 6.3. Realización de la práctica..... | 34 |

| | |
|--|-----------|
| Lab. 7 - BGPv4: Route Reflector y Confederation | 35 |
| 7.1. Objetivo de la práctica | 35 |
| 7.2. Introducción a iBGP..... | 35 |
| 7.3. Confederation..... | 35 |
| 7.4. Route Reflection..... | 36 |
| 7.5. Realización de la práctica..... | 38 |
| Anexo A – Comandos para OSPF y BGP..... | 39 |
| A.1. OSPF commands | 39 |
| A.2. BGP commands..... | 41 |
| Anexo B – Simulador GNS3 | 45 |

Entorno del laboratorio

En aquest capítol introductorí hi ha una descripció general de la configuració de l'entorn que es farà servir per fer les pràctiques de laboratori. Al botar el PC hi ha que seleccionar la imatge “xarxes”. Aquesta imatge s’ha confeccionat a partir de la distribució de Linux de mida reduïda anomenada slitaz (<http://www.slitaz.org>).

Configuració bàsica

Para acceder:

Usuari i password: xc / xc

Superusuari: root / root

El funcionament habitual és obrir la sessió com a usuari “xc” i en la consola canviar a root si ho necessiten. Els icones de les aplicacions que es faran servir habitualment estan a la part de sota de l'escriptori:

- menú desplegable
- consola x11
- navegador web (midori)
- editor
- wireshark
- calculadora
- navegador del sistema de fitxers.

Per configurar el PC per DHCP cal executar la següent comanda com a superusuari. Això és necessari per poder accedir al servidor pclabxc per fer els minicontrols.

```
xarxes# udhcpc -i e0
```

Interfícies dels PCs

Per a fer les pràctiques de xarxes utilitzareu els següents ports de comunicacions dels PCs (vegeu la Figura 1):

- **ttyS0** (COM1 en windows): Aquí hi connectareu la consola per poder configurar els routers i commutadors CISCO.
- **e0, e1, e2**: son tres targes Ethernet en les posicions que indica la Figura 1.

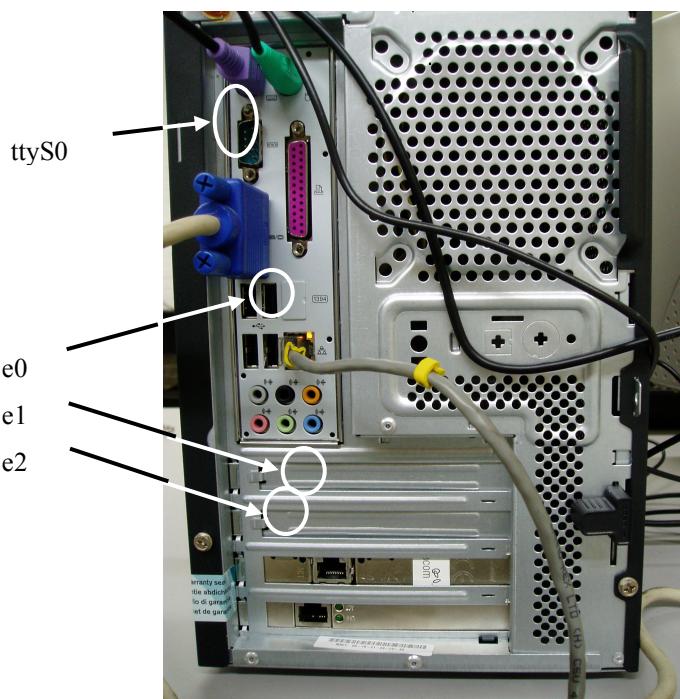


Figura 1: Ports de comunicació dels PCs del laboratori que farem servir en les pràctiques.

Recordatorio de configuración de IP en Linux y CISCO IOS

Como práctica de repaso se va a configurar una red con encaminamiento dinámico RIP. Todos los dispositivos de la red son PC con linux. Para los que están dibujados como routers se usarán routers CISCO.

Configuración de interfaces linux

Ejemplo de asignación de una dirección IP.

```
ifconfig Interface @IP netmask mask broadcast broadcast
```

Ejemplos de configuración de una ruta estática.

```
route [add|del] -host IP-host gw IP-gateway  
route [add|del] -net IP-network netmask mask gw IP-gateway  
route [add|del] default gw IP-gateway
```

Configuración de routers

Para la configuración de las IP en las interfaces del router, se necesita un PC conectado al router a través del cable consola. Luego en un terminal Linux se usa la aplicación **minicom** para interaccionar con el router. Ejemplo de asignación de una dirección IP a un interfaz Ethernet de un router.

```
router# configure terminal  
router(config)# interface ethernet0  
router(config-if)# ip address 10.0.0.10 255.255.255.0  
router(config-if)# no shutdown  
router(config-if)# exit  
router(config) #
```

En el caso de interfaz serie, hay que configurar también la velocidad de transmisión ya que se permiten varias diferentes. Esta velocidad pero solo hay que configurarla en uno de los dos routers conectados entre sí (el otro se configura de acuerdo al primero). En concreto hay que comprobar cuál de los conectores lleva la etiqueta DCE (el otro será DTE) y solo en este añadir el siguiente comando a la hora de configurar una dirección IP (el resto sigue igual al ejemplo anterior).

```
router(config-if)# clockrate 56000
```

Lab. 1 – Configuración de IPv6 y encaminamiento RIPng

1.1. Introducción a IPv6

Una dirección IPv6 está formada por 128 bits. Las direcciones se clasifican en diferentes tipos: unicast, multicast y anycast. En esta práctica solo se consideran direcciones unicast.

1.1.1. Formato

Las direcciones unicast generalmente se dividen en dos grupos lógicos: los primeros 64bits identifican el prefijo de red (routing-prefix o subnet prefix), y son usados para encaminamiento; los últimos 64bits identifican el interface de red del host (interfaceID).

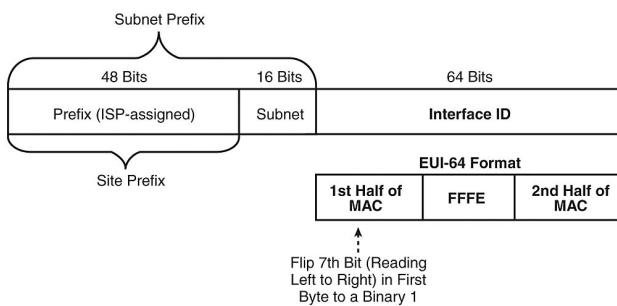


Figura 2: Formato dirección IPv6

Una dirección IPv6 se representa mediante ocho grupos de cuatro dígitos hexadecimales, cada grupo representando 16 bits (dos octetos). Los grupos se separan mediante puntos (:). Un ejemplo de dirección IPv6 podría ser:

2001:0000:85a3:0000:0000:8a2e:0370:7334

Esta representación completa puede ser simplificada de varias maneras, eliminando partes de la representación.

- Los ceros iniciales de cada grupo pueden omitirse, aunque cada grupo debe contener al menos un dígito hexadecimal.
- Uno o más grupos de ceros pueden ser sustituidos por dos puntos. Esta sustitución puede realizarse únicamente una vez en la dirección. En caso contrario, obtendríamos una representación ambigua. Si pueden hacerse varias sustituciones, debemos hacer la de mayor número de grupos; si el número de grupos es igual, debemos hacer la situada más a la izquierda. Con esta regla, reduciríamos la dirección del ejemplo a:

2001:0:85a3::8a2e:370:7334

1.1.2. Ambitos

Toda dirección IPv6, excepto la dirección indefinida (::), tiene un “ámbito” (scope en inglés), que determina en qué partes de la red es válida. En direccionamiento unicast:

- Las direcciones **link-local** y la dirección de loopback tienen ámbito de enlace local, es decir, deben ser usadas en la red directamente conectada. Una dirección IPv6 Link-Local comienza con el prefijo FE80::/10 (los primeros 10 bits), luego los bits del 11 hasta 64 (los siguientes 54 bits) se configuran con valores de ceros. Los restantes 64 bits son de interfaceID. Estas direcciones usan por lo tanto el formato siguiente: FE80 :: InterfaceID.
- Las direcciones **Unique Local Address** (ULA) se utilizan para comunicaciones locales. Son enruteables sólo dentro de un ámbito cooperativo (similar a los rangos de direcciones privadas 10/8, 172.16/12, y 192.168/16 en IPv4). Las direcciones incluyen una secuencia pseudoaleatoria en el routing-prefix para minimizar el riesgo de conflictos en la interconexión de plataformas diferentes o si los paquetes se desvían a Internet (son únicas en todo el mundo).
- El resto de direcciones, excepto aquellas privadas o reservadas, tienen ámbito **global**, que significa que son mundialmente enruteables y pueden ser usadas para conectarse a direcciones de ámbito global en cualquier lugar.

1.1.3. Metodos de configuración de IPv6

Hay dos maneras para configurar una IPv6:

- **Statefull** es cuando se asigna manualmente o bien cuando se obtiene de un servidor DHCPv6 (junto a otros parámetros como Gateway, nombre, dominio, etc.).
- **Stateless** es cuando se configura automáticamente (autoconfiguración). El routing-prefix de la dirección global se obtiene de un router conectado a la misma red a través de la nueva funcionalidad Network Discovery y se completa con un interfaceID formado de la conversión de la MAC a 64 bits. Juntamente a un routing-prefix global, se envía también una ruta para salir de la red. De la misma manera, el routing-prefix de la dirección link-local será fe80::/64 y el interfaceID de la conversión a 64 bits de la MAC de la interfaz. Opcionalmente, el router puede enviar también un routing-prefix ULA si este se está usando en este ámbito.

1.2. Configuración en Linux

1.2.1. Configuración de IPv6 en Linux

Las versiones más modernas de Linux usan por defecto el método stateless. Por lo tanto una IPv6 link-local ya está configurada y la global se configuraría a través del Network Discovery (al recibir un ICMP Router Advertisement del router de su red). La diferencia es que en este caso hay que activar la interfaz (**ifconfig interface up**) para que reciba estos mensajes.

Si se quisiera asignar una IPv6 global manualmente el comando es

```
ifconfig interface add @IPv6/prefixlength
```

Por ejemplo, para asignar la dirección 2001:0db8:0:f101::1/64 a la interfaz e0, el comando es

```
ifconfig e0 add 2001:0db8:0:f101::1/64
```

En algunas versiones de Linux, asignar una IPv6 a una interfaz no es suficiente para que esta se active. Así que puede que se necesite activarla con el comando **ifconfig interface up**.

Al contrario que en IPv4, en IPv6 una misma interfaz puede tener varias direcciones. Por lo tanto, si se quiere modificar una dirección previamente asignada, no basta con enviar un nuevo ifconfig con la nueva dirección ya que la interfaz se quedaría con las dos IPv6 (es decir no se sobreescrive). En IPv6 hay que eliminar la dirección antigua antes de poner la nueva con el siguiente comando

```
ifconfig interface del @IPv6/prefixlength
```

1.2.2. Configuración de rutas estaticas en Linux

Recordar que si se usa la asignación stateless, además de una IPv6, se obtiene también una ruta estatica para salir de la red.

En el caso pero que se quisiera añadir una ruta estatica a la tabla de encaminamiento, el comando es

```
route -A inet6 add IPv6network/prefixlength gw IPv6address [dev interface]
```

Substituyendo **del** con **add** se elimina la ruta de la tabla.

Por ejemplo, con el siguiente comando se añade una ruta para alcanzar las direcciones globales 2000::/3 a través del gateway 2001:0db8:0:f101::1

```
route -A inet6 add 2000::/3 gw 2001:0db8:0:f101::1
```

En el caso de querer configurar una ruta por defecto por el gateway 2001:0db8:0:f101::1

```
route -A inet6 add default gw 2001:0db8:0:f101::1
```

1.2.3. Verificación

Para visualizar la dirección asignada a una interfaz se puede seguir usando **ifconfig** o bien el comando **ip**.

```
ifconfig interface      [|grep "inet6 addr:"]
ip -6 addr show dev interface
```

Con el comando **ip -6** se consigue visualizar exclusivamente información relativa a IPv6, mientras que **ifconfig** enseña toda la información general de la interfaz. Si se quiere ver solo la parte relativa a IPv6 se puede usar **grep** como indicado en los comandos anteriores.

Para visualizar rutas IPv6 se pueden usar los comandos **ip** o **route** de la siguiente manera

```
ip -6 route show [dev interface]
route -A inet6
```

Para verificar la conectividad entre interfaces IPv6 se puede usar **ping6** y **traceroute6**. Por ejemplo

```
ping6 2003:ff::1
ping6 -I eth0 fe80::2e0:18ff:fe90:9205
traceroute6 3ffe:b00:c18::1::10
```

Cabe mencionar que en el caso de ping hacia una dirección link-local (segunda línea en el ejemplo anterior) hay que especificar por que interfaz hay que transmitir usando la opción **-i interface**.

1.3. Configuración en routers CISCO

1.3.1. Configuración

Tambien en los routers se pueden configurar direcciones IPv6 en modo statefull y stateless.

Si se quiere configurar manualmente, en el caso de link-local (ambos campos routing-prefix e interfaceID), el comando es

```
router(config-if)# ipv6 address ipv6address/prefixlength link-local
```

En el caso de IPv6 global, hay dos posibilidades. Bien se pueden definir ambos campos routing-prefix e interfaceID

```
router(config-if)# ipv6 address ipv6address/prefixlength
```

O asignar solo el routing-prefix y que interfaceID venga de la conversión a 64 bits de la MAC de la interfaz.

```
router(config-if)# ipv6 address ipv6network/prefixlength eui-64
```

Si en cambio se quiere configurar una IPv6 link-local automáticamente (el routing-prefix será fe80::/64 e interfaceID la conversión de la MAC), el comando es

```
router(config-if)# ipv6 enable
```

Finalmente si se quiere configurar automáticamente una IPv6 global hay que usar la función Network Discovery. El Network Discovery usa mensajes ICMP

- El ICMP Router Advertisement que usa el router para anunciar por una interfaz el routing-prefix de su red. Para activar el envío de estos mensajes, hay que activar la función de routing IPv6 del router.

```
router(config)# ipv6 unicast-routing
```

- El ICMP Router Solicitation que usa el router u host que no tiene configurada una IPv6 global y quiere autoconfigurarse con el routing-prefix global recibido por el ICMP Router Advertisement. La IPv6 se completa con un interfaceID que viene de la conversión de la MAC. Por defecto, este comando configura también una IPv6 link-local usando fe80::64 y completando con la MAC.

```
router(config)# interface ethernet0
router(config-if)# ipv6 address autoconfig
router(config-if)# no shutdown
```

Si se quisiera desactivar el envío de los ICMP Router Advertisement de una interfaz concreta, se usa el comando siguiente

```
router(config-if)# ipv6 nd ra suppress
```

Recordar que el mismo comando con el **no** delante vuelve a activarlo.

1.3.2. Verificación

| | |
|------------------------------|--|
| R# show ipv6 interface | Permite ver el estado de las interfaces IPv6 |
| R# show ipv6 interface brief | Permite ver el estado resumido de las interfaces IPv6 |
| R# ping ipv6address | Hace un ping a la IPv6 ipv6address. Si esta es una link-local, hay que luego especificar la interfaz de salida |
| R# debug ipv6 nd | Permite ver el intercambio de mensajes Network Discovery |

1.4. Configuració de RIPng en routers CISCO

1.4.1. Introducción

RIPng [RFC2080] es la adaptación a IPv6 del protocolo de encaminamiento RIP. Las características de RIPng son:

- Es un protocolo de encaminamiento dinámico basado en el enfoque vector distancia
- Usa una única métrica para determinar la mejor ruta que es menor número de saltos (redes que se tienen que cruzar) para llegar a un prefijo destino.
- A través del intercambio periódico (30 segundos) de mensajes RIP que contienen los prefijos que se quieren distribuir en el sistema, los routers construyen sus tablas de encaminamiento
- Usa split-horizon para evitar bucles
- Usa poison-reverse y triggered-update para mejorar el tiempo de convergencia cuando hay un cambio en el sistema

1.4.2. Configuración básica

Para activar el RIPng en un router, hay que primero activar encaminamiento dinámico basado en IPv6 (este comando sirve para cualquier protocolo de encaminamiento) y luego crear el proceso RIP asignándole un nombre

```
router(config)# ipv6 unicast-routing
router(config)# ipv6 router rip ProcessName
```

Luego hay que activar las interfaces que se quieren involucrar en el encaminamiento dinámico. Involucrar en el sentido que i) la interfaz pertenece a una red IPv6 que se quiere anunciar o ii) la interfaz envía y recibe mensajes RIPng.

```
router(config)# interface Ethernet0
router(config-if)# ipv6 rip ProcessName enable
```

1.4.3. Verificación

| | |
|------------------------------------|---|
| R# show ipv6 route | Permite ver la tabla de encaminamiento |
| R# show ipv6 route rip | Permite ver solo las entradas RIPng de la tabla de encaminamiento |
| R# show ipv6 rip | Permite ver información de encaminamiento solo de RIPng |
| R# show ipv6 rip Process database | Permite ver información de la base de datos del proceso de RIPng |
| R# show ipv6 rip Process next-hops | Permite ver información de los gateways |
| R# debug ipv6 rip | Permite ver el intercambio de mensajes RIPng |

1.5. Ejemplo

Suponiendo el ejemplo de la Figura 3, se quieren configurar las direcciones IPv6 indicadas y activar encaminamiento RIPng entre los routers. En PC1 se quiere configurar manualmente una IPv6 global mientras en PC2 se configura stateless. Entre los dos routers solo se usan IPv6 link-local.

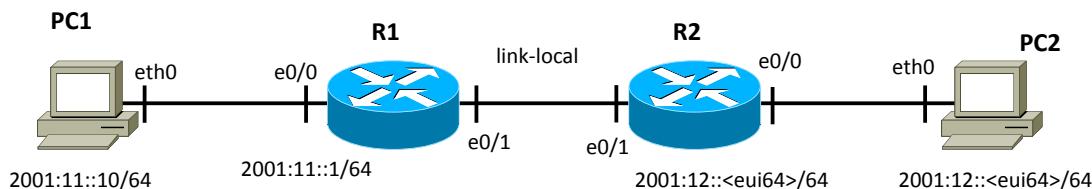


Figura 3: Ejemplo de red IPv6

Los comandos para PC1 son los siguientes

```
pc1# ifconfig eth0 add 2001:11::10/64
pc1# ifconfig eth0 up
pc1# route -A add default gw 2001:11::1
```

En el caso de PC2 solo hay que activar la interfaz y esperar el ICMP Router Advertisement del router.

```
pc2# ifconfig eth0 up
```

Para la configuración de IPv6 el router R1 los comandos son los siguientes

```
R1(config)# interface e0/0
R1(config-if)# ipv6 nd ra suppress
R1(config-if)# ipv6 address 2001:11::1/64
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)# interface e0/1
R1(config-if)# ipv6 enable
R1(config-if)# no shutdown
```

R2 es parecido a R1 pero hay que activar el envío de los ICMP router advertisement y en la configuración de la IPv6 global, solo se asigna el routing-prefix `2001:12::/64` y se añade la opción eui-64 para configurar la interfaceID de la MAC.

```
R2(config)# ipv6 unicast-routing
R2(config)# interface e0/0
R2(config-if)# ipv6 address 2001:12::/64 eui-64
R2(config-if)# no shutdown
R2(config-if)# exit
R2(config)# interface e0/1
R2(config-if)# ipv6 enable
R2(config-if)# no shutdown
```

Finalmente el RIPng en R1 se activa con los siguientes comandos (en R2 se usan los mismos).

```
R1(config)# ipv6 unicast-routing
R1(config)# ipv6 router rip ejemplo
R1(config)# interface e0/0
R1(config-if)# ipv6 rip ejemplo enable
R1(config)# interface e0/1
R1(config-if)# ipv6 rip ejemplo enable
```

1.6. Realización de la práctica

Cada grupo necesita 2 routers conectados por un cable serie y tres PCs.

1.6.1. Parte 1

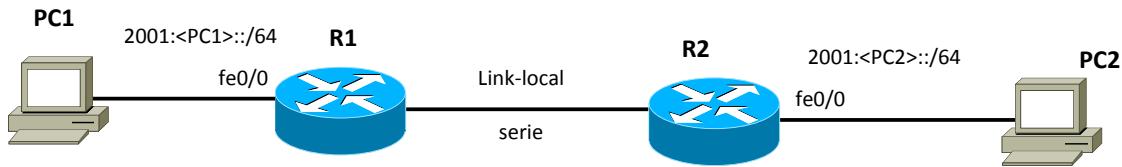


Figura 4: Red de la primera parte de la práctica

La Figura 4 representa la red de esta primera parte. Notar que para esta práctica se ha decidido elegir dos direcciones globales donde parte de las IPv6 depende del número del PC (<PC1> y <PC2>); de esta manera se asegura que no haya duplicidades. Para esta primera parte se necesitan dos PCs.

Seguir los siguientes pasos:

1. Configurar la IPv6 de PC1 y PC2 en modo stateless.
2. Activar el ICMP router advertisement en R1 para que configure y configurar la interfaz fastethernet de R1 (la que conecta PC1) con el routing-prefix de la figura y el interfaceID como conversión de la MAC a 64 bits.
3. Hacer lo mismo para R2.
4. Configurar las IPv6 de tipo link-local exclusivamente entre R1 y R2
5. Verificar que hay conectividad entre los dos routers y entre los routers y sus respectivos PCs.
6. Activar RIPng en todos los routers.
7. Comprobar que en las tablas de encaminamiento de todos los routers aparecen los prefijos. Comprobar que aparecen correctamente las entradas que corresponden a las redes directamente conectadas y las redes aprendidas por RIPng.
8. Comprobar que hay conectividad entre cualquier pareja de PCs.
9. Usar los comandos de verificación de RIPng e intentar interpretar los resultados.

1.6.2. Parte 2

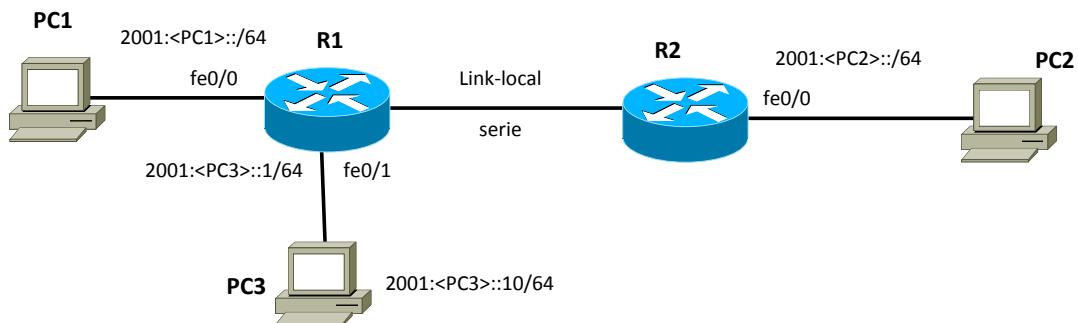


Figura 5: Red de la segunda parte de la práctica

La Figura 5 representa la red de esta segunda parte. En concreto se ha añadido el PC3. Este PC3 se quiere configurar manualmente.

Seguir los siguientes pasos:

1. Configurar la IPv6 global de PC3 y una ruta por defecto.
2. Configurar la IPv6 global de R1 hacia PC3 y activar el RIP también en esta interfaz.
3. Comprobar que hay conectividad entre todos los PCs y que las tablas de encaminamiento de los dos routers son correctas.

1.6.3. Parte 3

En esta tercera parte, hay que interconectar los routers R2 de cada dos grupos usando la interfaz FastEthernet fe0/1 y seguir los siguientes pasos:

1. Configurar la interfaz fe0/1 del router como link local.
2. Activar el RIP también en esta interfaz.
3. Comprobar que hay conectividad entre todos los PCs y que las tablas de encaminamiento son correctas.

Lab. 2 - Intra-Domain Routing: OSPF

2.1. Introducción a OSPF (RFC 2328)

Las características básicas son:

- Estandarizado por el IETF con el objetivo de tener un protocolo IGP no propietario de altas prestaciones.
- Es un protocolo de tipo *link state*: Esto significa que el router monitoriza y envía al resto de routers de la red información sobre las redes directamente conectadas y routers vecinos (*link state* se refiere a esa información). Las redes pueden ser de 4 tipos: Point-to-point, Broadcast, non-broadcast multiacces (NBMA), o point-to-multipoint.
- Cada router mantiene una base de datos con información de la topología de la red. Cada entrada de la base de datos consiste en la información recibida de algún router.
- Cada router envía su información local a todos los demás routers de la red usando *flooding*. Estos mensajes se denominan *Link State Advertisements* (LSAs). El encaminamiento por *flooding* consiste básicamente en enviar los datagramas por todas las interfaces excepto por la que ha llegado el mensaje. De este modo, el mensaje se propaga por toda la red, sin necesidad de usar tablas de encaminamiento.
- Los routers usan el algoritmo *Shortest Path First* (SPF), para calcular las entradas de encaminamiento óptimas, en función de la información almacenada en la base de datos.
- La métrica es adimensional (no representa el número de saltos). La métrica infinito es 0xFFFF.
- Existe un protocolo de *hello*, que consiste en enviar paquetes de señalización periódicamente. Este protocolo permite descubrir los routers vecinos, y saber si alguno de ellos deja de ser accesible.
- Para reducir el número de *floodings* en las redes broadcast con más de 1 router se elige un *Designated Router* (DR) y un *Backup Designated Router* (BDR). El DR es el único router del dominio broadcast que envía LSAs al resto de la red.
- Cada router se identifica con un número de 32 bits llamado *Router ID* (RID). Normalmente se escoge la dirección IP de mayor valor del router. Si se asigna una dirección a la interfaz de loopback, se escoge ésta aunque no sea la de mayor valor. Es recomendable asignar una dirección IP al loopback para que no cambie el RID en caso de cambiar las direcciones del router.
- Para la elección del DR y BDR se puede usar una prioridad (por defecto vale 1, si es igual a 0 significa que el router no puede ser elegido DR, BDR). En caso de igual prioridad, se escoge el router de mayor RID.
- El protocolo permite agrupar un conjunto de redes y routers contiguos en una “área”. El uso de múltiples áreas incrementa la escalabilidad y reduce el tráfico generado por el protocolo.
- Debe existir siempre el área 0, que hace de *backbone*, al cual se conectan todas las otras áreas. Si hay áreas no conectadas directamente al área 0, o existe alguna discontinuidad en el área 0 deben definirse *Virtual Links*.
- Los routers pueden ser *Internal Routers* (IR), si tienen todas las interfaces en la misma área; *Area Border Routers* (ABR) si tienen interfaces en más de un área; o *Autonomous System Boundary Routers* (ASBR) si anuncian rutas de otros protocolos de routing (estático, RIP, BGP, etc).

2.2. Configuración de OSPF en un router CISCO

2.2.1. Configuración básica en un área

Para configurar el algoritmo de encaminamiento OSPF en una única área (por ejemplo el RA de la Figura 6), los pasos a seguir son los siguientes.

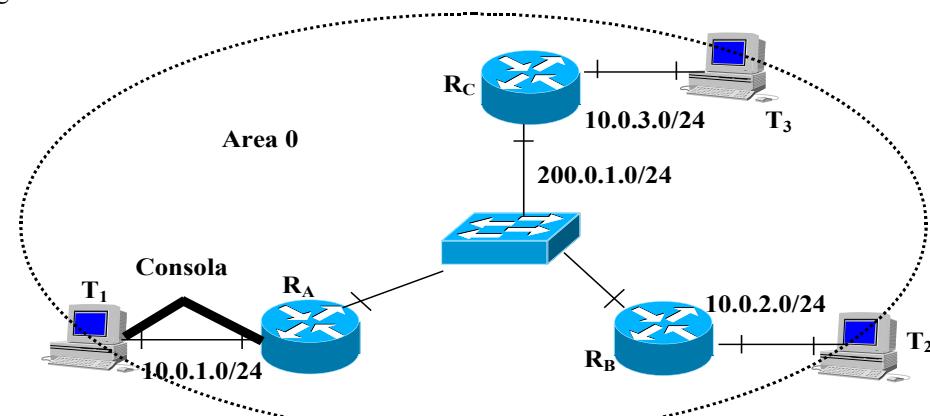


Figura 6: Red área única.

Primero conviene configurar una IP en la interfaz de loopback para identificar el routerID. En un router CISCO debe ser distinta de la red 127.0.0.0/8, pues esta red está reservada como *host loopback* y no permiten su uso. La IP de la interfaz loopback de un router puede ser tanto una IP pública como una IP privada.

```
Router# configure terminal
Router(config)# interface loopback0
Router(config-if)# ip address 192.168.0.7 255.255.255.0
```

El comando “**router ospf process-id**” para crear un proceso OSPF en el router. “*process-id*” es un identificador del proceso OSPF para el caso de que haya múltiples procesos OSPF ejecutándose en el router y es un número escogido por el administrador del sistema. Para indicar las redes que se deben anunciar se usa el comando “**network NetID WildcardMask area area-id**”. El comando “**network**” indica las interfaces que van a enviar o procesar mensajes de encaminamiento.

```
R_A# configure terminal
R_A(config)# router ospf 1
R_A(config-router)# network 200.0.1.0 0.0.0.255 area 0
R_A(config-router)# network 10.0.1.0 0.0.0.255 area 0
```

2.2.2. Configuración en múltiples áreas

Para configurar el algoritmo de encaminamiento OSPF en más de un área los pasos a seguir son los siguientes.

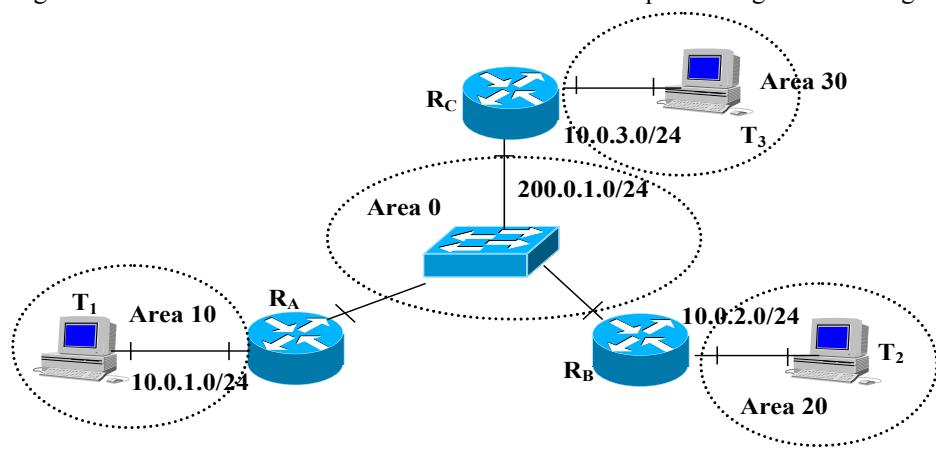


Figura 7: Red multi-área.

Si hay más de un área, siempre debe haber un área 0 que haga de backbone (troncal). Hay que configurar el área de backbone (área 0) y a continuación el resto de áreas (diseño jerárquico).

A las rutas que se generan dentro de un área se les llama **intra-area-routes** y aparecerán en la tabla de encaminamiento identificadas con la letra **O**. A las rutas aprendidas de otra área se les llama **inter-area-routes** o **summary-routes** y aparecerán en la tabla de encaminamiento identificadas con la letra **O IA**. A las rutas inyectadas desde otros protocolos de encaminamiento (usando redistribución de rutas) se les llama **external-routes** y aparecerán en la tabla de encaminamiento identificadas con la letra **O E1** (tipo 1 significa que el coste es la suma del protocolo interno más el externo) o **O E2** (tipo 2 significa que el coste es siempre el del protocolo externo). Por defecto OSPF siempre redistribuye con tipo 2.

En el caso de la Figura 7, la configuración del router R_A es la siguiente.

```
R_A(config)# interface e0
R_A(config-if)# ip address 10.0.1.1 255.255.255.0
R_A(config-if)# no shutdown
R_A(config-if)# exit
R_A(config)# interface e1
R_A(config-if)# ip address 200.0.1.1 255.255.255.0
R_A(config-if)# no shutdown
R_A(config-if)# exit
R_A(config)# router ospf
R_A(config-router)# network 200.0.1.0 0.0.0.255 area 0
R_A(config-router)# network 10.0.1.0 0.0.0.255 area 10
```

Para configurar un área como stub hay que añadir este comando en todos los routers que tienen por lo menos una interfaz en el área

```
area number stub
```

Para un área totalmente stub el comando es

```
area number stub no-summary
```

2.3. Modificación del comportamiento de OSPF

2.3.1. Prioridad y métricas

Para modificar la prioridad de un router en la elección del DR/BDR.

```
ip ospf priority number
```

donde “number” es un número entre 1 y 255. Prioridad 0 implica que el router no puede ser elegido DR o BDR, el valor por defecto es 1 y a mayor valor el router es elegido como DR o BDR.

La métrica (o coste) por defecto usada en OSPF es el ancho de banda. En un router CISCO el coste de un enlace se calcula como $10^8/\text{bandwidth}$ (bps). Por ejemplo si tenemos un enlace Ethernet a 10 Mbps el coste sería $10^8/10^7=10$, mientras que un modem a 56 Kbps tendría un coste de $10^8/(56*10^3)=1785$. Aquellos enlaces que tiene un ancho de banda superior a 100 Mbit/s, por ejemplo 1 Gbit/s, tendrán un coste de 1.

El SPF es un algoritmo de mínimo coste. Podemos modificar el coste de un enlace de tres maneras (en la configuración específica de la interfaz):

- modificando el valor del coste en la interfaz de ese enlace con el comando:

```
ip ospf cost cost
```

donde “cost” tiene un valor entre 1 y 65535.

- Modificando el ancho de banda de referencia (por defecto 10^8) con el comando:

```
auto-cost reference-bandwidth value
```

donde value es un valor en Mbit/s. Generalmente se fija como referencia el ancho de banda de la interfaz más rápida del sistema, de manera que esta tenga 1 y las demás costes más grandes.

- modificando el valor del bandwidth en la interfaz que permite calcular el coste con el comando:

```
bandwidth value
```

Con este comando no se cambia la velocidad real del enlace, solo el coste usado por SPF.

Se pueden cambiar los valores de periodicidad de los temporizadores de paquetes Hello: hello-interval (tiempo entre paquetes hello, por defecto es 10 s) y dead-interval (tiempo que considera que el enlace ha caído, por defecto es 40 s):

```
R#(config)# interface s0
R#(config-if)# ip ospf hello-interval 30
R#(config-if)# ip ospf dead-interval 120
```

2.3.2. Sumarización de rutas

Se pueden sumarizar las redes entre áreas. Las redes dentro de un área deben asignarse de forma que sean contiguas. La summarización se especifica en los ABR.

```
Router(config)# router ospf 1
Router(config-router)# area 1 range 200.0.1.0/19
```

Este comando suma las redes del área 1 en una única entrada 200.0.1.0/19 (/19 = 255.255.255.224.0).

2.3.3. Distribución de la ruta por defecto

Si se quiere que el protocolo OSPF inyecte la ruta por defecto y la anuncie a todos los routers OSPF se puede usar el comando “**default-information originate**”.

```
Router# configure terminal
Router(config)# ip route 0.0.0.0/0 150.0.0.1           -> define una ruta por defecto
Router(config)# router ospf 1
Router(config-router)# default-information originate
```

2.4. Verificación

| | |
|----------------------------------|---|
| R# show ip route | Permite ver la tabla de encaminamiento |
| R# show ip route ospf | Permite ver la tabla de encaminamiento solo para entradas OSPF |
| R# show ip ospf interface | Lista información relacionada con una interfaz que usa OSPF. Permite comprobar si las interfaces pertenecen al área a la que se suponen deberían pertenecer. También permite averiguar si una interfaz es DR, BDR o DROTHER (no es ni DR ni BDR), su prioridad y si la red es de tipo BMA o NBMA. |
| R# show ip ospf | Lista el número de veces que el algoritmo SPF (Short-First Path) se ha ejecutado |
| R# show ip ospf neighbor | Lista información acerca de los vecinos OSPF por cada interfaz |
| R# show ip ospf database | Lista los contenidos de la DB topológica |
| R# debug ip ospf "op" | Donde “op” son distintas opciones permiten debuguear la distintas operaciones que ejecuta OSPF (adjacency, events, etc) |

2.4.1. Ejemplos

Notar que el comando “**show ip ospf interface**” permite verificar gran parte de la información sobre una interfaz. Eso incluye el router-id, el DR, el BDR, su prioridad, sus adyacencias, los costes, etc. Otro comando bastante útil que nos permite averiguar información sobre OSPF es “**show ip ospf database**”, que nos proporciona información sobre la base de datos. El router tendrá una base de datos por cada área en la que participa. Esta base de datos, por área, es jerárquica y tiene varios niveles. Nos interesan 3 de los niveles para un sistema multi-área sin conexión a otro AS.

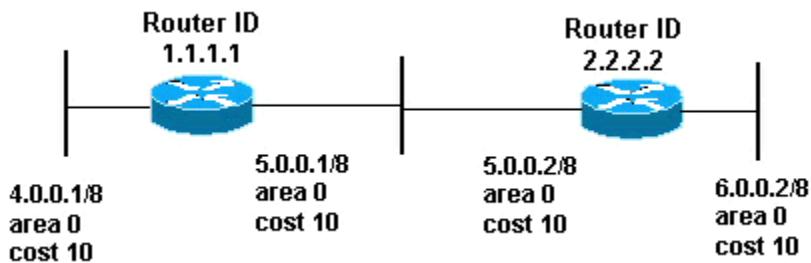


Figura 8: Red ejemplo para la base de datos OSPF.

- **Router Link State Database:** incluye la información de los enlaces del router que advierte el mensaje. En el caso de la Figura 8, el router RID=1.1.1.1 enviaría un LSA a RID=2.2.2.2 indicando que tiene 2 enlaces: el enlace 4.0.0.0/8 y el enlace 5.0.0.0/8. Por tanto, en la Router Link State Database de 2.2.2.2 veríamos:

```

r2.2.2.2#show ip ospf database

OSPF Router with ID (2.2.2.2) (Process ID 2)

        Router Link States (Area 0)

Link ID      ADV Router      Age      Seq#          Checksum      Link count
1.1.1.1      1.1.1.1        107      0x80000018      0x7966        2
2.2.2.2      2.2.2.2        106      0x80000015      0x6770        2
  
```

Notar que campo “Link ID” contiene el RID del router que ha enviado el LSA, mientras que el campo “ADV router” también contiene el RID del que ha enviado el mensaje. La información importante es que tienen 2 enlaces cada uno de ellos. Esto es debido a que cada uno de ellos ha ejecutado el comando network sobre 2 subredes. Los campos Age y Seq# indican respectivamente desde hace cuando se recibido este LSA y cuantas actualizaciones del mismo LSA se han recibido. En el caso de Seq#, el numero de secuencia inicial es siempre 0x80000000. En el caso de age, cada 1800 segundos (valor por defecto) el router que ha originado este LSA debe enviar un refresh del mismo LSA a todos los demás routers. Si pasado un MaxAge (3600 segundos, por defecto), un router no ha recibido un refresh, borra el LSA de su base de datos (y actualiza la tabla de encaminamiento).

Ejecutando este comando

```

r2.2.2.2#show ip ospf database router 1.1.1.1

(Link ID) Network/subnet number: 4.0.0.0
(Link Data) Network Mask: 255.0.0.0
Number of TOS metrics: 0
TOS 0 Metrics: 10
  
```

veríamos que ha recibido de r1.1.1.1 información sobre la red 4.0.0.0/8 con coste=10 (metrics).

- **Network Link State Database:** incluye información sobre los routers que hay en una red. Por lo tanto es una indicación de que RID hay en cada red.

```

r2.2.2.2#show ip ospf database

OSPF Router with ID (2.2.2.2) (Process ID 2)

        Network Link States (Area 0)

Link ID      ADV Router      Age      Seq#          Checksum      Link count
2.2.2.2      2.2.2.2        ----
2.2.2.2      1.1.1.1        ----
  
```

En este caso, la información que encontraremos en el campo “Link ID” no será el RID como antes, sino la IP@ del DR de la red. En cambio, el campo “ADV router” contiene el RID del que ha enviado el mensaje, y por tanto, el RID del router que es parte de la red. Eso significa que a partir del campo “ADV router” podemos obtener todos los RID de los routers de una misma red broadcast, ya que su campo “Link ID” será el mismo.

2.5. Realización de la práctica

La práctica tiene dos partes. En la primera parte se configura una red de un área, en la segunda parte se configuran 3 áreas.

2.5.1. Parte 1 – Única área

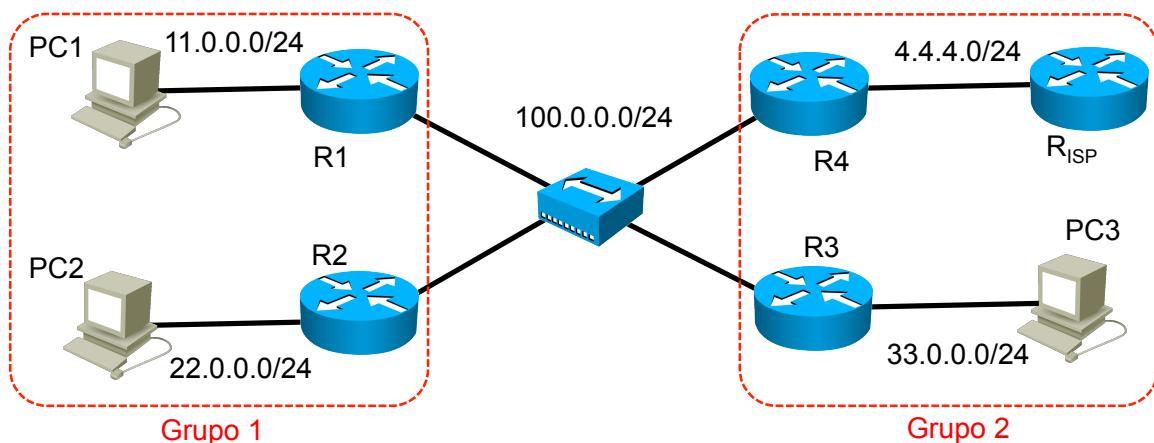


Figura 9: Topología de la red de la parte 1.

Configurar la red de la Figura 9 siguiendo los pasos que se indican a continuación (es importante respetar el orden indicado):

1. Configurar todas las interfaces de los routers y añadir una interfaz de loopback en cada router. Asignaremos una dirección $10.0.<\text{PC}\#>.1/30$ para el loopback de cada router. Escoger como número $\text{PC}\#$, el número del ordenador, es decir, si tiene 115, escoger la red $10.0.115.1/30$ para la loopback de ese router.
2. Configurar las IPs de los PCs y una ruta por defecto (usar los comandos **ifconfig** y **route add** de Linux). Comprobar con ping que hay conectividad entre interfaces de una misma red.
3. Configurar OSPF (comando **network**) en los routers en una única área 0. Comprobar con ping que todos los routers se ven.
4. El PC llamado Internet representa el router del ISP. Configurar la ruta por defecto en R4 y redistribuirla a los demás routers usando OSPF. Comprobar con ping que hay conectividad entre los PCs e Internet.
5. Comprobar que en las tablas de encaminamiento de todos los routers aparecen las redes. Comprobar que aparecen correctamente las entradas que corresponden a las redes directamente conectadas y las redes de la misma área. Interpretar las métricas.
6. Usar los comandos de verificación de ospf (por ejemplo **show ip ospf interface**), e intentar interpretarlos (por ejemplo RouterID, DR y BDR en los enlaces broadcast, routers adyacentes, etc.).
7. Probar de desconectar algún enlace y comprobar cómo las tablas se actualizan en pocos segundos (conectarlo otra vez).
8. Hacer un shutdown del enlace DR y comprobar la elección del nuevo DR y del nuevo BDR. Asignar prioridades a alguna de las interfaces para fijar la nueva elección.

2.5.2. Parte 2 – Múltiples áreas

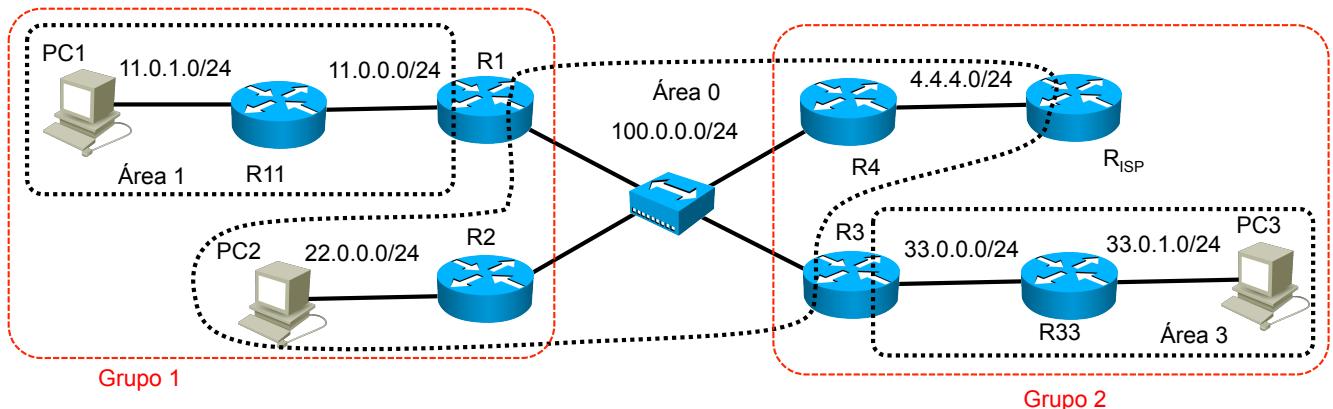


Figura 10: Topología de la red de la parte 2.

La red se divide ahora en 3 áreas con las direcciones mostradas en la Figura 10. Configurar la red siguiendo los pasos que se indican a continuación:

1. Modificar la configuración de la red y de las @IP anteriores según el nuevo esquema:
 - a. Conectar un nuevo router entre R1 y PC1 (R11) y uno entre R3 y PC3 (R33).
 - b. Modificar las direcciones IP de PC1 y PC3 y configurar sus nuevas rutas por defecto
 - c. Asignar IPs al nuevo router
 - d. Comprobar conectividad con ping
2. Modificar el OSPF de acuerdo con las áreas indicadas en la figura (comando “**no network**” para eliminar las redes que se anunciaban, comando “**network**” para activarlas). Comprobar con ping que todos los routers se ven.
3. Comprobar que en las tablas de encaminamiento de todos los routers aparecen las redes. Comprobar que aparecen correctamente las entradas que corresponden a las redes directamente conectadas, redes de la misma área y redes inter-área. Interpretar las métricas.
4. Cada grupo puede probar a modificar el tipo de la nueva área creada. Comparar como quedan las tablas de encaminamiento de los routers (sobre todo R2 y R4) cuando la nueva área es normal, stub o totalmente stub.
5. Activar la summarización de rutas en el área 1 y 3 (pensar bien como sumarizar las dos redes de cada área), y comprobar cómo quedan las tablas de encaminamiento.
6. Usar los comandos de verificación de ospf, e intentar interpretarlos (por ejemplo RouterID, DR y BDR en los enlaces broadcast, routers adyacentes, etc.).

Lab. 3 – MultiProtocol Label Switching (MPLS)

3.1 Objetivos de la práctica

El objetivo de la presente práctica es familiarizarse con la tecnología y los conceptos de MPLS (Multiprotocol Label Switching), así como su configuración.

3.2 Introducción a MPLS

La tecnología MPLS (Multiprotocol Label Switching) también es conocida como tecnología de la capa 2.5, porque realiza un encapsulado intermedio entre la capa de enlace (capa 2) y la capa de red (capa 3). En este encapsulado se introduce una etiqueta de 4 bytes, que permite a los routers utilizar técnicas de conmutación. El utilizar el etiquetado por debajo de capa 3, permite que MPLS pueda funcionar independientemente del protocolo de capa 3 utilizado, de ahí lo de “multiprotocol”. Esta arquitectura de etiquetado es flexible y permite anidar etiquetas, es decir, introducir una trama MPLS dentro de otra.

El objetivo de MPLS es separar la parte de encaminamiento (plano de control) de la parte de conmutación (plano de forwarding) en el reenvío de los paquetes, de forma que mientras la parte de encaminamiento es compleja y lenta (tiempos de convergencia, cálculo de rutas), se realiza independientemente de la parte de conmutación, que es rápida y simple.

De forma muy simplificada, se podría decir que los routers inicialmente calculan todas las rutas (usando protocolos de routing IP) a los destinos y luego intercambiando etiquetas establecen los circuitos virtuales (llamado Label Switched Path, LSP) entre cualquier origen y cualquier destino para empezar a conmutar. Las etiquetas introducidas a los paquetes cuando entran en la red MPLS están asociadas al LSP que seguirá el paquete en la red hacia un destino determinado y estas etiquetas se introducen en el paquete (*label push*), antes de la cabecera de capa 3. Las etiquetas que se añaden sólo tienen significado local al nodo MPLS (el router) y van cambiando salto a salto (*label swap*). Así de esta manera, el paquete entra en la red (a través de los routers MPLS frontera) y se le añade una etiqueta según el LSP para su destino, el paquete es conmutado dentro de la red (a través de los routers MPLS internos) cambiando en cada salto la etiqueta y finalmente sale de la red MPLS (a través de los routers MPLS frontera) próximo al destino, quitándole la etiqueta (*label pop*).

Las principales aplicaciones de MPLS son funciones de ingeniería de tráfico (a los flujos de cada usuario se les asocia una etiqueta diferente), routing basados en políticas (Policy Routing), servicios de VPN, servicios que requieren QoS, etc y según la aplicación.

3.2.1 Terminología MPLS

La terminología más importante y necesaria para esta práctica de MPLS es:

- **Forwarding Equivalence Class (FEC):** 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.
- **Label Switched Path (LSP):** camino que siguen los paquetes que pertenecen a la misma FEC, es equivalente a un circuito virtual.
- **Label Switching Router (LSR):** router que puede encaminar paquetes en función del valor de la etiqueta MPLS
- **Label Distribution Protocol (LDP):** protocolo utilizado para distribución de etiquetas MPLS.
- **Label Information Base (LIB):** la tabla de etiquetas que manejan los LSR. Relaciona la pareja (interfaz de entrada - etiqueta de entrada) con (interfaz de salida - etiqueta de salida).
- **Forwarding Information Base (FIB):** 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.
- **Label Forwarding Information Base (LFIB):** 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.
- **Penultimate Hop Popping (PHP):** 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.

3.2.2 Ejemplo de funcionamiento

En MPLS, el protocolo LDP está encargado de distribuir las etiquetas juntamente a los prefijos para que los routers construyen las tablas LIB and LFIB. En la Figura 11 se ilustra un ejemplo. Si se aplicara PHP, el router R2 podría hacer directamente label pop (es decir eliminar la etiqueta MPLS) ya que el prefijo destino se encuentra conectado directamente al siguiente router R1. Por defecto, CISCO aplica PHP.

Cabe destacar que MPLS necesita conectividad IP entre todos los routers para funcionar, es decir se necesita un protocolo de encaminamiento (por ejemplo OSPF) para que los routers conozcan todos los prefijos.

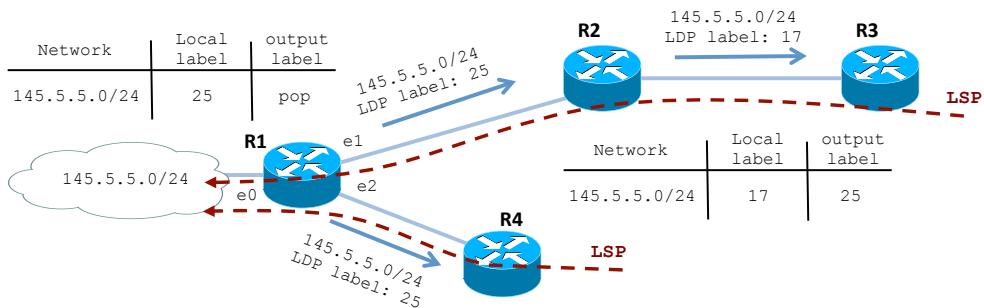


Figura 11: Ejemplo de distribución de etiquetas en MPLS.

3.2.3 Configuración de MPLS

Para activar el forwarding MPLS, en CISCO se necesita usar el mecanismo Cisco Express Forwarding (CEF). CEF es el método usado por CISCO para crear las FIB y, por consiguiente, las LFIB para MPLS. Se activa con el comando **ip cef** en configuración global.

Se necesita luego activar el protocolo LDP en los routers para el intercambio de etiquetas. Hay que destacar que en CISCO se pueden usar dos protocolos distintos, LDP y TDP (Tag Distributel Protocol). TDP es el protocolo propietario CISCO anterior a MPLS. El comando es **mpls label protocol ldp**.

Finalmente, hay que activar MPLS en cada interfaz con el comando **mpls ip**.

Ejemplo para el caso de R1 en la Figura 11.

```
R1(config)# ip cef
R1(config)# mpls label protocol ldp
R1(config)# interface e1
R1(config-if)# mpls ip
R1(config)# interface e2
R1(config-if)# mpls ip
```

3.2.4 Verificación MPLS

| | |
|--------------------------------------|---|
| R# show ip route | Permite ver la tabla de encaminamiento |
| R# show ip route @IP | Permite ver el camino hasta la @IP conocido por el router |
| R# show ip cef | Permite ver la tabla de forwarding |
| R# show mpls interfaces | Permite ver que interfaces usan MPLS y su estado |
| R# show mpls ldp discovery | Permite obtener información de LDP local y de los vecinos |
| R# show mpls ldp neighbor | Permite ver las adyacencias LDP y conocer su estado |
| R# show mpls ldp bindings | Permite ver la tabla LIB |
| R# show mpls forwarding-table | Permite ver la tabla LFIB |

3.3 MPLS Traffic Engineering (MPLS-TE)

El funcionamiento de MPLS se puede extender con ingeniería de tráfico (Traffic Engineering, TE). En el caso tradicional, el coste asociado a cada interfaz es un valor estático, asignado manualmente o automáticamente según la tecnología usada. Este sistema crea un uso no óptimo de los recursos disponibles. En efecto, todos los datagramas irían siempre y exclusivamente por los caminos de coste mínimo creando posibles zonas de congestión. Por el contrario, TE correlaciona los flujos entre dos routers apropiadamente para permitir el uso eficiente de la red.

Para la distribución del estado real de cada interfaz (y no simplemente su coste inicial), las extensiones TE se han aplicado al protocolo de encaminamiento dinámico OSPF, creando por lo tanto el nuevo protocolo OSPF-TE.

Con la información actualizada sobre el estado de la red distribuida por OSPF-TE, MPLS-TE puede computar caminos LSP optimizando los recursos de la red y capaces de cumplir con determinados requerimientos de los clientes. En este caso se dice que MPLS usa un algoritmo de encaminamiento llamado Constraint Based Routing (CBR), es decir, dadas las restricciones de los clientes, CBR calcula el mejor camino LSP posible.

Para crear el LSP, MPLS usa un protocolo de señalización llamado RSVP-TE (Resource Reservation Protocol with Traffic Engineering) que reserva los recursos necesarios entre routers.

Cabe destacar que en el caso de MPLS-TE, se suele llamar Tunnel el LSP creado por RSVP-TE.

3.3.1 Ejemplo de funcionamiento

En el ejemplo de la Figura 12, un cliente de la red 145.5.5.0/24 quiere establecer un camino de 30 Mbps hasta la red 200.0.0.0/24. El router R1 se configura para que establezca un LSP de 30 Mbps hasta R3. Con la información obtenida por OSPF-TE, R1 calcula con CBR el camino que permite esta capacidad hasta R3. R1 ejecuta RSVP-TE para crear este LSP de 30 Mbps y reservar estos recursos en cada router: 1) el mensaje PATH se lanza a lo largo del camino determinado por

CBR hasta R3 y comprueba que realmente hay suficiente recursos; 2) el mensaje RESV de vuelta a R1 reserva estos recursos y actualiza las tablas LIB. Luego, OSPF-TE distribuye el nuevo estado de cada interfaz a todos los routers.

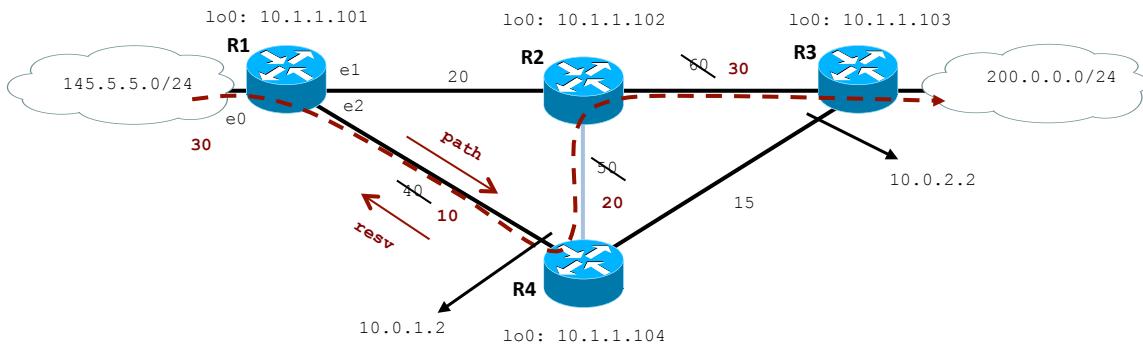


Figura 12: Ejemplo de MPLS-TE.

3.3.2 Activación de las extensiones TE

Se necesita activar las extensiones TE en todos los routers, en las interfaces y en OSPF. En el caso del router R4 de la Figura 12 sería la siguiente.

Comandos para activar MPLS-TE

```
R4(config)# mpls traffic-eng tunnels      -> habilita TE en el router
```

Comandos para activar las extesiones TE en las interfaces

```
R4(config)# interface e1
R4(config-if)# mpls traffic-eng tunnels      -> habilita TE en una interfaz
R4(config)# interface e2
R4(config-if)# mpls traffic-eng tunnels
```

Comandos para activar las extensiones TE en OSPF. En el ejemplo, OSPF está configurado con una única área y se usa como RID la interfaz de loopback0. Notar que es imprescindible que la mascara de la @IP usada como RID sea /32 (255.255.255.255).

```
R4(config)# router ospf 10          -> se entra en OSPF (definido anteriormente como proc. 10)
R4(config-router)# mpls traffic-eng area 0      -> habilita TE en el área 0
R4(config-router)# mpls traffic-eng router-id loopback0      -> habilita TE a través del RID loopback0
```

En RSVP-TE se pueden fijar dos parámetros para cada interfaz: 1) el máximo ancho de banda reservables para todos los túneles, 2) el máximo ancho de banda reservable para cada túnel. El comando es **ip rsvp bandwidth interface-bitrate [single-tunnel-bitrate]**. Los valores de interface-bitrate y single-tunnel-bitrate son en kbps (el segundo valor es opcional y se puede omitir). Por ejemplo, R4 se podría configurar de esta manera

```
R4(config)# interface e1
R4(config-if)# ip rsvp bandwidth 4000 1000      -> asigna 4000 kpbs de capacidad total a la interfaz e2
                                                y cada tunnel es de 1000 kbps
R4(config)# interface e2
R4(config-if)# ip rsvp bandwidth 5000      -> asigna 5000 kpbs de capacidad total a la interfaz e2
                                                y cada tunnel puede ser de cualquier capacidad
```

3.3.3 Creación del LSP (tunnel) en MPLS-TE

Para establecer un túnel, se necesitan diferentes pasos. Considerando la Figura 12, se quiere configurar un túnel de 100 kbps con origen R1 y destino R3. Los comandos se detallan a continuación.

```
R1(config)# interface tunnel10          -> se crea el túnel llamado tunnel10
R1(config-if)# ip unnumbered loopback0      -> se asigna la interfaz de loopback0 como ip del tunel
R1(config-if)# tunnel mode mpls traffic-eng      -> habilita el modo MPLS-TE en el túnel
R1(config-if)# tunnel destination 10.1.1.103      -> se crea el túnel hasta la loopback de R3
R1(config-if)# tunnel mpls traffic-eng bandwidth 100      -> bitrate del túnel en kbps
R1(config-if)# tunnel mpls traffic-eng autoroute      -> se anuncia el túnel por OSPF
```

Estos comandos crean un túnel tunnel10 de R1 a R3 (10.1.1.103), es decir los túneles son unidireccionales. Si se quisiera establecer un túnel también en el sentido contrario, habría que aplicar los mismos comandos en R3 con destino R1. El nombre del túnel es arbitrario; la única restricción es que no se puede repetir el mismo nombre para dos túneles con una misma origen.

El túnel se puede crear de manera dinámica o explícita. En el caso dinámico, el camino se determina según el conocimiento que tienen los routers (proporcionado por OSPF-TE). En el caso explícito, se pueden fijar una directrices a la hora de escoger el camino. El comando es **tunnel mpls traffic-eng path-option número {dynamic|explicit {name nombre-camino}}**. A continuación se ilustra un ejemplo.

```
R1(config-if)# tunnel mpls traffic-eng path-option 1 explicit name LP1      -> túnel explícito llamado LP1
R1(config-if)# tunnel mpls traffic-eng path-option 2 dynamic      -> túnel dinámico
```

Cabe notar que el parámetro *número* en path-option indica el orden con el que se intenta establecer el túnel. En el ejemplo, antes se probaría con el camino explicito (opción 1) y, si no se pudiera establecer, entonces se probaría con la opción 2, dinámica.

En el caso de camino explicito, hay que asignar un nombre al camino (LP1 en el ejemplo) y luego indicar las directrices para establecerlo a través del comando **ip explicit-path name nombre-camino**. Por ejemplo, se puede indicar por que @IP tiene que pasar el túnel con el comando **next address @IP**. A continuación se define un camino explicito R1-R4-R3.

```
R1(config)# ip explicit-path name LP1
R1(cfg-ip-exp1-path)# next-address 10.0.1.2      -> @IP de la primera interfaz del camino explicito
R1(cfg-ip-exp1-path)# next-address 10.0.2.2      -> @IP de la segunda interfaz del camino explicito
R1(cfg-ip-exp1-path)# next-address 10.1.1.103     -> loopback final incluida en el camino explicito
```

Finalmente, a un túnel se le puede asignar una prioridad para manejar la relación con otros túneles a través del comando **tunnel mpls traffic-eng priority setup-priority hold-priority**. Los dos parámetros definen esta prioridad (siendo 0 la más alta y 7 la más baja). El segundo parámetro *hold-priority* determina la prioridad del túnel una vez establecido. El primer parámetro *setup-priority* determina la prioridad del túnel a la hora de establecerlo; si no se encontrasen recursos suficientes, esta parámetro da la posibilidad de eliminar aquellos túneles que tienen un *hold-priority* más alto.

MPLS permite aplicar balanceo de carga entre túneles que tienen mismo origen y destino. Por defecto CISCO aplica un balanceo de carga que depende del ancho de banda de los túneles; si un túnel tunnel1 tiene 10 Mbps y un segundo túnel tunnel2 con mismo origen y destino tiene 2 Mbps, MPLS aplicaría un balanceo de carga con un factor 5 a 1 entre tunnel1 y tunnel2.

3.1.1 Verificación MPLS-TE

| | |
|---|---|
| R# show mpls traffic-eng tunnels destination @IP | Visualiza los detalles de los túneles hasta @IP |
| R# show mpls traffic-eng tunnels brief | Visualiza los túneles creados |
| R# show ip rsvp sender | Visualiza las sesiones RSVP establecidas |

3.2 Realización de la práctica

Para la realización de esta práctica se usarán los router CISCO 1841 que tienen MPLS implementado en su IOS.

3.2.1 Parte 1

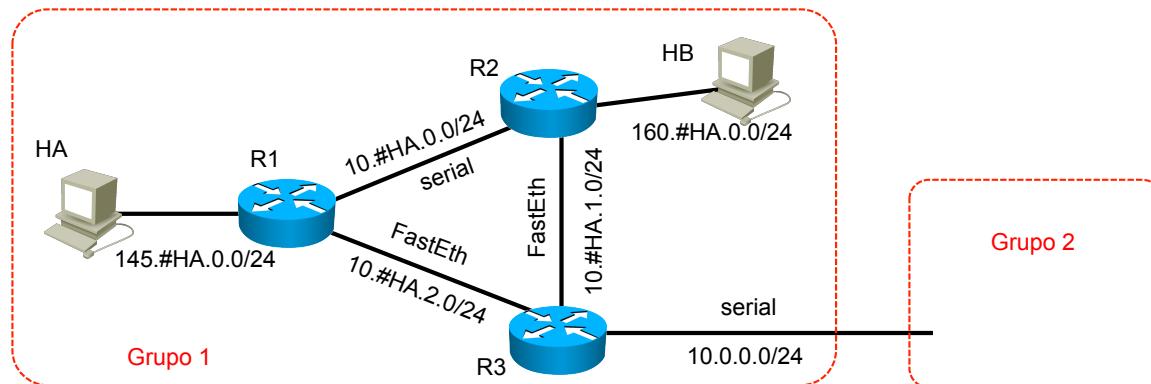


Figura 13: Red de la parte 1 de práctica.

Configurar la red de la Figura 13 siguiendo los pasos que se indican a continuación (es importante respetar el orden indicado):

1. Configurar las direcciones de loopback en R1-R3 como 1.#HA.1.X/32 donde #HA es el número del PC que hace de HA y X es el número del router RX.
2. Configurar las direcciones IP en todas las interfaces de la red que aparece en la figura de forma que hay conectividad entre dos interfaces vecinas. Notar que entre R1 y R2 se usa el enlace serie y también entre los routers R3 de los dos grupos. Recordar que para los enlaces serie hay que configurar la velocidad de transmisión¹.
3. Configurar encaminamiento interno (OSPF) en los routers. Distribuir las redes y las loopbacks por OSPF usando una única área. No distribuir OSPF hacia los hosts². Comprobar que hay conectividad entre los hosts.
4. Activar MPLS en los routers R1, R2 y R3.
5. Comprobar el estado de MPLS y de LDP y comprobar las tablas de encaminamiento, FIB, LIB y FLIB.

¹ El comando es **clockrate bitrate**. Para esta practica poner bitrate a 56000.

² Comando **passive-interface interface**. En el caso de R3, seria **R3(config-router)# passive-interface fastethernet1/0**

3.2.2 Parte 2

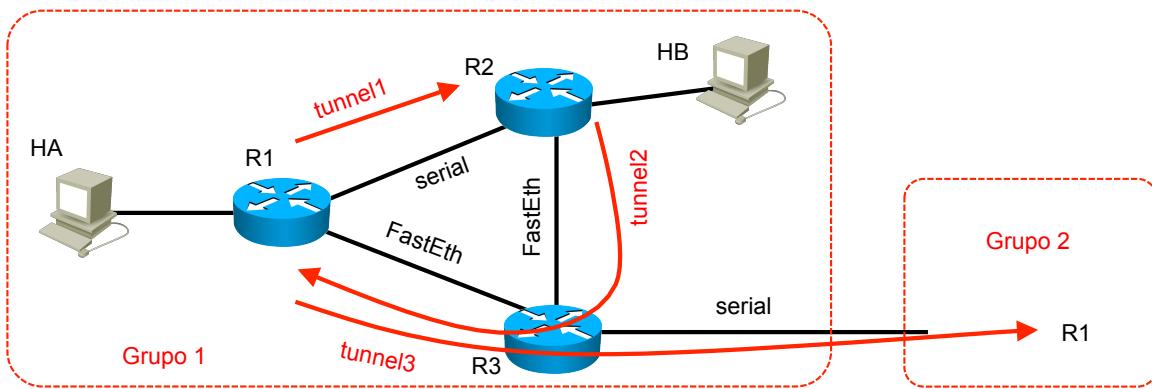


Figura 14: Red de la parte 2 de la práctica.

Manteniendo la misma topología, activar los 3 túneles con MPLS-TE como en la Figura 14 siguiendo los pasos que se indican a continuación (es importante respetar el orden indicado):

1. Manteniendo la configuración anterior, activar las extensiones TE en MPLS, en las interfaces (solo las que van hacia otro routers, no las que van a los hosts) y en OSPF como en explicado en la Sección 3.3.2.
2. Configurar RSVP con ancho de banda máximo de 1000 kbps en las interfaces fastethernet y de 100 kbps en las series como indicado en la Sección 3.3.2. No configurar el ancho de banda máximo por túnel.
3. Activar el túnel Tunnel1 de 50 kbps entre R1 y R2 entre las interfaces de loopbacks de R1 y R2. Comprobar que el túnel está activo.
4. Activar el túnel Tunnel2 de 100 kbps entre R2 y R1 entre las interfaces de loopbacks de R2 y R1. Comprobar que el túnel está activo.
5. Comprobar, usando Traceroute, que un ping entre HA y HB pasa por los túneles.
6. Activar el túnel Tunnel3 de 50 kbps entre los routers R1 de cada grupo usando las interfaces de loopback. Comprobar que el túnel está activo.

Lab. 4 - Inter-Domain Routing: BGPv4

4.1. Introducción a BGPv4 (RFC 4271)

Las características básicas son:

- Es un protocolo de encaminamiento externo que permite crear rutas entre sistemas autónomos (AS). En cada AS puede operar cualquier encaminamiento interno tipo RIP u OSPF.
- Un router que tiene un proceso BGP activo se llama *BGP speaker*. Para poder intercambiar información de encaminamiento BGP, dos routers vecinos (dos BGP speakers) deben establecer una sesión BGP a través del puerto 179 de TCP. En este caso estos dos routers se llaman *peers* o *neighbors*.
- BGP es un protocolo de tipo *path vector*. Es decir BGP recae en la categoría general de los protocolos *vector distancia* como RIP donde la mejor ruta es la que tiene menos saltos hasta el destino. BGP tiene pero algunos mecanismos adicionales. La información de encaminamiento BGP es una secuencia de números que identifican los diferentes ASes que hay que atravesar para llegar a un AS destino. Esta información evita la creación de bucles en las rutas. BGP además permite crear políticas de encaminamiento a través de una serie de atributos.
- Un AS puede ser de tipo *stub*, *multihomed* o de *transito*. Stub cuando un AS tiene una única sesión BGP abierta con otro AS y solo recibe y transmite su tráfico. Multihomed en el caso que un AS tenga mas de un AS conectado por BGP (por si uno falla) pero no deja que trafico de un AS pase por el con destino otro AS. De *transito* cuando el AS proporciona servicio de transito entre dos ASes.
- Una sesión BGP que conecta dos routers de dos AS distintos se llama BGP externo (eBGP). En el caso que el AS sea de transito, los routers del AS que mantienen un eBGP deben tambien establecer una sesión BGP entre ellos, llamada BGP interna (iBGP), para que estos puedan redistribuir la información BGP entre los ASes.
- Hay cuatro tipos de mensajes BGP: *open*, *update*, *keepalive* y *notification*. *open* se utiliza para el establecimiento de la sesión BGP; *update* cuando hay una modificación de una ruta o se ha encontrado una nueva ruta; periódicamente dos routers vecino se envían mensajes de *keepalive* para para verificar que la sesión BGP sigue activa; *notification* notifica el cierre de una sesión BGP debido a algún error.

4.2. Configuración de BGP en un router CISCO

4.2.1. Configuración básica

El comando “**router bgp AS-number**” crea un proceso BGP en el router donde “*AS-number*” es el numero que identifica el sistema autónomo (en Internet es un numero que asigna RIR).

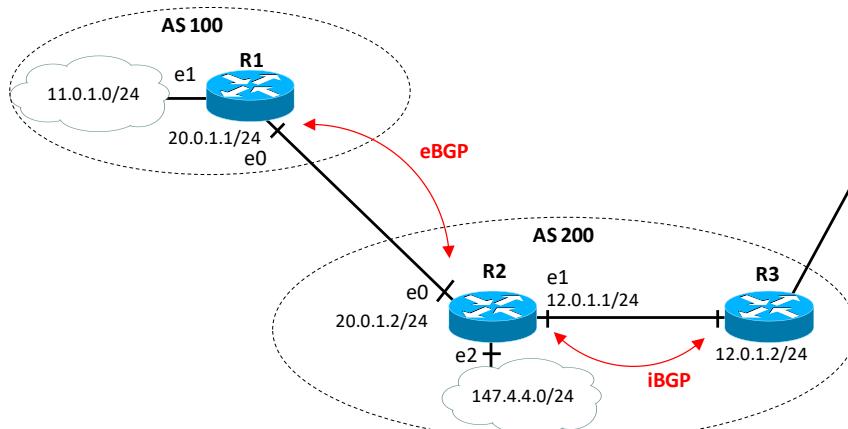


Figura 15: Ejemplo de red con diferente AS.

Para crear una sesión eBGP con un router vecino como el de la Figura 15, se usa el comando “**neighbor @IP remote-as AS-number**” donde @IP es la dirección IP del router vecino y AS-number es su numero de sistema autónomo. Notar que no hace falta ejecutar este comando cada vez que se quiere modificar o añadir algún prefijo.

| | |
|--|-----------------|
| R1(config)# router bgp 100 | -> el propio AS |
| R1(config-router)# neighbor 20.0.1.2 remote-as 200 | -> el otro AS |

| |
|--|
| R2(config)# router bgp 200 |
| R2(config-router)# neighbor 20.0.1.1 remote-as 100 |

La creación de una sesión iBGP sigue los mismos pasos del caso eBGP con la diferencia que en esta caso el sistema autónomo será el mismo numero. Por ejemplo para el iBGP de R2 con R3 será:

| |
|----------------------------|
| R2(config)# router bgp 200 |
|----------------------------|

```
R2(config-router)# neighbor 12.0.1.2 remote-as 200 -> mismo AS
```

Para anunciar redes internas a cualquier router BGP vecino, se usa el comando “**network #net mask #mask**” donde **#net** es la red que se quiere anunciar y **#mask** su máscara (opcional pero necesario si la clase es distinta de la máscara). Por ejemplo para que el router R1 anuncie por BGP la red interna 11.0.1.0 con máscara 255.255.255.0 se usará el comando:

```
R1(config)# router bgp 100
R1(config-router)# network 11.0.1.0 mask 255.255.255.0
```

Fijaros que:

- Solo se anuncian por BGP aquellas redes que están conectadas directamente al router y que se quieren distribuir a otros AS
- No se anuncian aquellas redes que tienen sesiones BGP (es decir, aquellas que interconectan routers y que por tanto son internas al AS).

4.2.2. Configuración con interfaz de loopback

Por defecto, la sesión BGP entre routers se establece mediante la dirección IP de la interfaz del router vecino. Sin embargo, CISCO proporciona el comando “**update-source #iFace**” que permite que cualquier interfaz indicada en **#iFace**, incluida la de loopback, pueda ser utilizada para establecer una sesión BGP.

En particular es muy común usar la interfaz de loopback para establecer las sesiones iBGP. La razón es que de esta forma se puede mantener activa una sesión BGP en caso de fallo. En efecto, si se usara la interfaz física para establecer la conexión TCP del BGP y en un dato momento esta fallase, la sesión BGP caería y habría que volver a activarla. En cambio si se usa la interfaz de loopback para crear la conexión TCP y la interfaz física falla, la sesión BGP se mantiene de todas maneras activa. Se da de esta manera tiempo al protocolo de encaminamiento interno del AS (como por ejemplo OSPF) de encontrar rutas alternativas a las rutas que han caido. Usar la interfaz de loopback tambien permite que los routers BGP puedan correr con múltiples vínculos entre ellos y de esta forma hacer balanceo de carga entre las rutas disponibles.

Un ejemplo de configuración de iBGP entre R2 y R3 usando las interfaces de loopback.

```
R2(config)# interface loopback0
R2(config-if)# ip address 2.2.2.1 255.255.255.252
R2(config-if)# no shutdown
R2(config-if)# exit
R2(config)# router bgp 200
R2(config-router)# neighbor 3.3.3.1 remote-as 200
R2(config-router)# neighbor 3.3.3.1 update-source loopback0
```

```
R3(config)# interface loopback0
R3(config-if)# ip address 3.3.3.1 255.255.255.252
R3(config-if)# no shutdown
R3(config-if)# exit
R3(config)# router bgp 200
R3(config-router)# neighbor 2.2.2.1 remote-as 200
R3(config-router)# neighbor 2.2.2.1 update-source loopback0
```

Para que haya conectividad entre los dos routers a través de la interfaz de loopback es necesario tener una ruta. Esto puede ser bien a través de un protocolo de **encaminamiento interno dinamico** (tipo OSPF, ver la Sección 3.2.3) que distribuía también la de loopback a los routers del AS o a través de una ruta estatica.

En el caso de ruta estatica, considerando el ejemplo de la Figura 15, hay que indicar en R2 que para alcanzar la interfaz de loopback de R3 hay que transmitir al gateway 12.0.1.2.

```
R2(config)# ip route 3.3.3.0 255.255.255.252 12.0.1.2
```

4.2.3. Uso de OSPF y BGP

Un AS generalmente tiene un protocolo de encaminamiento interno activo que gestiona las tablas de encaminamiento de todos los routers. Este protocolo puede bien ser estatico o, en su mayoria, dinamico usando OSPF. En este caso el OSPF se debe configurar antes que el BGP de manera que los dos estén bien sincronizados. Tambien es importante que la o las interfaces conectadas a otros ASes no anuncien las redes internas por OSPF. Por esta razón se usa el comando “**passive-interface #iFace**” donde **#iFace** es la interfaz por donde no se envían información OSPF. Por ejemplo en el caso de R2 de la figura, la configuración del OSPF seria:

```
R2# configure terminal
R2(config)# router ospf 10
R2(config-router)# passive-interface e0
R2(config-router)# network 20.0.1.0 0.0.0.255 area 0
R2(config-router)# network 12.0.1.0 0.0.0.255 area 0
R2(config-router)# network 147.4.4.0 0.0.0.255 area 0
R2(config-router)# network 2.2.2.0 0.0.0.3 area 0 -> se incluye la red de la intefaz de loopback
R2(config-router)# exit
```

Y luego se puede activar el BGP interno a través de la interfaz de loopback y el externo.

```
R2(config)# router bgp 200
R2(config-router)# neighbor 3.3.3.1 remote-as 200          -> iBGP con R3
R2(config-router)# neighbor 3.3.3.1 update-source loopback0
R2(config-router)# neighbor 20.0.1.1 remote-as 100          -> eBGP con R1
R2(config-router)# network 147.4.4.0 0.0.0.255            -> red local que se anuncia por BGP
```

4.3. Verificación

| | |
|---------------------------------|--|
| R# show ip route | Permite ver la tabla de encaminamiento |
| R# show ip bgp | Permite ver la tabla de encaminamiento de BGP |
| R# show ip bgp neighbors | Lista los routers vecino conectados por BGP |
| R# show ip bgp paths | Lista los paths establecidos por BGP |
| R# show ip bgp summary | Lista el estado de las sesiones BGP |
| R# clear ip bgp * | Resetea las sesiones BGP |
| R# debug ip bgp "op" | Donde "op" son distintas opciones permiten debuguear la distintas operaciones que ejecuta BGP (events, keepalive, updates, etc.) |

4.4. Realización de la práctica

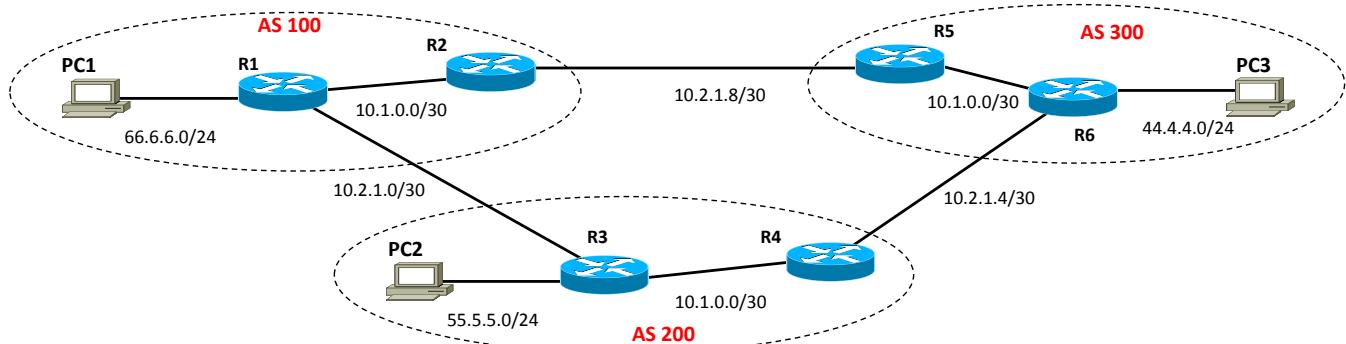


Figura 16: Topología de la red para esta práctica.

Configurar la red de la Figura 16 siguiendo los pasos que se indican a continuación (es importante respetar el orden indicado):

1. Configurar todas las interfaces de los routers y añadir una interfaz de loopback en cada router. Asignar una dirección $10.0.<\text{PC}\#>.1/32$ para el loopback de cada router R_n . Escoger como número $\text{PC}\#$, el número del ordenador, es decir, si tiene 115, escoger la red $10.0.115.1/30$ para la loopback de ese router.
2. Asignar las direcciones indicadas en la Figura 16. Configurar una ruta por defecto en los tres PCs. Comprobar con ping que hay conectividad entre interfaces de una misma red.
3. Activar OSPF en cada AS (comando **network**). Notar que hay que configurar el **passive-interface** en las interfaces que van hacia los PCs y hacia los routers de los otros ASes. Comprobar que los distintos ASes no ven las redes internas de los otros ASes. Comprobar con ping que los routers y el PC de un mismo AS se ven. Verificar el estado de las sesiones BGP.
4. Activar iBGP entre los routers del AS a través de sus interfaces de loopback. Comprobar que la sesión iBGP está activa.
5. Activar eBGP entre los routers de los distintos ASes. Comprobar que la sesión BGP entre routers están activas.
6. Comprobar que hay conectividad entre todos los PCs.
7. Comprobar que en las tablas de encaminamiento de todos los routers aparecen las redes. Comprobar que aparecen correctamente las entradas que corresponden a las redes directamente conectadas, las redes aprendida por OSPF y las redes aprendida por BGP.
8. Usar los comandos de verificación de BGP e intentar interpretarlos.
9. Desconectar el enlace entre R_1 y R_3 e interpretar lo que ocurre.

Recordar de ejecutar el comando **clear ip bgp *** para reenviar los mensajes update bgp y refrescar las tablas BGP.

Lab. 5 - BGPv4 manipulación de atributos

5.1. Objectivo de la práctica

Para implementar políticas de encaminamiento con BGP se usa el filtrado de rutas y la manipulación de atributos. Hay varias herramientas que permiten manipular y filtrar direcciones (no son excluyentes unas con otras):

- BGP route maps
- Prefix lists
- Identificación y filtrado de rutas basado en Network Layer Routing Information (NLRI)
- Identificación y filtrado de rutas basado en AS-PATH

En este Lab no veremos todos estos aspectos. Nos centraremos en los “route maps” como herramienta básica para la manipulación de los atributos Local Pref y MED.

5.2. Filtrado de rutas y manipulación de atributos

Los Route maps se usan en BGP para controlar y modificar la información de la tabla de encaminamiento BGP, para definir las condiciones por las cuales una ruta es distribuida entre dos routers y para modificar los atributos incluidos en los mensajes BGP. El comando route-map es una de las herramientas que permiten realizar dichas funcionalidades y se define de la siguiente manera:

```
route-map map-tag [permit | deny] [seq-number]
match: comando que especifica el criterio que debe ser comprobado
set: comando que indica la acción a ejecutar si el match aplica
```

donde “map-tag” es el nombre (label) que asignamos al map y “seq-number” indica la posición de la cláusula con respecto a otras cláusulas del mismo route-map (label), es decir:

```
route-map My-Map permit 10
  ! Primer conjunto de condiciones y acciones
Route-map My-Map permit 20
  ! Segundo conjunto de condiciones y acciones
```

Fijaros que es muy parecido a un if-else:

```
if condición then acción
  else if condición then acción
    else acción
```

ya que cada cláusula permit seq-number tiene una acción y condición y si no se cumple se pasa a la siguiente cláusula de forma secuencial. Si una de ellas se cumple, entonces se sale del if/elseif. Notar que las cláusulas del route-map están numeradas como 10, 20, 30, ... Hay dos motivos para numerar las cláusulas de esta manera: 1) Borrar fácilmente una cláusula sin afectar otra cláusula, 2) Insertar nuevas cláusulas entre dos cláusulas existentes.

Cada route-map tiene dos tipos de comandos:

- **Match**: selecciona rutas que la cláusula debe aplicar. Hay varias maneras de seleccionar las rutas. La más sencilla es usar Access Lists (ACLs). Si hay varios match en una cláusula, todas deben cumplirse (AND) para que se ejecute el comando set. Si no hay una cláusula match, entonces se ejecuta el set siempre (sobre todos los mensajes BGP recibidos o enviados). Algunas de las opciones para el comando match son las siguientes:

- **Match ip address** [address | acl-number]
- **Match metric** [metric]
- **Match as-path** [as-path-access-list]
- **Match community** [community]

Una de las opciones más usada es “match ip address acl-number”, que nos permite ejecutar la acción set a un conjunto de redes definido por un ACL.

- **Set**: modifica la información que será redistribuida en el protocolo objeto del route-map. Si hay varios comandos set, entonces se ejecutan todos si el match se cumple. Algunas de las opciones son las siguientes
- **Set localPref** [LocalPref]
 - **Set metric** [metric]
 - **Set as-path** [as-path]
 - **Set community** [community]

Notar que el objetivo del comando set es definir que atributos se manipulan por las condiciones definidas en el match.

Por ejemplo para modificar la tabla BGP cada vez que se recibe un mensaje BGP con la ruta 1.1.1.0/24 de forma que el next-hop sea 12.3.3.4 y el LocalPref sea 200 se haría:

```
route-map My-Map-1 permit 10
match ip address 1
set local-preference 200
set next-hop 12.3.3.4

access-list 1 permit 1.1.1.0 0.0.0.255
```

El comando **match** comprueba usando el ACL 1 que el mensaje BGP contiene la ruta 1.1.1.0/24 y con el comando **set** modificará la entrada en la tabla de encaminamiento con los valores establecidos en el script. Cuidado, en este script todavía no hemos dado la orden de que queremos modificar la tabla de encaminamiento BGP, es decir, todavía falta asignar el route-map e indicar la acción a realizar. El script solo indica que quiere realizar estas acciones, no sobre qué conexión BGP la tiene que realizar (ver más adelante como se asigna la acción). Las cláusulas pueden permitir/denegar y las ACLs también. Por consiguiente hay que conjugar las posibles combinaciones de la cláusula con el ACL.

- Si usas un ACL en una cláusula de route-map permit, las rutas que son permitidas por el ACL son redistribuidas.
- Si usas un ACL en una cláusula de route-map deny, las rutas que son permitidas por el ACL no son redistribuidas.
- Si usas un ACL en una cláusula de route-map permit or deny, y las rutas son denegadas por el ACL, entonces el comando map del route-map no se ejecuta y se evalúa la siguiente cláusula del route-map.

En conclusión la potencia del route-map con ACLs está básicamente en route-map permit y ACL permit/deny.

5.3. Ejemplo

El router R₂ recibe un mensaje BGP del AS34 en la que se le anuncia las redes 147.23.23.0/24 y 185.7.12.0/16. El atributo LocalPref tiene su valor defecto (por ejemplo 100). Queremos que cuando recibimos este mensaje BGP, el router R₂ actualice su tabla BGP y cambie el valor por defecto del LocalPref por el valor 200 para la red 147.23.23.0/24 y por el valor 240 para la red 185.7.12.0/16. Por otra parte, queremos reenviar a un router vecino perteneciente al AS68 con dirección IP=2.2.2.2, la ruta 185.7.12.0/16 via E-BGP añadiendo al mensaje 185.7.12.0/16 el atributo MED=75. La ruta 147.23.23.0/24 no anuncia ningún MED.

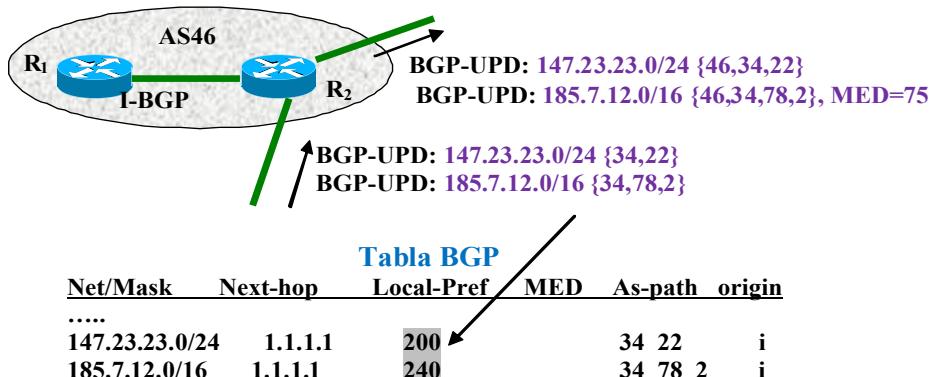


Figura 17: Ejemplo.

En el router R₂ configuraremos:

```
R2(config)# router bgp 46
R2(config-router)# neighbor 1.1.1.1 remote-as 34
R2(config-router)# neighbor 2.2.2.2 remote-as 68
R2(config-router)# neighbor 1.1.1.1 route-map My-Map-1 in
R2(config-router)# neighbor 2.2.2.2 route-map My-Map-2 out

R2(config)# route-map My-Map-1 permit 10
R2(config)# match ip address 1
R2(config)# set local-preference 200

R2(config)# route-map My-Map-1 permit 20
R2(config)# match ip address 2
R2(config)# set local-preference 240

R2(config)# route-map My-Map-2 permit 10
R2(config)# match ip address 2
R2(config)# set metric 75

R2(config)# access-list 1 permit 147.23.23.0 0.0.0.255
R2(config)# access-list 2 permit 185.7.12.0 0.0.0.255.255
```

Notar que para asignar el route-map se usa el comando “**neighbor IP@ route-map MapName [in|out]**” en dicha asignación, indicamos la dirección IP del router vecino BGP, el nombre del route-map que queremos que actúe y la dirección en la que queremos que actúe.

La dirección **in** es para mensajes BGP que entran en el router. Su efecto es que las condiciones-acciones del route-map se incluirán en la Tabla BGP. Es decir, si se cumple la condición match, el set actuará modificando la entrada de la tabla BGP sobre la/s ruta/s de la condición.

La dirección **out** es para mensajes BGP que salen del router. Su efecto es que las condiciones-acciones del route-map se incluirán en el mensaje BGP que sale. Es decir, si se cumple la condición match, el set actuará modificando los atributos anunciados en el mensaje BGP sobre la/s ruta/s de la condición.

5.4. Eliminación de route-map

Un route-map siempre tiene dos pasos: 1) Crear el route-map, 2) Aplicar el route-map a una sesión BGP de un router.

Si se quiere anular el efecto de un route-map en un router, basta entonces eliminar su aplicación a la sesión BGP. Usando el mismo ejemplo de la Figura 17, basta poner el mismo comando pero con el no delante:

```
R2(config-router)# no neighbor 2.2.2.2 route-map My-Map-2 out
```

Si se quiere eliminar del todo un route-map, entonces, usando el mismo ejemplo de la Figura 17, hay que poner el no delante del comando route-map. Si se quiere eliminar solo una línea, hay que añadir **permit #linea**.

```
R3(config)# no route-map Peer-R2
```

Si se quiere eliminar un access-list, usando el mismo ejemplo de la Figura 17, basta poner el no delante:

```
R3(config)# no access-list 1
```

5.5. Realización de la práctica

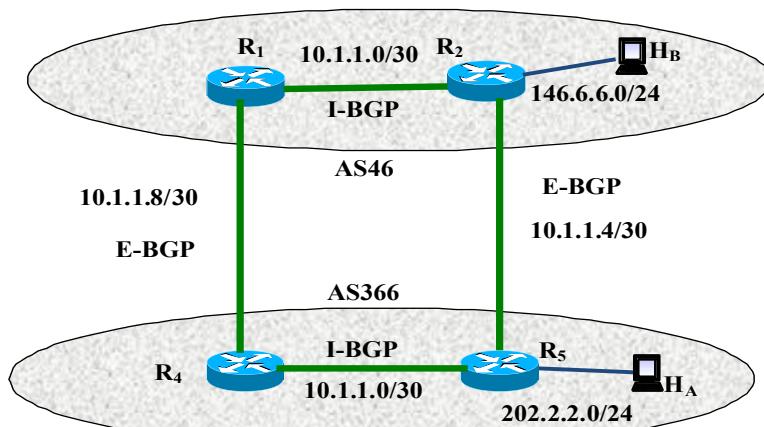


Figura 18: Red de la práctica 4.

5.5.1. Primera parte

Configurar la red de la Figura 18 siguiendo los pasos que se indican a continuación (es importante respetar el orden indicado):

1. Configurar las direcciones IP, encaminamiento interno (OSPF) y externo (BGP) de la red que aparece en la siguiente figura de forma que funcione correctamente.
2. Verificar las entradas en las tablas de encaminamiento BGP y comprobar que cada router de un dominio tiene dos rutas posibles para la red pública del otro dominio. Comprobar que se elige correctamente la mejor ruta entre las dos según los criterios BGP.
3. Haz un ping desde el host H_A al host H_B y confirma que el camino que han seguido los paquetes son:
 Echo request: H_A → R₅ → R₂ → H_B
 Echo reply: H_B → R₂ → R₅ → H_A
4. Nos situamos en el AS46 (no podeis reprogramar manualmente el router del AS366). Escribe los scripts necesarios para que el ping desde el host H_A al host H_B siga el siguiente camino:
 Echo request: H_A → R₅ → R₄ → R₁ → R₂ → H_B
 Echo reply: H_B → R₂ → R₁ → R₄ → R₅ → H_A
5. Comprobar las tablas de encaminamiento de R₁ y R₄ y razonar porque ahora solo hay dos entradas.

Lab. 6 - BGPv4 manipulación de atributos con comunidades

6.1. Objectivo de la práctica

Aprender a realizar políticas activas más complejas que las realizadas en la práctica anterior. Para ello se usará el atributo “community” como herramienta básica para realizar dichas políticas.

6.2. Uso de comunidades (community)

Otro manera para realizar políticas de encaminamiento consiste en el uso del atributo “community”. Para definir comunidades, usaremos las mismas herramientas que en la sección anterior: route-maps y ACLs. La idea es muy sencilla: el AS que define la comunidad tiene que exportar rutas con la comunidad definida. Por tanto tiene que crear un route-map que asignará a un vecino BGP. En dicho route-map tiene que filtrar (ACL) que rutas van a exportar esa comunidad. Por otro lado, el que reciba el update, tiene que crear un route-map para detectar la comunidad y fijar la acción. Veamos un ejemplo. Supongamos que el AS78 ha acordado con el AS46 que el router R2 fije un Local Pref = 175 en el enlace R3-R2 cuando reciba la comunidad 78:500.

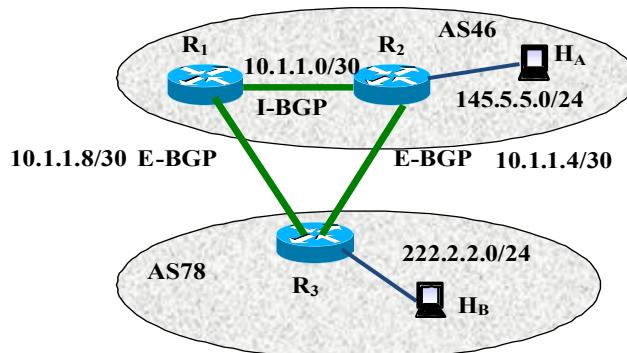


Figura 19: Ejemplo de uso de comunidades.

En el ejemplo de la Figura 18, la configuración del router R3 sería la siguiente.

```
R3(config)# router bgp 78
R3(config-router)# neighbor 10.1.1.9 remote-as 46
R3(config-router)# neighbor 10.1.1.5 remote-as 46
R3(config-router)# neighbor 10.1.1.5 send-community
R3(config-router)# neighbor 10.1.1.5 route-map Peer-R2 out

R3(config)# route-map Peer-R2 permit 10
R3(config)# match ip address 1
R3(config)# set community 78:500

R3(config)# access-list 1 permit 222.2.2.0 0.0.0.255
```

En la primera parte del script, establecemos las conexiones BGP con los routers del AS46. Además permitimos el envío de comunidades al router R2 (@IP = 10.1.1.5). Finalmente, asignamos el route-map Peer-R2 con la opción out (si se cumple el route-map entonces el mensaje BGP incluirá el atributo fijado en dicho route-map). El route-map Peer-R2 indica que si se detecta la red 222.2.2.0/24 se incluya la comunidad 78:500 en el mensaje BGP de salida. Ahora, tenemos que configurar el router R2 para que ejecute las acciones que queramos realizar cuando detecte la comunidad 78:500.

```
R2(config)# router bgp 46
R2(config-router)# neighbor 10.1.1.1 remote-as 46
R2(config-router)# neighbor 10.1.1.6 remote-as 78
R2(config-router)# neighbor 10.1.1.6 route-map Peer-R3 in

R2(config)# route-map Peer-R3 permit 10
R2(config)# match community 1
R2(config)# set local-preference 175

R2(config)# route-map Peer-R3 permit 20

R2(config)# ip community-list 1 permit 78:500
```

Una vez configuradas las conexiones BGP asignamos el route-map Peer-R3 de entrada (si detectamos un mensaje BGP de entrada que cumple el route-map modificamos la tabla BGP de acuerdo al route-map). El route-map Peer-R3 con permit 10 nos indica que si se cumple la condición indicada en el ACL ip community list se asigne un Local-Pref de 175. El route-map Peer-R3 con permit 20 es necesario para no descartar el resto de mensajes BGP que no cumplen el route-map.

6.3. Realización de la práctica

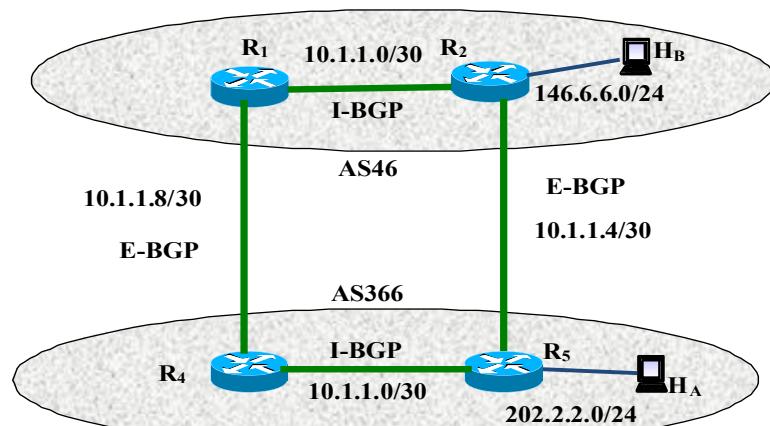


Figura 20: Red de la práctica 4.

6.3.1. Primera parte

Configurar la red de la Figura 18 siguiendo los pasos que se indican a continuación (es importante respetar el orden indicado):

1. Configurar las direcciones IP, encaminamiento interno (OSPF) y externo (BGP) de la red que aparece en la siguiente figura de forma que funcione correctamente.
2. Verificar las entradas en las tablas de encaminamiento BGP y comprobar que cada router de un dominio tiene dos rutas posibles para la red pública del otro dominio. Comprobar que se elige correctamente la mejor ruta entre las dos según los criterios BGP.
3. Haz un ping desde el host H_A al host H_B y confirma que el camino que han seguido los paquetes son:
 - Echo request: H_A → R₅ → R₂ → H_B
 - Echo reply: H_B → R₂ → R₅ → H_A
4. Definir una política de encaminamiento usando comunidades para que el ping desde el host H_A al host H_B siga el siguiente camino:
 - Echo request: H_A → R₅ → R₄ → R₁ → R₂ → H_B
 - Echo reply: H_B → R₂ → R₁ → R₄ → R₅ → H_A
5. Verificar las entradas en las tablas de encaminamiento BGP y comprobar que las rutas elegidas son las correctas.
6. Sustituir el prefijo de la red 202.2.2.0/24 con 147.7.7.0/24. Modificar oportunamente la configuración de IP, OSPF y BGP del AS366 manteniendo la selección de ruta del punto 2. ¿Se necesita modificar algo en el AS46?

Lab. 7 - BGPv4: Route Reflector y Confederation

7.1. Objetivo de la práctica

El objetivo de esta práctica es aprender a configurar AS sin el requisito de malla completa (full-mesh) para las sesiones iBGP. Para ello se aprenderán los dos métodos conocidos: router reflector y confederation.

7.2. Introducción a iBGP

Uno de los requisitos que se han visto es que se debe establecer una malla completa de sesiones iBGP entre routers BGP (Figura 21). En otras palabras, cada router BGP en un AS debe tener sesiones iBGP con todos los otros routers BGP del AS. Esto hace que no pueda haber ningún bucle en iBGP porque se exige que:

- toda la información que se envía por iBGP se aprende directamente desde el router que ha obtenido la información por eBGP
- toda información recibida por iBGP solo se puede reenviar por eBGP.

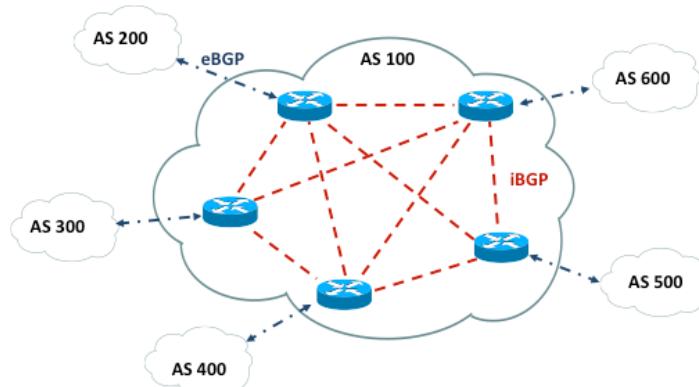


Figura 21: Malla completa de sesiones iBGP.

Si N es el número de routers BGP de un AS, el requisito de full-mesh hace que cada router deba mantener $N-1$ sesiones iBGP activas al mismo tiempo por un total de $N(N-1)/2$ sesiones iBGP para todo el AS. Existen dos métodos para relajar este requisito que son Route Reflector y Confederation que veremos a continuación.

7.3. Confederation

La implementación de BGP con confederación reduce la full-mesh de iBGP dentro de un AS. El truco consiste en dividir un AS en múltiples sub-ASes (confederación de AS). Cada sub-AS se comporta como un AS con internamente una full-mesh de iBGP pero solo algunas sesiones eBGP con el resto de sub-ASes del AS. Esta sesión eBGP realmente son interna al AS y se les suele llamar eiBGP (o también confederation BGP, cBGP). Esta configuración permite que atributos internos al AS como next hop, metric y local preference se mantengan para todos los sub-ASes. De cara al exterior, el AS se seguiría viendo como un único AS.

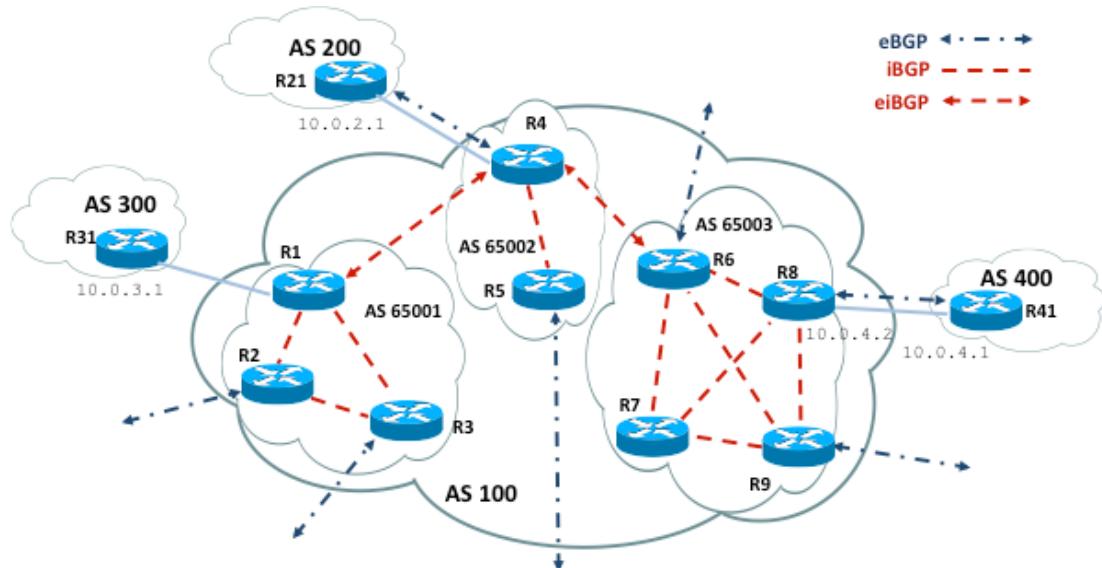


Figura 22: Ejemplo de confederación de AS.

Para configurar una confederación se necesitan dos comandos. El primero es “**bgp confederation identifier AS-number**” donde AS-number es su número de AS. El segundo es “**bgp confederation peers AS₁-number AS₂-number etc.**”, donde AS_n-number son los números que identifican los sub-ASes que hacen parte de la confederación. Generalmente para estos sub-ASes se usan número de AS del rango privado 64512-65535.

En el ejemplo de la Figura 22, la configuración del router R1 sería la siguiente. Se supone que las direcciones de loopback se asignan siguiendo el criterio 10.100.X.1/30 donde X es el número del router (no obstante, el comando **update-source dummy0** se omite para simplificar el ejemplo).

| | |
|--|--------------------------------------|
| R1(config)# router bgp 65001 | -> pertenece al sub-AS privado 65001 |
| R1(config-router)# bgp confederation identifier 100 | -> número AS real |
| R1(config-router)# bgp confederation peers 65002 | -> está conectado al sub-AS 65002 |
| R1(config-router)# neighbor 10.0.3.1 remote-as 300 | -> eBGP con el AS300 |
| R1(config-router)# neighbor 10.100.2.1 remote-as 65001 | -> iBGP del mismo sub-AS |
| R1(config-router)# neighbor 10.100.3.1 remote-as 65001 | -> iBGP del mismo sub-AS |
| R1(config-router)# neighbor 10.100.4.1 remote-as 65002 | -> eiBGP con otro sub-AS |

Para el router R4.

| | |
|--|---|
| R4(config)# router bgp 65002 | -> está conectado al sub-AS 65001 y 65003 |
| R4(config-router)# bgp confederation identifier 100 | |
| R4(config-router)# bgp confederation peers 65001 65003 | |
| R4(config-router)# neighbor 10.0.2.1 remote-as 200 | |
| R4(config-router)# neighbor 10.100.1.1 remote-as 65001 | |
| R4(config-router)# neighbor 10.100.5.1 remote-as 65002 | |
| R4(config-router)# neighbor 10.100.6.1 remote-as 65003 | |

Para el router R8.

| | |
|--|--|
| R8(config)# router bgp 65003 | |
| R8(config-router)# bgp confederation identifier 100 | |
| R8(config-router)# neighbor 10.0.4.1 remote-as 400 | |
| R8(config-router)# neighbor 10.100.6.1 remote-as 65003 | |
| R8(config-router)# neighbor 10.100.7.1 remote-as 65003 | |
| R8(config-router)# neighbor 10.100.9.1 remote-as 65003 | |

Y por ultimo el R418.

| | |
|---|--|
| R41(config)# router bgp 400 | -> un router externo ve el AS como 100 |
| R41(config-router)# neighbor 10.0.4.2 remote-as 100 | |

7.4. Route Reflection

Otra solución posible para relajar el requisito de full-mesh para iBGP es usar Route Reflector (RR). En este caso se divide el AS en clusters y para cada cluster se elige un router BGP que haga de router RR³. El resto de routers que no son RR se llaman clientes. Dentro de cada cluster se configuran sesiones iBGP entre clientes y RR, pero no entre clientes (se configura una topología a estrella de iBGP con RR como centro). Entre RR de cluster diferente se configuran sesiones iBGP a malla completa. La característica de los router RR es que estos pueden reenviar mensajes recibidos por iBGP a otros vecinos iBGP, mientras para los clientes sigue valida la regla que no pueden. En concreto el router RR sigue estas reglas al recibir un mensaje BGP:

- Si el mensaje BGP proviene de un vecino no cliente (por ejemplo otro RR), entonces el RR la refleja a todos sus clientes dentro de su cluster.
- Si el mensaje BGP proviene de un cliente, el RR lo refleja a todos los vecinos clientes y no clientes.
- Si el mensaje BGP se aprende de un vecino eBGP, éste se envía a todos los vecinos clientes y no clientes.

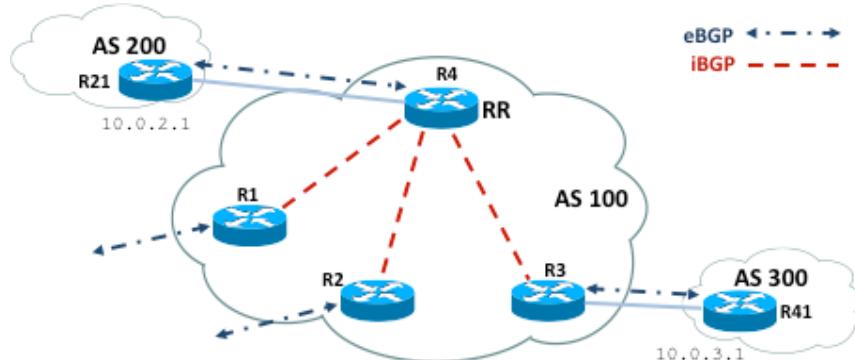


Figura 23: Ejemplo de un Route Reflector en un AS.

³ También se podrían configurar dos o más RR por cluster pero en esta práctica solo se considera el caso de un RR por cluster

En el ejemplo de la Figura 23, el router R4 es el único RR de este AS. Por lo tanto todos los demás router BGP son clientes y deben establecer una sesión iBGP con el RR. Cuando un cliente recibe un mensaje eBGP de otro AS, este lo reenvia solamente al RR. El RR a su vez, enviará este mensaje a los otros AS por eBGP como es habitual pero tambien a todos los demás clientes de su mismo AS (por eso se llama reflector). Como los clientes reciben el mensaje por iBGP, solo pueden reenviarlo por eBGP.

Para esta configuración se necesita añadir en la configuración de los vecinos iBGP del router elegido como RR el comando **“neighbor @IP-neighbor route-reflector-client”**.

En el ejemplo de la Figura 23, la configuración del router R4 sería la siguiente. Como en el caso anterior se supone que las direcciones de loopback se asignan siguiendo el criterio 10.100.X.1/30 donde X es el número del router (el comando **update-source dummy0** se omite para simplificar el ejemplo).

```
R4(config)# router bgp 100
R4(config-router)# neighbor 10.0.2.1 remote-as 200
R4(config-router)# neighbor 10.100.1.1 remote-as 100      -> iBGP con otro router
R4(config-router)# neighbor 10.100.1.1 route-reflector-client   -> se especifica que es un cliente
R4(config-router)# neighbor 10.100.2.1 remote-as 100
R4(config-router)# neighbor 10.100.2.1 route-reflector-client
R4(config-router)# neighbor 10.100.3.1 remote-as 100
R4(config-router)# neighbor 10.100.3.1 route-reflector-client
```

Y para el router R3.

```
R3(config)# router bgp 100
R3(config-router)# neighbor 10.0.3.1 remote-as 300
R3(config-router)# neighbor 10.100.4.1 remote-as 100      -> iBGP con el RR
```

En el ejemplo de la Figura 24 hay tres routers RR y por lo tanto se han definido tres clusters. Dentro de cada clusters, los clientes mantienen una sesión iBGP con su RR, mientras los RR mantienen una full-mesh entre ellos.

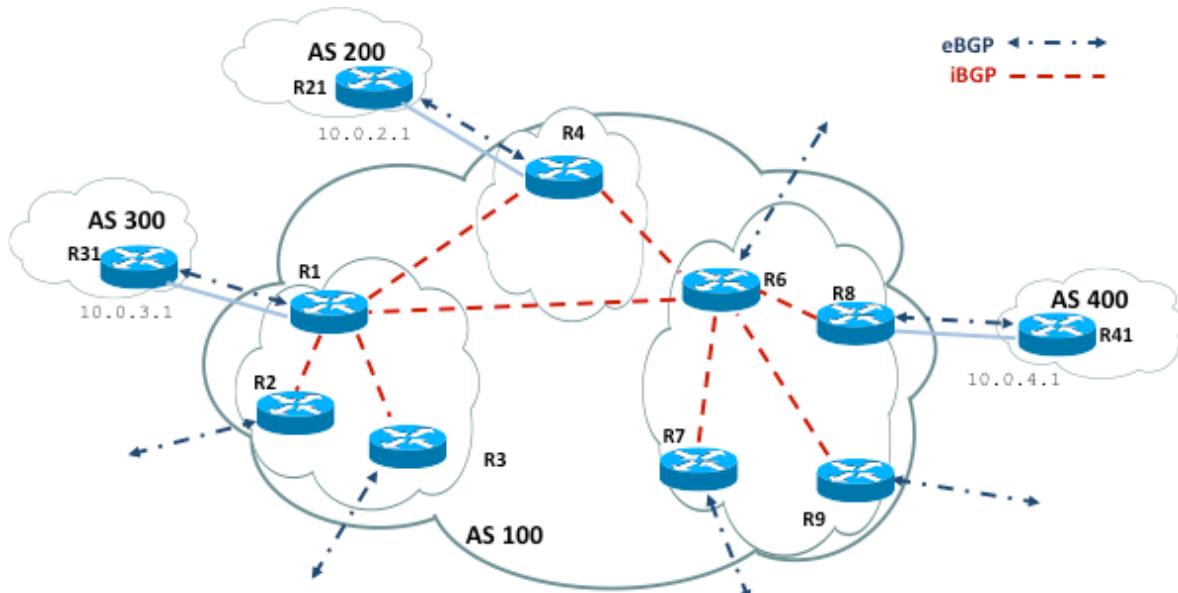


Figura 24: Ejemplo de tres Route Reflector en un AS.

En este caso la configuración del router R1 sería la siguiente.

```
R1(config)# router bgp 100
R1(config-router)# neighbor 10.0.3.1 remote-as 300
R1(config-router)# neighbor 10.100.2.1 remote-as 100
R1(config-router)# neighbor 10.100.2.1 route-reflector-client
R1(config-router)# neighbor 10.100.3.1 remote-as 100
R1(config-router)# neighbor 10.100.3.1 route-reflector-client
R1(config-router)# neighbor 10.100.4.1 remote-as 100      -> iBGP con otro RR
R1(config-router)# neighbor 10.100.6.1 remote-as 100      -> iBGP con otro RR
```

La del R4 sería la siguiente.

```
R4(config)# router bgp 100
R4(config-router)# neighbor 10.0.2.1 remote-as 200
R4(config-router)# neighbor 10.100.1.1 remote-as 100
R4(config-router)# neighbor 10.100.6.1 remote-as 100
```

Y finalmente para R8.

```
R8(config)# router bgp 100
R8(config-router)# neighbor 10.0.4.1 remote-as 400
R8(config-router)# neighbor 10.100.6.1 remote-as 100
```

Cabe destacar que para evitar bucles, BGP define dos nuevos atributos cuando se usa RR:

- **Originator-id:** Este es un atributo opcional. Un RR crea este atributo. Su función es guardar el identificador del router (RID) que originó la ruta. De este modo, si debido a una inadecuada configuración una ruta es anunciada a su router origen, dicha información será ignorada.
- **Cluster-list:** Atributo de una ruta en el que se van añadiendo los cluster-id del cluster al que pertenece cada RR por el que va pasando la ruta. Su objetivo y funcionamiento es parecido al caso de AS-path. Es decir es útil para evitar bucles en el caso de múltiples RR en el interior de un mismo cluster, ya que un RR puede detectar si su cluster-id se encuentra ya en la lista y evitar así un bucle ignorando la ruta.

7.5. Realización de la práctica

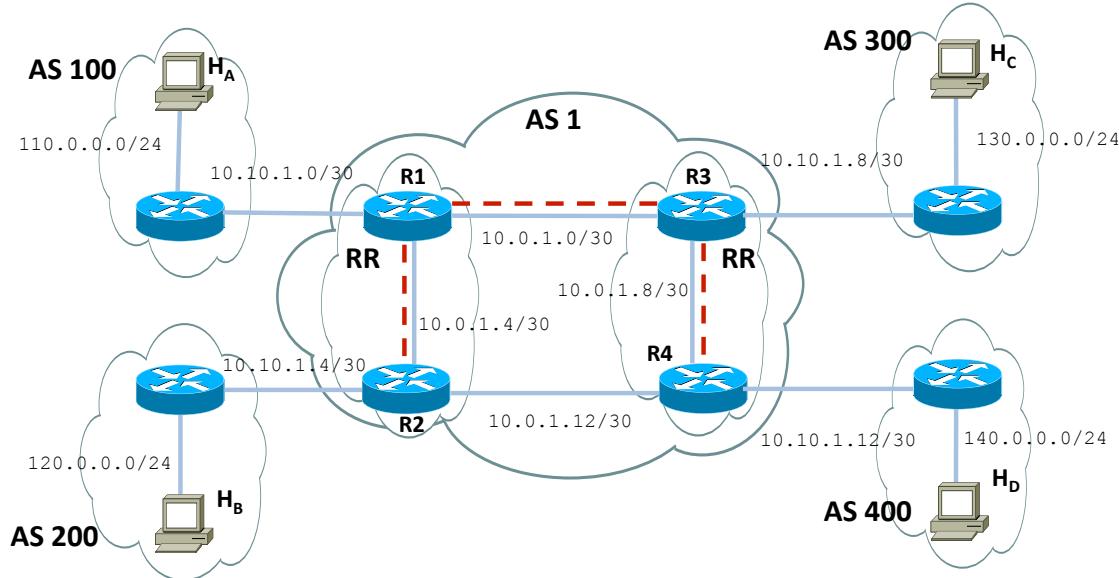


Figura 25: Red de la práctica.

Configurar la red de la Figura 25 siguiendo los pasos que se indican a continuación (es importante respetar el orden indicado):

1. Dividirse en 2 sub-grupos, cada uno con 6 PCs. Un grupo se ocupará de la configuración del AS 100, AS 200 y mitad del AS 1 y el otro grupo de la configuración del AS 300, AS 400 y la otra mitad del AS 1.
2. Configurar las direcciones IP en todas las interfaces de la red que aparece en la figura de forma que hay conectividad entre dos interfaces vecinas.
3. Configurar encaminamiento interno (OSPF) en los routers del AS 1. Comprobar que dos routers no vecinos del AS 1 pueden hacerse ping. Fijarse que no hace falta OSPF en los AS 100, 200, 300 y 400 ya que solo hay 1 router.
4. Activar el encaminamiento interno del BGP en AS 1, configurando R3 y R1 como routers RR para las sesiones iBGP. Configurar R2 y R4 como clientes. Comprobar que las sesiones están creadas.
5. Activar eBGP entre AS 1 y AS 100 y 200, lo mismo entre AS 1 y AS 300 y 400. Comprobar que todo funciona como esperado.
6. Verificar las entradas en las tablas de encaminamiento BGP y comprobar que los hosts pueden hacerse ping entre ellos.

Anexo A – Comandos para OSPF y BGP

A.1. OSPF commands

A.1.1. OSPF router

To start OSPF process you have to specify the OSPF router. Enable or disable the OSPF process.

router ospf ProcessNumber

To define a RID for OSPF.

```
ospf router-id A.B.C.D
passive interface #iFace
timers spf <0-4294967295> <0-4294967295>
refresh group-limit <0-10000>
refresh per-slice <0-10000>
refresh age-diff <0-10000>
auto-cost reference-bandwidth <1-4294967>
network A.B.C.D netmask area A.B.C.D
network A.B.C.D netmask area <0-4294967295>
```

This command specifies the OSPF enabled interface. If the interface has an address of 10.0.0.1/8 then the command below provides network information to the ospf routers

```
router ospf
network 10.0.0.0 netmask area 0
```

the network command's mask length should be the same as the interface address's mask.

A.1.2. OSPF interface

ip ospf cost <1-65535>

Set link cost for the specified interface. The cost value is set to router-LSA's metric field and used for SPF calculation.

ip ospf dead-interval <1-65535>

Set number of seconds for RouterDeadInterval timer value used for Wait Timer and Inactivity Timer. This value must be the same for all routers attached to a common network. The default value is 40 seconds.

ip ospf hello-interval <1-65535>

Set number of seconds for HelloInterval timer value. Setting this value, Hello packet will be sent every timer value seconds on the specified interface. This value must be the same for all routers attached to a common network. The default value is 10 seconds.

ip ospf network (broadcast|non-broadcast|point-to-multipoint|point-to-point)

Set explicitly network type for specified interface.

ip ospf priority <0-255>

Set RouterPriority integer value. Setting higher value, router will be more eligible to become Designated Router. Setting the value to 0, router is no longer eligible to Designated Router. The default value is 1.

ip ospf retransmit-interval <1-65535>

Set number of seconds for RxmtInterval timer value. This value is used when retransmitting Database Description and Link State Request packets. The default value is 5 seconds.

ip ospf transmit-delay

Set number of seconds for InfTransDelay value. LSAs' age should be incremented by this value when transmitting. The default value is 1 seconds.

A.1.3. Redistribute routes to OSPF

default-information originate

default-information originate metric <0-16777214>

default-information originate metric <0-16777214> metric-type (1|2)

default-information originate metric <0-16777214> metric-type (1|2) route-map word

default-information originate always

default-information originate always metric <0-16777214>

```
default-information originate always metric <0-16777214> metric-type (1|2)
no default-information originate
distribute-list name out (kernel|connected|static|rip|ospf)
default-metric <0-16777214>
distance <1-255>
distance ospf (intra-area|inter-area|external) <1-255>
```

A.1.4. Showing OSPF information

```
show ip ospf
show ip ospf interface [#iFace]
show ip ospf neighbor
show ip ospf neighbor #iFace
show ip ospf neighbor detail
show ip ospf neighbor #iFace detail
show ip ospf database
show ip ospf database (asbr-summary|external|network|router|summary)
show ip ospf database (asbr-summary|external|network|router|summary) link-state-id
show ip ospf database (asbr-summary|external|network|router|summary) link-state-id adv-router adv-router
show ip ospf database (asbr-summary|external|network|router|summary) adv-router adv-router
show ip ospf database (asbr-summary|external|network|router|summary) link-state-id self-originate
show ip ospf database (asbr-summary|external|network|router|summary) self-originate
show ip ospf database max-age
show ip ospf database self-originate
show ip ospf refresher
show ip ospf route
```

A.2. BGP commands

A.2.1 BGP router

First of all you must configure BGP router with router bgp command. To configure BGP router, you need AS number. AS number is an identification of autonomous system. BGP protocol uses the AS number for detecting whether the BGP connection is internal one or external one.

router bgp *asn*

Enable a BGP protocol process with the specified *asn*. After this statement you can input any BGP Commands. You cannot create different BGP process under different *asn* without specifying multiple-instance.

no router bgp *asn*

Destroy a BGP protocol process with the specified *asn*.

bgp router-id *A.B.C.D*

This command specifies the router-ID. In that case default router ID value is selected as the largest IP Address of the interfaces.

A.2.2. BGP route

network *A.B.C.D* mask *X.Y.Z.W*

This command adds the announcement network.

router bgp 1

network 10.0.0.0 255.0.0.0

This configuration example says that network 10.0.0.0/8 will be announced to all neighbors. Some vendors' routers don't advertise routes if they aren't present in their IGP routing tables; bgp doesn't care about IGP routes when announcing its routes.

A.2.3. Defining Peer

neighbor *peer* remote-as *asn*

Creates a new neighbor whose remote-as is *asn*. *peer* can be an IPv4 address or an IPv6 address.

router bgp 1

neighbor 10.0.0.1 remote-as 2

In this case my router, in AS-1, is trying to peer with AS-2 at 10.0.0.1.

This command must be the first command used when configuring a neighbor. If the remote-as is not specified, bgp will complain like this:

can't find neighbor 10.0.0.1

A.2.4. BGP Peer commands

In a router bgp clause there are neighbor specific configurations required.

neighbor *peer* shutdown

no neighbor *peer* shutdown

Shutdown the peer. We can delete the neighbor's configuration by no neighbor peer remote-as as-number but all configuration of the neighbor will be deleted. When you want to preserve the configuration, but want to drop the BGP peer, use this syntax.

neighbor *peer* ebgp-multipath

no neighbor *peer* ebgp-multipath

neighbor *peer* description ...

no neighbor *peer* description ...

Set description of the peer.

neighbor *peer* version *version*

Set up the neighbor's BGP version. *version* can be 4, 4+ or 4-. BGP version 4 is the default value used for BGP peering. BGP version 4+ means that the neighbor supports Multiprotocol Extensions for BGP-4. BGP version 4- is similar but the neighbor speaks the old Internet-Draft revision 00's Multiprotocol Extensions for BGP-4. Some routing software is still using this version.

neighbor *peer* interface *ifname*

no neighbor *peer* interface *ifname*

When you connect to a BGP peer over an IPv6 link-local address, you have to specify the ifname of the interface used for the connection.

neighbor peer next-hop-self
no neighbor peer next-hop-self

This command specifies an announced route's nexthop as being equivalent to the address of the bgp router.

neighbor peer update-source
no neighbor peer update-source
neighbor peer default-originate
no neighbor peer default-originate

bgp's default is to not announce the default route (0.0.0.0/0) even it is in routing table. When you want to announce default routes to the peer, use this command.

neighbor peer port port
neighbor peer send-community
neighbor peer weight weight

This command specifies a default weight value for the neighbor's routes.

neighbor peer maximum-prefix number

A.2.5. Peer filtering

neighbor peer distribute-list name [in|out]

This command specifies a distribute-list for the peer. *direct* is in or out.

neighbor peer prefix-list name [in|out]
neighbor peer filter-list name [in|out]
neighbor peer route-map name [in|out]

Apply a route-map on the neighbor. *direct* must be in or out.

A.2.6. Show IP BGP

show ip bgp
show ip bgp A.B.C.D
show ip bgp X:X::X:X

This command displays BGP routes. When no route is specified it display all of IPv4 BGP routes.

```
BGP table version is 0, local router ID is 10.1.1.1
  Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
  Origin codes: i - IGP, e - EGP, ? - incomplete
```

| Network | Next Hop | Metric | LocPrf | Weight | Path |
|--------------------------|----------|--------|--------|--------|------|
| *> 1.1.1.1/32 | 0.0.0.0 | 0 | | 32768 | i |
| Total number of prefixes | | | | | |

show ip bgp regexp line

This command display BGP routes using AS path regular expression.

show ip bgp community community
show ip bgp community community exact-match

This command display BGP routes using *community*.

show ip bgp community-list word
show ip bgp community-list word exact-match

This command display BGP routes using community list.

show ip bgp summary
show ip bgp neighbor [peer]
clear ip bgp peer

Clear peers which have addresses of X.X.X.X.

clear ip bgp peer soft in

Clear peer using soft reconfiguration.

show debug

```
debug event
debug update
debug keepalive
no debug event
no debug update
no debug keepalive
```

A.2.7. IP Access List

```
access-list name permit ipv4-network
access-list name deny ipv4-network
```

Basic filtering is done by access-list as shown in the following example.

```
access-list filter deny 10.0.0.0/9
access-list filter permit 10.0.0.0/8
```

A.2.8. Route Map

Route map is a very useful function. There is a match and set statement permitted in a route map.

```
route-map test permit 10
match ip address 10
set local-preference 200
```

This means that if a route matches ip access-list number 10 it's local-preference value is set to 200.

```
route-map route-map-name permit priority
match ip address access_list
```

Matches the specified *access_list*.

```
match ip next-hop ipv4_addr
```

Matches the specified *ipv4_addr*.

```
match aspath as_path
```

Matches the specified *as_path*.

```
match metric metric
```

Matches the specified *metric*.

```
match community community_list
```

Matches the specified *community_list*.

```
set ip next-hop ipv4_address
```

Set the BGP nexthop address.

```
set local-preference local_pref
```

Set the BGP local preference.

```
set weight weight
```

Set the route's weight.

```
set metric metric
```

Set the BGP attribute MED.

```
set as-path prepend as_path
```

Set the BGP AS path to prepend.

```
set community community
```

Set the BGP community attribute.

```
set ipv6 next-hop global ipv6_address
```

Set the BGP-4+ global IPv6 nexthop address.

```
set ipv6 next-hop local ipv6_address
```

Set the BGP-4+ link local IPv6 nexthop address.

Anexo B – Simulador GNS3

GNS3 es un simulador grafico de redes (<http://www.gns3.net/>) de código abierto y libre distribución que se puede utilizar en múltiples sistemas operativos, incluyendo Windows, Linux y MacOS X. En particular, permite crear redes a través de un entorno grafico usando dispositivos de red que emulan CISCO IOS y Juniper JunOS.

GNS3 es una excelente herramienta complementaria a los laboratorios de red. También se puede utilizar para experimentar o verificar las configuraciones de equipos de red antes de implementarlo más adelante en routers reales.

Actualmente su versión más estable es la 0.7.4. La versión “all-in-one” proporciona todas las herramientas útiles para el simulador excepto los sistemas operativos a simular como CISCO IOS, JunOS, etc. Estos sistemas operativos no son de libre distribución y el usuario de GNS3 debe proporcionar las imágenes de los sistemas operativos que necesita simular.

A continuación se ilustran algunos ejemplos.

La Figura 26 muestra un ejemplo de red creada a partir del entorno grafico de GNS3. El menu de selección de los dispositivos disponibles se encuentra en la ventana de la izquierda. Una vez arrastrados en la ventana central, estos dispositivos se pueden conectar entre si a través de enlaces que pueden usar tecnologías distintas, bien fastethernet, gigabitethernet, serial, etc.

La Figura 27 muestra como se puede acceder a la consola de un router. El router es realmente emulado, así que está corriendo realmente el sistema operativo del router y lo que aparece es la consola para configurarlo. Como las imágenes de los sistemas operativos son las reales, los comandos de configuración son exactamente iguales a los de un dispositivo real.

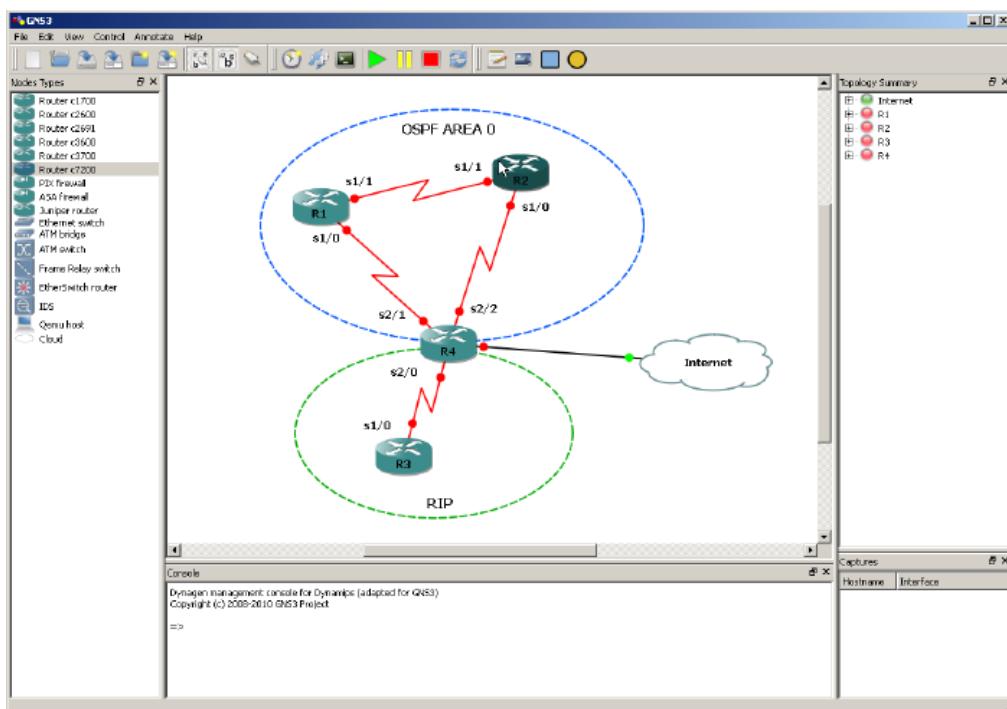


Figura 26: GNS3 permite crear redes a través de un entorno grafico.

Anexo B – Simulador GNS3

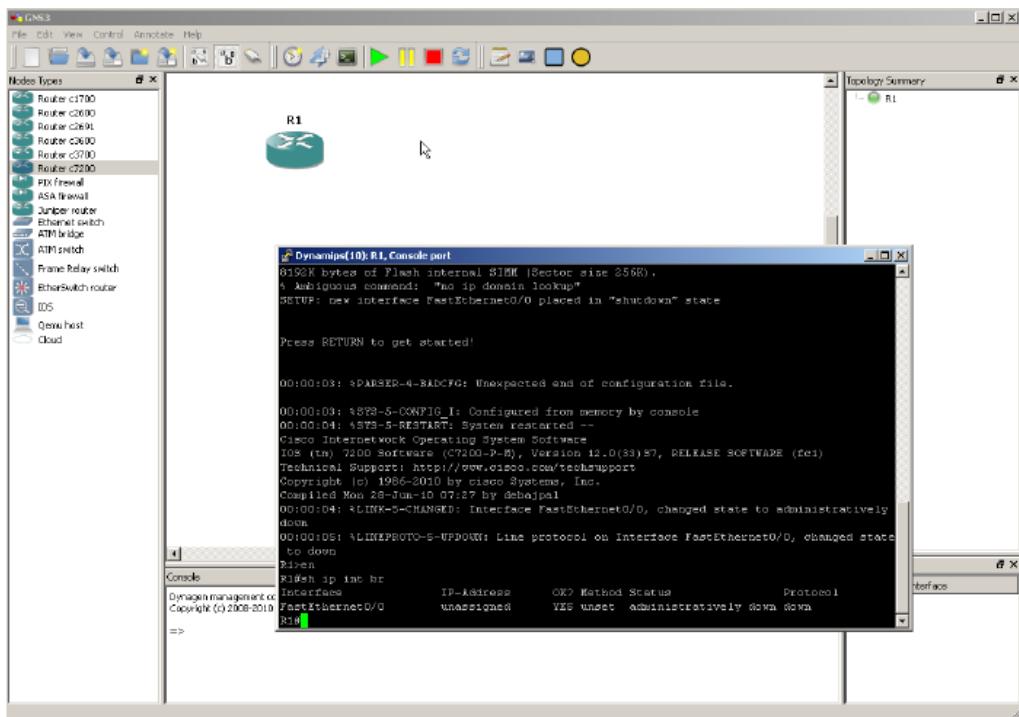


Figura 27: Acceso a la consola de configuración del router R1.