Facultat Informàtica de Barcelona FIB

Grau en Enginyeria Informàtica

Laboratori de Protocols d'Internet (PI-Grau)

José M. Barceló Ordinas, Llorenç Cerdà Alabern

Tardor de 2023

Índex

Entorn del laboratori (imatge xarxes)

En aquest capítol introductori 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 s'ha de seleccionar la imatge "xarxes". Aquesta imatge s'ha confeccionat a partir de la distribució de Linux de mida reduïda anomenada Alpine (http://alpinelinux.org).

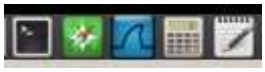
1. Informació bàsica

Usuari i password: xc / xc

Superusuari i password: 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:



Aquestes són, per ordre des de l'esquerra: consola, navegador web, wireshark, calculadora i editor.

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.

```
# udhcpc -i e0
```

2. 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 targetes ethernet. El sistema operatiu dóna els noms `eth0`, `eth1`, `eth2` a aquestes targetes. Hi ha el problema, però, que la posició física de la targeta amb el mateix nom pot canviar d'un PC a un altre. Perquè la posició de les targetes es correspongui amb la seva posició física en tots els PCs, les imatges fan servir la comanda ifrename/iftab en la fase de boot. Amb aquesta comanda s'anomenen les interfícies `eth0`, `eth1`, `eth2` amb els noms `e0`, `e1`, `e2`, de forma que quedin en les posicions que indica la Figura 1. Tenir en compte, doncs, que tot i que en alguns punts d'aquest manual es fan servir els noms per defecte (`eth0`, `eth1`, `eth2`), cal fer servir els noms `e0`, `e1`, `e2` segons la targeta que es faci servir (que podem identificar per la seva posició física en el PC).

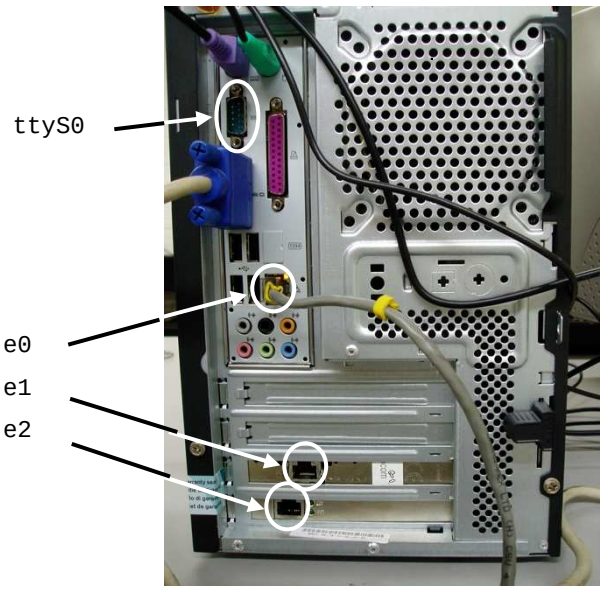


Figura 1: Ports de comunicació dels PCs del laboratori que farem servir en les pràctiques.

2.1. Identificació del nom de les interfícies ethernet

En alguns PCs del laboratori s'ha canviat alguna targeta ethernet que havia deixat de funcionar, i al botar el nom ja no es correspon amb e0, e1 i e2, tal com s'ha descrit anteriorment. A continuació s'explica un mètode senzill per determinar el nom de les interfícies, i quina és la seva ubicació física en el PC.

Primer cal determinar el nom que ha assignat Linux al botar. Per això basta executar "ifconfig –a", tal com es motra a continuació:

```
xc# ifconfig –a
eth3        Link encap:Ethernet  HWaddr 08:00:27:4E:4C:C7
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:10 Base address:0xd020

eth1        Link encap:Ethernet  HWaddr 08:00:27:BE:7D:7F
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:9 Base address:0xd240

eth4        Link encap:Ethernet  HWaddr 08:00:27:5A:83:86
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:11 Base address:0xd260
```

Del bolcat podem veure que, en aquest exemple, el nom de les interfícies és eth3, eth1 i eth4. Ara queda determinar quina és la posició física en el PC. Per fer-ho tindrem en compte que al cable que connecta el PC a la xarxa del laboratori arriben contínuament packets del switch on està connectat. Per tant, només hem de capturar paquets amb tcpdump, si n'arriba algun, vol dir que estem fent la captura en la interfície on hi ha connectat el cable de xarxa. Per exemple, per determinar si el cable de xarxa està connectat en eth3 executariem:

```
xc# ifconfig eth3 up
xc# tcpdump –ni eth3
20:00:14.229940 STP 802.1d, Config, Flags [none], bridge-id
833e.00:11:5c:05:f5:40.8004, length 43
20:00:14.516673 STP 802.1d, Config, Flags [none], bridge-id
8004.00:11:5c:05:f5:40.8004, length 43
^C
```

Del bolcat anterior podem verure que efectivament el cable està en eth3. A continuació desconnectaríem el cable, el connectaríem en una altra targeta, i repetiríem les commandes anteriors amb el nom d'una altra interfície, per exemple eth1. Si arriba tràfic, vol dir que la targeta on està el cable és eth1, i la que queda seria eth4

3. Zebra

GNU Zebra es un software de libre distribución que gestiona protocolos de encaminamiento TCP/IP (RIPv1v2, OSPFv2, BGPv4). El diseño de zebra está basado en modularidad y ejecuta un proceso para cada protocolo de encaminamiento (daemon). Cada daemon (**ripd, ospfd, bgpd, zebra**) tiene su propio fichero de configuración y su interfaz de configuración de terminal. Eso significa que cuando configuras direcciones IP, rutas estáticas, etc tienes que acceder al daemon "zebra", cuando configuras comandos de RIP tienes que acceder al daemon de "ripd", etc.

Zebra emula los comandos de CISCO para protocolos de encaminamiento, de forma que quitando algunas pequeñas diferencias, para configurar OSPF o RIP o BGP se ejecutan los mismos comandos que el IOS de CISCO. De esta forma, cada PC linux se convierte en un PC router con encaminamiento controlado con comandos similares al IOS. En el Anexo tenéis los comandos usados por Zebra y que usareis en el Lab.

El siguiente comando ejecuta los daemons de zebra, ripd, ospfd, bgpd:

xarxes% su

Passw: +root

xarxes-root% /etc/init.d/quaggua start

Para configurar direcciones IP, rutas estáticas, visualizar tablas de encaminamiento, etc, hay que abrir un "xterm" y ejecutar:

xarxes% telnet localhost zebra

password: zebra

zebra> enable

password: zebra

zebra# → ya estamos en modo privilegiado

zebra# configure terminal → podemos ejecutar cualquier comando IOS de este daemon

En el caso de que queramos usar OSPF y configurar comandos de OSPF hay que ejecutar el daemon correspondiente:

xarxes% telnet localhost ospfd

password: zebra

zebra> enable

zebra# → ya estamos en modo privilegiado

zebra# configure terminal → podemos ejecutar cualquier comando IOS de este daemon

Ídem para BGP. Zebra no se comporta "exactamente" igual a IOS CISCO. Las variaciones son mínimas. Ej. CISCO IOS usa "router ospf process-id" donde process-id es el identificador de proceso. Sirve para lanzar varios daemons OSPF. Sin embargo, zebra permite lanzar un solo proceso, con lo que el comando es "router ospf" sin usar process-id. Para ver las diferencias consultar el manual de zebra con los comandos OSPF. Al final del manual teneis un Apéndice con los principales comandos de configuración. Para más información, consultar la página Web de Zebra (http://www.zebra.org/ o www.quagga.net/).

Eines per repassar les pràctiques

Les pràctiques que es fan en les sessions presencials de laboratori es poden fer també a casa amb les eines que s'expliquen a continuació. És convenient fer-les també a casa si després de la sessió presencial de laboratori queden dubtes o no s'ha tingut temps d'acabar la pràctica.

1. Per a les pàctiques amb els PCs

Són les practiques d'OSPF i BGP.

En el següents enllaços podeu trobar 2 màquines virtuals (MV) creades des de VirtualBox (https://www.virtualbox.org) on hi ha instal·lat un Linux amb el mateix programari que la imatge que teniu en el laboratori. Hi ha 2 distribucions. La distribució Slitaz és més compacta i fa servir menys recursos. Es feia servir anteriorment per fer les pràctiques, però s'ha canviat per la distribució Alpine perquè es varen canviar els PCs i Slitaz no detectava les targetes ethernet.

Distribució Alpine: https://studies.ac.upc.edu/FIB/grau/XC/alpine-xarxes.ova

Distribució Slitaz: https://studies.ac.upc.edu/FIB/grau/XC/ slitaz50-xarxes.ova

Per importar-la des de VirtualBox:

Fitxer → Importar màquina virtual

Per a tenir múltiples VMs, clonar la imatge amb virtualbox tantes vegades com faci falta:

Seleccionar la imatge → clone → MARCAR L'OPCIÓ: "reinitialize the mac address of all network cards" → Linked clone

Podeu crear una xarxa de MVs i connectar-les per a repasar la pràctica del laboratori.

Veureu que la MV està configurada amb 4 targetes ethernet. Pequè dues MVs tinguin una targeta en la mateixa xarxa, cal anar a paràmetres → xarxa → nom, i posar el mateix nom en les dues MVs.

2. Per a les pràctiques d'STP

En el següent enllaç us podeu descarregar el simulador packettracer de CISCO. Només us heu de registrar per poder descarregar-vos el simulador sense cost.

https://www.netacad.com/about-networking-academy/packet-tracer/

El model del routers que hi ha en els racks és 1841, els commutadors són 2950. Podeu fer les practiques d'STP. Packettracer no implementa VRRP i VMPS (no podreu fer aquestes parts). Tampoc us servirá per fer les practiques de BGP, doncs packettracer només permet configurar connexions eBGP (no iBGP).

Lab 1. Spanning Tree Protocol (STP)

1. Objetivo de la pràctica

Aprender el funcionamiento de Spanning Tree Protocol (STP) y su configuración básica en switches cisco.

2. Introducción a STP

Objetivo: construir una topología libre de bucles (en árbol) con los caminos óptimos (de mínimo coste). Los puertos que no forman parte del árbol se bloquean. El coste de un puerto se calcula en base a la velocidad de transmisión:

Velocidad de transmisión	Coste
10 Gbps	2
1 Gbps	4
100Mbps	19
10 Mbps	100

Para construir el árbol STP se eligen (en este orden):

Un *Root Bridge* (RB) en todo el dominio broadcast. Nota: se usa indistintamente la palabra switch o bridge.Un *Root Port* en cada switch que no sea RB, que permita enviar tráfico hacia el RB. Garantiza que haya un árbol que una todos los switches.
Un *Designated Port* en cada dominio de colisiones (o segmento). Garantiza que todos los dominios de colisiones sean accesibles. El *switch* que posee el *designated port* se llama el *designated bridge* del segmento.

Los puertos que no se eligen como *Root Ports* o *Designated Ports*, quedan bloqueados (ver la Figura 2). Para la elección del *Root Bridge, Root Port, Designated Port*:

- Los switches se identifican con un *Bridge ID* (BID) formado por una prioridad (configurable manualmente) y una de las MACs del switch. El BID tiene 8 bytes: 2 de prioridad más 6 de la dirección MAC. A menor valor del BID, mayor prioridad.
- Se envían mensajes de señalización: *Bridge Protocol Data Unit* (BPDU).
- Inicialmente las BPDUs se envían cada 2 segundos a la dirección multicast: 01-80-C2-00-00-00. Los campos de las BPDUs usados en el cálculo del árbol son:
- *Root BID* (8 bytes)
- *Root Path Cost* (4 bytes): Se incrementa con el coste del puerto donde se recibe.
- *Sender BID* (8 bytes): BID del switch que envía la BPDU.
- *Port ID* (2 bytes): ID del puerto donde se transmite la BPDU (todos los puertos de un mismo switch han de tener IDs distintos). Análogamente al BID, el port ID está formado por una prioridad (byte más significativo) y un número de puerto (byte menos significativo).

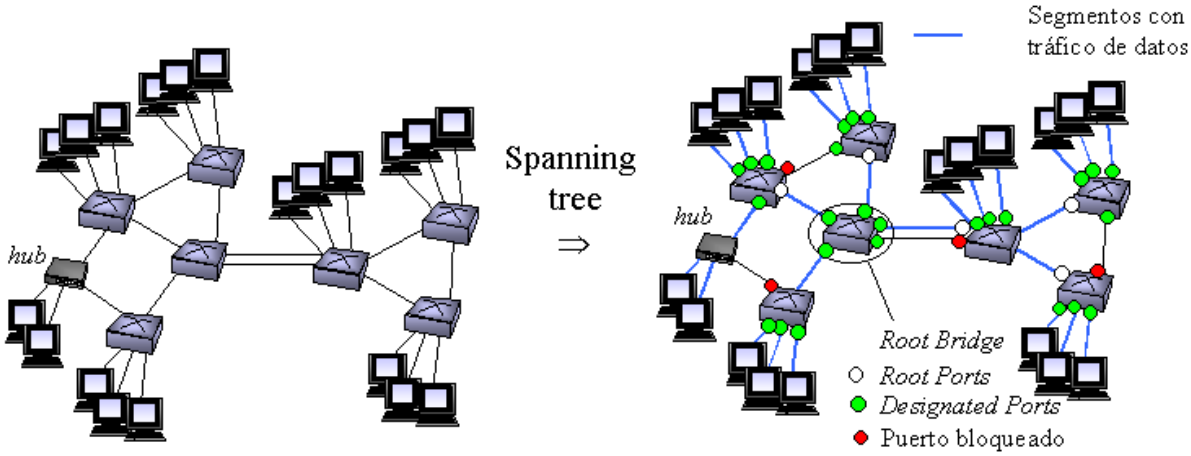


Figura 2: Formación de un árbol STP.

2.1. Elección del Root Bridge (RB):

- Inicialmente todos los switches suponen que son *Root Bridge*, todos sus puertos son *Designated Ports* y generan BPDUs con *Root BID = Sender BID*.
- Si un switch recibe una BPDU con un *Root BID* menor, deja de generar BPDUs y asume ese BID como *Root BID*. En poco tiempo sólo genera BPDUs el RB. Los demás switches modifican el *Root Path Cost*, *Sender BID* y *Port ID* antes de enviar la BPDU. El RB envía BPDUs por todos los puertos. Los demás switches sólo reenvían los BPDUs recibidos por el *Root Port*.
- Las prioridades del switch y de los puertos se pueden configurar manualmente. Inicialmente la prioridad del switch vale 0x8000 (32768), y de los puertos 0x80 (128). Hay que configurar la prioridad del switch más pequeña en el switch que se desee sea el RB.
- La elección del RB es importante: debe ser el switch más céntrico, para tener un árbol en forma de estrella (ver la Figura 3).

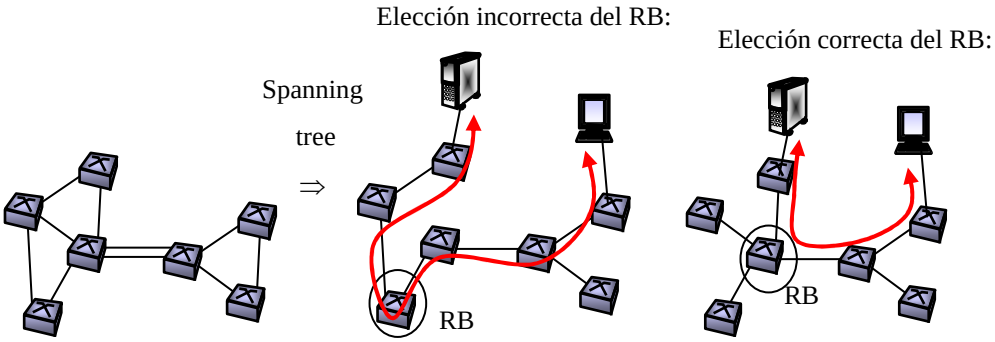


Figura 3: Elección del *Root Bridge*.

2.2. Elección del Root Port:

- Cada switch que no sea el RB selecciona un puerto como *Root Port*.
- Para la elección se compara la información contenida en las BPDUs recibidas en cada puerto. El puerto escogido es el que ha recibido una BPDU que cumple la primera de la siguiente secuencia de condiciones (ver la Figura 4):

    1. Menor *Root BID* (hacia el Root Bridge).
    2. Menor *Root Path Cost* (camino óptimo hacia el *Root Bridge*).
    3. Menor *Sender BID*
    4. Menor *Port ID*

2.3. Elección del Designated Port (acceso a un dominio de colisiones):

- Todos los puertos del RB son *Designated Ports*, exceptuando aquellos que formen un bucle de nivel 1 (por ejemplo, dos puertos conectados a un mismo hub, o entre ellos con un cable cruzado).

Para los demás switches:

- Los puertos por los que no se reciben BPSUs son *Desingated Ports*.
- Los puertos por los que se reciben BPDUs y no son *Root Ports*: se compara la información de las BPDUs recibidas y enviadas por ese puerto. El puerto es *Designated Port* si cumple la siguiente secuencia de condiciones (ver la Figura 4):
    1. Menor *Root BID* (hacia el *Root Bridge*).
    2. Menor *Root Path Cost* (camino óptimo hacia el *Root Bridge*).
    3. Menor *Sender BID*
    4. Menor *Port ID*

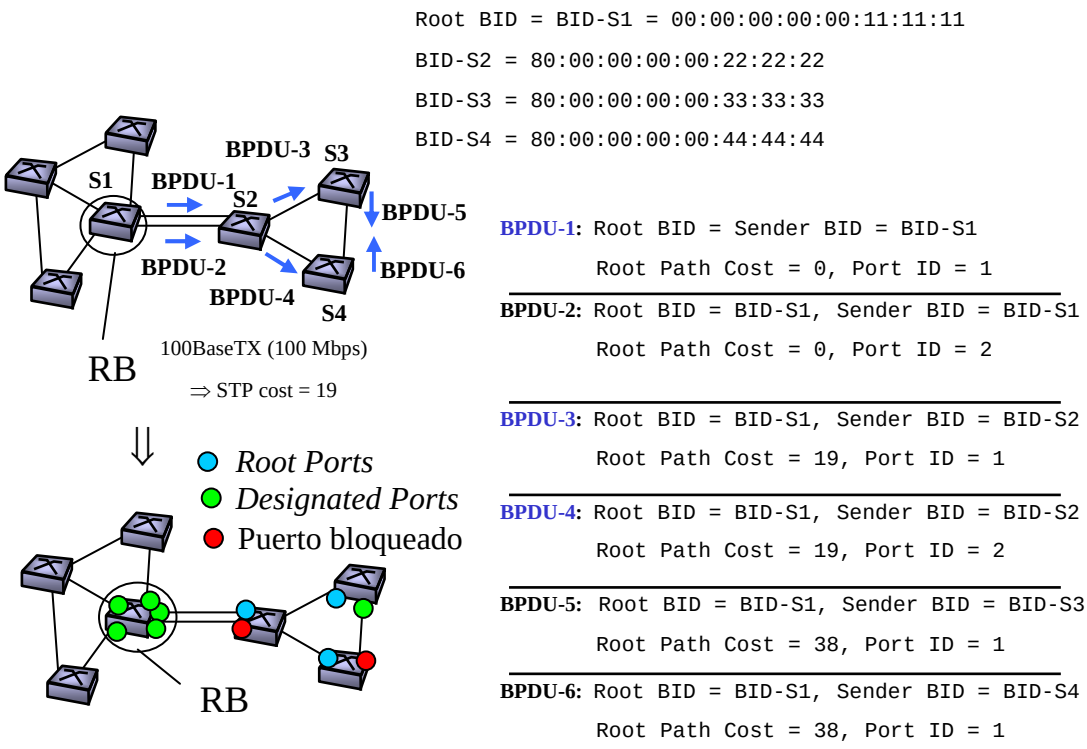


Figura 4: Ejemplo de elección del *Root port* y *Designated port*.

2.4. Estados de STP

Los puertos pueden estar en uno de los siguientes estados (ver la Figura 5):

- **Blocking** – No se hace *forwarding* de las tramas. Se escuchan BPDUs. Es el estado inicial.
- **Listening** - No se hace *forwarding* de las tramas. Se escuchan/transmiten BPDUs (construcción del árbol).
- **Learning** - No se hace *forwarding* de las tramas. Se aprenden direcciones. Se escuchan/transmiten BPDUs
- **Forwarding** - Se hace *forwarding* de las tramas y se aprenden direcciones. Se escuchan/transmiten BPDUs
- **Disabled** - No se hace *forwarding* de las tramas ni se escuchan/transmiten BPDUs. Puerto *shutdown*.

En los cambios de estado intervienen los siguientes temporizadores:

- *Hello*: tiempo entre BPDUs enviadas por un switch *Root Bridge* (2 segundos).
- *Forward:* tiempo en los estados de *listening/learning* (15 segundos).
- *Max Age*.: Tiempo máximo en que se guarda una BPDU (20 segundos). Si en ese tiempo no se reciben otras BPDUs, se pasa al estado de construcción del árbol STP (*listening*).

Transición entre estados:

1. Inicialización o *no shutdown*.
2. Puerto seleccionado como *Root* o *Designated*, o expira el *timer Max. Age* (20 segundos).
3. Expira el *timer forwarding* (15 segundos).
4. El puerto deja de ser *Root* o *Designated*. Inicialmente todos los switches suponen que son *Root Bridge* y todos sus puertos *Designated Ports*.
5. *Shutdown*

Específicos de CISCO:

6. *Port Fast*: Pensado para el caso de tener un host conectado directamente al switch. Para evitar retardos, el puerto se pone directamente en *forwarding* después de inicializar el puerto. Si el switch detecta un bucle, vuelve directamente a blocking.
7. *UplinkFast*: Pensado para *edge switches* (conmutadores de acceso). El switch tiene en cuenta los enlaces redundantes para sustituir rápidamente un *Root Port* en caso de fallo.

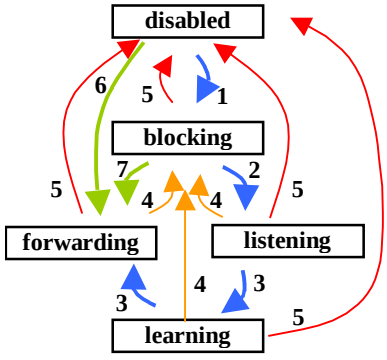


Figura 5: Diagrama de estados

Transiciones que ocurren al botar un switch:

1. Inicialmente supone que es *Root Bridge* y todos sus puertos *Designated Ports.*
2. Sus puertos pasan al estado de *listening*. En este estado el switch envía y escucha BPDUs, eligiéndose los *Root* y *Designated ports* (tal como se explica en las secciones 2.2 y 2.3).
3. Los puertos que dejan de ser *Designated ports* pasan al estado de *blocking*.
4. Después de estar 15 segundos en el estado de *listening*, el switch supone que la configuración de los puertos ha convergido, y pasa al estado de *learning*. En este estado empieza a inicializar la tabla MAC, pero continua 15 segundos más sin enviar tramas para asegurar la convergencia.
5. Después de estar 15 segundos en el estado de *learning*, los *Root* y *Designated ports* pasan al estado de *forwarding*: funcionamiento normal de puerto.

.
Transiciones que ocurren al cambiar la topología:

1. Un puerto bloqueado que no recibe BPDUs durante 20 segundos pasa al estado de *listening*.
2. Si el switch detecta que un Root Port pierde el link ethernet (se desconecta el puerto), también pondrá inmediatamente los puertos bloqueados en estado *listening*

Este comportamiento implica que la red estará entre 30 y 50 segundos en reaccionar a un cambio de topología.

2.5. Topology Change Notification

La entradas de la tabla MAC tienen un *max-age-timer* por defecto de 300 segundos (5 minutos). Es decir, si no se ve llegar una trama con dirección fuente igual a la de la tabla durante 300 segundos, la entrada se elimina. Esto implica que si se produce un cambio en la topología del STP, pueden hacer falta hasta 5 minutos hasta que los switches encaminen correctamente las tramas de las estaciones afectadas. Para resolver este problema, STP usa los mensajes llamados *Topology Change Notification BPDU* (TCN-BPDU).

El funcionamiento es el siguiente:

- Cuando se produce un cambio de topología, se envían TCN-BPDUs.
- Al recibir una TCN-BPDU, el switch cambia el *max-age-timer* de las entradas de tabla MAC al *Forward-timer* (15 segundos).

3. Configuración de un switch CISCO 2950

3.1. Configuración remota (telnet)

Para poder acceder al switch remotamente con una sesión telnet hace falta asignar una dirección IP a una VLAN y activar un password para acceder al *privileged Exec* y para los terminales vty:

```
Switch(config)#interface vlan 1
Switch(config-if)#ip add 192.168.10.2 255.255.255.0
Switch(config-if)#no shutdown
Switch(config-if)#exit
Switch(config)#enable password cisco
Switch(config)#line vty 0 4
Switch(config-line)#login
Switch(config-line)#password cisco
```

3.2. Configuración de VLANs en un switch Cisco 2950 (IOS)

Los switches 2950 solo disponen de VLAN basadas en el estándar IEEE802.1Q.

Las VLANs se identifican por un número (*vlan-id*) y un nombre (*vlan-name*). Si no se asigna ningún nombre, el nombre por defecto es VLAN*vlan-id.* Por defecto todos los puertos del *switch* tienen una configuración estática y pertenecen a la VLAN1.

- Para crear una VLAN:

```
Sw# configure term
Sw(config)# vlan VLAN-ID
Sw(config-vlan)# name NAME
Sw(config-vlan)# end
```

- Para añadir una interfaz a una VLAN:
  El puerto debe estar en modo *access*. Para cambiar el modo de un puerto:

```
Switch(config-if)# switchport mode {access | dynamic {auto | desirable} | trunk}
```

Descripción:

access	Set the port to access mode (either static-access or dynamic-access depending on the setting of the switchport access vlan inteface configuration command). The port is set to access unconditionally and operates as a nontrunking, single VLAN interface that transmits and receives nonencapsulated (non-tagged) frames. An access port can be assigned to only one VLAN.
dynamic auto	Set the interface trunking mode dynamic parameter to auto to specify that the interface convert the link to a trunk link.
dynamic desirable	Set the interface trunking mode dynamic parameter to desirable to specify that the interface actively attempt to convert the link to a trunk link.
Trunk	Set the port to trunk unconditionally. The port is a trunking VLAN Layer 2 interface. The port transmits and receives encapsulated (tagged) frames that identify the VLAN of origination. A trunk is a point-to-point link between two switches or between a switch and a router.

The default mode is **dynamic desirable**.

Para asignar la interfaz a la VLAN:

```
Switch(config-if)# switchport access vlan vlan-id
```

- Para configurar un puerto como *trunk*:
  Sólo puede ser 802.1Q.

```
Switch(config-if)# switchport mode trunk
```

- Para activar VTP en un dominio:

```
Switch(config)# vtp {domain domain-name | mode {client | server | transparent} | password
password}
```

- Para desactivar VTP:

```
Switch(config)# no vtp
```

- Verificación:

```
Switch# show vlan
Switch# show interfaces switchport
Switch# show vtp status
```

3.3. Configuración de STP en un switch Cisco 2950 (IOS)

- Por defecto STP está activado. Para activar/desactivar:

```
Switch(config)# spanning-tree vlan vlan-id
```

- Para configurar el coste y prioridad de un puerto:

```
Switch(config-if)# spanning-tree [vlan vlan-id] cost cost
Switch(config-if)# spanning-tree [vlan vlan-id] port-priority priority
```

- Para configurar STP de una vlan:

```
Switch(config)# spanning-tree vlan vlan-id {forward-time seconds | hello-time seconds | max-
age seconds | priority priority | {root {primary | secondary} [diameter net-diameter [hello-
time seconds]]}}
```

Descripción:

vlan-id	VLAN range associated with a spanning-tree instance. You can specify a single VLAN identified by VLAN ID number, a range of VLANs separated by a hyphen, or a series of VLANs separated by a comma. The range is 1 to 4094 when the enhanced software image (EI) is installed and 1 to 1005 when the standard software image (SI) is installed.
forward-time *seconds*	Set the forward-delay time for the specified spanning-tree instance. The forwarding time determines how long each of the listening and learning states last before the interface begins forwarding. The range is 4 to 30 seconds
hello-time *seconds*	Set the interval between hello bridge protocol data units (BPDUs) sent by the root switch configuration messages. The range is 1 to 10 seconds.
max-age *seconds*	Set the interval between messages the spanning tree receives from the root switch. If a switch does not receive a BPDU message from the root switch within this interval, it recomputes the spanning-tree topology. The range is 6 to 40 seconds.
priority *priority*	Set the switch priority for the specified spanning-tree instance. This setting affects the likelihood that the switch is selected as the root switch. A lower value increases the probability that the switch is selected as the root switch. The range is 0 to 61440 in increments of 4096. Valid priority values are 4096, 8192, 12288, 16384, 20480, 24576, 28672, 32768, 36864, 40960, 45056, 49152, 53248, 57344, and 61440. All other values are rejected.

root primary	Force this switch to be the root switch.
root secondary	Set this switch to be the root switch should the primary root switch fail.
diameter *net-diameter*	Set the maximum number of switches between any two end stations. The range is 2 to 7.

- Para habilitar la opción *uplink-fast* y *portfast*:

```
Switch(config)# spanning-tree uplinkfast
Switch(config)# spanning-tree portfast default
Switch(config-if)# spanning-tree portfast
```

- Verificación:

```
Switch# show spanning-tree active
Switch# show spanning-tree detail
Switch# show spanning-tree summary
Switch# show spanning-tree vlan vlan-id …
Switch# show spantree interface interface-id …
Switch# debug spanning-tree bpdu
```

Por ejemplo, para ver el estado de STP en los puertos con la VLAN8 activa:

```
Switch#sh spanning-tree vlan 8

VLAN0008
  Spanning tree enabled protocol ieee
  Root ID    Priority    32776
             Address     000a.b7e6.7e00
             Cost        19
             Port        25 (GigabitEthernet0/1)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32776  (priority 32768 sys-id-ext 8)
             Address     000f.f7d5.e040
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 300

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gi0/1            Root FWD 19        128.25   P2p
Gi0/2            Altn BLK 19        128.26   P2p
```

3.4. Configuración de Etherchannel

El switch 2950 permite agrupar puertos, incrementando así el ancho de banda e introduciendo redundancia. Los dos puertos deben pertenecer a la misma VLAN o ser trunk.

Para configurar un etherchannel primero se crea con el comando **interface port-channel** *port-channel-number* y después se asignan los puertos con el comando **channel-group**. Para la formación del etherchannel puede usarse el *Port Aggregation Protocol* (PAgP).

Ejemplo de configuración:

```
Switch(config)# interface port-channel 1
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# channel-group 1 mode on
```

Las opciones del comando **channel-group** son:

```
channel-group channel-group-number mode {auto|desirable|on}
on Force the interface to channel without PAgP
auto Enable PAgP only if a PAgP device is detected.
desirable Unconditionally enable PAgP.
```

Verificación:

```
Switch# show etherchannel [channel-group-number] {detail | load-balance | port | port-channel
| summary}
Switch# show pagp [channel-group-number]
```

4. Realitación de la práctica

4.1. Configuración de VLANs y VLAN de gestión

1. Configurar los puertos que muestra la Figura 6 con 2 VLANs (VLAN2 y VLAN3). Los enlaces entre switches están en **modo trunk** y los enlaces con los PCs donde hay el cable consola están en la VLAN2 y VLAN3 (ver el apartado 3.2 para la configuración de los puertos) en **modo access**. Al haber dos VLANS, se genera una instancia Spanning-tree por cada VLAN (MSTP).

2. Asignar una IP a cada switch en la VLAN2 y configurar telnet (apartado 3.1). Las IP's serán 10.0.0.x, con x=1, 2, 3 ó 4 si el conmutador está etiquetado como C1, C2, C3 ó C4. Iniciar una una sesión telnet desde los PCs con el cable consola con cada uno de los 4 switches, para poder ver su configuración.

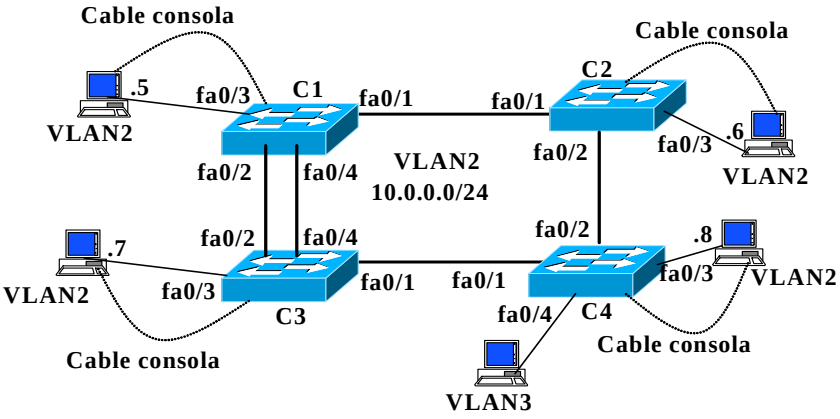


Figura 6

4.2. Ajustes de STP

1. Identificar con los comandos show (ejecutándolos en los 4 switches) quién es el RB (*root bridge*) de la VLAN2 y de la VLAN3 y qué puertos están bloqueados. El RB ¿es el mismo switch para las 2 VLANs? ¿Por qué?

2. Comprobar que enlace ha quedado bloqueado en cada VLAN. ¿Es el mismo enlace para las 2 VLANs? ¿Por qué? Hazte un esquema con la topología resultante para cada VLAN.

3. Cambiar las prioridades del spanning-tree de las VLANs para forzar que C4 sea el RB de las dos VLANs. Comprobar de nuevo que enlace ha quedado bloqueado en cada VLAN. ¿Es el mismo enlace para las 2 VLANs? ¿Por qué?

4. Cambiar las prioridades de spanning tree de las VLANs para que el enlace C1-C2 esté bloqueado para la VLAN3 y no para la VLAN2, y los dos enlaces C1-C3 estén bloqueados para la VLAN2 y no para la VLAN3. Para la VLAN3 habrá uno de los 2 enlaces C1-C3 bloquedo. Configurar los switches para que el puerto fa0/2 sea el qué se bloquee. De este modo VLAN2 y VLAN3 compartirían los enlaces C2-C4 y C3-C4, mientras que el enlace C1-C2 sólo lo usaría VLAN2, y C1-C3 sólo lo usaría VLAN3 (a través de los puertos fa0/4-fa0/4).

4.3. Configuración de etherchannels

1. Quitar la configuración en modo trunk i configurar los puertos cómo muestra la Figura 7 en la VLAN2. Es decir, agregar los puertos fa0/2 y fa0/4 para formar un Eherchannel.

2. Comprobar con los comandos show quién es el RB y qué puertos está bloqueado. Observa la etiqueta que se muestra para la agregación del puerto, así cómo el coste asignado al puerto agregado. Observa que no se ha bloqueado ningún puerto y que se comportan como un solo enlace STP.

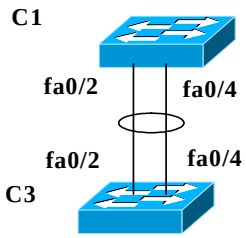


Figura 7

4.4. *Broadcast-storms* (opcional)

1. Configurar dos conmuntadores sólo con dos puertos en la VLAN2 tal cómo muestra la siguiente figura, pero sin etherchannel.

2. Comprobar con los comandos show quién es el RB y qué puerto está bloqueado.

3. Comprobar que el puerto bloqueado se activa si desconectamos el otro, y observar los estados por los qué pasa STP con los comandos show.

4. Configurar otro puerto en la misma VLAN, conectar un PC, desactivar STP y provocar un *broadcast storm*.

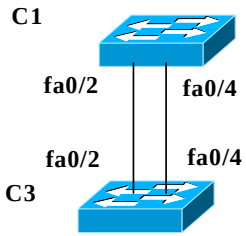


Figura 8

Lab 2. Virtual Router Redundancy Protocol (VRRP) y VLAN Membership Policy Server (VMPS)

1. Objetivo de la práctica

- Aprender el funcionamiento de redundancia de routers (VRRP) y configuración de VLANs dinámicas en switches cisco (VMPS).

2. Virtual Router Redundancy Protocol (VRRP)

VRRP es un estándar (RFC-2338) del protocolo HSRP propietario de CISCO. VRRP permite introducir un router de *backup* y hacer *load balancing*. Por ejemplo, en la Figura 9 todos los PCs tendrían el mismo router cómo router por defecto. Si este cayera, todos los PCs quedarían sin poder salir de su red.

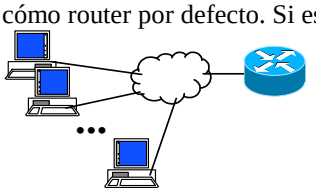


Figura 9

Figura 10

VRRP permite configurar dos o más routers (por ejemplo R1 y R2 en la Figura 10) de forma que una de sus interfaces comparta una dirección IP y una dirección MAC "virtual". La dirección virtual, que debe ser distinta a la de la interfaz del router, es vista por todos los PCs como si fuera la dirección de un único router. Los dos routers se envían mensajes de *advertisement* periódicamente entre ellos (por defecto, cada 1 segundo). Uno de ellos es el router *máster* y el otro *backup*. Si el *backup* deja de recibir *advertisement* del máster, pasa a ser el máster. Se pueden configurar más de una dirección IP virtual, de forma que un grupo de PCs tengan a R1 como *máster*, y R2 *backup* y viceversa, haciendo así balanceo de carga.

A la interfaz que van a compartir los routers hay que asignar dos (o más) direcciones: la dirección única del router y la dirección (o direcciones) "virtual". La MAC virtual es siempre 00-00-5E-00-01-VRID, donde VRID es el *Virtual Router Identifier*, es decir, un número que identifica el grupo de routers que comparten la interfaz.

3. Configuración de VRRP en un router CISCO

Una posible configuración para el ejemplo de la Figura 10 sería la siguiente:

```
Configuración del R1
1.  R1(config)# interface f0/0
2.  R1(config-if)# ip address 10.1.0.1 255.255.255.0
3.  R1(config-if)# vrrp 1 priority 200
4.  R1(config-if)# vrrp 1 authentication text pi
5.  R1(config-if)# vrrp 1 ip 10.1.0.10
6.  R1(config-if)# vrrp 2 priority 100
7.  R1(config-if)# vrrp 2 ip 10.1.0.20
8.  R1(config-if)# no shutdown

Configuración del R2
9.  R2(config)# interface f0/0
10. R2(config-if)# ip address 10.1.0.2 255.255.255.0
11. R2(config-if)# vrrp 1 priority 100
12. R2(config-if)# vrrp 1 authentication text pi
13. R2(config-if)# vrrp 1 ip 10.1.0.10
14. R2(config-if)# vrrp 2 priority 200
15. R2(config-if)# vrrp 2 ip 10.1.0.20
16. R1(config-if)# no shutdown
```

Con esta configuración R1 sería el máster del grupo 1 con IP virtual 10.1.0.10, y R2 sería el máster del grupo 2 con IP virtual 10.1.0.20. Como se puede observar para que R1 sea master del VRRP-1 primero se activa una prioridad entre 1 y 254 y luego se le asigna una IP virtual. Al ser la prioridad 200 del grupo VRRP 1 más alta en R1 que la prioridad 100 del grupo VRRP 1 del router R2, el R1 será master del grupo VRRP 1 con IP virtual 10.1.0.10 mientras que el R2 será backup. Se puede razonar de la misma manera con el grupo VRRP 2 pero esta vez será R2 el master y R1 el backup.

La autenticación es opcional y sirve para que todos los routers de un grupo envíen mensajes autenticado con el mismo string. Por lo tanto, cuando se configura un grupo, ej. VRRP-1, tienen que tener el mismo string como parámetro de entrada (pi en el ejemplo anterior).

En caso de que **la interfaz del router con el switch sea un trunk**, vrrp debe configurarse en cada sub-interfaz que se desee. Por ejemplo, para **configurar la sub-interfaz correspondiente a una VLAN** con número de VLAN 1 los comandos serían:

```
1.  R1(config)# interface f0/0
2.  R1(config-if)# no shutdown
3.  R1(config-if)# interface f0/0.1
4.  R1(config-if)# encapsulation dot1q 1
5.  R1(config-if)# ip address 10.1.0.1 255.255.255.0
6.  R1(config-if)# vrrp 1 ...
```

Verificación:

```
1.  Router# show vrrp [brief]
```

4. VLAN Membership Policy Server (VMPS)

La configuración de VLANs dinámicas en switches CISCO se hace a través de VMPS. VMPS usa el paradigma cliente-servidor con UDP (puerto por defecto del servidor: 1589).

Inicialmente los clientes configuran los puertos con una VLAN dinámica. Con esta configuración, no se asigna ninguna VLAN al puerto. Cuando el switch recibe la primera trama de un host conectado a un puerto con una VLAN dinámica, el switch se conecta al servidor para descubrir la VLAN a la que pertenece el host. Para ello VMPS hace un mapeo dirección-MAC ⇒ VLAN y devuelve la VLAN a la que pertenece el host. Si el host se desconecta del puerto y se conecta a otro distinto, VMPS reconfigura dinámicamente las VLANs de los puertos.

La mayoría de switches de CISCO pueden configurar sus puertos con VLANs dinámicas. Sin embargo, sólo los switches de gama alta (por ejemplo, series Catalyst 5000/6000) implementan un servidor VMPS. Para hacer esta práctica usaremos una implementación de libre distribución (OpenVMPS) que puede descargarse de:

`http://sourceforge.net/projects/vmps`

OpenVMPS crea un daemon que se puede ejecutar desde un PC con linux. OpenVMPS lee un fichero de configuración que tiene el mismo formato que el fichero de configuración VMPS de CISCO. En la siguiente URL hay una descripción de VMPS de CISCO:

`http://www.cisco.com/en/US/docs/switches/lan/catalyst5000/catos/5.x/configuration/guide/vmps.html`

Si el fichero de configuración es vlan.db, para iniciar el daemon OpenVMPS hay que ejecutar simplemente:

```
linux# ./vmpsd –f vlan.db
```

Para reiniciar el daemon (hace falta si deseamos cambiar el fichero de configuración) ejecutar *killall vmpsd*, y volver a ejecutar el comando anterior.

4.1. Fichero de configuración de VMPS

La Figura 11 muestra un ejemplo de configuración de VMPS distribuido con OpenVMPS (se ha añadido un número de línea). Las líneas que empiezan con el carácter '!' son comentarios. A continuación hay una breve descripción:

- Hay que definir un *VMPS domain* (línea 6), este dominio debe coincidir con el dominio VTP del switch. VTP (Virtual Trunking Protocol) es un protocolo propietario de CISCO que permite propagar la configuración de las VLANs a todos los switches de un mismo dominio. Por ejemplo, al crear una VLAN en un servidor VTP, la VLAN se propaga a todo el dominio.
- VMPS puede operar en modo *open* o *secure* (línea 7). En modo *open*: Si una MAC no está definida, se le asigna una VLAN por defecto (línea 8). En modo *secure*: Si una MAC no está definida se bloquea el puerto. Para desbloquear un puerto hay que ejecutar los comandos *shutdown / no shutdown*.
- En la sección *MAC Addresses* (línea 11) se asignan las VLANs a las que pertenecen las direcciones MAC. Puede usarse `--NONE--` para denegar explícitamente el acceso a cualquier VLAN. Notar que para identificar las VLANs se usa el VLAN-name, no el VLAN-id. En el switch deben haberse creado las VLANs con el mismo nombre que el indicado en esta sección del fichero de configuración.
- Una VLAN se puede restringir a un switch específico, o a un grupo de puertos de un switch. Para ello hay que especificar:
  1. Los puertos permitidos (sección *Port Groups*, línea 24). Por ejemplo, la línea 31 especifica el puerto 2/4 del switch 10.0.0.1, y la línea 32 especifica todos los puertos del switch 10.0.0.2.
  2. Las VLANs a las que se les aplicará alguna restricción (sección *VLAN groups*, la línea 34).
  3. La asociación entre las definiciones anteriores (sección *VLAN port Policies*, línea 43).

```
1.    !vmps domain <domain-name> - The VMPS domain must be defined.
2.    !vmps mode { open | secure } - The default mode is open.
3.    !vmps fallback <vlan-name>
4.    !vmps no-domain-req { allow | deny } - The default value is allow.
5.    !
6.    vmps domain mydomain
7.    vmps mode open
8.    vmps fallback --NONE--
9.    vmps no-domain-req deny
10.   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
11.   !MAC Addresses
12.   ! address <addr> vlan-name <vlan_name>
13.   !
14.   vmps-mac-addrs
15.   !
16.   address 0010.a49f.30e1 vlan-name --DEFAULT--
17.   ! disabled - no access
18.   address 0010.a49f.30e2 vlan-name --NONE--
19.   ! vlan TEST restricted
20.   address 0010.a49f.30e3 vlan-name TEST
21.   ! vlan TEST1 unrestricted
22.   address 0010.a49f.30e4 vlan-name TEST1
23.   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
24.   !Port Groups
25.   !vmps-port-group <group-name>
26.   ! default-vlan <vlan-name>
27.   ! fallback-vlan <vlan-name>
28.   ! device <device-id> { port <port-name> | all-ports }
29.   !
30.   vmps-port-group myswitch
31.   device 10.0.0.1 port 2/4
32.   device 10.0.0.2 all-ports
33.   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
34.   !VLAN groups
35.   !vmps-vlan-group <group-name>
36.   ! vlan-name <vlan-name>
37.   !
38.   vmps-vlan-group myvlans
39.   vlan-name TEST
40.   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
41.   !VLAN port Policies
42.   !vmps-port-policies {vlan-name <vlan_name> | vlan-group <group-name> }
43.   ! { port-group <group-name> | device <device-id> port <port-name> }
44.   !
45.   vmps-port-policies vlan-group myvlans
46.   port-group myswitch
```

Figura 11: Ejemplo de configuración VMPS.

4.2. Configuración de los clientes de VMPS en un switch 2950 (IOS)

Debe configurarse:

1. La dirección IP del switch.
2. El dominio VTP.
3. La dirección del servidor VMPS.
4. Configurar las interfaces con VLANs dinámicas.

- Para asignar la dirección IP del switch:
  Hay que entrar en modo de configuración de la subinterface de la VLAN a la que queremos asignar la dirección IP:

```
Switch(config)# interface vlan vlan-id
Switch(config-if)# ip address ip-add mask
```

- Para definir el dominio VTP:

```
Switch(config)# vtp domain domain-name
```

- Para asignar la dirección IP del VMPS:

```
Switch(config)# vmps server ip-addr
```

- Para reconfirmar con el VMPS la configuración de las VLANs:

```
Switch# vmps reconfirm
```

- Para configurar un puerto con VLAN dinámica:

```
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan dynamic
```

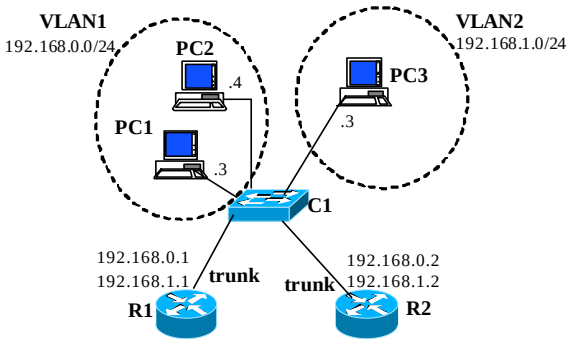
- Verificación:

```
Switch# show vlan
Switch# show vmps
```

5. Realización de la práctica - Configuración de VRRP

1. Configurar la siguiente red.

2. Añadir con 3 grupos VRRP virtuales, 2 en VLAN1 con direcciones virtuales 192.168.0.{11,12} y 1 en VLAN2 con dirección virtual 192.168.1.11. Hacer la configuración de forma que en VLAN1 R1 sea master de PC1 y backup de PC2 y viceversa. A su vez, en VLAN2 R1 y R2 son un grupo de routers virtuales de PC3 (R1 máster y R2 backup). Los PCs han de tener como ruta por defecto la dirección virtual de un grupo VRRP. Comprobar la configuración con show vrrp.



3. Comprobar que los PCs tienen conectividad con toda la red. Capturar los VRRP *advertizements* con tcpdump e interpretarlos. Comprobar la configuración con los comandos show.

4. Comprobar que si cae el enlace de un router (desconectando el router del switch), el *backup router* substituye al máster y los PCs no pierden la conectividad. Comprobar la configuración con show vrrp.

6. Realización de la práctica - Configuración de VMPS

1. Asignar estáticamente el puerto en el que hay conectado el PC1 a la VLAN2 (ver la Figura 12). Para el PC2 queremos que la asignación de la VLAN sea dinámica. Poner por nombre **lab2 a la VLAN2**.

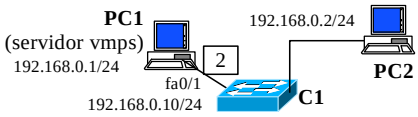


Figura 12

2. Configurar todos los demás puertos del switch con VLANs dinámicas y asignar el VTP domain "**lab**".

3. Asignar las direcciones IP que muestra la Figura 12 al PC1 y al switch.

4. Configurar PC1 como servidor VMPS del switch.

5. Ejecutar **`tcpdump -ni eth0 udp`** en una consola del PC1 para capturar el tráfico intercambiado entre el switch y el servidor vmps (en vez de eth0, poner la interface que corresponda).

6. Activar el daemon vmpsd en el PC1 con el comando: **`vmpsd -d -l 0x0707 -f vlan.db`**. El comando está en el directorio vmpsd del usuario root (directorio /root/vmpsd).

7. Configurar el PC2 con la dirección IP indicada en la Figura 12. Comprobar haciendo ping desde el PC2 al PC1 que el switch contacta con el servidor de vmps. Sin embargo el PC2 no tendrá acceso al switch, pues todavía no se ha configurado el fichero vlan.db para dar acceso a PC2.

8. Hacer una copia del fichero vlan.db a vlan-1.db (cp vlan.db vlan-1.db), y modificar vlan-1.db para que el PC2 se configure en la VLAN2. Para modificar vlan-1.db se puede usar el editor *vi* o leafpad. Ejecutar de nuevo el daemon vmpsd con el nuevo fichero de configuración. Usar los siguientes valores para la configuración global:

```
vmps domain lab
vmps mode open
vmps fallback --NONE--
vmps no-domain-req deny
```

9. Hacer ping de nuevo desde el PC2. Continuará sin poder acceder, pues el switch recuerda que el acceso de la MAC del PC2 se ha denegado.

10. Ordenar al switch el refresco del vmps y hacer ping de nuevo desde el PC2. Comprobar que el switch contacta de nuevo con el servidor al recibir la primera trama del PC2, y se asigna correctamente la vlan. Comprobar que al cambiar de puerto el PC2 y hacer ping, el switch asigna el puerto donde hay el PC2 a la VLAN correspondiente.

11. Cambiar la configuración del servidor vmps para que el PC2 sólo pueda conectarse a uno de los puertos 0/2, 0/3, 0/4, 0/5 del switch. En caso de conectarse a otro puerto deseamos que el puerto del switch quede bloqueado. Comprobar que se ha conseguido la configuración deseada.

Lab 3. Intra-Domain Routing: OSPF

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 cuatro 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. Si no es DR/BDR es DROTHER.
- 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 siembre 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. Configuración de OSPF en un router CISCO

2.1. Configuración básica en un área

Primero conviene configurar una IP en la interfaz de loopback. En un router CISCO debe ser distinta de la red 127.0.0.0/8, pues esta red está reservada como *host loopback* y los routers CISCO no permiten su uso. En Quaggua, la interfaz loopback se llama dummy0. La IP de la interface loopback de router puede ser tanto una IP pública como una IP privada, y deberá ser advertida por el protocolo de encaminamiento como una subred más de la red corporativa.

```
Router# configure terminal
Router(config)# interface dummy0
Router(config-if)# ip address 192.168.0.7/24
```

Para configurar el algoritmo de encaminamiento OSPF en un área (por ejemplo el área 0), los pasos a seguir son los siguientes:

- Si usamos **CISCO OS**: el comando "**router ospf** *process-id*" permite 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" tiene dos efectos: (i) activa la interfaz como OSPF (escuchar en la multicast 224.0.0.5), indicando que esa interfaz va a enviar o procesar mensajes de encaminamiento, y ii) indica

que esa red tiene que ser anunciada por otras interfaces que hayan activado OSPF usando también el comando network:

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

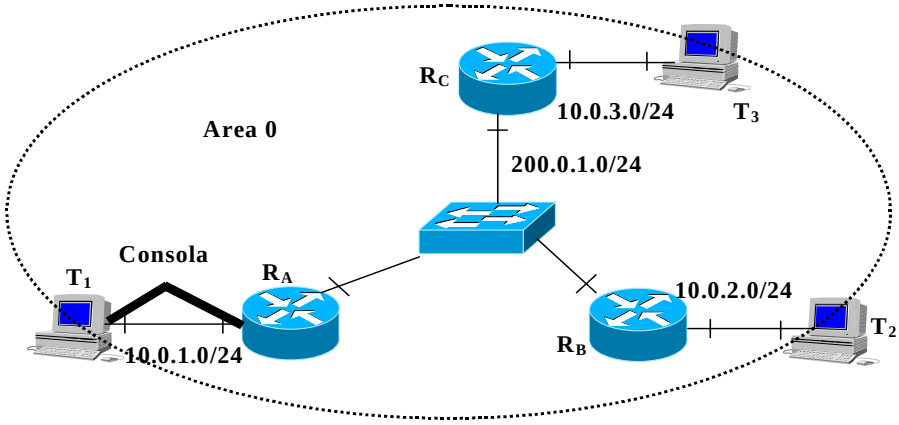


Figura 13: Red OSPF single-area

- Si usamos **zebra** el comando "**router ospf**" permite acceder al proceso OSPF en el router. Para indicar las redes que se deben anunciar se usa el comando "**network** *NetID/Mask* **area** *area-id*". El comando "network" indica las interfaces que van a enviar o procesar mensajes de encaminamiento:

```
RA# configure terminal
RA(config)# router ospf
RA(config-router)# network 200.0.1.0/24 area 0
RA(config-router)# network 10.0.1.0/24 area 0
```

A partir de ahora, todos los comandos que usemos para los Labs de encaminamiento (OSPF, BGP y Peering) indicarán el uso de ZEBRA y no el original de CISCO OSPF.

2.2. Modificación del comportamiento de OSPF

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

```
RA(config-if)# 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/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 $10^8/(56*10^3)=1785$. El SPF es un algoritmo de mínimo coste. Podemos modificar el coste de un enlace de dos maneras: (1) modificando el valor del coste en la interfaz de ese enlace con el comando:

```
RA(config-if)# ip ospf cost cost
```

donde *cost* tiene un valor entre 1 y 65535 o (2) modificando el valor del bandwidth en la interfaz que permite calcular el coste con el comando:

```
RA(config-if)# 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):

```
RA(config)# interface s0
RA(config-if)# bandwidth 2048000  → 2,048 Mbps
RA(config-if)# ip ospf cost 488 → equivalente al comando anterior
RA(config-if)# ip ospf hello-interval 30
RA(config-if)# ip ospf dead-interval 120
```

2.3. Configuración en áreas múltiples

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

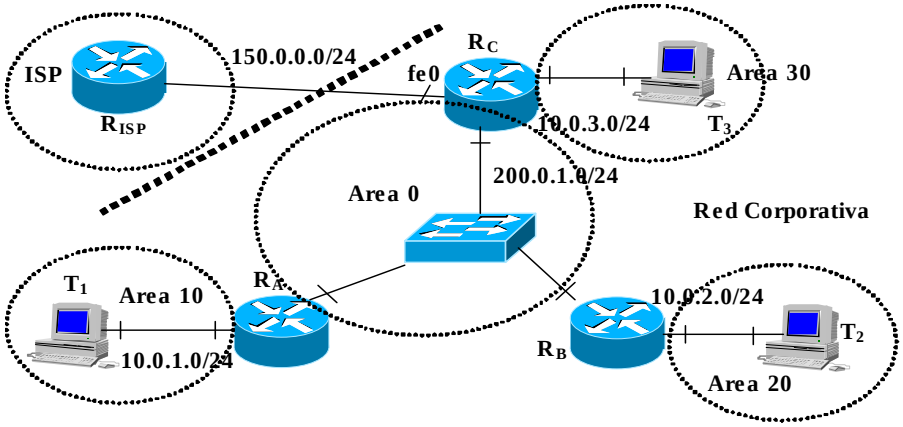


Figura 14: Red OSPF multi-area

Si hay más de un área, siempre debe haber un área 0 que haga de backbone. Debemos configurar el área de backbone (área 0) y a continuación el resto de áreas (diseño jerárquico), Figura 14.

```
RA(config)# interface e0
RA(config-if)# ip address 10.0.1.1/24
RA(config-if)# no shutdown
RA(config-if)# exit
RA(config)# interface e1
RA(config-if)# ip address 200.0.1.1/24
RA(config-if)# no shutdown
RA(config-if)# exit
RA(config)# router ospf 1
RA(config-router)# network 200.0.1.0/24 area 0
RA(config-router)# network 10.0.1.0/24 area 10
```

Como información adicional (no lo usaremos en nuestro Lab), mencionar que en los routers CISCO, 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.

2.4. Sumarización de rutas

Sumarización entre áreas: Las redes dentro de un área deben asignarse de forma que sean contiguas. La sumarización se especifica en los ABR:

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

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

2.5. Distribución de la ruta por defecto

Si queremos que el protocolo OSPF inyecte la ruta por defecto y la anuncie a todos los routers OSPF podemos usar el comando "**default-information originate**". También es importante que la o las interfaces conectadas a otros AS 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. El comando passive-interface tiene el efecto de mantener la interfaz como OSPF, activada con el comando network, pero sin enviar mensajes OSPF por ella. De esta forma, el router R_{ISP} no recibirá ningún mensaje OSPF por parte del router Rc en la red corporativa. Con ellos conseguimos que los dos dominios, ISP y Red Corporativa, estén aislados desde el punto de vista de OSPF. Es decir, cada uno de ellos tiene su encaminamiento intra-domain OSPF sin que se solapen dichos dominios. Para ver como se configira, volvamos a la Figura 14. Por ejemplo, a la configuración en el router Rc habría que añadir:

```
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)# network 150.0.0.1 area 0
Router(config-router)# default-information originate
Router(config-router)# passive-interface fe0
```

2.6. Verificación

```
R# show ip protocols   Permite ver que protocolos de encaminamiento hay activos listando
                       parámetros tales como temporizadotes, métricas, filtros, etc
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 interfaz 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)
```

El comando "show ip ospf interface" nos permite verificar gran parte de la información sobre una interfaz. Eso incluye, el Router-ID (RID), el DR, el BDR, su prioridad, sus adyacencias, sus vecinos, 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.

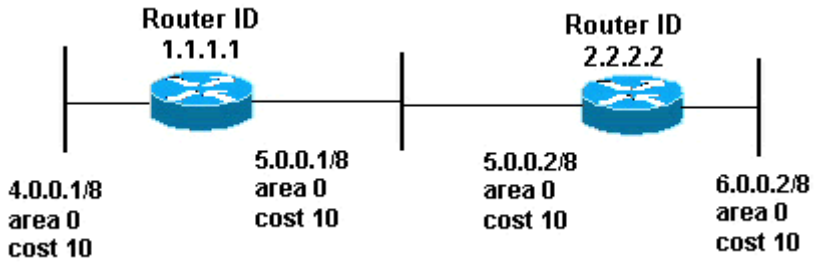


Figura 15: Red ejemplo base datos OSPF

- Router Link State Database: incluye la información de los enlaces del router que advierte el mensaje. En el caso de la figura 15, 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.

  Ejecutando el comando "show ip ospf database router 1.1.1.1" en el router r2.2.2.2 veríamos que ha recibido de r1.1.1.1 información sobre la red 4.0.0.0/24 con coste=10 (metrics).

```
(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
```

- 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 ha 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.

- Summary Link State Database: incluye información sumarizada de las redes que hay en otras áreas.

```
        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" es la dirección de red (IP_{subnet}) perteneciente a otras áreas. El campo "ADV router" contiene el RID del router que ha enviado el mensaje LSA.

3. Realización de la práctica

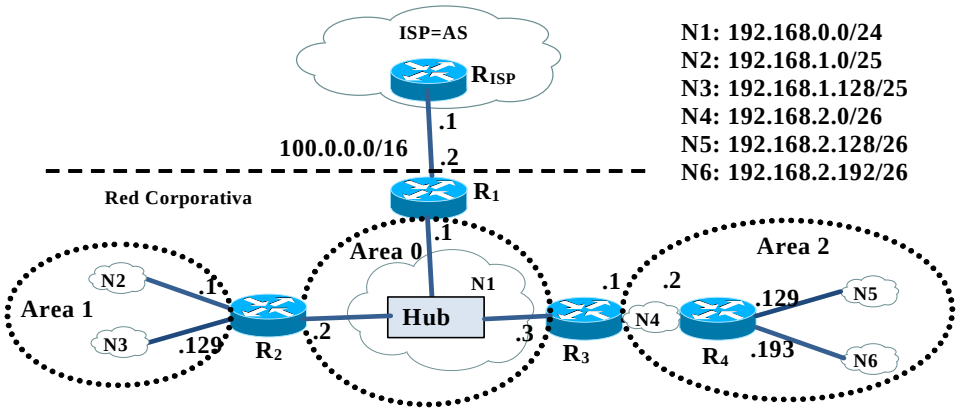


Figura 16: Topología de la red de la parte 1.

Para realizar la práctica cada grupo necesita 5 routers. Configurar la red siguiendo los pasos que se indican a continuación (es importante respetar el orden indicado):

1. Configurar todas las interfaces y la interfaz dummy (loopback, lo, si no hay dummy) de los routers. Asignaremos una dirección 10.0.<PC#>.1/30 para dummy de cada router R_x (x=1,2,…,5). Escoger como R_x el número de PC eliminando la centena, es decir, si el PC tiene número 115, escoger la red 10.0.15.0/30 como red para la dummy de ese router.

2. Configurar OSPF en los routers. Comprobar con ping que todos los routers se ven.

3. R_{ISP} representa el router del ISP. Configurar con R_{ISP} OSPF y asegurarse de que el OSPF del ISP está aislado del OSPF de la red corporativa usando passive-interfaces. Configurar la ruta por defecto en R_1 y

redistribuirla a los demás routers usando OSPF. Comprobar con ping que hay conectividad entre los routers Rx, x=1,2,3,4, y el router R_{ISP}.

4. 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.

5. Identificar que routers son ABR y cuales son internos. Identificar cuantas bases de datos tiene cada router. Activar la sumarización de rutas para que en las tablas haya el mínimo número de rutas. Comprobar la sumarización.

6. Usar los comandos de verificación de ospf (ej. show ip ospf interface), e intentar interpretarlos (ej. RID, DR y BDR en los enlaces broadcast, routers adyacentes, etc.).
    - Show ip ospf interface
    - Show ip route
    - Show ip ospf database
    - Show ip ospf neighbors
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.

9. Asignar prioridades a alguna de las interfaces para fijar una nueva elección.

- **Escribe al final de esta sección tanto el script de configuración de uno de los routers conectados al Hub (IP's, OSPF) así como la tabla de encaminamiento de los routers involucrados.**
- **Escribe también el forma de la Base de Datos OSPF.**

Lab 4. Inter-Domain Routing: BGPv4

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 también 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 mejor 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.

1.1. Best Path Selection

BGP guarda en la tabla BGP los prefijos y sus atributos (incluidos los recibidos de los BGP peers). Para ver la tabla BGP hay que ejecutar “show ip bgp”. Con esta información BGP determina la mejor ruta para cada prefijo. La mejor ruta se usa para determinar la entradas añadidas a la tabla de encaminamiento, y los mensajes BGP enviados a los BGP peers. El algoritmo de selección de la mejor ruta (*Best Path Selection*) sigue el siguiente orden:

1. Highest WEIGHT.
2. Highest LOCAL_PREF.
3. Locally originated via a network or aggregate BGP subcommand or through redistribution from an IGP.
4. Shortest AS_PATH.
5. Lowest origin type.
6. Lowest multi−exit discriminator (MED).
7. Prefer eBGP over iBGP paths.
8. Lowest IGP metric to the BGP next hop.
9. When both paths are external, prefer the path that was received first.
10. Prefer BGP routers with the lowest router ID.

2. Configuración de BGP en un router CISCO

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).

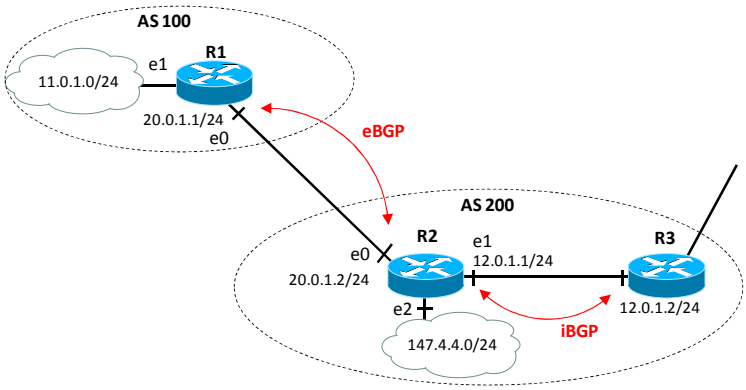


Figura 17: Ejemplo de red con diferente AS.

Para crear una sesión eBGP con un router vecino como el de la figura 17, 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.

```
R1# configure terminal
R1(config)# interface e0
R1(config-if)# ip address 20.0.1.1/24
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)# router bgp 100                          → el propio AS
R1(config-router)# neighbor 20.0.1.2 remote-as 200  → el otro AS
```

```
R2# configure terminal
R2(config)# interface e0
R2(config-if)# ip address 20.0.1.2/24
R2(config-if)# no shutdown
R2(config-if)# exit
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 mascara (opcional pero necesario si la clase es distinta de la mascara). Por ejemplo para que el router R1 anuncie por BGP la red interna 11.0.1.0 se usará el comando:

```
R1(config)# router bgp 100
R1(config-router)# network 11.0.1.0/24
```

Fijaros que:

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

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 una interfaz de loopback** (la interfaz dummy en Quagga) **para establecer las sesiones iBGP**. La razón es que de esta forma se puede mantener activa una sesión BGP en caso de fallo. En caso de tener sesione eBGP, tiene menos sentido, ya que no es normal encontrar caminos alternativos entre AS.

En efecto, si se usara la interfaz física para establecer la conexión TCP del BGP y 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 caído. Usar la interfaz de loopback también 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# configure terminal
R2(config)# interface dummy0
R2(config-if)# ip address 2.2.2.1/24
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 dummy0
```

```
R3# configure terminal
R3(config)# interface dummy0
R3(config-if)# ip address 3.3.3.1/24
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 dummy0
```

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 que distribuía también la de loopback a los routers del AS o a través de una ruta estática. En este último caso por ejemplo hay que indicar en R2 que para alcanzar la interfaz de loopback de R3 hay que transmitir a la 12.0.1.2.

```
R2(config)# ip route 3.3.3.0/30  12.0.1.2
```

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 estático o, en su mayoría, dinámico usando OSPF. En este caso el OSPF se debe configurar antes que el BGP de manera que los dos estén bien sincronizados. También es importante que la o las interfaces conectadas a otros AS 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 17, la configuración del OSPF seria:

```
R2# configure terminal
R2(config)# router ospf 10
R2(config-router)# network  20.0.1.0/24 area 0
R2(config-router)# network  12.0.1.0/24 area 0
R2(config-router)# network 147.4.4.0/24 area 0
R2(config-router)# network   2.2.2.0/24 area 0  → se incluye la red de la intefaz de
                                                        loopback
R2(config-router)# passive-interface e0
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 dummy0
R2(config-router)# neighbor 20.0.1.1 remote-as 100                → eBGP con R1
R2(config-router)# network 12.0.1.0/24
```

2.4. Verificación

```
R# show ip protocols       Permite ver que protocolos de encaminamiento hay activos listando
                           parámetros tales como temporizadores, métricas, filtros, etc
R# show ip route           Permite ver la tabla de encaminamiento
R# show ip bgp             Permite ver la tabla 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.)
```

3. Realización de la práctica

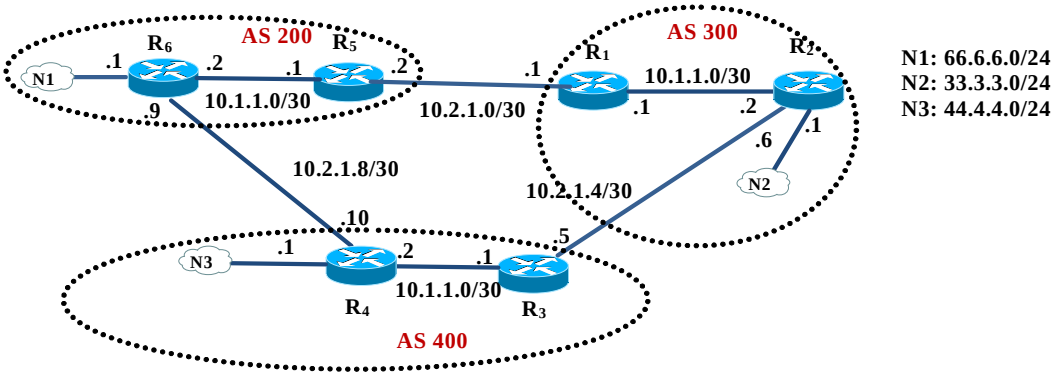


Figura 18: Topología de la red para esta práctica.

Para realizar la práctica cada grupo necesita 6 routers. Configurar la red siguiendo los pasos que se indican a continuación: Usar OSPF en AS300 y AS400 para que distribuya sólo las direcciones privadas, y BGP para que distribuya sólo las direcciones públicas.

1. Configurar todas las interfaces de los routers y añadir una interfaz de loopback en cada router. Asignaremos una dirección 10.0.<PC#>.1/30 para el loopback de cada router R_x, x=1,..6. Escoger como R_x el número de PC eliminando la centena, es decir, si el PC tiene número 115, escoger la red 10.0.15.0/30 como red para la loopback de ese router.

2. Asignar las direcciones IP's indicadas en la Figura 18. Comprobar con ping que hay conectividad entre interfaces de una misma red.

3. Activar OSPF en AS300 y AS400. Comprobar que los distintos AS's no ven las redes internas de los otros AS's. Comprobar con ping que los routers y el PC de un mismo AS se ven.

4. Activar iBGP entre los routers del AS300 y AS400 a través de sus interfaces de loopback. Comprobar que la sesión i-BGP se ha establecido. Para ello ejecutar "show ip bgp summary" y comprobar que el estado no es IDLE ni ACTIVE: cuando está ESTABLISED muestra un entero que indica cuantos prefijos se han aprendido del bgp-peer.

5. Activar eBGP entre los routers de los distintos AS's. Comprobar que las sesiones e-BGP se ha establecido. Para ello ejecutar "show ip bgp summary" y comprobar que el estado es ESTABLISHED.

6. Comprobar que hay conectividad entre todos los routers con IPs públicas.

7. Comprobar que en las tablas de encaminamiento de todos los routers aparecen las redes públicas. Comprobar que aparecen correctamente las entradas que corresponden a las redes directamente conectadas, las redes aprendida por OSPF y las redes aprendidas por BGP.

8. Usar los comandos de verificación de BGP, e intentar interpretarlos.
   - Show ip route
   - Show ip bgp
   - Show ip bgp neighbors
   - Show ip bgp paths
   - Show ip bgp summary

9. Probar de desconectar el enlace entre R1 y R2 e interpretar lo que ocurre.

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

Escribe a continuación tanto el script de configuración de uno de los routers conectados a un PC (IP's, OSPF y BGP) así como la tabla BGP de los routers involucrados.

Lab 5. BGPv4 attribute manipulation – Local-Pref and Communities

1. Objetivo de la práctica

Para implementar políticas de encaminamiento con BGP se usan 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 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.

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 clausula con respecto a otras clausulas 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
elseif condición then acción
else acción

ya que cada clausula permit seq-number tiene una acción y condición y si no se cumple se pasa a la siguiente clausula de forma secuencial. Si una de ellas se cumple, entonces se sale del if/elseif. Notar que las clausulas del route-map están numeradas como 10, 20, 30, etc. Hay dos motivos para numerar las clausulas de esta manera:

- Borrar fácilmente una clausula sin afectar otra clausula.
- Insertar nuevas clausulas entre dos clausulas existentes.

Cada route-map tiene dos tipos de comandos:

- ***Match***: selecciona rutas que la clausula debe aplicar. Hay varias maneras de seleccionar las rutas. La más sencilla es usar ACLs (Access Lists). Si hay varios match en una clausula, todas deben cumplirse (AND) para que se ejecute el comando set. Si no hay una clausula 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 para el comando match 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
route-map My-Map-1 permit 20
!
 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 BGP 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 que conexión BGP la tiene que realizar (ver más adelante como se asigna la acción). Las clausulas pueden permitir/denegar y las ACLs también. Por consiguiente hay que conjugar las posibles combinaciones de la clausula con el ACL.

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

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

Ejemplo de uso: el router R_2, figura 19, 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 R_2 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 vía 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.

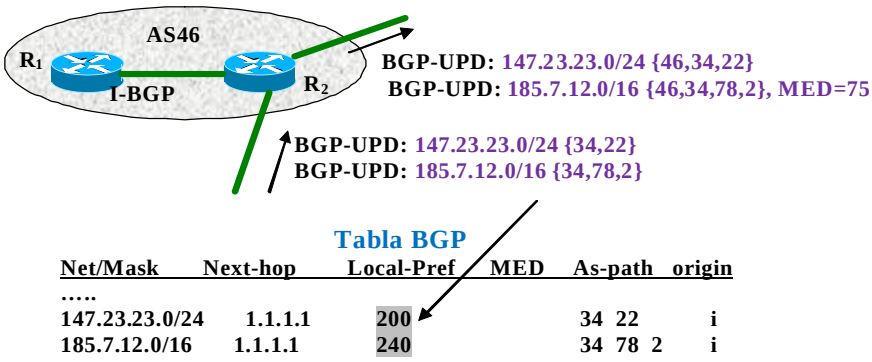


Figura 19: Mensajes Update BGP

En el router R_2 configuraremos:

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

!
route-map My-Map-1 permit 10
match ip address 1
set local-preference 200
!
route-map My-Map-1 permit 20
match ip address 2
set local-preference 240
route-map My-Map-1 permit 30
!
route-map My-Map-2 permit 10
match ip address 2
set metric 75
route-map My-Map-2 permit 20
!
access-list 1 permit   147.23.23.0    0.0.0.255
access-list 2 permit  185.7.12 0      0.0.255.255
```

Notar que para asignar el route-map usamos 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.

3. Uso de comunidades.

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, figura 20, que el AS78 ha acordado con el AS46 que el router R_2 fije un Local Pref = 175 en el enlace R_3-R_2 cuando reciba la comunidad 78:500.

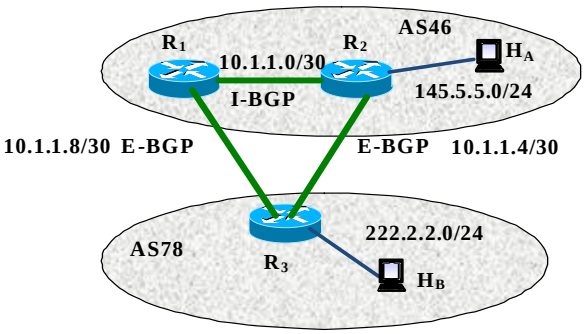


Figura 20: Comunidades en BGP

R3

```
router bgp 78
neighbor 10.1.1.9 remote-as 46
neighbor 10.1.1.5 remote-as 46
neighbor 10.1.1.5 send-community
neighbor 10.1.1.5 route-map Peer-R2 out
!
route-map Peer-R2 permit 10
match ip address 1
set community 78:500
route-map Peer-R2 permit 20
!
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 R_2 (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 R_2 para que ejecute las acciones que queramos realizar cuando detecte la comunidad 78:500.

R2

```
router bgp 46
neighbor 10.1.1.1 remote-as 46
neighbor 10.1.1.6 remote-as 78
neighbor 10.1.1.6 route-map Peer-R3 in
!
route-map Peer-R3 permit 10
match community 1
set local-preference 175
route-map Peer-R3 permit 20
!
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 acorde 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.

4. Realización de la práctica

Paso 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. NOTA: no usar las interfaces de loopback, para configurar más rápidamente OSPF/BGP. Usar OSPF para que distribuya sólo las direcciones privadas, y BGP para que distribuya sólo las direcciones públicas. Comprobar que las sesiones BGP se han establecido. Para ello ejecutar "**show ip bgp summary**" y comprobar que el estado es ESTABLISHED.

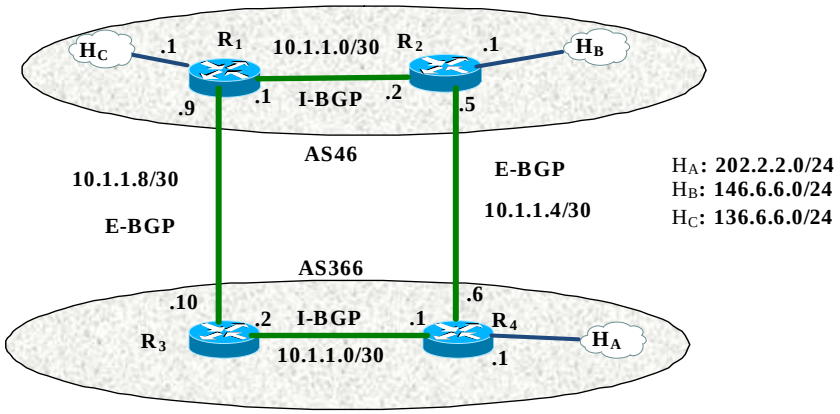


Figura 21: Práctica BGP

Escribe los scripts para los routers R_1, R_2, R_3, R_4.

Paso 2: Definir una política de encaminamiento usando el atributo Local-Pref y comunidades que realice la siguiente función:

Si somos el administrador del AS366:

- Configurar R_4 para que los paquetes provenientes del la red H_A y con destino la red 136.6.6.0/24 (H_C) deben salir por el enlace que une los routers R_4-R_3 y R_3-R_1.
- Configurar R_4 para que forzar al AS46 a que los paquetes que provengan de la red 136.6.6.0/24 (H_C) hacia el AS366 (H_A) vayan por el enlace que une los routers R_1-R_2 y R_2-R_4.

Si somos el administrador del AS46:

- Configurar R_1 para que la política definida por el AS366 tenga efecto. Comprobar con tcpdump y ping que los paquetes siguen las rutas deseadas. NOTA: en un router con varias interfaces (y direcciones IP) se puede usar ping –I <dirección IP interfaz> para enviar paquetes con la dirección IP fuente igual a <dirección IP interfaz>.

Escribe al final de esta sección los scripts para los routers involucrados en los cambios e indica la tabla BGP de dichos routers.

Comandos de visualización de rutas:

- show ip route
- show ip route @IP_{red}
- show ip bgp
- show ip bgp @IP_{red}

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

Scripts

Lab 6. BGPv4 Route-Reflectors

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 confederación.

2. Introducción a iBGP

Uno de los requisitos que se han visto es que se debe establecer una malla completa de sesiones iBGP entre router BGP (ver Figura 22).En otras palabras, cada router BGP en un AS debe tener sesiones iBGP con todos los otros routers BGP del AS. Esto hace que las tablas de encaminamiento no puedan producir 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.

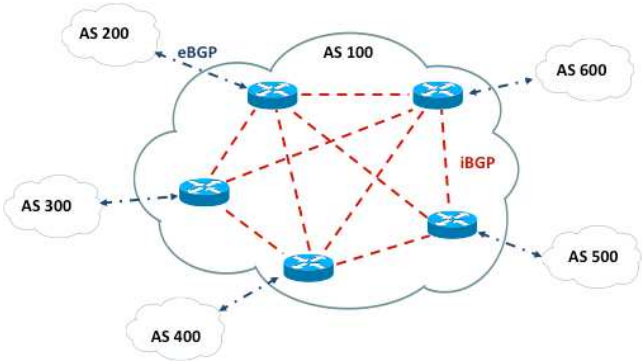


Figura 22: Malla completa de sesiones iBGP.

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 iBGPpara todo el AS. Existen dos métodos para relajar este requisito que son Route Reflector y Confederación que veremos a continuación.

3. Reflectores de Rutas (RFC 4456)

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 RR[1]. 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 la 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.

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 reenvía solamente al RR. El RR a su vez, enviará este mensaje a los otros AS por eBGP como es habitual pero también 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.

[1] Tambien se podrían configurar dos o mas RR por cluster pero en esta práctica solo se considera el caso de un RR por cluster

Para esta configuración se necesita añadir en la configuración del router elegido como RR el comando "**neighbor** *@IP-neighbor* **route-reflector-client**".

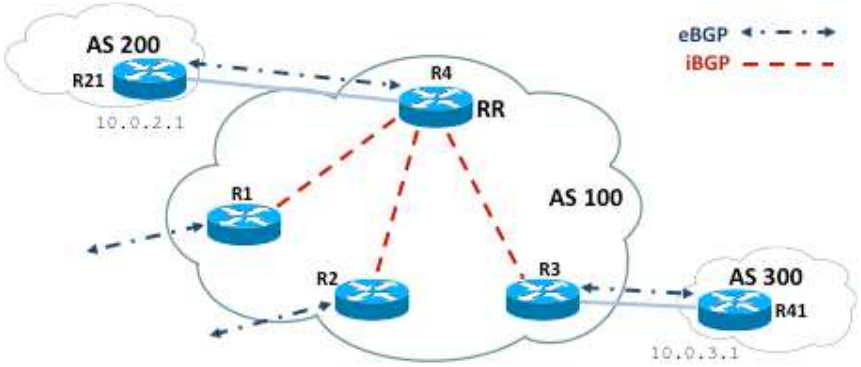


Figura 23: **Ejemplo de un Route Reflector en un AS.**

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 siguiente ejemplo, 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.

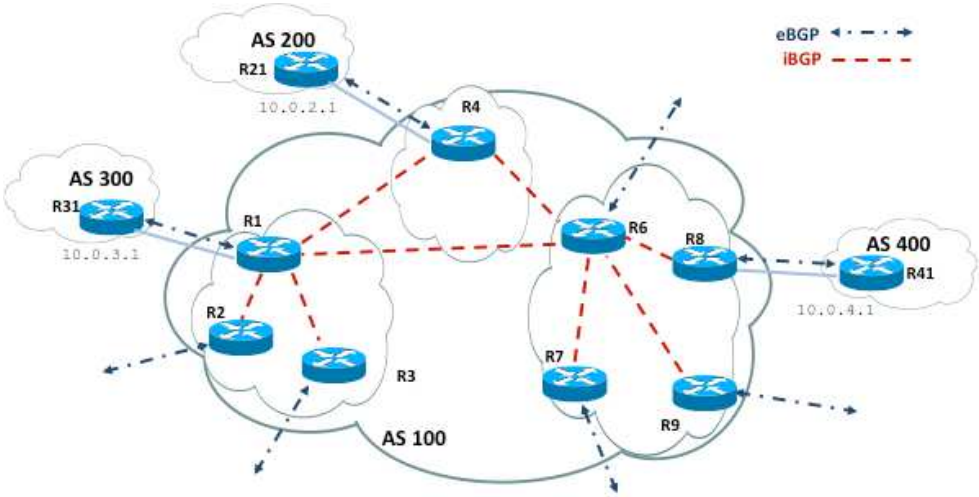


Figura 24: Ejemplo de tres Route Reflector en un AS.

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.

4. Confederaciones (RFC 5065)

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 confederación 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.

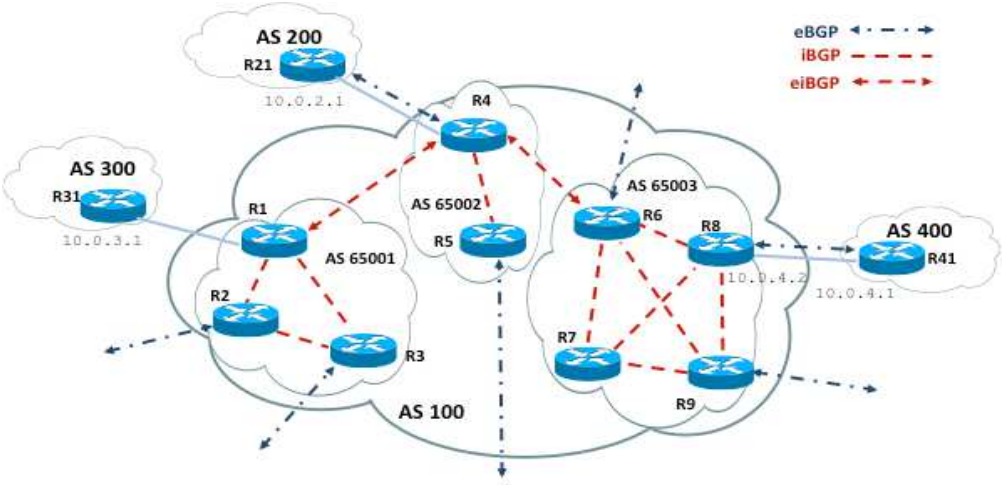


Figura 25: Ejemplo de confederación de AS.

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***AS_1-number AS_2-number etc.*", donde AS_n-number son los números que identifican los sub-ASes que forman 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 25, 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
R4(config-router)# bgp confederation identifier 100
R4(config-router)# bgp confederation peers 65001 65003-> está conectado al sub-AS 65001 y
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 R41.

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

4.1. Aislamiento de OSPF entre Sub-ASs distintos

En el esquema anterior el next-hop de las rutas anunciadas por eBGP serán las IPs de las conexiones con los routers externos. Por lo tanto, todos los sub-ASes deberán tener conectividad con esas IPs. Si se desea poder aislar los protocolos de routing internos (ej. OSPF) de sub-ASes distintos, de forma que esas IPs no tengan que propagarse entre todos ellos, puede usarse el comando "neighbor BGPpeer next-hop-self". Con este comando cuando el router envía el mensaje BGP de update hacia BGPpeer, se pone a si mismo como next-hop. Así pues, usando esta configuración en las conexiones eiBGP podemos conseguir que los routers dentro de un subAS sólo tengan que tener conectividad con las IPs de las conexiones eiBGP de ese subAS. Permitiendo así, aislar los protocolos OSPF de subASes distintos (con el comando passive-interface).

Por ejemplo, usando esta técnica en el router R4 de la Figura 25 la configuración sería la siguiente.

```
R4(config)# router bgp 65002
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 10.100.1.1 next-hop-self -> eiBGP
R4(config-router)# neighbor 10.100.5.1 remote-as 65002
R4(config-router)# neighbor 10.100.6.1 remote-as 65003
R4(config-router)# neighbor 10.100.6.1 next-hop-self -> eiBGP
```

5. Realización de la práctica

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

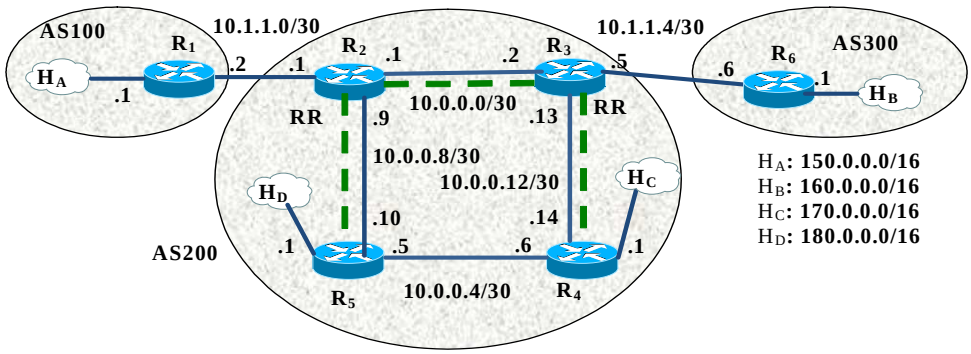


Figura 26: Red de la práctica.

NOTA: no usar las interfaces de loopback, para configurar más rápidamente OSPF/BGP. Usar OSPF para que distribuya las direcciones privadas y públicas, y BGP para que distribuya sólo las direcciones públicas. Usar passive-interfaces para aislar AS's y confederaciones.

1. Configurar las direcciones IP en todas las interfaces de la red que aparece en la figura de forma que haya conectividad entre dos interfaces vecinas.
2. Configurar encaminamiento interno (OSPF) en los routers del AS 200.
3. Activar eBGP entre R1-R2 y R3-R6. Establecer sólo las conexiones iBGP que aparecen con líneas discontinuas (R5-R2, R2-R3, R3-R4). Comprobar que las sesiones BGP se han establecido. Para ello ejecutar “show ip bgp neighbor” y comprobar que el estado es ESTABLISHED. Comprobar en que en las tablas BGP y de encaminamiento no aparecen todas las redes públicas, razonar porqué.
4. Configurar R2 y R3 como routers RR para las sesiones iBGP del AS 200. Configurar R4 y R5 como clientes de R3 y R2, respectivamente. Comprobar que ahora las tablas BGP y de encaminamiento tienen todas las redes públicas.
5. Borrar la configuración relativa a los RRs. Establecer una confederación con 2 sub-ASs: uno formado por R2-R5 (64512) y otro por R3-R4 (64513). Comprobar que las tablas BGP y de encaminamiento tienen todas las redes públicas.
6. Usar “neighbor BGPpeer next-hop-self” entre R2 y R3, tal como se explica en el apartado 4.1. Comprobar cómo cambia el next-hop en las tablas BGP al añadir esta configuración.

- NOTA: en un router con varias interfaces (y direcciones IP) se puede usar ping –I <dirección IP interfaz> para enviar paquetes con la dirección IP fuente igual a <dirección IP interfaz>.

Escribe a continuación los scripts de configuración.

APENDICE

OSPF router

To start OSPF process you have to specify the OSPF router. As of this writing, ospfd does not support multiple OSPF processes.

router ospf Command

Enable or disable the OSPF process. ospfd does not yet support multiple OSPF processes. So you can not specify an OSPF process number.

ospf router-id a.b.c.d OSPF Command

passive interface interface OSPF Command

timers spf <0-4294967295> <0-4294967295> OSPF Command

refresh group-limit <0-10000> OSPF Command

refresh per-slice <0-10000> OSPF Command

refresh age-diff <0-10000> OSPF Command

auto-cost refrence-bandwidth <1-4294967> OSPF Command

network a.b.c.d/m area a.b.c.d OSPF Command

network a.b.c.d/m area <0-4294967295> OSPF Command

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 10.0.0.0/8 area 0

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

OSPF interface

ip ospf cost <1-65535> Interface Command

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> Interface Command

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> Interface Command

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) Interface Command

Set explicitly network type for specifed interface.

ip ospf priority <0-255> Interface Command

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> Interface Command

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 Interface Command

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

Redistribute routes to OSPF

default-information originate	OSPF Command
default-information originate metric <0-16777214>	OSPF Command
default-information originate metric <0-16777214> metric-type (1|2)	OSPF Command
default-information originate metric <0-16777214> metric-type (1|2) route-map word	OSPF Command
default-information originate always	OSPF Command
default-information originate always metric <0-16777214>	OSPF Command
default-information originate always metric <0-16777214> metric-type (1|2)	OSPF Command
no default-information originate	OSPF Command
distribute-list NAME out (kernel|connected|static|rip|ospf	OSPF Command
default-metric <0-16777214>	OSPF Command
distance <1-255>	OSPF Command
distance ospf (intra-area|inter-area|external) <1-255>	OSPF Command
router zebra	Command

Showing OSPF information

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

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* Command

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

no router bgp *asn* Command

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

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

This command specifies the router-ID. If bgpd connects to zebra it gets interface and address information. In that case default router ID value is selected as the largest IP Address of the interfaces. When router zebra is not enabled bgpd can't get interface information so router-id is set to 0.0.0.0. So please set router-id by hand.

BGP route

network *A.B.C.D/M* BGP

This command adds the announcement network.

```
router bgp 1
 network 10.0.0.0/8
```

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.

Defining Peer

neighbor *peer* **remote-as** *asn* BGP

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, bgpd will complain like this:

```
can't find neighbor 10.0.0.1
```

BGP Peer commands

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

neighbor *peer* **shutdown** BGP

no neighbor *peer* **shutdown** BGP

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-multihop** BGP

no neighbor *peer* **ebgp-multihop** BGP

neighbor *peer* **description ...** BGP

no neighbor *peer* **description ...** BGP

Set description of the peer.

neighbor *peer* **version** *version* BGP

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* BGP

no neighbor *peer* **interface** *ifname* BGP

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** BGP

no neighbor *peer* **next-hop-self** BGP

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

neighbor *peer* **update-source** BGP

no neighbor *peer* **update-source** BGP

neighbor *peer* **default-originate** BGP

no neighbor *peer* **default-originate** BGP

bgpd'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* BGP

neighbor *peer* **port** *port* BGP

neighbor *peer* **send-community** BGP

neighbor *peer* **send-community** BGP

neighbor *peer* **weight** *weight* BGP

no neighbor *peer* **weight** *weight* BGP

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

neighbor *peer* **maximum-prefix** *number* BGP

no neighbor *peer* **maximum-prefix** *number* BGP

Peer filtering

neighbor *peer* **distribute-list** *name* **[in|out]** BGP

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

neighbor *peer* **prefix-list** *name* **[in|out]** BGP command

neighbor *peer* **filter-list** *name* **[in|out]** BGP command

neighbor *peer* **route-map** *name* **[in|out]** BGP

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

Show IP BGP

show ip bgp Command

show ip bgp *A.B.C.D* Command

show ip bgp *X:X::X:X* Command

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
```

More Show IP BGP

show ip bgp regexp *line* Command

This command display BGP routes using AS path regular expression (see Display BGP Routes by AS Path).

show ip bgp community *community* Command

show ip bgp community *community* **exact-match** Command

This command display BGP routes using *community* (see Display BGP Routes by Community).

show ip bgp community-list *word* Command

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

This command display BGP routes using community list (see Display BGP Routes by Community).

show ip bgp summary Command

show ip bgp neighbor [*peer***]** Command

clear ip bgp *peer* Command

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

clear ip bgp *peer* **soft in** Command

Clear peer using soft reconfiguration.

show debug Command

debug event Command

debug update Command

debug keepalive Command

no debug event Command

no debug update Command

no debug keepalive

IP Access List

access-list *name* **permit** *ipv4-network* Command

access-list *name* **deny** *ipv4-network* Command

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
```

Route Map

Route map is a very useful function in zebra. 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 Command

route-map *route-map-name* **permit** *priority* Command

Route Map Match Command

match ip address *access_list* Route-map Command

Matches the specified *access_list*

match ip next-hop *ipv4_addr* Route-map Command

Matches the specified *ipv4_addr*.

match aspath *as_path* Route-map Command

Matches the specified *as_path*.

match metric *metric* Route-map Command

Matches the specified *metric*.

match community *community_list* Route-map Command

Matches the specified *community_list*

Route Map Set Command

set ip next-hop *ipv4_address* Route-map Command

Set the BGP nexthop address.

set local-preference *local_pref* Route-map Command

Set the BGP local preference.

set weight *weight* Route-map Command

Set the route's weight.

set metric *metric* Route-map Command

Set the BGP attribute MED.

set as-path prepend *as_path* Route-map Command

Set the BGP AS path to prepend.

set community *community* Route-map Command

Set the BGP community attribute.

set ipv6 next-hop global *ipv6_address* Route-map Command

Set the BGP-4+ global IPv6 nexthop address.

set ipv6 next-hop local *ipv6_address* Route-map Command

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