

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

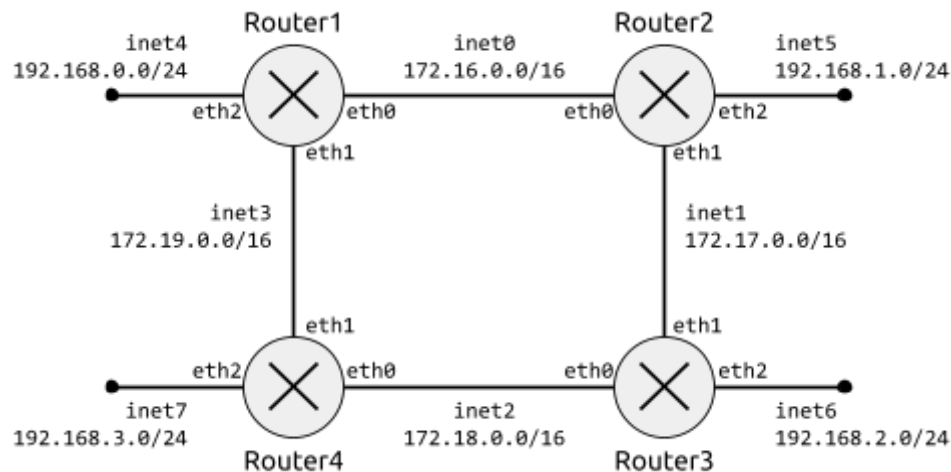
- Preparación del entorno

- 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`).

VM1:

`$ 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 02:46
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 02:46
R(n) 192.168.2.0/24	172.16.0.2	3 172.16.0.2	0 02:46
R(n) 192.168.3.0/24	172.19.0.4	2 172.19.0.4	0 02:57

`$ 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, 00:00:45
R>* 172.18.0.0/16 [120/2] via 172.19.0.4, eth1, 00:00:26
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, 00:00:45
R>* 192.168.2.0/24 [120/3] via 172.16.0.2, eth0, 00:00:38
R>* 192.168.3.0/24 [120/2] via 172.19.0.4, eth1, 00:00:26
```

`$ 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

```

VM2:

```
$ 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	02:53
R(n) 172.19.0.0/16	172.16.0.1	2	172.16.0.1	0	02:55
C(i) 192.168.0.0/24	0.0.0.0	1	self	0	
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	02:53
R(n) 192.168.3.0/24	172.17.0.3	3	172.17.0.3	0	02:53

```
$ 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

```

K>* 0.0.0.0/0 via 192.168.0.3, eth0
C>* 127.0.0.0/8 is directly connected, lo
K>* 169.254.0.0/16 is directly connected, eth0
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, 00:02:18
R>* 172.19.0.0/16 [120/2] via 172.16.0.1, eth0, 00:02:26
C>* 192.168.0.0/24 is directly connected, eth0
C>* 192.168.1.0/24 is directly connected, eth2
R>* 192.168.2.0/24 [120/2] via 172.17.0.3, eth1, 00:02:18
R>* 192.168.3.0/24 [120/3] via 172.17.0.3, eth1, 00:02:07

```

```
$ ip route
```

```
default via 192.168.0.3 dev eth0
```

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

```
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 dev eth0 proto kernel scope link src 192.168.0.2
```

```
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;

```
$ 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	02:57
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	02:37
R(n) 192.168.0.0/24	172.17.0.2	2 172.17.0.2	0	02:57
R(n) 192.168.1.0/24	172.17.0.2	2 172.17.0.2	0	02:57
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	02:37

\$ 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
K>* 169.254.0.0/16 is directly connected, eth1
R>* 172.16.0.0/16 [120/2] via 172.17.0.2, eth1, 00:03:28
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, 00:03:16
R>* 192.168.0.0/24 [120/2] via 172.17.0.2, eth1, 00:03:28
R>* 192.168.1.0/24 [120/2] via 172.17.0.2, eth1, 00:03:28
C>* 192.168.2.0/24 is directly connected, eth2
R>* 192.168.3.0/24 [120/2] via 172.18.0.4, eth0, 00:03:16

\$ ip route

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 2
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:

\$ 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	172.18.0.3	2 172.18.0.3	0	02:56
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	02:41
R(n) 192.168.1.0/24	172.18.0.3	3 172.18.0.3	0	02:56
R(n) 192.168.2.0/24	172.18.0.3	2 172.18.0.3	0	02:56

```
C(i) 192.168.3.0/24 0.0.0.0 1 self 0
```

```
$ 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

```
K>* 0.0.0.0/0 via 172.16.0.3, eth0
```

```
C>* 127.0.0.0/8 is directly connected, lo
```

```
K>* 169.254.0.0/16 is directly connected, eth0
```

```
C>* 172.16.0.0/16 is directly connected, eth0
```

```
R>* 172.17.0.0/16 [120/2] via 172.18.0.3, eth0, 00:04:55
```

```
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, 00:04:55
```

```
R>* 192.168.1.0/24 [120/3] via 172.18.0.3, eth0, 00:04:55
```

```
R>* 192.168.2.0/24 [120/2] via 172.18.0.3, eth0, 00:04:55
```

```
C>* 192.168.3.0/24 is directly connected, eth2
```

```
$ ip route
```

```
default via 172.16.0.3 dev eth0
```

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

```
172.16.0.0/16 dev eth0 proto kernel scope link src 172.16.0.4
```

```
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.

Copia una captura de pantalla de Wireshark con mensajes RIP que muestre el formato de un mensaje Response.

No.	Time	Source	Destination	Protoc	Length	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	0.00025073	172.16.0.1	224.0.0.9	RIPv2	66	Request
3	0.00090792	172.16.0.2	172.16.0.1	RIPv2	166	Response
4	0.00278308	172.16.0.1	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.9 for any sources
Frame 3: 166 bytes on wire (1328 bits), 166 bytes captured (1328 bits) on interface 0						
Ethernet II, Src: CadmusCo_a2:06:d2 (08:00:27:a2:06:d2), Dst: CadmusCo_b9:6d:34 (08:00:27:b9:6d:34)						
Destination: CadmusCo_b9:6d:34 (08:00:27:b9:6d:34)						
Source: CadmusCo_a2:06:d2 (08:00:27:a2:06:d2)						
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: 132						
Checksum: 0x43cf [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: 192.168.0.0, Metric: 1						
IP Address: 192.168.1.0, Metric: 1						
IP Address: 192.168.2.0, Metric: 2						
IP Address: 192.168.3.0, Metric: 3						

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.

VM4:

`$ ip link set eth1 down`

VM1:

`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	02:33
R(n) 172.18.0.0/16	172.19.0.4	16	172.19.0.4	0	01:50
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	02:33
R(n) 192.168.2.0/24	172.16.0.2	3	172.16.0.2	0	02:33
R(n) 192.168.3.0/24	172.19.0.4	16	172.19.0.4	0	01:50

Ahora necesita más saltos para llegar a las redes del router 4.

Ejercicio 5. Los servicios de Quagga pueden configurarse de forma interactiva mediante un interfaz VTY (Virtual Teletype), de forma similar a los encaminadores comerciales. Para activar el interfaz VTY, hay

que añadir el comando password al fichero de configuración del servicio deseado. 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 interfaz VTY de ripd con telnet y configurarlo. Teclea '?' para mostrar la ayuda asociada.

Ejemplo de sesión:

```
$ telnet 127.0.0.1 ripd
Trying 127.0.0.1...
Connected to 127.0.0.1.
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> ?
  echo      Echo a message back to the vty
  enable    Turn on privileged mode command
  exit      Exit current mode and down to previous mode
  help      Description of the interactive help system
  list      Print command list
  quit      Exit current mode and down to previous mode
  show      Show running system information
  terminal   Set terminal line parameters
  who       Display who is on vty
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)# 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
Connection closed by foreign host.
$ _
```

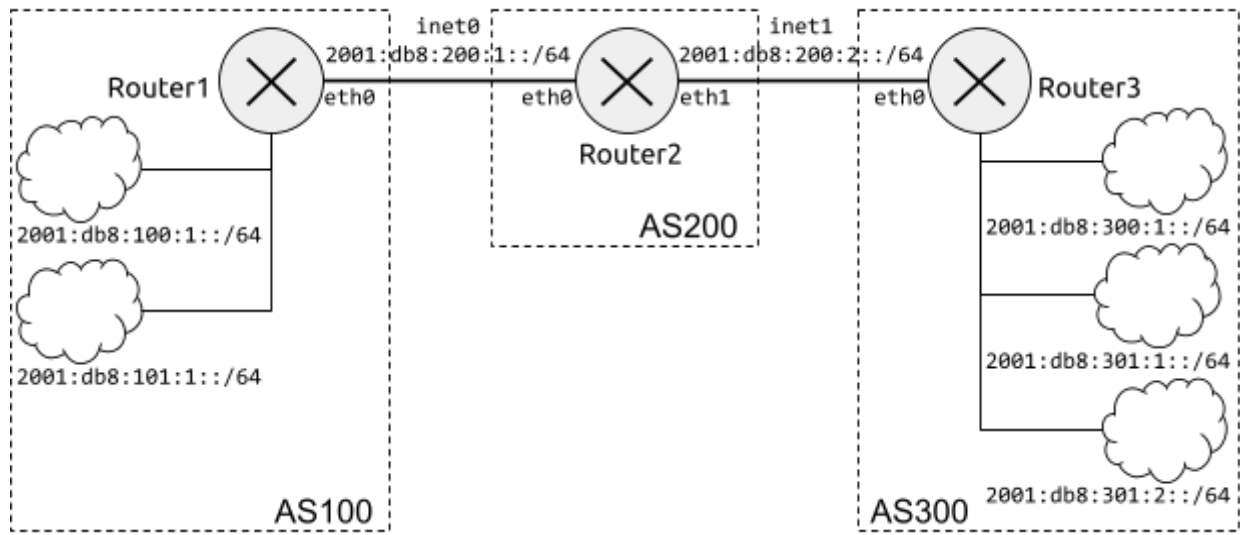
Nota: Para poder escribir la configuración en ripd.conf, el usuario quagga debe tener los permisos

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:db8:100:1::/64, 2001:db8:101:1::/64
AS200	Transit	
AS300	Stub	2001:db8:300:1::/64, 2001:db8:301:1::/64

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
```

```
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 100
```

```

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
# Anunciar un prefijo de red agregado
network 2001:db8:200::/47
# 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 100
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 8. 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 reenvío de IPv6 con el comando ip (ip -6 route).

```

VM1:
$ 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:db8:200::/47
           2001:db8:200:1::2
                   0      0 200 i
*> 2001:db8:300::/47
           2001:db8:200:1::2
                   0 200 300 i

```

Total number of prefixes 3

```
$ 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:db8:200::/47 [20/0] via fe80::a00:27ff:fea2:6d2, eth0, 00:01:25
```

```
K * 2001:db8:200:1::/64 is directly connected, eth0
```

```
C>* 2001:db8:200:1::/64 is directly connected, eth0
```

```
B>* 2001:db8:300::/47 [20/0] via fe80::a00:27ff:fea2:6d2, eth0, 00:00:55
```

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

```
$ 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:db8:200:1::/64 dev eth0 metric 1024 pref medium
```

```
2001:db8:200::/47 via fe80::a00:27ff:fea2:6d2 dev eth0 proto zebra metric 1024 pref medium
```

```
2001:db8:300::/47 via fe80::a00:27ff:fea2:6d2 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:

```
$ 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:db8:200::/47
```

```
::
```

```
0
```

```
32768 i
```

```
*> 2001:db8:300::/47
```

```
2001:db8:200:2::3
```

```
0
```

```
0 300 i
```

Total number of prefixes 3

```
$ 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:feb9:6d34, eth0, 00:03:12
```

```
K * 2001:db8:200:1::/64 is directly connected, eth0
```

```

C>* 2001:db8:200:1::/64 is directly connected, eth0
K * 2001:db8:200:2::/64 is directly connected, eth1
C>* 2001:db8:200:2::/64 is directly connected, eth1
B>* 2001:db8:300::/47 [20/0] via fe80::a00:27ff:fe48:35c1, eth1, 00:02:54
C>* fd00:0:0:a::/64 is directly connected, eth1
C * fe80::/64 is directly connected, eth1
C>* fe80::/64 is directly connected, eth0

$ 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:feb9:6d34 dev eth0 proto zebra metric 1024 pref medium
2001:db8:200:1::/64 dev eth0 proto kernel metric 256 pref medium
2001:db8:200:1::/64 dev eth0 metric 1024 pref medium
2001:db8:200:2::/64 dev eth1 proto kernel metric 256 pref medium
2001:db8:200:2::/64 dev eth1 metric 1024 pref medium
2001:db8:300::/47 via fe80::a00:27ff:fe48:35c1 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
fd00:0:0:a::/64 dev eth1 proto kernel metric 256 expires 2591801sec pref medium
fe80::/64 dev eth0 proto kernel metric 256 pref medium
fe80::/64 dev eth1 proto kernel metric 256 pref medium

```

VM3:

```

$ 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:db8:200::/47	2001:db8:200:2::2	0	200	i	
*> 2001:db8:300::/47	::	0	32768	i	

Total number of prefixes 3

```

$ 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:fe9f:2272, eth0, 00:04:12
B>* 2001:db8:200::/47 [20/0] via fe80::a00:27ff:fe9f:2272, eth0, 00:04:12
K * 2001:db8:200:2::/64 is directly connected, eth0

```

```

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

$ 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:fe9f:2272 dev eth0 proto zebra metric 1024 pref medium
2001:db8:200:2::/64 dev eth0 proto kernel metric 256 pref medium
2001:db8:200:2::/64 dev eth0 metric 1024 pref medium
2001:db8:200::/47 via fe80::a00:27ff:fe9f:2272 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
fd00:0:0:a::/64 dev eth0 proto kernel metric 256 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 que muestre el formato de un mensaje UPDATE.

No.	Time	Source	Destination	Protoc	Length	Info
30	124.977820	fe80::a00:27ff:fe9f:2272	2001:db8:200:2::3	ICMPv6	86	Neighbor Solicitation for 2001:db8:200:2::3 from 08:00:27:9f:22:72
31	124.978698	2001:db8:200:2::3	fe80::a00:27ff:fe9f:2272	ICMPv6	78	Neighbor Advertisement 2001:db8:200:2::3 (sol)
32	125.974886	2001:db8:200:2::3	2001:db8:200:2::2	TCP	94	46598 > bgp [SYN] Seq=0 Win=28800 Len=0 MSS=1440 SACK_PERM=1 TSval=1147896 TSecr=0 WS=128
33	125.974922	2001:db8:200:2::2	2001:db8:200:2::3	TCP	74	bgp > 46598 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
34	129.982148	fe80::a00:27ff:fe4e:fe80	a00:27ff:fe9f:2272	ICMPv6	86	Neighbor Solicitation for fe80::a00:27ff:fe9f:2272 from 08:00:27:48:35:c1
35	129.982199	fe80::a00:27ff:fe9f:fe80	a00:27ff:fe4e:fe80	ICMPv6	78	Neighbor Advertisement fe80::a00:27ff:fe9f:2272 (sol)
36	129.985890	fe80::a00:27ff:fe9f:fe80	a00:27ff:fe4e:fe80	ICMPv6	86	Neighbor Solicitation for fe80::a00:27ff:fe48:35c1 from 08:00:27:9f:22:72
37	129.987171	fe80::a00:27ff:fe4e:fe80	a00:27ff:fe9f:2272	ICMPv6	78	Neighbor Advertisement fe80::a00:27ff:fe48:35c1 (sol)
38	132.298321	2001:db8:200:2::2	2001:db8:200:2::3	TCP	94	36534 > bgp [SYN] Seq=0 Win=28800 Len=0 MSS=1440 SACK_PERM=1 TSval=1159496 TSecr=0 WS=128
39	132.299133	2001:db8:200:2::3	2001:db8:200:2::2	TCP	94	bgp > 36534 [SYN, ACK] Seq=0 Ack=1 Win=28560 Len=0 MSS=1440 SACK_PERM=1 TSval=1154221 TSecr=1159496 WS=128
40	132.299182	2001:db8:200:2::2	2001:db8:200:2::3	TCP	86	36534 > bgp [ACK] Seq=1 Ack=1 Win=28800 Len=0 TSval=1159497 TSecr=1154221
41	132.299776	2001:db8:200:2::2	2001:db8:200:2::3	BGP	147	OPEN Message
42	132.300287	2001:db8:200:2::3	2001:db8:200:2::2	TCP	86	bgp > 36534 [ACK] Seq=1 Ack=62 Win=28672 Len=0 TSval=1154222 TSecr=1159498
43	132.301443	2001:db8:200:2::3	2001:db8:200:2::2	BGP	166	OPEN Message, KEEPALIVE Message
44	132.301470	2001:db8:200:2::2	2001:db8:200:2::3	TCP	86	36534 > bgp [ACK] Seq=62 Ack=81 Win=28800 Len=0 TSval=1159500 TSecr=1154223
45	132.303426	2001:db8:200:2::2	2001:db8:200:2::3	BGP	124	KEEPALIVE Message, KEEPALIVE Message
46	132.303478	2001:db8:200:2::3	2001:db8:200:2::2	BGP	105	KEEPALIVE Message
47	132.344111	2001:db8:200:2::2	2001:db8:200:2::3	TCP	86	36534 > bgp [ACK] Seq=100 Ack=100 Win=28800 Len=0 TSval=1159542 TSecr=1154225
48	133.303817	2001:db8:200:2::2	2001:db8:200:2::3	BGP	265	UPDATE Message, UPDATE Message
49	133.304669	2001:db8:200:2::3	2001:db8:200:2::2	BGP	177	UPDATE Message

▶ Internet Protocol Version 6, Src: 2001:db8:200:2::3 (2001:db8:200:2::3), Dst: 2001:db8:200:2::2 (2001:db8:200:2::2)
 ▶ Transmission Control Protocol, Src Port: bgp (179), Dst Port: 36534 (36534), Seq: 100, Ack: 100, Len: 91
 ▼ Border Gateway Protocol - UPDATE Message
 Marker: ffffffffffffffffffffffffffffffff
 Length: 91
 Type: UPDATE Message (2)
 Unfeasible routes length: 0 bytes
 Total path attribute length: 68 bytes
 ▼ Path attributes
 ▶ ORIGIN: IGP (4 bytes)
 ▶ AS_PATH: 300 (10 bytes)
 ▶ MULTI_EXIT_DISC: 0 (7 bytes)
 ▶ MP_REACH_NLRI (47 bytes)

Hemos hecho service bgpd stop y luego service bgpd start