

# 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 `curso` es `curso`.

## Contenidos

### Parte I. Protocolo interior: RIP

- Preparación del entorno

- Configuración del protocolo RIP

### Parte II. Protocolo exterior: BGP

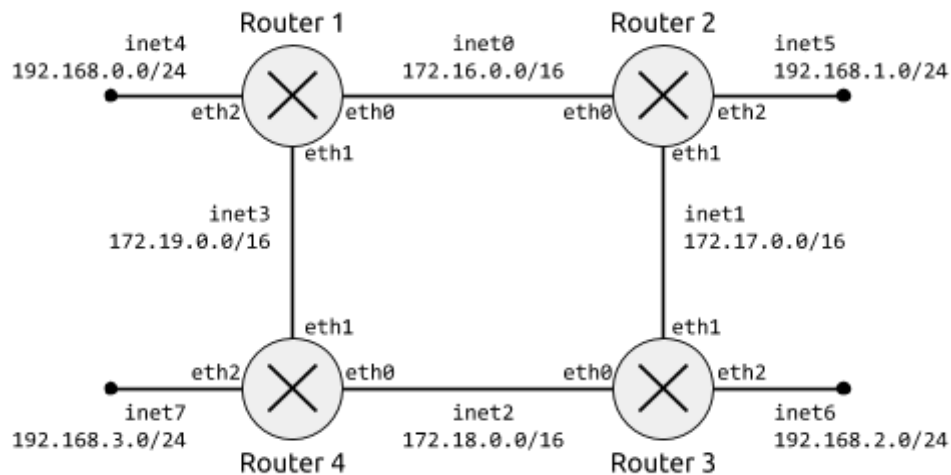
- Preparación del entorno

- 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.

## 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`:

```
# 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 `vttysh` (`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)`.

*Copia los comandos usados y su salida.*

### [VM1]

**Comando:** `vttysh -c "show ip rip"`

**Salida:**

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 02:40
R(n) 172.18.0.0/16	172.19.0.4	2 172.19.0.4	0 02:42
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 02:40
R(n) 192.168.2.0/24	172.16.0.2	3 172.16.0.2	0 02:40
R(n) 192.168.3.0/24	172.19.0.4	2 172.19.0.4	0 02:42

**Comando:** `vttysh -c "show ip route"`

**Salida:**

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, 00:00:59
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, 00:00:59
R>* 192.168.2.0/24 [120/3] via 172.16.0.2, eth0, 00:00:52
```

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

**Comando:** ip route

**Salida:**

```
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
```

Vemos que en la tabla de encaminamiento IPv4 también se han añadido las rutas con protocolo zebra

**[VM2]**

**Comando:** vtysh -c "show ip rip"

**Salida:**

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 02:32
R(n) 172.19.0.0/16	172.16.0.1	2 172.16.0.1	0 02:47
R(n) 192.168.0.0/24	172.16.0.1	2 172.16.0.1	0 02:47
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 02:32
R(n) 192.168.3.0/24	172.17.0.3	3 172.17.0.3	0 02:32

**Comando:** vtysh -c "show ip route"

**Salida:**

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, 00:08:32
R>* 172.19.0.0/16 [120/2] via 172.16.0.1, eth0, 00:08:41
R>* 192.168.0.0/24 [120/2] via 172.16.0.1, eth0, 00:08:41
C>* 192.168.1.0/24 is directly connected, eth2
R>* 192.168.2.0/24 [120/2] via 172.17.0.3, eth1, 00:08:32
R>* 192.168.3.0/24 [120/3] via 172.17.0.3, eth1, 00:08:30
```

**Comando:** ip route

**Salida:**

```
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 192.168.1.2
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
```

### [VM3]

**Comando:** vtysh -c "show ip rip"

#### **Salida:**

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	02:47
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	02:46
R(n) 192.168.0.0/24	172.17.0.2	3	172.17.0.2	0	02:47
R(n) 192.168.1.0/24	172.17.0.2	2	172.17.0.2	0	02:47
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	02:46

**Comando:** vtysh -c "show ip route"

#### **Salida:**

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
K>* 169.254.0.0/16 is directly connected, eth1
R>* 172.16.0.0/16 [120/2] via 172.17.0.2, eth1, 00:08:34
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, 00:08:31
R>* 192.168.0.0/24 [120/3] via 172.17.0.2, eth1, 00:08:34
R>* 192.168.1.0/24 [120/2] via 172.17.0.2, eth1, 00:08:34
C>* 192.168.2.0/24 is directly connected, eth2
R>* 192.168.3.0/24 [120/2] via 172.18.0.4, eth0, 00:08:31
```

**Comando:** ip route

#### **Salida:**

```
169.254.0.0/16 dev eth0 scope link metric 1002
169.254.0.0/16 dev eth1 scope link metric 1003
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
```

**[VM4]****Comando:** vtysh -c "show ip rip"**Salida:**

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	02:44
R(n) 172.17.0.0/16	172.18.0.3	2	172.18.0.3	0	02:29
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	02:44
R(n) 192.168.1.0/24	172.18.0.3	3	172.18.0.3	0	02:29
R(n) 192.168.2.0/24	172.18.0.3	2	172.18.0.3	0	02:29
C(i) 192.168.3.0/24	0.0.0.0	1	self	0	

**Comando:** vtysh -c "show ip route"**Salida:**

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

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

&gt; - 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, 00:08:36
R>* 172.17.0.0/16 [120/2] via 172.18.0.3, eth0, 00:08:36
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, 00:08:36
R>* 192.168.1.0/24 [120/3] via 172.18.0.3, eth0, 00:08:36
R>* 192.168.2.0/24 [120/2] via 172.18.0.3, eth0, 00:08:36
C>* 192.168.3.0/24 is directly connected, eth2
```

**Comando:** ip route**Salida:**

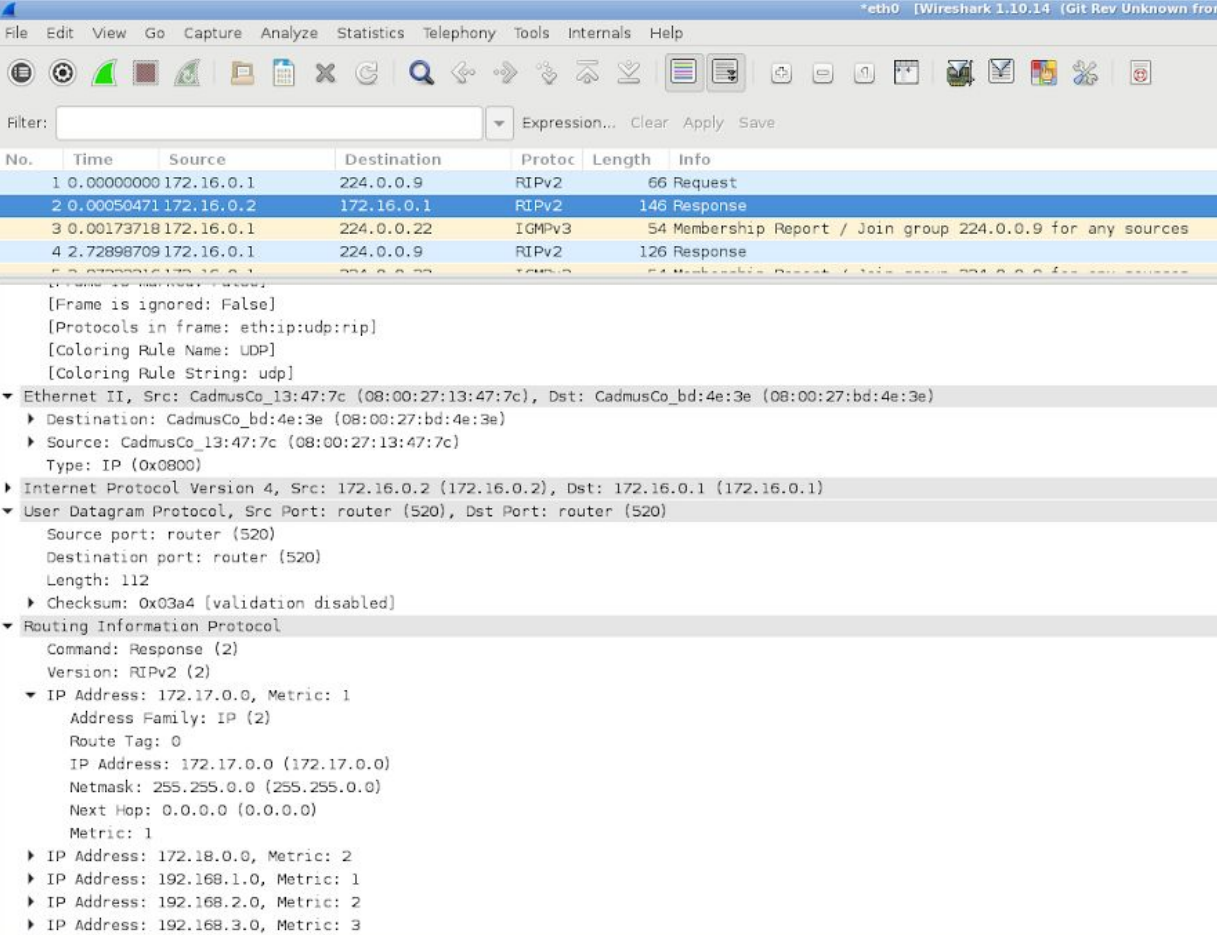
```
172.16.0.0/16 via 172.19.0.1 dev eth1 proto zebra metric 2
172.17.0.0/16 via 172.18.0.3 dev eth0 proto zebra metric 2
172.18.0.0/16 dev eth0 proto kernel scope link src 172.18.0.4
172.19.0.0/16 dev eth1 proto kernel scope link src 172.19.0.4
192.168.0.0/24 via 172.19.0.1 dev eth1 proto zebra metric 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 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.

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

### Captura en eth0 de MV1



Filter:  Expression... Clear Apply Save

No.	Time	Source	Destination	Protoc	Length	Info
1	0.00000000	172.16.0.1	224.0.0.9	RIPv2	66	Request
2	0.00050471	172.16.0.2	172.16.0.1	RIPv2	146	Response
3	0.00173718	172.16.0.1	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.9 for any sources
4	2.72898709	172.16.0.1	224.0.0.9	RIPv2	126	Response

▼ Ethernet II, Src: CadmusCo\_13:47:7c (08:00:27:13:47:7c), Dst: CadmusCo\_bd:4e:3e (08:00:27:bd:4e:3e)  
 ▶ Destination: CadmusCo\_bd:4e:3e (08:00:27:bd:4e:3e)  
 ▶ Source: CadmusCo\_13:47:7c (08:00:27:13:47:7c)  
 Type: IP (0x0800)

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

▼ User Datagram Protocol, Src Port: router (520), Dst Port: router (520)  
 Source port: router (520)  
 Destination port: router (520)  
 Length: 112  
 ▶ Checksum: 0x03a4 [validation disabled]

▼ 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  
 ▶ IP Address: 192.168.1.0, Metric: 1  
 ▶ IP Address: 192.168.2.0, Metric: 2  
 ▶ IP Address: 192.168.3.0, Metric: 3

- Encapsulado: Se encapsula en UDP en el puerto 520
- Es un request periódico con el vector de distancias
- Dirección origen: 172.16.0.2 (VM2 por la red0)
- Dirección destino: 172.16.0.1 (VM1 por la red0)
- Version: RIPv2
- Info ruta (p.e de la primera):
  - Dirección de red: 172.17.0.0
  - Mascara de red: 255.255.0.0
  - Siguiente salto 0.0.0.0 (Soy yo mismo)
  - Distancia: 1

**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.

Copia los comandos usados y su salida.

**[VM1]**

**Comando:** `ip link set dev eth1 down`

**Comando:** `vttysh -c "show ip rip"`

**Salida:**

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	02:58
R(n) 172.18.0.0/16	172.16.0.2	3 172.16.0.2	0	02:58
R(n) 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	02:58
R(n) 192.168.2.0/24	172.16.0.2	3 172.16.0.2	0	02:58
R(n) 192.168.3.0/24	172.16.0.2	4 172.16.0.2	0	02:58

Observamos que tras un rato para ir a la red 172.19.0.0/16 tiene que dar 4 saltos (pasar por todos los routers cuando antes podía ir via eth1 yendo directamente a VM4

**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.

## 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::/47 (Tiene la 2001:db8:100:1::/64 y 2001:db8:101:1::/64)
AS200	Transito	
AS300	Stub	2001:db8:300::/47 (Tiene la 2001:db8:300:1::/64, la 2001:db8:301:1::/64 y la 2001:db8:301:2::/64)

**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.
- 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:

```
[VM1]
# 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

[VM2]

# Activar el encaminamiento BGP en el AS 200
router bgp 200
# Establecer el identificador de encaminador BGP
bgp router-id 0.0.0.2
# Añadir el encaminador BGP vecino en el AS 200
neighbor 2001:db8:200:1::1 remote-as 100
neighbor 2001:db8:200:2::3 remote-as 300
# Empezar a trabajar con direcciones IPv6
address-family ipv6
# Activar IPv6 en el encaminador BGP vecino
neighbor 2001:db8:200:1::1 activate
neighbor 2001:db8:200:2::3 activate
# Dejar de trabajar con direcciones IPv6
exit-address-family
```

```
[VM3]
# Activar el encaminamiento BGP en el AS 300
router bgp 300
# Establecer el identificador de encaminador BGP
bgp router-id 0.0.0.3
# Añadir el encaminador BGP vecino en el AS 200
neighbor 2001:db8:200:2::2 remote-as 200
# Empezar a trabajar con direcciones IPv6
address-family ipv6
# Anunciar un prefijo de red agregado
network 2001:db8:300::/47
# Activar IPv6 en el encaminador BGP vecino
neighbor 2001:db8:200:2::2 activate
# Dejar de trabajar con direcciones IPv6
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).

*Copia los comandos usados y su salida.*

**[VM1]**

**Comando:** vtysh -c "show ipv6 bgp"

**Salida:**

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

**Comando:** vtysh -c "show ipv6 route"

**Salida:**

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:fe13:477c, eth0, 00:01:16

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

**Comando:** ip -6 route

**Salida:**

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:db8:300::/47 via fe80::a00:27ff:fe13:477c dev eth0 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
```

## [VM2]

**Comando:** vtysh -c "show ipv6 bgp"

### Salida:

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

**Comando:** vtysh -c "show ipv6 route"

### Salida:

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:febd:4e3e, eth0, 00:01:47
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:fe18:ce44, eth1, 00:01:27
C * fe80::/64 is directly connected, eth1
C>* fe80::/64 is directly connected, eth0
```

**Comando:** ip -6 route

### Salida:

```
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:febd:4e3e dev eth0 proto zebra metric 1024 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:fe18:ce44 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

```

### [VM3]

**Comando:** vtysh -c "show ipv6 bgp"

#### **Salida:**

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:** vtysh -c "show ipv6 route"

#### **Salida:**

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:fe30:a3fe, eth0, 00:01:32

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

C>\* fd00:0:0:a::/64 is directly connected, eth0

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

**Comando:** ip -6 route

#### **Salida:**

```

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:fe30:a3fe 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
fd00:0:0:a::/64 dev eth0 proto kernel metric 256 pref medium
fe80::/64 dev eth0 proto kernel metric 256 pref medium

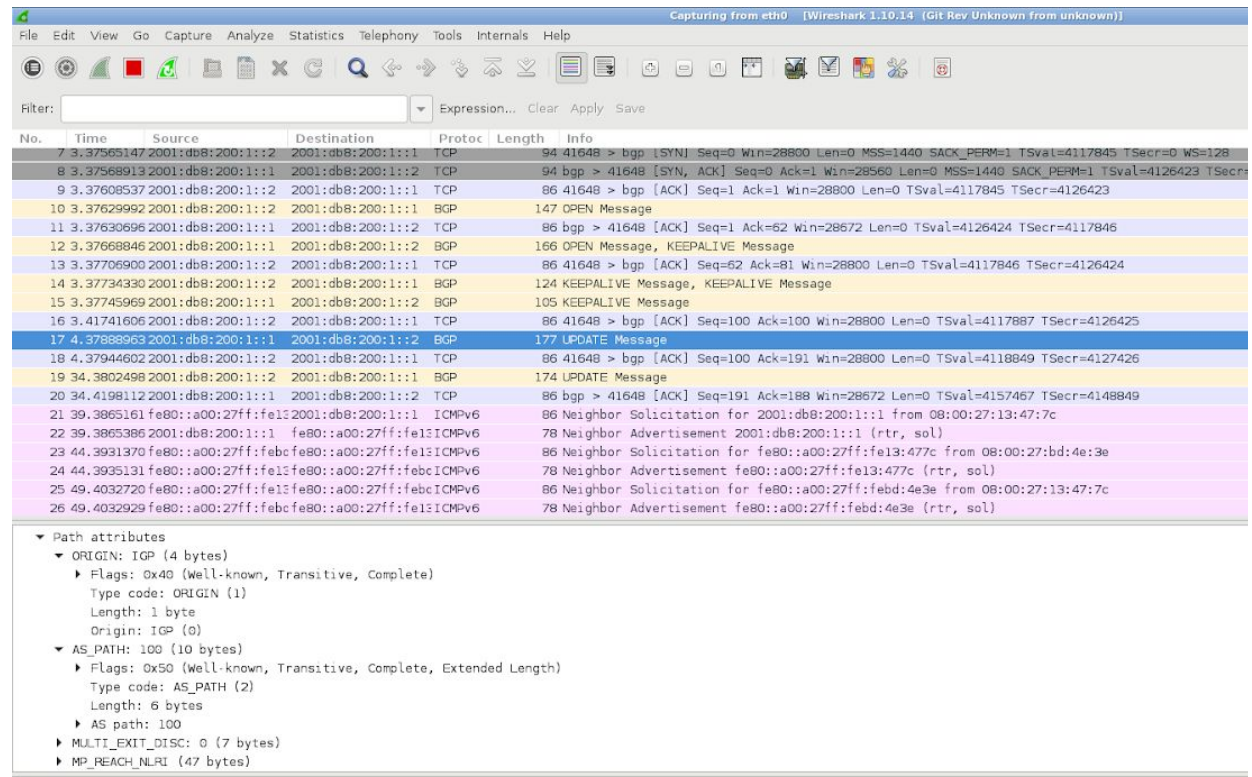
```

**Ejercicio 11.** 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.

Para forzar el open se ha hecho:

```
service bgpd stop  
service bgpd start
```



No.	Time	Source	Destination	Protoc	Length	Info
7	3.37565147	2001:db8:200:1::2	2001:db8:200:1::1	TCP	94	41648 > bgp [SYN] Seq=0 Win=28800 Len=0 MSS=1440 SACK_PERM=1 TSval=4117845 TSecr=0 WS=128
8	3.37568913	2001:db8:200:1::1	2001:db8:200:1::2	TCP	94	bgp > 41648 [SYN, ACK] Seq=0 Ack=1 Win=28560 Len=0 MSS=1440 SACK_PERM=1 TSval=4126423 TSecr=
9	3.37608537	2001:db8:200:1::2	2001:db8:200:1::1	TCP	86	41648 > bgp [ACK] Seq=1 Ack=1 Win=28800 Len=0 TSval=4117845 TSecr=4126423
10	3.37629992	2001:db8:200:1::2	2001:db8:200:1::1	BGP	147	OPEN Message
11	3.37630696	2001:db8:200:1::1	2001:db8:200:1::2	TCP	86	bgp > 41648 [ACK] Seq=1 Ack=62 Win=28672 Len=0 TSval=4126424 TSecr=4117846
12	3.37668846	2001:db8:200:1::1	2001:db8:200:1::2	BGP	166	OPEN Message, KEEPALIVE Message
13	3.37706900	2001:db8:200:1::2	2001:db8:200:1::1	TCP	86	41648 > bgp [ACK] Seq=62 Ack=81 Win=28800 Len=0 TSval=4117846 TSecr=4126424
14	3.37734330	2001:db8:200:1::2	2001:db8:200:1::1	BGP	124	KEEPALIVE Message, KEEPALIVE Message
15	3.37745969	2001:db8:200:1::1	2001:db8:200:1::2	BGP	105	KEEPALIVE Message
16	3.41741606	2001:db8:200:1::2	2001:db8:200:1::1	TCP	86	41648 > bgp [ACK] Seq=100 Ack=100 Win=28800 Len=0 TSval=4117887 TSecr=4126425
17	4.37888963	2001:db8:200:1::1	2001:db8:200:1::2	BGP	177	UPDATE Message
18	4.37944602	2001:db8:200:1::2	2001:db8:200:1::1	TCP	86	41648 > bgp [ACK] Seq=100 Ack=191 Win=28800 Len=0 TSval=4118849 TSecr=4127426
19	34.3802498	2001:db8:200:1::2	2001:db8:200:1::1	BGP	174	UPDATE Message
20	34.4198112	2001:db8:200:1::1	2001:db8:200:1::2	TCP	86	bgp > 41648 [ACK] Seq=191 Ack=188 Win=28672 Len=0 TSval=4157467 TSecr=4148849
21	39.3865161	fe80::a00:27ff:fe13:2001:db8:200:1::1	fe80::a00:27ff:fe13:2001:db8:200:1::1	ICMPv6	86	Neighbor Solicitation for 2001:db8:200:1::1 from 08:00:27:13:47:7c
22	39.3865386	2001:db8:200:1::1	fe80::a00:27ff:fe13:2001:db8:200:1::1	ICMPv6	78	Neighbor Advertisement 2001:db8:200:1::1 (rtr, sol)
23	44.3931370	fe80::a00:27ff:febc:fe80::a00:27ff:fe13:2001:db8:200:1::1	fe80::a00:27ff:febc:fe80::a00:27ff:fe13:2001:db8:200:1::1	ICMPv6	86	Neighbor Solicitation for fe80::a00:27ff:fe13:477c from 08:00:27:bd:4e:3e
24	44.3935131	fe80::a00:27ff:fe13:fe80::a00:27ff:febc:2001:db8:200:1::1	fe80::a00:27ff:fe13:fe80::a00:27ff:febc:2001:db8:200:1::1	ICMPv6	78	Neighbor Advertisement fe80::a00:27ff:fe13:477c (rtr, sol)
25	49.4032720	fe80::a00:27ff:fe13:fe80::a00:27ff:febc:2001:db8:200:1::1	fe80::a00:27ff:fe13:fe80::a00:27ff:febc:2001:db8:200:1::1	ICMPv6	86	Neighbor Solicitation for fe80::a00:27ff:febd:4e3e from 08:00:27:13:47:7c
26	49.4032929	fe80::a00:27ff:febc:fe80::a00:27ff:fe13:2001:db8:200:1::1	fe80::a00:27ff:febc:fe80::a00:27ff:fe13:2001:db8:200:1::1	ICMPv6	78	Neighbor Advertisement fe80::a00:27ff:febd:4e3e (rtr, sol)

▼ Path attributes

- ▼ ORIGIN: IGP (4 bytes)
  - Flags: 0x40 (Well-known, Transitive, Complete)
  - Type code: ORIGIN (1)
  - Length: 1 byte
  - Origin: IGP (0)
- ▼ AS\_PATH: 100 (10 bytes)
  - Flags: 0x50 (Well-known, Transitive, Complete, Extended Length)
  - Type code: AS\_PATH (2)
  - Length: 6 bytes
  - AS path: 100
- MULTI\_EXIT\_DISC: 0 (7 bytes)
- MP\_REACH\_NLRI (47 bytes)