

Práctica 1.5. RIP y BGP

Objetivos

En esta práctica se afianzan los conceptos elementales del encaminamiento. En particular, se estudia un protocolo de encaminamiento interior y otro exterior: RIP (*Routing Information Protocol*) y BGP (*Border Gateway Protocol*).

Existen muchas implementaciones de los protocolos de encaminamiento. En esta práctica vamos a utilizar Quagga, que actualmente implementa RIP (versiones 1 y 2), RIPng, OSPF, OSPFv3, IS-IS y BGP. Quagga está estructurado en diferentes servicios (uno para cada protocolo) controlados por un servicio central (Zebra) que hace de interfaz entre la tabla de reenvío del *kernel* y las tabla de encaminamiento de cada protocolo.

Todos los ficheros de configuración han de almacenarse en el directorio `/etc/quagga`. La sintaxis de estos ficheros es sencilla y está disponible en <http://quagga.net>. Revisar especialmente la correspondiente a RIP y BGP en <https://www.quagga.net/docs/quagga.html>. Además, en `/usr/share/doc/quagga-0.99.22.4` hay ficheros de ejemplo.



Activar el **portapapeles bidireccional** (menú Dispositivos) en las máquinas virtuales.

Usar la opción de Virtualbox (menú Ver) para realizar **capturas de pantalla**.

La **contraseña** del usuario `cursoresdes` es `cursoresdes`.

Contenidos

Parte I. Protocolo interior: RIP

- Preparación del entorno

- Configuración del protocolo RIP

Parte II. Protocolo exterior: BGP

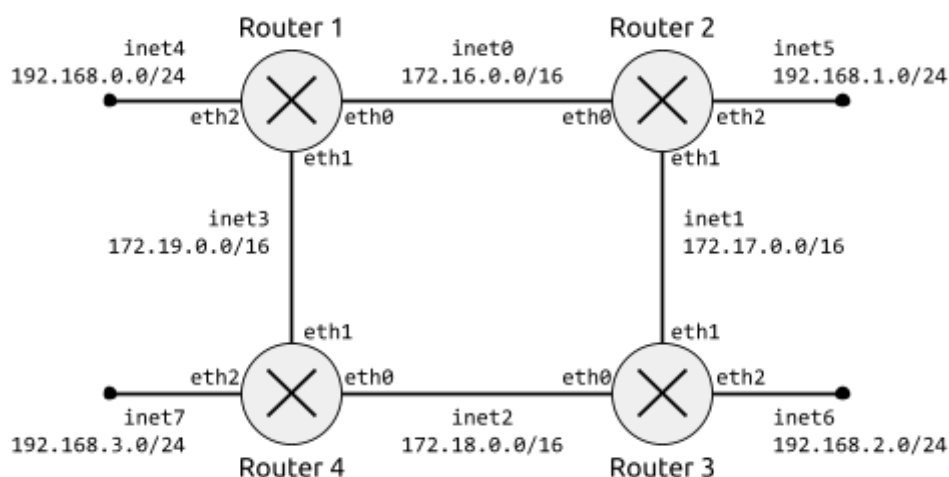
- Preparación del entorno

- Configuración del protocolo BGP

Parte I. Protocolo interior: RIP

Preparación del entorno

Configuraremos la topología de red que se muestra en la siguiente figura, donde cada encaminador (Router1...Router4) tiene tres interfaces, cada uno conectado a una red diferente.:



Al igual que en prácticas anteriores, usaremos la herramienta vtopo1 para construir automáticamente esta topología. A continuación se muestra el contenido del fichero de configuración de la topología:

```
netprefix inet
machine 1 0 0 1 3 2 4
machine 2 0 0 1 1 2 5
machine 3 0 2 1 1 2 6
machine 4 0 2 1 3 2 7
```

Para facilitar la configuración de las máquinas, la siguiente tabla muestra las direcciones de cada uno de los interfaces de los encaminadores:

Máquina virtual	Interfaz	Dirección de red	Dirección IP
Router1	eth0	172.16.0.0/16	172.16.0.1
	eth1	172.19.0.0/16	172.19.0.1
	eth2	192.168.0.0/24	192.168.0.1
Router2	eth0	172.16.0.0/16	172.16.0.2
	eth1	172.17.0.0/16	172.17.0.2
	eth2	192.168.1.0/24	192.168.1.2
Router3	eth0	172.18.0.0/16	172.18.0.3
	eth1	172.17.0.0/16	172.17.0.3
	eth2	192.168.2.0/24	192.168.2.3
Router4	eth0	172.18.0.0/16	172.18.0.4
	eth1	172.19.0.0/16	172.19.0.4
	eth2	192.168.3.0/24	192.168.3.4

Configurar todos los encaminadores según la figura y tabla anterior. Además, activar el reenvío de paquetes IPv4 igual que en la práctica 1.1. Después, comprobar:

- Que los encaminadores adyacentes son alcanzables, por ejemplo, Router1 puede hacer *ping* a Router2 y Router4.
- Que la tabla de reenvío de cada encaminador es la correcta e incluye una entrada para cada una de las tres redes a las que está conectado.

Configuración del protocolo RIP

Ejercicio 1. Configurar RIP en todos los encaminadores para que intercambien información:

- Crear un fichero ripd.conf en /etc/quagga con el contenido que se muestra a continuación.
- Iniciar el servicio RIP (y Zebra) con `service ripd start`.

Contenido del fichero /etc/quagga/ripd.conf:

```
# Activar el encaminamiento por RIP
router rip
# Definir la versión del protocolo que se usará
version 2
# Habilitar información de encaminamiento en redes asociadas a los interfaces
network eth0
network eth1
network eth2
```

Ejercicio 2. Consultar la tabla de encaminamiento de RIP y de Zebra en cada encaminador con el comando `vttysh` (`sudo vtysh -c "show ip rip"` y `sudo vtysh -c "show ip route"`). Comprobar también la tabla de reenvío de IPv4 con el comando `ip` (`ip route`).

Copia los comandos usados y su salida.

Router1:

```
[cursoredes@localhost ~]$ sudo vtysh -c "show ip rip"
```

Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP

Sub-codes:

(n) - normal, (s) - static, (d) - default, (r) - redistribute,
(i) - interface

Network	Next Hop	Metric From	Tag Time
C(i) 172.16.0.0/16	0.0.0.0	1 self	0
R(n) 172.17.0.0/16	172.16.0.2	2 172.16.0.2	0 02:57
R(n) 172.18.0.0/16	172.19.0.4	2 172.19.0.4	0 02:57
C(i) 172.19.0.0/16	0.0.0.0	1 self	0
C(i) 192.168.0.0/24	0.0.0.0	1 self	0
R(n) 192.168.1.0/24	172.16.0.2	2 172.16.0.2	0 02:57
R(n) 192.168.2.0/24	172.16.0.2	3 172.16.0.2	0 02:57
R(n) 192.168.3.0/24	172.19.0.4	2 172.19.0.4	0 02:57

```
[cursoredes@localhost ~]$ sudo vtysh -c "show ip route"
```

Codes: K - kernel route, C - connected, S - static, R - RIP,

O - OSPF, I - IS-IS, B - BGP, A - Babel,

> - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo

C>* 172.16.0.0/16 is directly connected, eth0

R>* 172.17.0.0/16 [120/2] via 172.16.0.2, eth0, 00:00:50

R>* 172.18.0.0/16 [120/2] via 172.19.0.4, eth1, 00:00:50

C>* 172.19.0.0/16 is directly connected, eth1

C>* 192.168.0.0/24 is directly connected, eth2

R>* 192.168.1.0/24 [120/2] via 172.16.0.2, eth0, 00:00:50

R>* 192.168.2.0/24 [120/3] via 172.16.0.2, eth0, 00:00:50

R>* 192.168.3.0/24 [120/2] via 172.19.0.4, eth1, 00:00:50

```
[cursoredes@localhost ~]$ ip route
```

```
172.16.0.0/16 dev eth0 proto kernel scope link src 172.16.0.1
```

```
172.17.0.0/16 via 172.16.0.2 dev eth0 proto zebra metric 2
```

```
172.18.0.0/16 via 172.19.0.4 dev eth1 proto zebra metric 2
```

```
172.19.0.0/16 dev eth1 proto kernel scope link src 172.19.0.1
```

```
192.168.0.0/24 dev eth2 proto kernel scope link src 192.168.0.1
```

```
192.168.1.0/24 via 172.16.0.2 dev eth0 proto zebra metric 2
```

```
192.168.2.0/24 via 172.16.0.2 dev eth0 proto zebra metric 3
```

192.168.3.0/24 via 172.19.0.4 dev eth1 proto zebra metric 2

Router2:

[cursoredes@localhost ~]\$ sudo vtysh -c "show ip rip"

Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP

Sub-codes:

(n) - normal, (s) - static, (d) - default, (r) - redistribute,
(i) - interface

	Network	Next Hop	Metric From	Tag Time
C(i)	172.16.0.0/16	0.0.0.0	1 self	0
C(i)	172.17.0.0/16	0.0.0.0	1 self	0
R(n)	172.18.0.0/16	172.17.0.3	2 172.17.0.3	0 02:43
R(n)	172.19.0.0/16	172.16.0.1	2 172.16.0.1	0 02:44
R(n)	192.168.0.0/24	172.16.0.1	2 172.16.0.1	0 02:44
C(i)	192.168.1.0/24	0.0.0.0	1 self	0
R(n)	192.168.2.0/24	172.17.0.3	2 172.17.0.3	0 02:43
R(n)	192.168.3.0/24	172.17.0.3	3 172.17.0.3	0 02:43

[cursoredes@localhost ~]\$ sudo vtysh -c "show ip route"

Codes: K - kernel route, C - connected, S - static, R - RIP,

O - OSPF, I - IS-IS, B - BGP, A - Babel,

> - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo

C>* 172.16.0.0/16 is directly connected, eth0

C>* 172.17.0.0/16 is directly connected, eth1

R>* 172.18.0.0/16 [120/2] via 172.17.0.3, eth1, 00:02:40

R>* 172.19.0.0/16 [120/2] via 172.16.0.1, eth0, 00:02:30

R>* 192.168.0.0/24 [120/2] via 172.16.0.1, eth0, 00:02:30

C>* 192.168.1.0/24 is directly connected, eth2

R>* 192.168.2.0/24 [120/2] via 172.17.0.3, eth1, 00:02:40

R>* 192.168.3.0/24 [120/3] via 172.17.0.3, eth1, 00:02:40

[cursoredes@localhost ~]\$ ip route

172.16.0.0/16 dev eth0 proto kernel scope link src 172.16.0.2

172.17.0.0/16 dev eth1 proto kernel scope link src 172.17.0.2

172.18.0.0/16 via 172.17.0.3 dev eth1 proto zebra metric 2

172.19.0.0/16 via 172.16.0.1 dev eth0 proto zebra metric 2

192.168.0.0/24 via 172.16.0.1 dev eth0 proto zebra metric 2

192.168.1.0/24 dev eth2 proto kernel scope link src 192.168.1.0

192.168.2.0/24 via 172.17.0.3 dev eth1 proto zebra metric 2

192.168.3.0/24 via 172.17.0.3 dev eth1 proto zebra metric 3

[cursoredes@localhost ~]\$ sudo vtysh -c "show ip rip"

Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP

Sub-codes:

(n) - normal, (s) - static, (d) - default, (r) - redistribute,
(i) - interface

	Network	Next Hop	Metric From	Tag Time
R(n)	172.16.0.0/16	172.17.0.2	2 172.17.0.2	0 02:31
C(i)	172.17.0.0/16	0.0.0.0	1 self	0
C(i)	172.18.0.0/16	0.0.0.0	1 self	0
R(n)	172.19.0.0/16	172.18.0.4	2 172.18.0.4	0 02:31
R(n)	192.168.0.0/24	172.17.0.2	3 172.17.0.2	0 02:31
R(n)	192.168.1.0/24	172.17.0.2	2 172.17.0.2	0 02:31
C(i)	192.168.2.0/24	0.0.0.0	1 self	0
R(n)	192.168.3.0/24	172.18.0.4	2 172.18.0.4	0 02:31

```
[cursoredes@localhost ~]$ sudo vtysh -c "show ip route"
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo
R>* 172.16.0.0/16 [120/2] via 172.17.0.2, eth1, 00:03:29
C>* 172.17.0.0/16 is directly connected, eth1
C>* 172.18.0.0/16 is directly connected, eth0
R>* 172.19.0.0/16 [120/2] via 172.18.0.4, eth0, 00:03:38
R>* 192.168.0.0/24 [120/3] via 172.17.0.2, eth1, 00:03:20
R>* 192.168.1.0/24 [120/2] via 172.17.0.2, eth1, 00:03:29
C>* 192.168.2.0/24 is directly connected, eth2
R>* 192.168.3.0/24 [120/2] via 172.18.0.4, eth0, 00:03:38
[cursoredes@localhost ~]$ ip route
172.16.0.0/16 via 172.17.0.2 dev eth1 proto zebra metric 2
172.17.0.0/16 dev eth1 proto kernel scope link src 172.17.0.3
172.18.0.0/16 dev eth0 proto kernel scope link src 172.18.0.3
172.19.0.0/16 via 172.18.0.4 dev eth0 proto zebra metric 2
192.168.0.0/24 via 172.17.0.2 dev eth1 proto zebra metric 3
192.168.1.0/24 via 172.17.0.2 dev eth1 proto zebra metric 2
192.168.2.0/24 dev eth2 proto kernel scope link src 192.168.2.3
192.168.3.0/24 via 172.18.0.4 dev eth0 proto zebra metric 2

Router4:
[cursoredes@localhost ~]$ sudo vtysh -c "show ip rip"
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
        (n) - normal, (s) - static, (d) - default, (r) - redistribute,
        (i) - interface

      Network      Next Hop      Metric From      Tag Time
R(n) 172.16.0.0/16  172.19.0.1      2 172.19.0.1      0 02:46
R(n) 172.17.0.0/16  172.18.0.3      2 172.18.0.3      0 02:57
C(i) 172.18.0.0/16  0.0.0.0         1 self            0
C(i) 172.19.0.0/16  0.0.0.0         1 self            0
R(n) 192.168.0.0/24 172.19.0.1      2 172.19.0.1      0 02:46
R(n) 192.168.1.0/24 172.18.0.3      3 172.18.0.3      0 02:57
R(n) 192.168.2.0/24 172.18.0.3      2 172.18.0.3      0 02:57
C(i) 192.168.3.0/24 0.0.0.0         1 self            0
[cursoredes@localhost ~]$ sudo vtysh -c "show ip route"
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

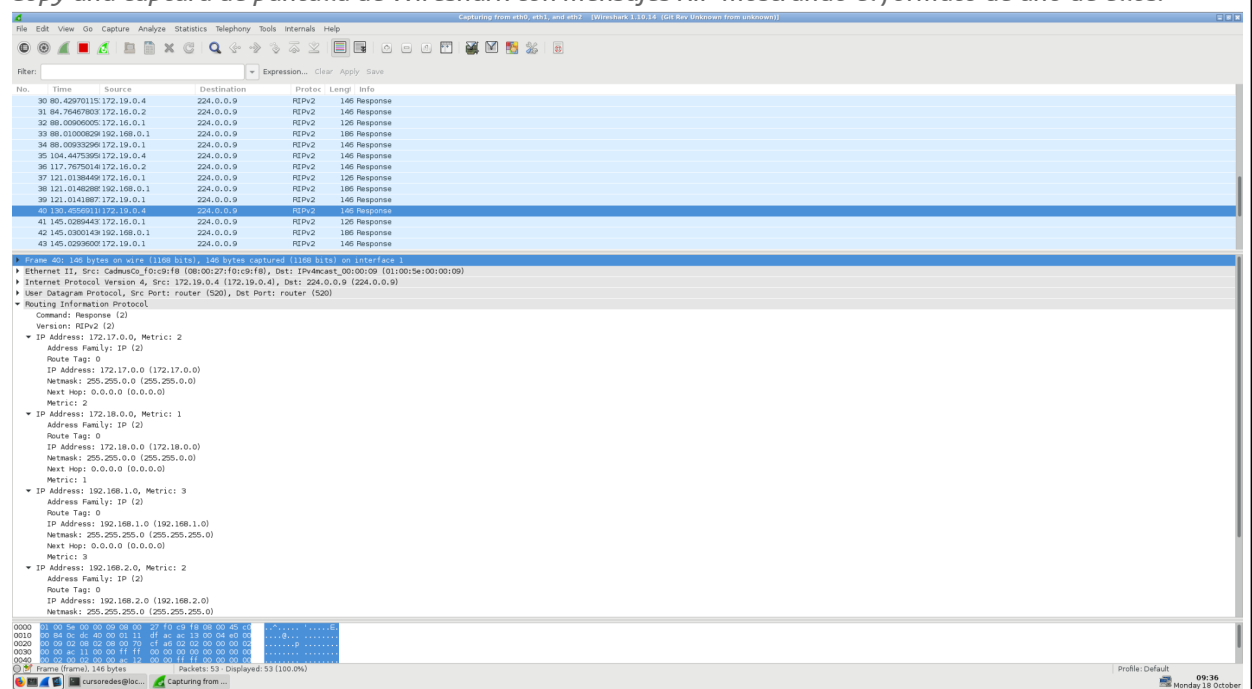
C>* 127.0.0.0/8 is directly connected, lo
R>* 172.16.0.0/16 [120/2] via 172.19.0.1, eth1, 00:04:23
R>* 172.17.0.0/16 [120/2] via 172.18.0.3, eth0, 00:04:40
C>* 172.18.0.0/16 is directly connected, eth0
C>* 172.19.0.0/16 is directly connected, eth1
R>* 192.168.0.0/24 [120/2] via 172.19.0.1, eth1, 00:04:23
R>* 192.168.1.0/24 [120/3] via 172.18.0.3, eth0, 00:04:32
R>* 192.168.2.0/24 [120/2] via 172.18.0.3, eth0, 00:04:40
C>* 192.168.3.0/24 is directly connected, eth2
[cursoredes@localhost ~]$ ip route
172.16.0.0/16 via 172.19.0.1 dev eth1 proto zebra metric 2
172.17.0.0/16 via 172.18.0.3 dev eth0 proto zebra metric 2
```

172.18.0.0/16 dev eth0 proto kernel scope link src 172.18.0.4
 172.19.0.0/16 dev eth1 proto kernel scope link src 172.19.0.4
 192.168.0.0/24 via 172.19.0.1 dev eth1 proto zebra metric 2
 192.168.1.0/24 via 172.18.0.3 dev eth0 proto zebra metric 3
 192.168.2.0/24 via 172.18.0.3 dev eth0 proto zebra metric 2
 192.168.3.0/24 dev eth2 proto kernel scope link src 192.168.3.4

Ejercicio 3. Con la herramienta wireshark, estudiar los mensajes RIP intercambiados, en particular:

- Encapsulado.
- Direcciones origen y destino.
- Campo de versión.
- Información para cada ruta: dirección de red, máscara de red, siguiente salto y distancia.

Copy una captura de pantalla de Wireshark con mensajes RIP mostrando el formato de uno de ellos.



Ejercicio 4. Eliminar el enlace entre Router1 y Router4 (por ejemplo, desactivando el interfaz eth1 en Router4). Comprobar que Router1 deja de recibir los anuncios de Router4 y que, pasados aproximadamente 3 minutos (valor de *timeout* por defecto para las rutas), ha reajustado su tabla.

Copia los comandos usados y su salida.

Router1:

`[cursoredes@localhost ~]$ sudo vtysh -c "show ip route"`

Codes: K - kernel route, C - connected, S - static, R - RIP,
 O - OSPF, I - IS-IS, B - BGP, A - Babel,
 > - selected route, * - FIB route

```
C>* 127.0.0.0/8 is directly connected, lo
C>* 172.16.0.0/16 is directly connected, eth0
R>* 172.17.0.0/16 [120/2] via 172.16.0.2, eth0, 00:03:07
C>* 172.19.0.0/16 is directly connected, eth1
C>* 192.168.0.0/24 is directly connected, eth2
R>* 192.168.1.0/24 [120/2] via 172.16.0.2, eth0, 00:16:55
```

```
[cursoredes@localhost ~]$ ip route
172.16.0.0/16 dev eth0 proto kernel scope link src 172.16.0.1
172.17.0.0/16 via 172.16.0.2 dev eth0 proto zebra metric 2
172.18.0.0/16 via 172.16.0.2 dev eth0 proto zebra metric 3
172.19.0.0/16 dev eth1 proto kernel scope link src 172.19.0.1
192.168.0.0/24 dev eth2 proto kernel scope link src 192.168.0.1
192.168.1.0/24 via 172.16.0.2 dev eth0 proto zebra metric 2
192.168.2.0/24 via 172.16.0.2 dev eth0 proto zebra metric 3
192.168.3.0/24 via 172.16.0.2 dev eth0 proto zebra metric 4
```

Como se ve, ya no aparece ninguna ruta recibida por la interfaz eth1 que no sea directa.

Ejercicio 5 (Opcional). Los servicios de Quagga pueden configurarse de forma interactiva mediante un terminal (telnet), de forma similar a los encaminadores comerciales. Configurar ripd vía VTY:

- Añadir "password asor" al fichero ripd.conf, desactivar el protocolo (no router rip) y comentar el resto de entradas. Una vez cambiado el fichero, reiniciar el servicio.
- Conectar al VTY de ripd y configurarlo. En cada comando se puede usar ? para mostrar la ayuda asociada.

```
$ telnet localhost ripd
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is Quagga (version 0.99.20.1)
Copyright © 1996-2005 Kunihiro Ishiguro, et al.
User Access Verification

Password: asor
localhost.localdomain> enable
localhost.localdomain# configure terminal
localhost.localdomain(config)# router rip
localhost.localdomain(config-router)# version 2
localhost.localdomain(config-router)# network eth0
localhost.localdomain(config-router)# write
Configuration saved to /etc/quagga/ripd.conf
localhost.localdomain(config-router)# exit
localhost.localdomain(config)# exit
localhost.localdomain# show running-config
Current configuration:
!
password asor
!
router rip
version 2
network eth0
!
line vty
!
end
localhost.localdomain# write
Configuration saved to /etc/quagga/ripd.conf
localhost.localdomain# exit
```

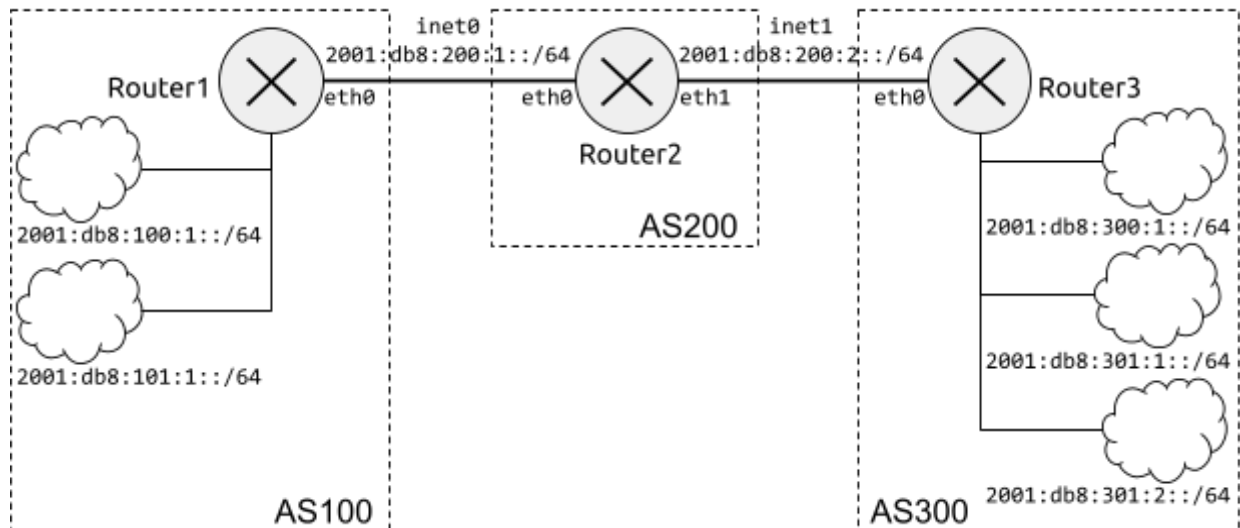
Nota: Para poder escribir la configuración en ripd.conf, el usuario quagga debe tener los permisos

adecuados sobre el fichero. Para cambiar el propietario del fichero, ejecutar el comando `chown quagga:quagga /etc/quagga/ripd.conf`.

Parte II. Protocolo exterior: BGP

Preparación del entorno

Configuraremos la topología de red con 3 AS, siendo uno de ellos el proveedor de los otros dos:



Nota: El prefijo `2001:db8::/32` está reservado para documentación y ejemplos (RFC 3849).

Crearemos esta topología (sin las redes internas de los AS) con la herramienta `vtopo1` y el siguiente fichero:

```
netprefix inet
machine 1 0 0
machine 2 0 0 1 1
machine 3 0 1
```

Para facilitar la configuración de las máquinas, la siguiente tabla muestra las direcciones de cada uno de los interfaces de los encaminadores:

Máquina virtual	Interfaz	Dirección de red	Dirección IP
Router1	eth0	2001:db8:200:1::/64	2001:db8:200:1::1
Router2	eth0	2001:db8:200:1::/64	2001:db8:200:1::2
	eth1	2001:db8:200:2::/64	2001:db8:200:2::2
Router3	eth0	2001:db8:200:2::/64	2001:db8:200:2::3

Configurar los encaminadores según se muestra en la figura anterior. Debe comprobarse la conectividad entre máquinas adyacentes.

Configuración del protocolo BGP

Ejercicio 6. Consultar la documentación de las clases de teoría para determinar el tipo de AS (*stub*, *multihomed* o *transit*) y los prefijos de red que debe anunciar. Recordar que el prefijo global de encaminamiento es de 48 bits y que los prefijos anunciados deben agregarse al máximo.

Número de AS	Tipo	Prefijos agregados
AS100	Stub	2001:db8:100::/47
AS200	Transit	
AS300	Stub	2001:db8:300::/47

Ejercicio 7. Configurar BGP en los encaminadores para que intercambien información:

- Crear un fichero `bgpd.conf` en `/etc/quagga` usando como referencia el que se muestra a continuación.
- Iniciar el servicio BGP (y Zebra) con `service bgpd start`.

Por ejemplo, el contenido del fichero `/etc/quagga/bgpd.conf` de Router1 en el AS 100 sería:

```
# Activar el encaminamiento BGP en el AS 100
router bgp 100
# Establecer el identificador de encaminador BGP
bgp router-id 0.0.0.1
# Añadir el encaminador BGP vecino en el AS 200
neighbor 2001:db8:200:1::2 remote-as 200
# Empezar a trabajar con direcciones IPv6
address-family ipv6
# Anunciar un prefijo de red agregado
network 2001:db8:100::/47
# Activar IPv6 en el encaminador BGP vecino
neighbor 2001:db8:200:1::2 activate
# Dejar de trabajar con direcciones IPv6
exit-address-family
```

Ejercicio 8. Consultar la tabla de encaminamiento de BGP y de Zebra en cada encaminador con el comando `vttysh` (`sudo vtysh -c "show ipv6 bgp"` y `sudo vtysh -c "show ipv6 route"`). Comprobar también la tabla de reenvío de IPv6 con el comando `ip` (`ip -6 route`).

Copia los comandos usados y su salida.

Router1:

```
[cursoredes@localhost ~]$ sudo vtysh -c "show ipv6 bgp"
```

```
BGP table version is 0, local router ID is 0.0.0.1
```

```
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, R Removed
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete
```

```
Network      Next Hop      Metric LocPrf Weight Path
*> 2001:db8:100::/47
                ::                0      32768 i
*> 2001:db8:300::/47
                2001:db8:200:1::2
                                0 200 300 i
```

```

Total number of prefixes 2
[cursoredes@localhost ~]$ sudo vtysh -c "show ipv6 route"
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv6, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

C>* ::1/128 is directly connected, lo
C>* 2001:db8:200:1::/64 is directly connected, eth0
B> 2001:db8:300::/47 [20/0] via fe80::a00:27ff:fedf:6548, eth0, 00:02:47
C>* fe80::/64 is directly connected, eth0
[cursoredes@localhost ~]$ ip -6 route
unreachable ::/96 dev lo metric 1024 error -113 pref medium
unreachable ::ffff:0.0.0.0/96 dev lo metric 1024 error -113 pref medium
2001:db8:200:1::/64 dev eth0 proto kernel metric 256 pref medium
unreachable 2002:a00::/24 dev lo metric 1024 error -113 pref medium
unreachable 2002:7f00::/24 dev lo metric 1024 error -113 pref medium
unreachable 2002:a9fe::/32 dev lo metric 1024 error -113 pref medium
unreachable 2002:ac10::/28 dev lo metric 1024 error -113 pref medium
unreachable 2002:c0a8::/32 dev lo metric 1024 error -113 pref medium
unreachable 2002:e000::/19 dev lo metric 1024 error -113 pref medium
unreachable 3ffe:ffff::/32 dev lo metric 1024 error -113 pref medium
fe80::/64 dev eth0 proto kernel metric 256 pref medium

Router2:
[cursoredes@localhost ~]$ sudo vtysh -c "show ipv6 bgp"
BGP table version is 0, local router ID is 0.0.0.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network      Next Hop      Metric LocPrf Weight Path
*> 2001:db8:100::/47
      2001:db8:200:1::1
          0          0 100 i
*> 2001:db8:300::/47
      2001:db8:200:2::3
          0          0 300 i

Total number of prefixes 2
[cursoredes@localhost ~]$ sudo vtysh -c "show ipv6 route"
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv6, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

C>* ::1/128 is directly connected, lo
B> 2001:db8:100::/47 [20/0] via fe80::a00:27ff:fedf:6548, eth0, 00:11:02
C>* 2001:db8:200:1::/64 is directly connected, eth0
C>* 2001:db8:200:2::/64 is directly connected, eth1
B>* 2001:db8:300::/47 [20/0] via fe80::a00:27ff:fedf:6548, eth1, 00:03:35
C * fe80::/64 is directly connected, eth1
C>* fe80::/64 is directly connected, eth0
[cursoredes@localhost ~]$ ip -6 route
unreachable ::/96 dev lo metric 1024 error -113 pref medium
unreachable ::ffff:0.0.0.0/96 dev lo metric 1024 error -113 pref medium
2001:db8:200:1::/64 dev eth0 proto kernel metric 256 pref medium
2001:db8:200:2::/64 dev eth1 proto kernel metric 256 pref medium

```

```

2001:db8:300::/47 via fe80::a00:27ff:fedf:6548 dev eth1 proto zebra metric 1024 pref medium
unreachable 2002:a00::/24 dev lo metric 1024 error -113 pref medium
unreachable 2002:7f00::/24 dev lo metric 1024 error -113 pref medium
unreachable 2002:a9fe::/32 dev lo metric 1024 error -113 pref medium
unreachable 2002:ac10::/28 dev lo metric 1024 error -113 pref medium
unreachable 2002:c0a8::/32 dev lo metric 1024 error -113 pref medium
unreachable 2002:e000::/19 dev lo metric 1024 error -113 pref medium
unreachable 3ffe:ffff::/32 dev lo metric 1024 error -113 pref medium
fe80::/64 dev eth0 proto kernel metric 256 pref medium
fe80::/64 dev eth1 proto kernel metric 256 pref medium

```

Router3:

```
[cursoredes@localhost ~]$ sudo vtysh -c "show ipv6 bgp"
```

BGP table version is 0, local router ID is 0.0.0.3

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, R Removed

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 2001:db8:100::/47	2001:db8:200:2::2	0	200	100	i
*> 2001:db8:300::/47	::	0	32768		i

Total number of prefixes 2

```
[cursoredes@localhost ~]$ sudo vtysh -c "show ipv6 route"
```

Codes: K - kernel route, C - connected, S - static, R - RIPng,

O - OSPFv6, I - IS-IS, B - BGP, A - Babel,

> - selected route, * - FIB route

```
C>* ::1/128 is directly connected, lo
```

```
B>* 2001:db8:100::/47 [20/0] via fe80::a00:27ff:fe17:e047, eth0, 00:01:43
```

```
C>* 2001:db8:200:2::/64 is directly connected, eth0
```

```
C>* fe80::/64 is directly connected, eth0
```

```
[cursoredes@localhost ~]$ ip -6 route
```

```

unreachable ::/96 dev lo metric 1024 error -113 pref medium
unreachable ::ffff:0.0.0.0/96 dev lo metric 1024 error -113 pref medium
2001:db8:100::/47 via fe80::a00:27ff:fe17:e047 dev eth0 proto zebra metric 1024 pref medium
2001:db8:200:2::/64 dev eth0 proto kernel metric 256 pref medium
unreachable 2002:a00::/24 dev lo metric 1024 error -113 pref medium
unreachable 2002:7f00::/24 dev lo metric 1024 error -113 pref medium
unreachable 2002:a9fe::/32 dev lo metric 1024 error -113 pref medium
unreachable 2002:ac10::/28 dev lo metric 1024 error -113 pref medium
unreachable 2002:c0a8::/32 dev lo metric 1024 error -113 pref medium
unreachable 2002:e000::/19 dev lo metric 1024 error -113 pref medium
unreachable 3ffe:ffff::/32 dev lo metric 1024 error -113 pref medium
fe80::/64 dev eth0 proto kernel metric 256 pref medium

```

Ejercicio 9. Con ayuda de la herramienta wireshark, estudiar los mensajes BGP intercambiados (OPEN, KEEPALIVE y UPDATE).

Copia una captura de pantalla de Wireshark con mensajes BGP mostrando el formato del mensaje UPDATE.

Wireshark 1.10.14 (Git Key Unknown from unknown)

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	2001:db8:2001::12	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation for 2001:db8:2001::1 from 08:00:27:df:65:48
2	0.000040008	2001:db8:2001::11	2001:db8:2001::12	ICMPv6	86	Neighbor Advertisement 2001:db8:2001::1 (sol, ovr) is at 08:00:27:df:65:48
3	0.000040008	2001:db8:2001::11	2001:db8:2001::12	TCP	174	LSQATE Message
4	0.040296495	2001:db8:2001::11	2001:db8:2001::12	TCP	86	60656 > bsp [ACK] Seq=1 Ack=89 Win=225 Len=0 TSval=1889990 TSecr=1883778
5	28.97846473	2001:db8:2001::12	2001:db8:2001::11	BGP	105	KEEPALIVE Message
6	28.97852869	2001:db8:2001::11	2001:db8:2001::12	TCP	86	60656 > bsp [ACK] Seq=1 Ack=108 Win=225 Len=0 TSval=1918928 TSecr=1912756
7	28.97918604	2001:db8:2001::11	2001:db8:2001::12	BGP	105	KEEPALIVE Message

Frame 3: 174 bytes on wire (1392 bits), 174 bytes captured (1392 bits) on interface 0

- Ethernet II, Src: CadmusCo_df:65:48 (08:00:27:df:65:48), Dst: CadmusCo_df:65:48 (08:00:27:df:65:48)
- Internet Protocol Version 6, Src: 2001:db8:2001::12 (2001:db8:2001::12), Dst: 2001:db8:2001::11 (2001:db8:2001::11)
- Transmission Control Protocol, Src Port: bsp (179), Dst Port: 60656 (60656), Seq: 1, Ack: 1, Len: 88
- Border Gateway Protocol - LSQATE Message
 - Marker: ffffffff
 - Length: 88
 - Types: LSQATE Message (2)
 - Unfeasible routes length: 0 bytes
 - Total path attribute length: 65 bytes
 - Path attributes
 - ORIGIN: IGP (4 bytes)
 - Flags: 0x40 (well-known, Transitive, Complete)
 - Type code: ORIGIN (1)
 - Length: 1 byte
 - Origin: IGP (0)
 - AS_PATH: 200 300 (16 bytes)
 - Flags: 0x50 (well-known, Transitive, Complete, Extended Length)
 - Type code: AS_PATH (2)
 - Length: 10 bytes
 - AS path: 200 300
 - AS path segment: 200 300
 - Path segment type: AS_SEQUENCE (2)
 - Path segment length: 2 ASes
 - Path segment value: 200 300
 - MP_REACH_NLRI (47 bytes)
 - Flags: 0x00 (optional, Non-transitive, Complete)
 - Type code: MP_REACH_NLRI (14)
 - Length: 44 bytes
 - Address family: IPv6 (2)
 - Subsequent address family identifier: Unicast (1)
 - Next hop network address (52 bytes)
 - Next hop: 2001:db8:2001::12 (16)
 - Next hop: fe80:a00:27ff:fe6f:6548 (16)
 - Subnetwork points of attachment: 0
 - Network layer reachability information (7 bytes)
 - 2001:db8:300::/47
 - MP Reach NLRI prefix length: 47
 - MP Reach NLRI prefix: 2001:db8:300::

0070 00 50 02 00 0a 02 02 00 00 00 c8 00 00 01 2c 80 .P.....
 0080 0e 2c 00 02 20 20 01 04 b8 02 00 00 01 00 00
 0090 00 00 00 00 00 02 fe 60 00 00 00 00 00 0a 00
 00a0 27 ff fe df 65 48 00 2f 20 01 04 b8 03 00eh/

Text Item (Text): 1 byte Packets: 39 - Displayed: 39 (100.0%)

Profile: Default 10:34 Monday 18 October