

# 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), RIPvng, 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 encaminamiento del *kernel* y la información 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

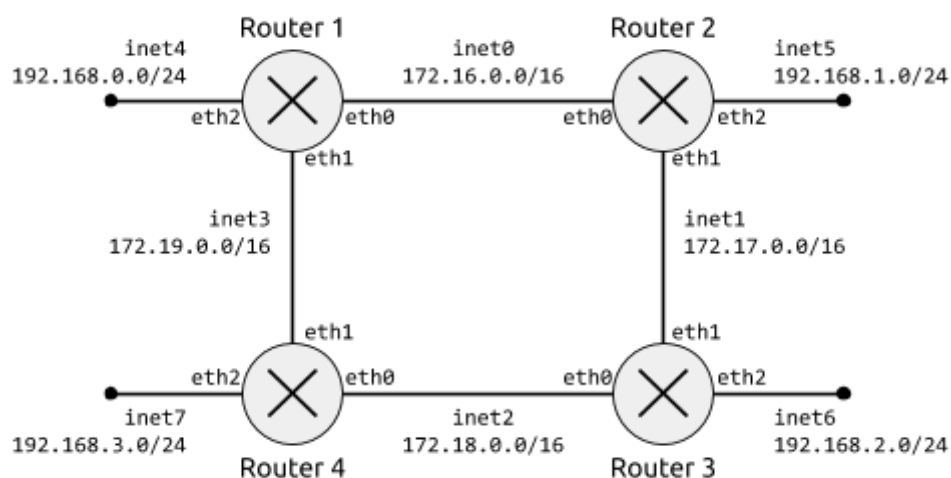
- Preparación del entorno

- Configuración del protocolo BGP

## Parte I. Protocolo interior: RIP

### Preparación del entorno

Configuraremos la topología de red que se muestra en la siguiente figura:



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

**Ejercicio 1.** 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 encaminamiento de cada encaminador es la correcta e incluye una entrada para cada una de las tres redes a las que está conectado.

```
En R1:
sudo ip link set eth0 up
sudo ip link set eth1 up
sudo ip link set eth2 up
sudo ip addr add 172.16.0.1/16 dev eth0
sudo ip addr add 172.19.0.1/16 dev eth1
sudo ip addr add 192.168.0.1/24 dev eth2
sudo sysctl -w net.ipv4.conf.all.forwarding=1
```

```
En R2:
sudo ip link set eth0 up
sudo ip link set eth1 up
sudo ip link set eth2 up
sudo ip addr add 172.16.0.2/16 dev eth0
sudo ip addr add 172.17.0.2/16 dev eth1
sudo ip addr add 192.168.1.2/24 dev eth2
sudo sysctl -w net.ipv4.conf.all.forwarding=1
```

```
En R3:
sudo ip link set eth0 up
sudo ip link set eth1 up
Sudo ip link set eth2 up
sudo ip addr add 172.18.0.3/16 dev eth0
sudo ip addr add 172.17.0.3/16 dev eth1
sudo ip addr add 192.168.2.3/24 dev eth2
sudo sysctl -w net.ipv4.conf.all.forwarding=1
```

```
En R4:
sudo ip link set eth0 up
sudo ip link set eth1 up
sudo ip link set eth2 up
sudo ip addr add 172.18.0.4/16 dev eth0
sudo ip addr add 172.19.0.4/16 dev eth1
sudo ip addr add 192.168.3.0/24 dev eth2
sudo sysctl -w net.ipv4.conf.all.forwarding=1
```

## Configuración del protocolo RIP

**Ejercicio 2.** 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`: (lo abro con `sudo gedit /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 3.** Consultar la tabla de encaminamiento de RIP y de Zebra en cada encaminador con el comando vtysh (sudo vtysh -c "show ip rip" y sudo vtysh -c "show ip route"). Comprobar también la tabla de encaminamiento de IPv4 con el comando ip (ip route).

En R1:

sudo vtysh -c "show ip rip"

```
[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	02:45
R(n)	172.18.0.0/16	172.19.0.4	2	172.19.0.4	0	02:52
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	02:45
R(n)	192.168.2.0/24	172.16.0.2	3	172.16.0.2	0	02:45
R(n)	192.168.3.0/24	172.19.0.4	2	172.19.0.4	0	02:52

En R2:

sudo vtysh -c "show ip rip"

```
[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	02:39
R(n)	172.19.0.0/16	172.16.0.1	2	172.16.0.1	0	02:58
R(n)	192.168.0.0/24	172.16.0.1	2	172.16.0.1	0	02:58
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	02:39
R(n)	192.168.3.0/24	172.16.0.1	3	172.16.0.1	0	02:58

En R3:

```
sudo vtysh -c "show ip rip"
```

---

```
[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	02:35
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	02:42
R(n)	192.168.0.0/24	172.17.0.2	3	172.17.0.2	0	02:35
R(n)	192.168.1.0/24	172.17.0.2	2	172.17.0.2	0	02:35
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	02:42

---

En R4:

```
sudo vtysh -c "show ip rip"
```

---

```
[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	02:52
R(n)	172.17.0.0/16	172.18.0.3	2	172.18.0.3	0	02:58
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	02:52
R(n)	192.168.1.0/24	172.19.0.1	3	172.19.0.1	0	02:52
R(n)	192.168.2.0/24	172.18.0.3	2	172.18.0.3	0	02:58
C(i)	192.168.3.0/24	0.0.0.0	1	self	0	

---

Ahora vamos a usar el otro comando mencionado...

En R1:

```
sudo vtysh -c "show ip route"
```

```
[cursoredes@localhost ~]$ sudo vtysh -c "show ip route"
Codes: K - kernel route, C - connected, S - static, R - RIP,
       0 - 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, 00:09:23
R>* 172.18.0.0/16 [120/2] via 172.19.0.4, eth1, 00:09:09
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, 00:09:23
R>* 192.168.2.0/24 [120/3] via 172.16.0.2, eth0, 00:09:15
R>* 192.168.3.0/24 [120/2] via 172.19.0.4, eth1, 00:09:09
```

---

En R2:

```
sudo vtysh -c "show ip route"
```

```
[cursoredes@localhost ~]$ sudo vtysh -c "show ip route"
Codes: K - kernel route, C - connected, S - static, R - RIP,
       0 - 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, 00:09:20
R>* 172.19.0.0/16 [120/2] via 172.16.0.1, eth0, 00:09:28
R>* 192.168.0.0/24 [120/2] via 172.16.0.1, eth0, 00:09:28
C>* 192.168.1.0/24 is directly connected, eth2
R>* 192.168.2.0/24 [120/2] via 172.17.0.3, eth1, 00:09:20
R>* 192.168.3.0/24 [120/3] via 172.16.0.1, eth0, 00:09:14
```

---



En R3:

```
sudo vtysh -c "show ip route"
```

```
[cursoredes@localhost ~]$ sudo vtysh -c "show ip route"
Codes: K - kernel route, C - connected, S - static, R - RIP,
       0 - 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, 00:09:24
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, 00:09:17
R>* 192.168.0.0/24 [120/3] via 172.17.0.2, eth1, 00:09:24
R>* 192.168.1.0/24 [120/2] via 172.17.0.2, eth1, 00:09:24
C>* 192.168.2.0/24 is directly connected, eth2
R>* 192.168.3.0/24 [120/2] via 172.18.0.4, eth0, 00:09:17
```

---

En R4:

```
sudo vtysh -c "show ip route"
```

```
[cursoredes@localhost ~]$ sudo vtysh -c "show ip route"
Codes: K - kernel route, C - connected, S - static, R - RIP,
       0 - 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, 00:09:21
R>* 172.17.0.0/16 [120/2] via 172.18.0.3, eth0, 00:09:21
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, 00:09:21
R>* 192.168.1.0/24 [120/3] via 172.19.0.1, eth1, 00:09:21
R>* 192.168.2.0/24 [120/2] via 172.18.0.3, eth0, 00:09:21
C>* 192.168.3.0/24 is directly connected, eth2
```

**Ejercicio 4.** 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.

Voy a ver los mensajes RIP que circulan por eth0.

#### RIPv2 REQUEST:

No.	Time	Source	Destination	Protoc	Lengj	Info
1	0.00000000	172.16.0.1	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.9 for any sources
2	0.00027737	172.16.0.1	224.0.0.9	RIPv2	66	Request
3	0.00131018	172.16.0.2	172.16.0.1	RIPv2	126	Response
4	2.92822860	172.16.0.1	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.9 for any sources

▶ Frame 2: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0

▶ Ethernet II, Src: 02:00:00:00:01:00 (02:00:00:00:01:00), Dst: IPv4mcast\_00:00:09 (01:00:5e:00:00:09)

▼ Internet Protocol Version 4, Src: 172.16.0.1 (172.16.0.1), Dst: 224.0.0.9 (224.0.0.9)

Version: 4  
Header length: 20 bytes

▶ Differentiated Services Field: 0xc0 (DSCP 0x30: Class Selector 6; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))

Total Length: 52  
Identification: 0xe390 (58256)

▶ Flags: 0x02 (Don't Fragment)  
Fragment offset: 0  
Time to live: 1  
Protocol: UDP (17)

▶ Header checksum: 0x094e [validation disabled]  
Source: 172.16.0.1 (172.16.0.1)  
Destination: 224.0.0.9 (224.0.0.9)

▶ User Datagram Protocol, Src Port: router (520), Dst Port: router (520)

▼ Routing Information Protocol

Command: Request (1)  
Version: RIPv2 (2)

▼ Address not specified, Metric: 16  
Address Family: Unspecified (0)  
Route Tag: 0  
Netmask: 0.0.0.0 (0.0.0.0)  
Next Hop: 0.0.0.0 (0.0.0.0)  
Metric: 16

#### RIPv2 RESPONSE:

No.	Time	Source	Destination	Protoc	Lengj	Info
1	0.00000000	172.16.0.1	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.9 for any sources
2	0.00027737	172.16.0.1	224.0.0.9	RIPv2	66	Request
3	0.00131018	172.16.0.2	172.16.0.1	RIPv2	126	Response
4	2.92822860	172.16.0.1	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.9 for any sources

▶ Frame 3: 126 bytes on wire (1008 bits), 126 bytes captured (1008 bits) on interface 0

▶ Ethernet II, Src: 02:00:00:00:02:00 (02:00:00:00:02:00), Dst: 02:00:00:00:01:00 (02:00:00:00:01:00)

▼ Internet Protocol Version 4, Src: 172.16.0.2 (172.16.0.2), Dst: 172.16.0.1 (172.16.0.1)

Version: 4  
Header length: 20 bytes

▶ Differentiated Services Field: 0xc0 (DSCP 0x30: Class Selector 6; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))

Total Length: 112  
Identification: 0xd39c (54172)

▶ Flags: 0x02 (Don't Fragment)  
Fragment offset: 0  
Time to live: 64  
Protocol: UDP (17)

▶ Header checksum: 0x0dfd [validation disabled]  
Source: 172.16.0.2 (172.16.0.2)  
Destination: 172.16.0.1 (172.16.0.1)

▶ User Datagram Protocol, Src Port: router (520), Dst Port: router (520)



```

▼ Routing Information Protocol
  Command: Response (2)
  Version: RIPv2 (2)
  ▼ IP Address: 172.17.0.0, Metric: 1
    Address Family: IP (2)
    Route Tag: 0
    IP Address: 172.17.0.0 (172.17.0.0)
    Netmask: 255.255.0.0 (255.255.0.0)
    Next Hop: 0.0.0.0 (0.0.0.0)
    Metric: 1
  ▼ IP Address: 172.18.0.0, Metric: 2
    Address Family: IP (2)
    Route Tag: 0
    IP Address: 172.18.0.0 (172.18.0.0)
    Netmask: 255.255.0.0 (255.255.0.0)
    Next Hop: 0.0.0.0 (0.0.0.0)
    Metric: 2
  ▼ IP Address: 192.168.1.0, Metric: 1
    Address Family: IP (2)
    Route Tag: 0
    IP Address: 192.168.1.0 (192.168.1.0)
    Netmask: 255.255.255.0 (255.255.255.0)
    Next Hop: 0.0.0.0 (0.0.0.0)
    Metric: 1
  ▼ IP Address: 192.168.2.0, Metric: 2
    Address Family: IP (2)
    Route Tag: 0
    IP Address: 192.168.2.0 (192.168.2.0)
    Netmask: 255.255.255.0 (255.255.255.0)
    Next Hop: 0.0.0.0 (0.0.0.0)
    Metric: 2

```

**Ejercicio 5.** 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.

**En R4:**

```
ip link set eth1 down
```

Puede observar que en la tabla de rutas de R1 ha desaparecido R4 (172.19.0.4)  
Muestro el antes y después de la tabla de rutas:

```

asorfemachine_1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

cursoredes@localhost:~
File Edit View Search Terminal Help

[cursoredes@localhost ~]$ sudo vtysh -c "show ip rip"
Codes: R - RIP, C - connected, S - Static, 0 - 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 02:56
R(n) 172.18.0.0/16 172.19.0.4        2 172.19.0.4         0 01:09
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 02:56
R(n) 192.168.2.0/24 172.16.0.2        3 172.16.0.2         0 02:56
R(n) 192.168.3.0/24 172.19.0.4        2 172.19.0.4         0 01:09

[cursoredes@localhost ~]$ sudo vtysh -c "show ip rip"
Codes: R - RIP, C - connected, S - Static, 0 - 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 02:49
R(n) 172.18.0.0/16 172.16.0.2        3 172.16.0.2         0 02:49
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 02:49
R(n) 192.168.2.0/24 172.16.0.2        3 172.16.0.2         0 02:49
R(n) 192.168.3.0/24 172.16.0.2        4 172.16.0.2         0 02:49

```

**Ejercicio 6 (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

**Ejercicio 7.** Configurar los encaminadores según se muestra en la figura anterior. Debe comprobarse la conectividad entre máquinas adyacentes.

```
R1:
ip link set eth0 up
ip addr add 2001:db8:200:1::1/64 dev eth0

R2:
ip link set eth0 up
ip link set eth1 up
ip addr add 2001:db8:200:1::2/64 dev eth0
ip addr add 2001:db8:200:2::2/64 dev eth1

R3:
ip link set eth0 up
ip addr add 2001:db8:200:2::3/64 dev eth0
```

## Configuración del protocolo BGP

**Ejercicio 8.** 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. Suponed que el RIR ha asignado a cada AS prefijos de longitud 48 y que los prefijos anunciados deben agregarse al máximo.

Número de AS	Tipo	Prefijos Agregados
AS100	Stub	2001:db8:100::/63
AS200	Transito (Conecta dos AS)	2001:db8:200::/62
AS300	Stub	2001:db8:300::/63

**Ejercicio 9.** 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. `sudo gedit /etc/quagga/bgpd.conf`
- 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:

**EN R1:**

```
# 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
```

---

**EN R2:**

```
router bgp 200
bgp router-id 0.0.0.2
neighbor 2001:db8:200:1::1 remote-as 100
neighbor 2001:db8:200:2::3 remote-as 300
address-family ipv6
neighbor 2001:db8:200:1::1 activate
neighbor 2001:db8:200:2::3 activate
exit-address-family
```

---

**EN R3:**

```
router bgp 300
bgp router-id 0.0.0.3
neighbor 2001:db8:200:2::2 remote-as 200
address-family ipv6
network 2001:db8:300::/47
neighbor 2001:db8:200:2::2 activate
Exit-address-family
```



**Ejercicio 10.** Consultar la tabla de encaminamiento de BGP y de Zebra en cada encaminador con el comando vtysh (sudo vtysh -c "show ipv6 bgp" y sudo vtysh -c "show ipv6 route"). Comprobar también la tabla de encaminamiento de IPv6 con el comando ip (ip -6 route).

**COMANDO:** `sudo vtysh -c "show ipv6 bgp"`

**R1:**

```
[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
```

**R2:**

```
[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
```

**R3:**

```
[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
```

COMANDO: **sudo vtysh -c "show ipv6 route"**

**R1:**

```
[cursoredes@localhost ~]$ sudo vtysh -c "show ipv6 route"
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       0 - 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::ff:fe00:200, eth0, 00:08:05
C>* fe80::/64 is directly connected, eth0
```

---

**R2:**

```
[cursoredes@localhost ~]$ sudo vtysh -c "show ipv6 route"
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       0 - 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::ff:fe00:100, eth0, 00:03:14
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::ff:fe00:300, eth1, 00:03:07
C>* fe80::/64 is directly connected, eth1
C>* fe80::/64 is directly connected, eth0
```

---

**R3:**

```
[cursoredes@localhost ~]$ sudo vtysh -c "show ipv6 route"
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       0 - 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::ff:fe00:201, eth0, 00:03:14
C>* 2001:db8:200:2::/64 is directly connected, eth0
C>* fe80::/64 is directly connected, eth0
```

**Ejercicio 11.** Con ayuda de la herramienta wireshark, estudiar los mensajes BGP intercambiados (OPEN, KEEPALIVE y UPDATE).

Solo he logrado capturar el mensaje KEEPALIVE:

No.	Time	Source	Destination	Protoc	Length	Info
1	0.00000000	2001:db8:200:1::2	2001:db8:200:1::1	BGP	105	KEEPALIVE Message
2	0.00004613	2001:db8:200:1::1	2001:db8:200:1::2	TCP	86	bgp > 51658 [ACK] Seq=1 Ack=20 Win=224 Len=0 TSval=3229815 TSecr=3227223
3	0.00047353	2001:db8:200:1::1	2001:db8:200:1::2	BGP	105	KEEPALIVE Message
4	0.00130539	2001:db8:200:1::2	2001:db8:200:1::1	TCP	86	51658 > bgp [ACK] Seq=20 Ack=20 Win=225 Len=0 TSval=3227224 TSecr=3229816
5	5.00048495	fe80::ff:fe00:100	2001:db8:200:1::2	ICMPv6	86	Neighbor Solicitation for 2001:db8:200:1::2 from 02:00:00:00:01:00
6	5.00159892	2001:db8:200:1::2	fe80::ff:fe00:100	ICMPv6	78	Neighbor Advertisement 2001:db8:200:1::2 (sol)
7	10.0091735	fe80::ff:fe00:200	fe80::ff:fe00:100	ICMPv6	86	Neighbor Solicitation for fe80::ff:fe00:100 from 02:00:00:00:02:00
▶ Frame 1: 105 bytes on wire (840 bits), 105 bytes captured (840 bits) on interface 0						
▶ Ethernet II, Src: 02:00:00:00:02:00 (02:00:00:00:02:00), Dst: 02:00:00:00:01:00 (02:00:00:00:01:00)						
▶ Internet Protocol Version 6, Src: 2001:db8:200:1::2 (2001:db8:200:1::2), Dst: 2001:db8:200:1::1 (2001:db8:200:1::1)						
▼ Transmission Control Protocol, Src Port: 51658 (51658), Dst Port: bgp (179), Seq: 1, Ack: 1, Len: 19						
Source port: 51658 (51658)						
Destination port: bgp (179)						
[Stream index: 0]						
Sequence number: 1 (relative sequence number)						
[Next sequence number: 20 (relative sequence number)]						
Acknowledgment number: 1 (relative ack number)						
Header length: 32 bytes						
Flags: 0x018 (PSH, ACK)						
Window size value: 225						
[Calculated window size: 225]						
[Window size scaling factor: -1 (unknown)]						
▶ Checksum: 0x767c [validation disabled]						
▶ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps						
▶ [SEQ/ACK analysis]						
▼ Border Gateway Protocol - KEEPALIVE Message						
Marker: ffffffffffffffffffffffffffffffff						
Length: 19						
Type: KEEPALIVE Message (4)						