

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

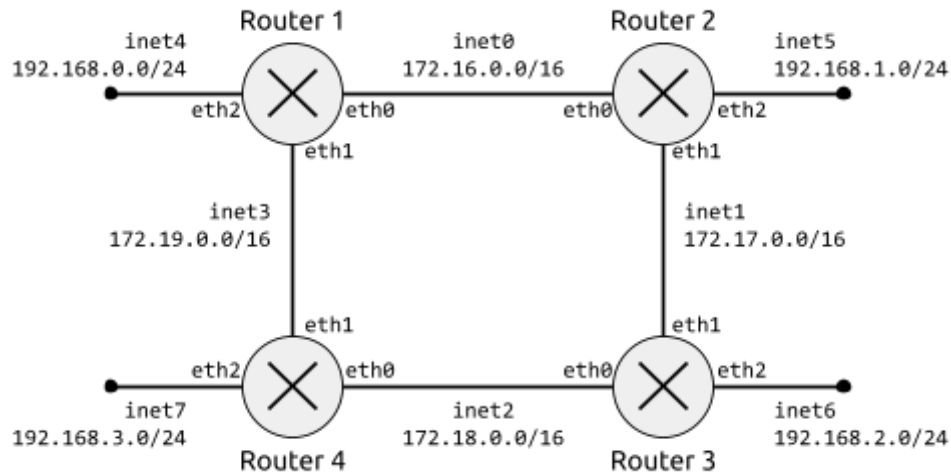
- Preparación del entorno

- 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.

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`.

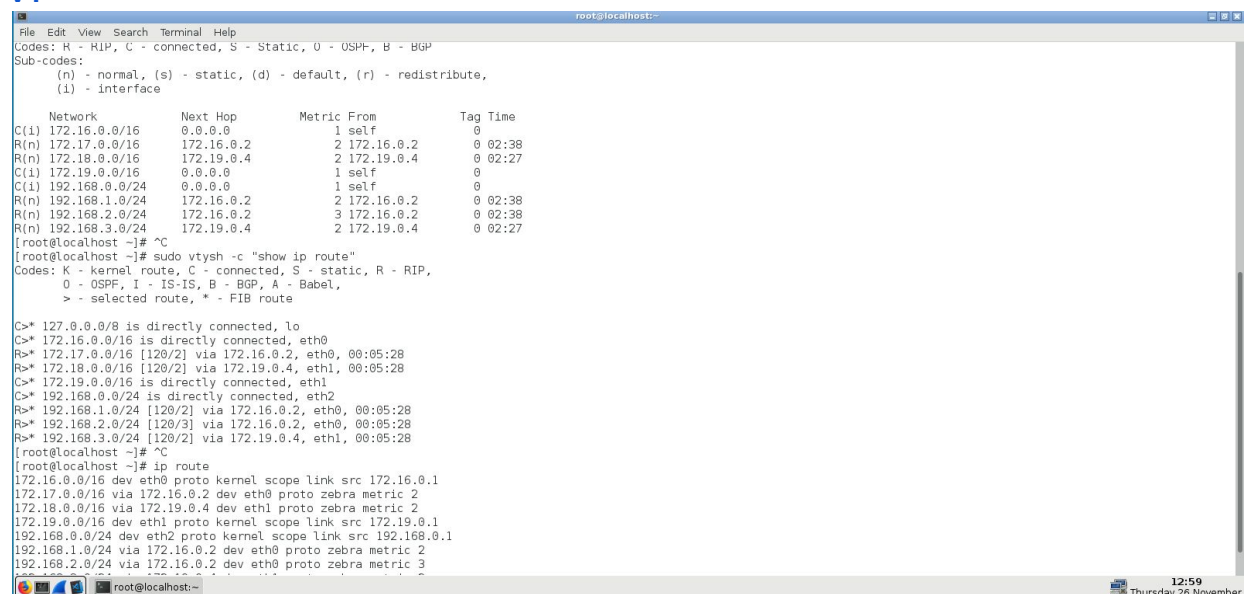
Contenido del fichero `/etc/quagga/ripd.conf`:

```
# 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 vttysh -c "show ip rip"` y `sudo vttysh -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.

V1



```
root@localhost:~#
File Edit View Search Terminal Help
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) 172.15.0.0/16 0.0.0.0        1 self           0
R(n) 172.17.0.0/16 172.16.0.2     2 172.16.0.2      0 02:38
R(n) 172.18.0.0/16 172.19.0.4     2 172.19.0.4      0 02:27
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 02:38
R(n) 192.168.2.0/24 172.16.0.2     3 172.16.0.2      0 02:38
R(n) 192.168.3.0/24 172.19.0.4     2 172.19.0.4      0 02:27
[root@localhost ~]# ^C
[root@localhost ~]# sudo vttysh -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>* 172.15.0.0/16 is directly connected, eth0
R>* 172.17.0.0/16 [120/2] via 172.16.0.2, eth0, 00:05:28
R>* 172.18.0.0/16 [120/2] via 172.19.0.4, eth1, 00:05:28
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, 00:05:28
R>* 192.168.2.0/24 [120/3] via 172.16.0.2, eth0, 00:05:28
R>* 192.168.3.0/24 [120/2] via 172.19.0.4, eth1, 00:05:28
[root@localhost ~]# ^C
[root@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
```

V2

V2 [Corriendo] - Oracle VM VirtualBox

Archivo

Máquina

Ver

Entrada

Dispositivos

Ayuda

```

[root@localhost ~]# service ripd start
Redirecting to /bin/systemctl start ripd.service
[root@localhost ~]# 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 02:59
R(n) 172.19.0.0/16 172.16.0.1      2 172.16.0.1    0 02:46
R(n) 192.168.0.0/24 172.16.0.1      2 172.16.0.1    0 02:46
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 02:59
R(n) 192.168.3.0/24 172.17.0.3      3 172.17.0.3    0 02:59
[root@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, 00:08:56
R>* 172.19.0.0/16 [120/2] via 172.16.0.1, eth0, 00:08:45
R>* 192.168.0.0/24 [120/2] via 172.16.0.1, eth0, 00:08:45
C>* 192.168.1.0/24 is directly connected, eth2
R>* 192.168.2.0/24 [120/2] via 172.17.0.3, eth1, 00:08:56
R>* 192.168.3.0/24 [120/3] via 172.17.0.3, eth1, 00:08:56
[root@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 192.168.1.2
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
[root@localhost ~]#

```

V3

V3 [Corriendo] - Oracle VM VirtualBox

Archivo

Máquina

Ver

Entrada

Dispositivos

Ayuda

root@localhost:~

File

Edit

View

Search

Terminal

Help

```

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 02:43
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 02:57
R(n) 192.168.0.0/24 172.18.0.4      3 172.18.0.4    0 02:57
R(n) 192.168.1.0/24 172.17.0.2      2 172.17.0.2    0 02:43
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 02:57
[root@localhost ~]# 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, 00:14:15
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, 00:14:29
R>* 192.168.0.0/24 [120/3] via 172.18.0.4, eth0, 00:14:05
R>* 192.168.1.0/24 [120/2] via 172.17.0.2, eth1, 00:14:15
C>* 192.168.2.0/24 is directly connected, eth2
R>* 192.168.3.0/24 [120/2] via 172.18.0.4, eth0, 00:14:29
[root@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 172.18.0.4 dev eth0 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
[root@localhost ~]#

```

V4

V4 [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

root@localhost:~

File Edit View Search Terminal Help

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	02:51
R(n)	172.17.0.0/16	172.18.0.3	2	172.18.0.3	0	02:39
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	02:51
R(n)	192.168.1.0/24	172.18.0.3	3	172.18.0.3	0	02:39
R(n)	192.168.2.0/24	172.18.0.3	2	172.18.0.3	0	02:39
C(i)	192.168.3.0/24	0.0.0.0	1	self	0	

[root@localhost ~]# 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, 00:16:37
R>* 172.17.0.0/16 [120/2] via 172.18.0.3, eth0, 00:17:00
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, 00:16:37
R>* 192.168.1.0/24 [120/3] via 172.18.0.3, eth0, 00:16:47
R>* 192.168.2.0/24 [120/2] via 172.18.0.3, eth0, 00:17:00
C>* 192.168.3.0/24 is directly connected, eth2

[root@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

[root@localhost ~]#

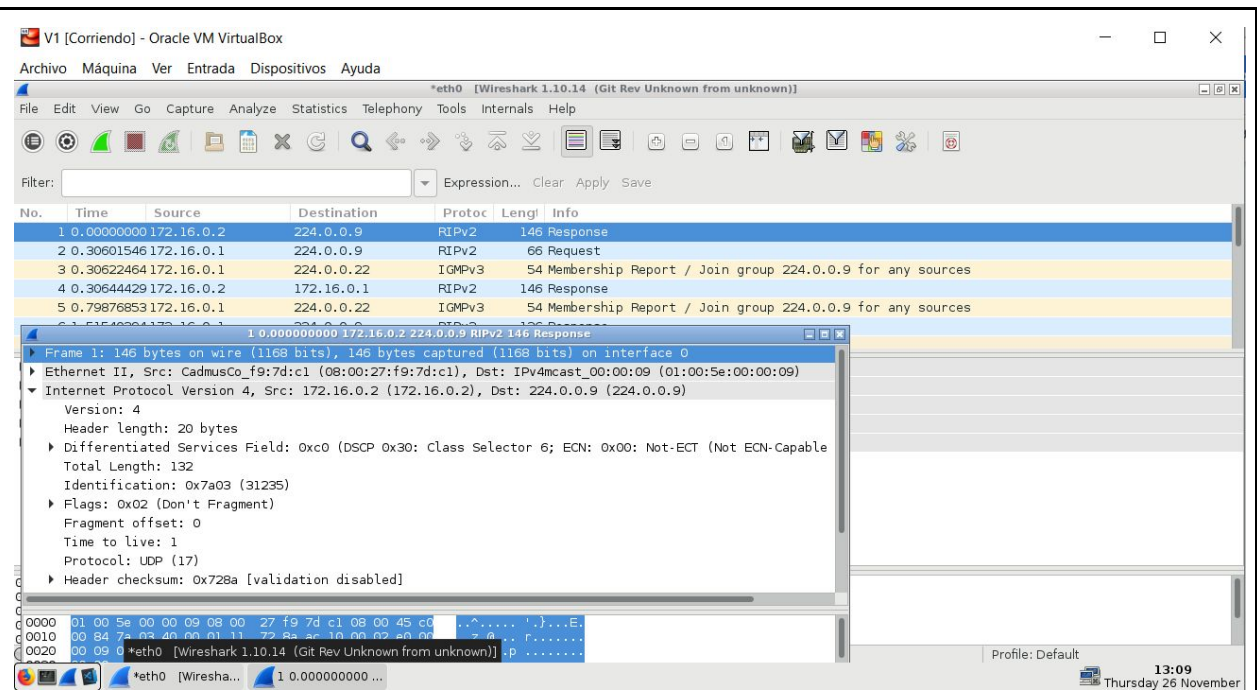
13:05 Thursday 26 November

Ejercicio 4. Con la herramienta wireshark, estudiar los mensajes RIP intercambiados, en particular:

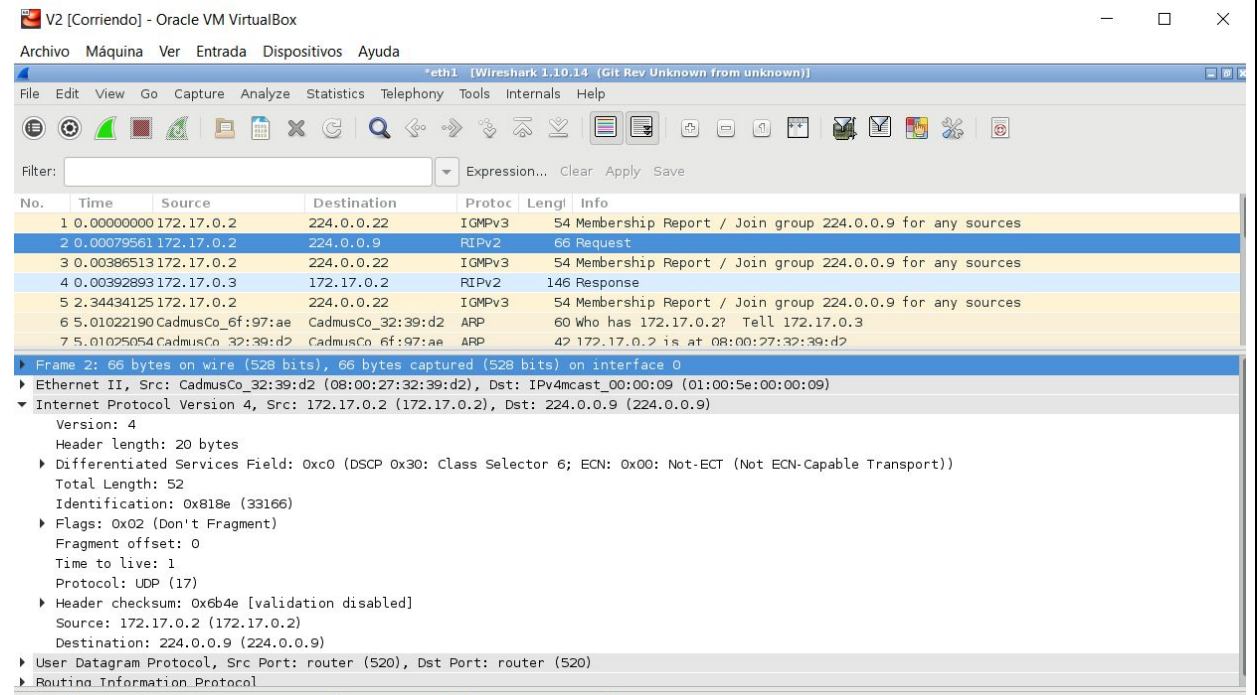
- Encapsulado.
- Direcciones origen y destino.
- Campo de versión.
- Información para cada ruta: dirección de red, máscara de red, siguiente salto y distancia.

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

V1 observado desde eth0



V2 observado eth1



V3 observado desde eth2

V3 [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

*eth2 [Wireshark 1.10.14 (Git Rev Unknown from unknown)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protoc	Length	Info
1	0.00000000	192.168.2.3	224.0.0.9	RIPv2	66	Request
2	0.00051124	192.168.2.3	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.9 for any sources
3	3.47985523	192.168.2.3	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.9 for any sources
4	5.92509255	192.168.2.3	224.0.0.9	RIPv2	186	Response
5	19.2485032	192.168.1.2	224.0.0.9	RIPv2	186	Response
6	19.9929549	192.168.3.4	224.0.0.9	RIPv2	186	Response
7	20.7586848	192.168.0.1	224.0.0.9	RIPv2	186	Response

Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0

Ethernet II, Src: CadmusCo_46:d9:86 (08:00:27:46:d9:86), Dst: IPv4mcast_00:00:09 (01:00:5e:00:00:09)

Internet Protocol Version 4, Src: 192.168.2.3 (192.168.2.3), Dst: 224.0.0.9 (224.0.0.9)

Internet Protocol Version 4, Src: 192.168.2.3 (192.168.2.3), 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: 0x5a79 (23161)

Flags: 0x02 (Don't Fragment)

Fragment offset: 0

Time to live: 1

Protocol: UDP (17)

Header checksum: 0x7bcb [validation disabled]

Source: 192.168.2.3 (192.168.2.3)

Destination: 224.0.0.9 (224.0.0.9)

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

Routing Information Protocol

0000 01 00 5e 00 00 09 08 00 27 46 d9 86 08 00 45 c0 ..^.....F....E.

0010 00 34 5a 79 40 00 01 11 7b cb c0 a8 02 03 e0 00 .4Zy@...{.....

0020 00 09 02 08 02 08 00 20 a2 e6 01 02 00 00 00 00Q.....

13:11 Thursday 26 November

V4 observado desde eth0

V4 [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

*eth0 [Wireshark 1.10.14 (Git Rev Unknown from unknown)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protoc	Length	Info
1	0.00000000	172.18.0.4	224.0.0.9	RIPv2	66	Request
2	0.00023577	172.18.0.4	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.9 for any sources
3	0.00061655	172.18.0.3	172.18.0.4	RIPv2	126	Response
4	1.02834562	172.18.0.3	224.0.0.9	RIPv2	126	Response
5	5.01547494	CadmusCo_f9:7d:c1	CadmusCo_f9:7d:c1	ARP	60	Who has 172.18.0.4? Tell 172.18.0.3
6	5.01550160	CadmusCo_f9:7d:c1	CadmusCo_f9:7d:c1	ARP	42	172.18.0.4 is at 08:00:27:f9:7d:c1
7	6.79923171	172.18.0.4	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.9 for any sources

Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0

Ethernet II, Src: CadmusCo_f9:7d:c1 (08:00:27:f9:7d:c1), Dst: IPv4mcast_00:00:09 (01:00:5e:00:00:09)

Internet Protocol Version 4, Src: 172.18.0.4 (172.18.0.4), 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: 0x3523 (13603)

Flags: 0x02 (Don't Fragment)

Fragment offset: 0

Time to live: 1

Protocol: UDP (17)

Header checksum: 0xb7b6 [validation disabled]

Source: 172.18.0.4 (172.18.0.4)

Destination: 224.0.0.9 (224.0.0.9)

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

Routing Information Protocol

0000 01 00 5e 00 00 09 08 00 27 f9 7d c1 08 00 45 c0 ..^.....F....E.

0010 00 34 5a 79 40 00 01 11 b7 b6 ac 12 00 04 e0 00 .45#@...{.....

0020 00 09 02 08 02 08 00 20 8c 51 01 02 00 00 00 00Q.....

0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00Q.....

0040 00 10 00 00 00 00 00 00 00 00 00 00 00 00 00Q.....

Frame (frame), 66 bytes Packets: 55 · Displayed: 55 (100.0%) · Dropped: 0 (0.0%) Profile: Default

13:13 Thursday 26 November

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.

```
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, 00:26:42
R>* 172.18.0.0/16 [120/2] via 172.19.0.4, eth1, 00:26:42
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, 00:26:42
R>* 192.168.2.0/24 [120/3] via 172.16.0.2, eth0, 00:26:42
R>* 192.168.3.0/24 [120/2] via 172.19.0.4, eth1, 00:26:42
```

```
[root@localhost ~]# 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	02:51
R(n) 172.18.0.0/16	172.19.0.4	2	172.19.0.4	0	02:14
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	02:51
R(n) 192.168.2.0/24	172.16.0.2	3	172.16.0.2	0	02:51
R(n) 192.168.3.0/24	172.19.0.4	2	172.19.0.4	0	02:14

```
[root@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
```

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

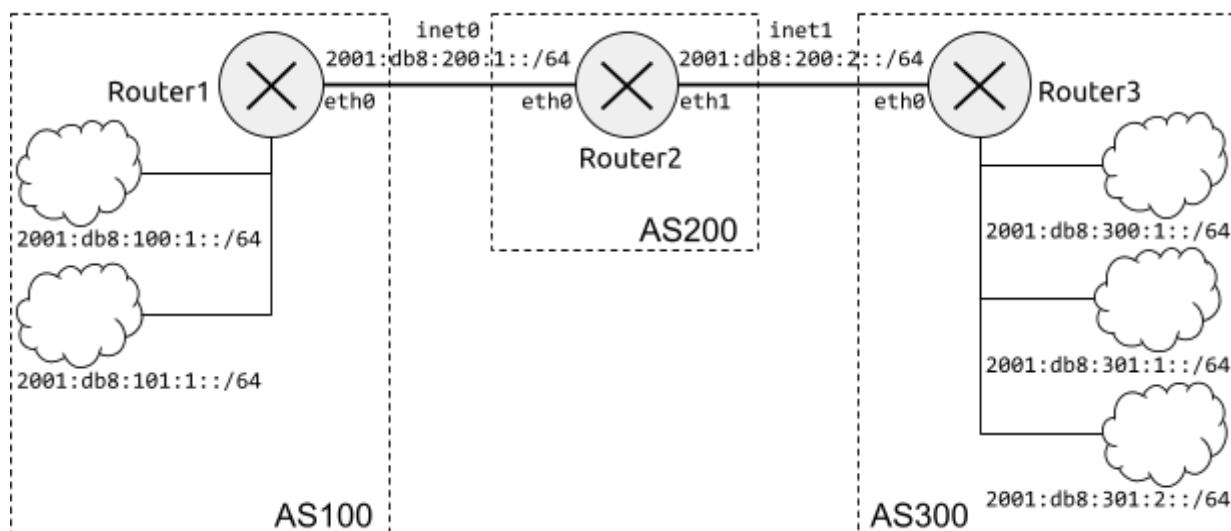
Password: asor
localhost.localdomain> enable
localhost.localdomain# configure terminal
localhost.localdomain(config)# router rip
localhost.localdomain(config-router)# version 2
localhost.localdomain(config-router)# network eth0
localhost.localdomain(config-router)# write
Configuration saved to /etc/quagga/ripd.conf
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
100	Stub (está conectado sólo a una AS)	2001:db8:100::/47
200	Tránsito	2001:db8:200::/47

300	Multihomed	2001:db8:300::/47
-----	------------	-------------------

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:

```
# 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 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.

V1

```
[root@localhost ~]# 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
```

```
Total number of prefixes 2
```

V2

```
[root@localhost ~]# 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

```

Total number of prefixes 2

V3

```
[root@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:db8:300::/47
::      0      32768 i

```

Total number of prefixes 1

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.

The image shows two screenshots of the Wireshark network protocol analyzer. The left screenshot displays a packet capture from the 'eth0' interface, showing a list of packets including BGP messages. The selected packet is a BGP UPDATE message, and its details are shown in the bottom pane, including the BGP header, the UPDATE message structure, and the payload. The right screenshot shows a similar packet capture, but with a different set of packets, including BGP OPEN, KEEPALIVE, and UPDATE messages. The selected packet is a BGP UPDATE message, and its details are shown in the bottom pane, including the BGP header, the UPDATE message structure, and the payload.