

1. Guión RIP

Alumnos: Luis Blazquez Miñambres 70910465Q

Samuel Gómez Sánchez

Nombre de la práctica: RIP

Asignatura: Redes de Computadores II

Fecha: 26 – 04 - 2019

Duración estimada de la práctica: 2 sesiones de 2h.

1.1. Cómo realizar un buen informe

- Al ir realizando todas las actividades que se proponen se han de documentar todos los pasos, adjuntando las órdenes (comandos o actividades) realizadas junto con las capturas de pantalla de su ejecución, explicando siempre la salida obtenida. No es suficiente con una captura de pantalla sin texto que acompañe a la imagen y lo contrario tampoco; es decir, adjuntar la orden (o comando) sin aportar prueba alguna de que se ejecutó realmente y sin una interpretación de la salida o resultado obtenido.
- En el informe se ha demostrar que se han realizado y entendido todas las actividades propuestas.
- En la entrega de la práctica se han de adjuntar todos los ficheros auxiliares que hayan sido utilizados y que no se incluyan en el informe. Por ejemplo: el escenario final obtenido, los ficheros de capturas del tráfico de red, scripts realizados para automatizar tareas, etc.

- Contestar en color verde para diferenciar claramente las respuestas del enunciado.

1.2.

1.3. Funcionamiento básico

1. En el fichero *RIPOSPF.rar* está definida una red como la que se muestra en la Figura 1. Descomprime el fichero de configuración del escenario *RIPOSPF.rar* en la carpeta correspondiente de GNS3.

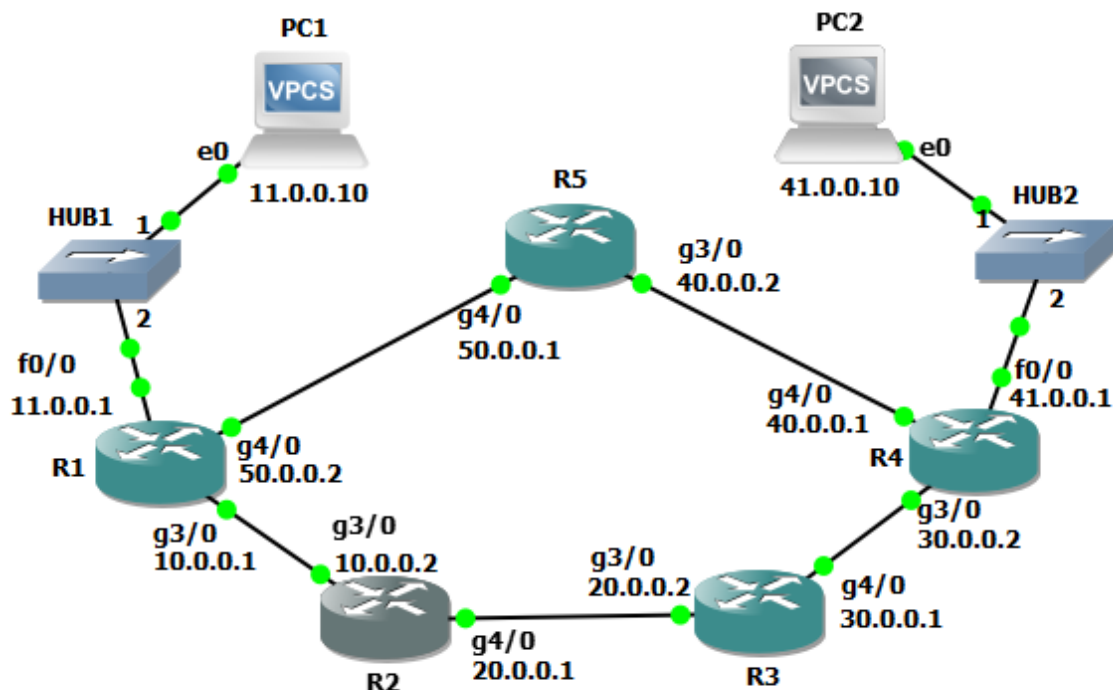


Figura 1: Escenario RIP y OSPF

Arranca todas las máquinas y abre una consola con cada una de ellas (orden `consolas`). Los equipos PC1 y PC2 tienen rutas por defecto a R1 y R4 respectivamente. Compruébalo con la orden `show o show ip` (Incluye aquí esa información).

RESPUESTA: Como podemos comprobar, PC1 tiene como dirección IP del router por defecto la dirección IP de la interfaz eth0 de R1. De la misma forma que PC2 tiene la dirección IP del router por defecto la de la interfaz eth0 de R2

```

PC1> show

NAME      IP/MASK      GATEWAY      MAC      LPORT  RHOST:PORT
PC1      11.0.0.10/8    11.0.0.1      00:50:79:66:68:00  10018  127.0.0.1:10019
fe80::250:79ff:fe66:6800/64

PC1> show ip

NAME      : PC1[1]
IP/MASK    : 11.0.0.10/8
GATEWAY    : 11.0.0.1
DNS        :
MAC        : 00:50:79:66:68:00
LPORT      : 10018
RHOST:PORT : 127.0.0.1:10019
MTU        : 1500

```

Imagen 1.1: Configuración del PC1

```

PC2> show

NAME      IP/MASK      GATEWAY      MAC      LPORT  RHOST:PORT
PC2      41.0.0.10/8    41.0.0.1      00:50:79:66:68:01  10020  127.0.0.1:10021
fe80::250:79ff:fe66:6801/64

PC2> show ip

NAME      : PC2[1]
IP/MASK    : 41.0.0.10/8
GATEWAY    : 41.0.0.1
DNS        :
MAC        : 00:50:79:66:68:01
LPORT      : 10020
RHOST:PORT : 127.0.0.1:10021
MTU        : 1500

```

Imagen 1.2: Configuración del PC2

Los *routers* no tienen configurada ninguna ruta, salvo la de las subredes a las que están directamente conectados. Compruébalo con la orden *show ip route* (Incluye aquí esa información).

RESPUESTA:

```

R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    50.0.0.0/8 is directly connected, GigabitEthernet4/0
C    10.0.0.0/8 is directly connected, GigabitEthernet3/0
C    11.0.0.0/8 is directly connected, FastEthernet0/0

```

Imagen 1.3: Tabla de rutas de R1

```

R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    20.0.0.0/8 is directly connected, GigabitEthernet4/0
C    10.0.0.0/8 is directly connected, GigabitEthernet3/0

```

Imagen 1.4: Tabla de rutas de R2

```

R3#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    20.0.0.0/8 is directly connected, GigabitEthernet3/0
C    30.0.0.0/8 is directly connected, GigabitEthernet4/0

```

Imagen 1.5: Tabla de rutas de R3

```

R5#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    50.0.0.0/8 is directly connected, GigabitEthernet4/0
C    40.0.0.0/8 is directly connected, GigabitEthernet3/0

```

Imagen 1.6: Tabla de rutas de R4

```

R4#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    40.0.0.0/8 is directly connected, GigabitEthernet4/0
C    41.0.0.0/8 is directly connected, FastEthernet0/0
C    30.0.0.0/8 is directly connected, GigabitEthernet3/0

```

Imagen 1.7: Tabla de rutas de R5

En los siguientes apartados se configurará RIP en los *routers* para que las tablas de encaminamiento permitan alcanzar cualquier punto de la red.

2. Para observar los mensajes que envíe R1 cuando se active RIP, arranca *wireshark* en todos los enlaces de R1. A continuación configura RIP en R1 para que exporte las rutas de las tres redes a las que está conectado.

```

config t
router rip
version 2
network 10.0.0.0
network 11.0.0.0
network 50.0.0.0
no auto-summary
exit
exit
wr

```

Activa la depuración de los mensajes rip: `debug ip rip`. Espera un minuto aproximadamente e interrumpe las capturas. Interrumpe también los mensajes de depuración: `no debug ip rip`

Analiza el comportamiento de R1 estudiando las capturas del tráfico y los mensajes de depuración para responder a las siguientes preguntas:

- a) Observa los mensajes REQUEST que se envían al arrancar *RIP* en R1 y analiza su contenido. ¿Son iguales en todas las interfaces? ¿Para qué se utilizan? ¿Qué rutas viajan en estos mensajes? ¿Quién responde a estos mensajes?

RESPUESTA: Los mensajes REQUEST en todas las interfaces son iguales en cuanto a contenido, la única diferencia que se aprecia de manera significativa es la dirección origen desde donde se manda el mensaje, que es para cada interfaz la suya.

Estos mensajes de REQUEST son mensajes de solicitud que se envían para que el resto de routers vecinos le envíen su tabla de rutas completa y él que las solicita pueda rellenarla, en este caso, es un mensaje de todas las rutas existentes ya que llevan en el campo *Command* el tipo de mensaje *Request (1)*. Se envía a la dirección multicast 224.0.0.9 cuando el router arranca y quiere rellenar su tabla de encaminamiento y a través del puerto 520 en UDP. Estos mensajes de solicitud, tal y como se puede observar en la capturas de Wireshark, todas llevan métrica 16 (que indica un destino inalcanzable) ya que aún no conocen el coste o número de saltos por los que hay que pasar para alcanzar cada destino y tienen familia de direcciones todas rellenas con 0.

Este mensaje lleva las redes que tiene R1 en sus tablas, es decir, aquellas redes a las que está directamente conectado (por sus interfaces eth0, eth3 y eth4) que son las que vienen al haber hecho la orden *show ip route* → 50.0.0.0, 11.0.0.0 y 10.0.0.0.

A este mensaje responderá el router o los routers que estén suscritos al grupo multicast 224.0.0.9 por la interfaz conectada a la interfaz por la que el router que inicia manda el mensaje de REQUEST y por tanto puedan recibir mensajes RIP para esta dirección IP. En este caso solo llegarían a ver el mensaje los routers vecinos del router que envía el mensaje ya que los mensajes del protocolo se envían con TTL=1

7	46.22432500(11.0.0.1	224.0.0.9	RIPv2	66 Request
9	54.45825300(11.0.0.1	224.0.0.9	RIPv2	66 Response
11	63.19262900(11.0.0.1	224.0.0.9	RIPv2	86 Response
16	89.22646400(11.0.0.1	224.0.0.9	RIPv2	86 Response
19	115.3803960(11.0.0.1	224.0.0.9	RIPv2	86 Response
23	144.0569640(11.0.0.1	224.0.0.9	RIPv2	86 Response

Imagen 2_a.1: Captura de mensajes de Wireshark en la interfaz eth0

▶	Frame 7: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▶	Ethernet II, Src: ca:01:2e:d8:00:00 (ca:01:2e:d8:00:00), Dst: IPv4mcast_09 (01:00:5e:00:00:09)
▼	Internet Protocol Version 4, Src: 11.0.0.1 (11.0.0.1), 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: 0x0000 (0)
▶	Flags: 0x00
	Fragment offset: 0
▶	Time to live: 2
	Protocol: UDP (17)
▶	Header checksum: 0xccef [validation disabled]
	Source: 11.0.0.1 (11.0.0.1)
	Destination: 224.0.0.9 (224.0.0.9)
	[Source GeoIP: Unknown]
	[Destination GeoIP: Unknown]
▶	User Datagram Protocol, Src Port: 520 (520), Dst Port: 520 (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

Imagen 2_a.2: Contenido del mensaje de REQUEST por la interfaz eth0

14	52.28506000(50.0.0.2)	224.0.0.9	RIPv2	66 Request
19	70.88122500(50.0.0.2)	224.0.0.9	RIPv2	86 Response
28	99.94482800(50.0.0.2)	224.0.0.9	RIPv2	86 Response
35	129.4689770(50.0.0.2)	224.0.0.9	RIPv2	86 Response
44	159.5032720(50.0.0.2)	224.0.0.9	RIPv2	86 Response

Imagen 2_a.3: Captura de mensajes de Wireshark en la interfaz eth4

▶	Frame 14: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▶	Ethernet II, Src: ca:01:2e:d8:00:70 (ca:01:2e:d8:00:70), Dst: IPv4mcast_09 (01:00:5e:00:00:09)
▼	Internet Protocol Version 4, Src: 50.0.0.2 (50.0.0.2), 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: 0x0000 (0)
▶	Flags: 0x00
	Fragment offset: 0
▶	Time to live: 2
	Protocol: UDP (17)
▶	Header checksum: 0xa5ee [validation disabled]
	Source: 50.0.0.2 (50.0.0.2)
	Destination: 224.0.0.9 (224.0.0.9)
	[Source GeoIP: Unknown]
	[Destination GeoIP: Unknown]
▶	User Datagram Protocol, Src Port: 520 (520), Dst Port: 520 (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

Imagen 2_a.4: Contenido del mensaje de REQUEST por la interfaz eth4

10	30.68424000(10.0.0.1)	224.0.0.9	RIPv2	66 Request
12	39.84154300(10.0.0.1)	224.0.0.9	RIPv2	66 Response
15	46.06310100(10.0.0.1)	224.0.0.9	RIPv2	66 Response
18	54.67660500(10.0.0.1)	224.0.0.9	RIPv2	86 Response
24	80.68998100(10.0.0.1)	224.0.0.9	RIPv2	86 Response
33	110.6683590(10.0.0.1)	224.0.0.9	RIPv2	86 Response
39	136.6778760(10.0.0.1)	224.0.0.9	RIPv2	86 Response

Imagen 2_a.5: Captura de mensajes de Wireshark en la interfaz eth3

▶	Frame 10: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▶	Ethernet II, Src: ca:01:2e:d8:00:54 (ca:01:2e:d8:00:54), Dst: IPv4mcast_09 (01:00:5e:00:00:09)
▼	Internet Protocol Version 4, Src: 10.0.0.1 (10.0.0.1), 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: 0x0000 (0)
▶	Flags: 0x00
	Fragment offset: 0
▶	Time to live: 2
	Protocol: UDP (17)
▶	Header checksum: 0xcdef [validation disabled]
	Source: 10.0.0.1 (10.0.0.1)
	Destination: 224.0.0.9 (224.0.0.9)
	[Source GeoIP: Unknown]
	[Destination GeoIP: Unknown]
▶	User Datagram Protocol, Src Port: 520 (520), Dst Port: 520 (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

Imagen 2_a.6: Contenido del mensaje de REQUEST por la interfaz eth3

R1#debug ip rip

RIP protocol debugging is on

*Mar 26 11:43:51.035: **RIP: sending v2 update to 224.0.0.9 via GigabitEthernet4/0 (50.0.0.2)**

*Mar 26 11:43:51.035: RIP: build update entries

*Mar 26 11:43:51.035: 10.0.0.0/8 via 0.0.0.0, metric 1, tag 0

*Mar 26 11:43:51.035: 11.0.0.0/8 via 0.0.0.0, metric 1, tag 0

R1#

*Mar 26 11:44:05.931: **RIP: sending v2 update to 224.0.0.9 via FastEthernet0/0 (11.0.0.1)**

*Mar 26 11:44:05.931: RIP: build update entries

*Mar 26 11:44:05.931: 10.0.0.0/8 via 0.0.0.0, metric 1, tag 0

*Mar 26 11:44:05.935: 50.0.0.0/8 via 0.0.0.0, metric 1, tag 0

R1#

*Mar 26 11:44:09.583: **RIP: sending v2 update to 224.0.0.9 via GigabitEthernet3/0 (10.0.0.1)**

*Mar 26 11:44:09.583: RIP: build update entries

*Mar 26 11:44:09.583: 11.0.0.0/8 via 0.0.0.0, metric 1, tag 0

*Mar 26 11:44:09.587: 50.0.0.0/8 via 0.0.0.0, metric 1, tag 0

R1#

*Mar 26 11:44:20.127: **RIP: sending v2 update to 224.0.0.9 via GigabitEthernet4/0 (50.0.0.2)**

*Mar 26 11:44:20.127: RIP: build update entries

*Mar 26 11:44:20.127: 10.0.0.0/8 via 0.0.0.0, metric 1, tag 0

*Mar 26 11:44:20.131: 11.0.0.0/8 via 0.0.0.0, metric 1, tag 0

R1#

*Mar 26 11:44:34.235: **RIP: sending v2 update to 224.0.0.9 via FastEthernet0/0 (11.0.0.1)**

*Mar 26 11:44:34.235: RIP: build update entries

*Mar 26 11:44:34.235: 10.0.0.0/8 via 0.0.0.0, metric 1, tag 0

*Mar 26 11:44:34.239: 50.0.0.0/8 via 0.0.0.0, metric 1, tag 0

R1#

*Mar 26 11:44:35.291: **RIP: sending v2 update to 224.0.0.9 via GigabitEthernet3/0 (10.0.0.1)**

*Mar 26 11:44:35.291: RIP: build update entries

*Mar 26 11:44:35.291: 11.0.0.0/8 via 0.0.0.0, metric 1, tag 0

*Mar 26 11:44:35.291: 50.0.0.0/8 via 0.0.0.0, metric 1, tag 0

R1#

*Mar 26 11:44:49.735: **RIP: sending v2 update to 224.0.0.9 via GigabitEthernet4/0 (50.0.0.2)**

*Mar 26 11:44:49.735: RIP: build update entries

*Mar 26 11:44:49.735: 10.0.0.0/8 via 0.0.0.0, metric 1, tag 0

*Mar 26 11:44:49.735: 11.0.0.0/8 via 0.0.0.0, metric 1, tag 0

R1#

*Mar 26 11:45:00.107: **RIP: sending v2 update to 224.0.0.9 via FastEthernet0/0 (11.0.0.1)**

*Mar 26 11:45:00.107: RIP: build update entries

*Mar 26 11:45:00.107: 10.0.0.0/8 via 0.0.0.0, metric 1, tag 0

*Mar 26 11:45:00.111: 50.0.0.0/8 via 0.0.0.0, metric 1, tag 0

R1#

*Mar 26 11:45:01.663: **RIP: sending v2 update to 224.0.0.9 via GigabitEthernet3/0 (10.0.0.1)**

*Mar 26 11:45:01.663: RIP: build update entries

*Mar 26 11:45:01.663: 11.0.0.0/8 via 0.0.0.0, metric 1, tag 0

*Mar 26 11:45:01.667: 50.0.0.0/8 via 0.0.0.0, metric 1, tag 0

R1#no debug ip rip

- b) Observa los mensajes RESPONSE que envía R1 periódicamente a través de cada una de sus interfaces. ¿Son iguales en todas las interfaces? ¿Qué rutas viajan en estos mensajes?

RESPUESTA: No todas las respuestas que manda son iguales en todas las interfaces, en cada interfaz manda las direcciones IP de su tabla de rutas excepto la de la red de la interfaz por la que la está mandando. Los mensajes que se envían son mensajes de respuesta periódicas o no solicitadas en las que manda todas las rutas de su tabla de rutas excepto la de la red por la que la manda , para actualizar sus tablas.

Por ejemplo, en las capturas que adjunto a continuación pertenecen a los mensajes RESPONSE por la interfaz eth3 de R1 , todas ellas son enviadas a la dirección multicast 224.0.0.9 a todos los routers vecinos por lo que son mensajes periódicos de actualización de tablas, de otra manera serían enviados a través de unicast a la dirección correspondiente. En las imágenes (1) y (2) se mandan los mensajes de respuesta al REQUEST con longitud de trama 66 en la que cada uno contiene una de las redes de la tabla de rutas de R1. En cambio en la imagen (3) es un mensaje RESPONSE que se enviará periódicamente de manera continuada (es una respuesta NO SOLICITADA) en los que se envían las dos rutas enviadas en la respuesta al REQUEST.

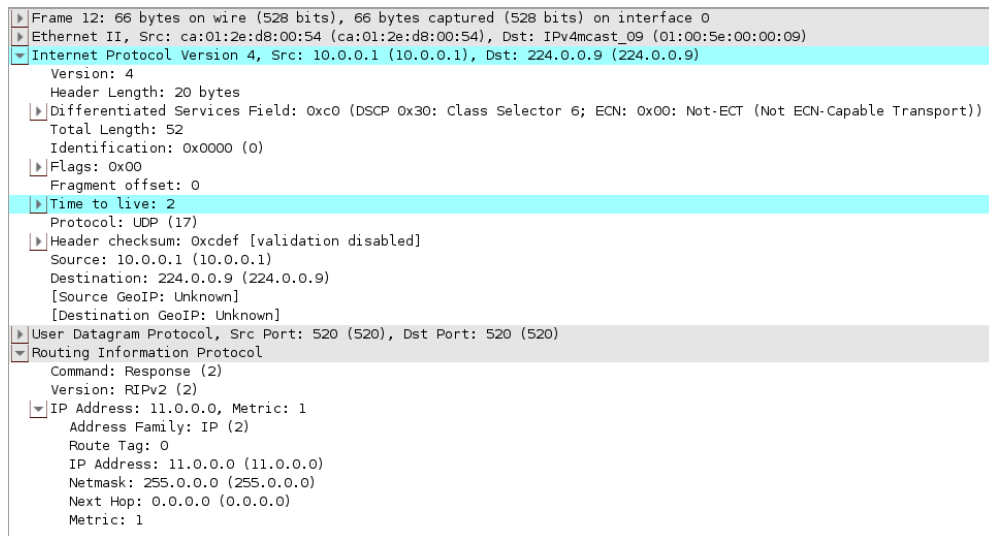


Imagen 2_b.1: Contenido del mensaje de RESPONSE con la dirección 11.0.0.0

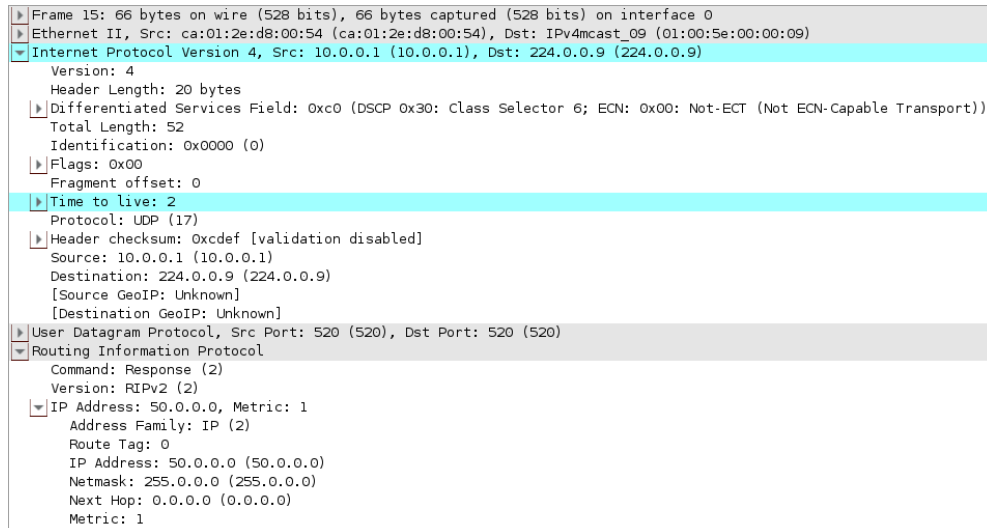


Imagen 2_b.2: Contenido del mensaje de RESPONSE con la dirección 50.0.0.0

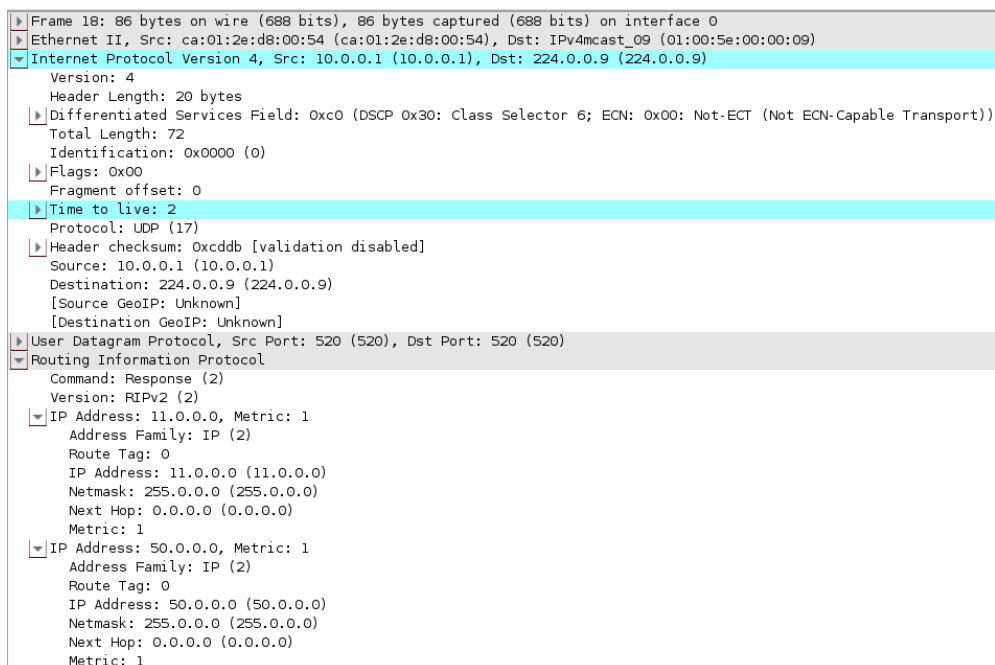


Imagen 2_b.3: Contenido del mensaje de RESPONSE periódico con 11.0.0.0 y 50.0.0.0

En esta red viajan las rutas con métrica 1, ya que las redes que viajan están directamente conectadas al router y son: 50.0.0.0, 11.0.0.0 y 10.0.0.0

- c) ¿Debería haber aprendido alguna ruta R1? Compruébalo consultando la tabla de encaminamiento mediante la orden `show ip route`.

RESPUESTA: No, no debería haber aprendido ninguna ruta nueva porque no ha recibido ningún mensaje de ningún otro router, ya que no hemos activado el protocolo RIP por ninguna de sus interfaces. En los mensajes de RESPONSE solo viajaban las rutas a las que R1 ya estaba directamente conectada y posteriormente los mensajes de RESPONSE periódicos cada 30 segundos.

- d) Ejecuta en R1 la orden `show ip protocols`. ¿Qué información nos facilita?
Adjunta y comenta la salida obtenida.

RESPUESTA: Nos facilita información acerca de los protocolos de red que tiene activados, en este caso es el protocolo de enrutamiento RIP, que hemos activado anteriormente. Aparece información relevante del protocolo como el tiempo restante hasta que se vuelva a mandar un RESPONSE periódico cada 30 segundos (en nuestro caso 23 segundos), el número de mensajes RIP recibidos y enviados por cada una de las interfaces y las redes contenidas en su tabla de rutas.

R1#show ip protocols

Routing Protocol is "rip"

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Sending updates every 30 seconds, **next due in 23 seconds**

Invalid after 180 seconds, hold down 180, flushed after 240

Redistributing: rip

Default version control: send version 2, receive version 2

Interface	Send	Recv	Triggered	RIP	Key-chain
FastEthernet0/0	2	2			
GigabitEthernet3/0	2	2			
GigabitEthernet4/0	2	2			

Automatic network summarization is not in effect

Maximum path: 4

Routing for Networks:

10.0.0.0

11.0.0.0

50.0.0.0

Routing Information Sources:

Gateway Distance Last Update

Distance: (default is 120)

En el resto de routers, este protocolo se encuentra desactivado por lo que la salida de este comando será vacía.

```
R2#show ip protocols  
R2#
```

Imagen 2_d.1: Ejemplo de que otros routers como R2 no tienen RIP activado

3. ¿Para qué sirve la orden `passive-interface <interface>`? ¿Sería útil configurar en R1 alguna de sus interfaces de este modo?. Incluye aquí la orden.

RESPUESTA: Según el manual de ordenes de IOS para RIP, indica las interfaces por las que no se distribuirán rutas, de manera que la interfaz pasada como parámetros junto a la orden será una interfaz por la que el router desactivará la distribución de rutas.

Sería útil configurar esta orden para la interfaz `eth0` tanto de R1 como de R4 , que son las interfaces por las que están conectados directamente a los equipos PC1 y PC2 respectivamente, ya que estos equipos no implementan RIP y no necesitan recibir las tablas de rutas ni van a enviar ningún mensaje relacionado con este protocolo a los routers. Por lo tanto es un gasto innecesario tener esas dos interfaces activadas para RIP sabiendo que no las necesitamos para que los routers se envíen información a través de este protocolo.

Utiliza esta orden en el resto de routers del escenario en los que sea necesario.

```
R1#enable  
R1#config terminal  
R1(config)#router rip  
R1(config-router)#passive-interface FastEthernet0/0  
R1(config-router)#exit  
R1#wr  
Building configuration...  
[OK]
```

```
R4#enable
R4#config terminal
R4(config)#router rip
R4(config-router)#passive-interface FastEthernet0/0
R4(config-router)#exit
R4(config)#exit
R4#wr
Building configuration...
```

4. Ve configurando y arrancando a continuación RIP en los encaminadores R2, R3 y R4, de uno en uno: primero en R2, luego en R3 y finalmente en R4. Cada vez que configures uno de esos encaminadores realiza las capturas que estimes conveniente, estudiándolas junto con los mensajes de depuración y responde a las siguientes cuestiones:

- a) Incluye las órdenes de configuración.

RESPUESTA: En cada router se ha añadido al parámetro *network* las redes a las que están directamente conectadas y por las que poder mandar los mensajes del protocolo RIP.

```
R2#enable
R2#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#router rip
R2(config-router)#version 2
R2(config-router)#network 10.0.0.0
R2(config-router)#network 20.0.0.0
R2(config-router)#no auto-summary
R2(config-router)#exit
R2(config)#exit
R2#wr
Building configuration...
```

Imagen 4_a.: Activación de RIP en el router R2

```
R3#enable
R3#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#router rip
R3(config-router)#version 2
R3(config-router)#network 20.0.0.0
R3(config-router)#network 30.0.0.0
R3(config-router)#no auto-summary
R3(config-router)#exit
R3(config)#exit
R3#wr
Building configuration...
```

Imagen 4_a.2: Activación de RIP en el router R3

```
R4#enable
R4#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R4(config)#router rip
R4(config-router)#version 2
R4(config-router)#network 30.0.0.0
R4(config-router)#network 40.0.0.0
R4(config-router)#network 41.0.0.0
R4(config-router)#no auto-summary
R4(config-router)#exit
R4(config)#exit
R4#wr
Building configuration...
[OK]
```

Imagen 4_a.3: Activación de RIP en el router R4

- b) Comprueba el envío de mensajes REQUEST. ¿Existe algún mensaje de RESPONSE a esos REQUEST? ¿Por qué?

RESPUESTA: podemos observar que al activar únicamente R2 primero no recibimos ningún mensaje RESPONSE a ese REQUEST porque no teníamos activado el protocolo RIP para los otros routers a los que estaba conectado exceptuando R1 porque los dos ya conocen la red 10.0.0.0 por lo que no tienen que actualizar sus tablas con nuevas rutas. Por tanto tras mandar el mensaje de REQUEST, que es recibido únicamente por R1, como único router que tiene activado RIP por su interfaz, los siguientes mensajes RESPONSE son únicamente de actualización periódicos como los anteriores aunque se puede distinguir que del mensaje nº 64 al mensaje nº70 cambia la métrica de las redes 50.0.0.0 y 11.0.0.0 que estaban a 1 y pasaron a 2.

En los routers R3 y R4 ocurre exactamente lo mismo como se puede observar en las capturas de WireShark por sus correspondientes interfaces, mandan un mensaje de REQUEST y más tarde un mensaje de RESPONSE periódico de actualización de tablas, cada una con sus métricas correspondientes para cada Router. Con la salvedad de que R3 envía un RESPONSE donde ha actualizado su tabla con las direcciones que ha recibido de R2 y lo mismo R4, la cual ha actualizado su tabla con las direcciones que ha recibido de R3, de manera que R4 conoce ya todas las redes del lado de su interfaz eth3.

17	91.666038	20.0.0.1	224.0.0.9	RIPv2	66 Request
19	97.896387	20.0.0.1	224.0.0.9	RIPv2	86 Response
23	122.217394	20.0.0.1	224.0.0.9	RIPv2	106 Response
30	153.120810	20.0.0.1	224.0.0.9	RIPv2	106 Response
35	184.112987	20.0.0.1	224.0.0.9	RIPv2	106 Response
41	213.086560	20.0.0.1	224.0.0.9	RIPv2	106 Response
47	239.624643	20.0.0.1	224.0.0.9	RIPv2	106 Response
52	269.168691	20.0.0.1	224.0.0.9	RIPv2	106 Response
57	296.814810	20.0.0.1	224.0.0.9	RIPv2	106 Response
64	324.874827	20.0.0.1	224.0.0.9	RIPv2	106 Response
70	353.475396	20.0.0.1	224.0.0.9	RIPv2	106 Response
76	381.889465	20.0.0.1	224.0.0.9	RIPv2	106 Response
81	411.126332	20.0.0.1	224.0.0.9	RIPv2	106 Response
88	441.392114	20.0.0.1	224.0.0.9	RIPv2	106 Response
93	470.672867	20.0.0.1	224.0.0.9	RIPv2	106 Response
99	500.776419	20.0.0.1	224.0.0.9	RIPv2	106 Response
105	528.452457	20.0.0.1	224.0.0.9	RIPv2	106 Response
109	557.651424	20.0.0.1	224.0.0.9	RIPv2	106 Response
116	587.721066	20.0.0.1	224.0.0.9	RIPv2	106 Response
121	618.820956	20.0.0.1	224.0.0.9	RIPv2	106 Response
127	648.584420	20.0.0.1	224.0.0.9	RIPv2	106 Response
133	676.790045	20.0.0.1	224.0.0.9	RIPv2	106 Response
140	705.796530	20.0.0.1	224.0.0.9	RIPv2	106 Response
145	736.284056	20.0.0.1	224.0.0.9	RIPv2	106 Response
150	763.906243	20.0.0.1	224.0.0.9	RIPv2	106 Response
155	791.278095	20.0.0.1	224.0.0.9	RIPv2	106 Response
161	818.660917	20.0.0.1	224.0.0.9	RIPv2	106 Response
166	845.067350	20.0.0.1	224.0.0.9	RIPv2	106 Response
173	875.751355	20.0.0.1	224.0.0.9	RIPv2	106 Response
178	903.792419	20.0.0.1	224.0.0.9	RIPv2	106 Response
183	930.549915	20.0.0.1	224.0.0.9	RIPv2	106 Response
190	960.774148	20.0.0.1	224.0.0.9	RIPv2	106 Response
195	988.282632	20.0.0.1	224.0.0.9	RIPv2	106 Response

Imagen 4_b.1: Captura del tráfico de wireshark por la interfaz eth3 del router R2

```

  Routing Information Protocol
    Command: Response (2)
    Version: RIPv2 (2)
  IP Address: 10.0.0.0, Metric: 1
    Address Family: IP (2)
    Route Tag: 0
    IP Address: 10.0.0.0
    Netmask: 255.0.0.0
    Next Hop: 0.0.0.0
    Metric: 1
  IP Address: 11.0.0.0, Metric: 2
    Address Family: IP (2)
    Route Tag: 0
    IP Address: 11.0.0.0
    Netmask: 255.0.0.0
    Next Hop: 0.0.0.0
    Metric: 2
  IP Address: 50.0.0.0, Metric: 2
    Address Family: IP (2)
    Route Tag: 0
    IP Address: 50.0.0.0
    Netmask: 255.0.0.0
    Next Hop: 0.0.0.0
    Metric: 2

```

Imagen 4_b.4: Contenido de los mensajes de RESPONSE periódicos de R2

488	2965.226027	30.0.0.1	224.0.0.9	RIPv2	66 Request
494	2981.660107	30.0.0.1	224.0.0.9	RIPv2	126 Response
502	3008.997055	30.0.0.1	224.0.0.9	RIPv2	126 Response
507	3036.398830	30.0.0.1	224.0.0.9	RIPv2	126 Response
516	3064.549600	30.0.0.1	224.0.0.9	RIPv2	126 Response
523	3093.125552	30.0.0.1	224.0.0.9	RIPv2	126 Response
531	3121.517679	30.0.0.1	224.0.0.9	RIPv2	126 Response

Imagen 4_b.2: Captura del tráfico de wireshark por la interfaz eth3 del router R3

```

▼ Routing Information Protocol
  Command: Response (2)
  Version: RIPv2 (2)
  ▼ IP Address: 10.0.0.0, Metric: 2
    Address Family: IP (2)
    Route Tag: 0
    IP Address: 10.0.0.0
    Netmask: 255.0.0.0
    Next Hop: 0.0.0.0
    Metric: 2
  ▼ IP Address: 11.0.0.0, Metric: 3
    Address Family: IP (2)
    Route Tag: 0
    IP Address: 11.0.0.0
    Netmask: 255.0.0.0
    Next Hop: 0.0.0.0
    Metric: 3
  ▼ IP Address: 20.0.0.0, Metric: 1
    Address Family: IP (2)
    Route Tag: 0
    IP Address: 20.0.0.0
    Netmask: 255.0.0.0
    Next Hop: 0.0.0.0
    Metric: 1
  ▼ IP Address: 50.0.0.0, Metric: 3
    Address Family: IP (2)
    Route Tag: 0
    IP Address: 50.0.0.0
    Netmask: 255.0.0.0
    Next Hop: 0.0.0.0
    Metric: 3

```

Imagen 4_b.3: Contenido de los mensajes de RESPONSE periódicos de R3

449	2734.613481	40.0.0.1	224.0.0.9	RIPv2	66 Request
451	2743.938561	40.0.0.1	224.0.0.9	RIPv2	66 Response
455	2758.046859	40.0.0.1	224.0.0.9	RIPv2	126 Response
456	2758.891602	40.0.0.1	224.0.0.9	RIPv2	166 Response
461	2788.292033	40.0.0.1	224.0.0.9	RIPv2	166 Response
468	2818.077437	40.0.0.1	224.0.0.9	RIPv2	166 Response
474	2845.405408	40.0.0.1	224.0.0.9	RIPv2	166 Response

Imagen 4_b.5: Captura del tráfico de wireshark por la interfaz eth3 del router R4

- ▼ Routing Information Protocol
 - Command: Response (2)
 - Version: RIPv2 (2)
 - ▼ IP Address: 10.0.0.0, Metric: 3
 - Address Family: IP (2)
 - Route Tag: 0
 - IP Address: 10.0.0.0
 - Netmask: 255.0.0.0
 - Next Hop: 0.0.0.0
 - Metric: 3
 - ▼ IP Address: 11.0.0.0, Metric: 4
 - Address Family: IP (2)
 - Route Tag: 0
 - IP Address: 11.0.0.0
 - Netmask: 255.0.0.0
 - Next Hop: 0.0.0.0
 - Metric: 4
 - ▼ IP Address: 20.0.0.0, Metric: 2
 - Address Family: IP (2)
 - Route Tag: 0
 - IP Address: 20.0.0.0
 - Netmask: 255.0.0.0
 - Next Hop: 0.0.0.0
 - Metric: 2
 - ▼ IP Address: 30.0.0.0, Metric: 1
 - Address Family: IP (2)
 - Route Tag: 0
 - IP Address: 30.0.0.0
 - Netmask: 255.0.0.0
 - Next Hop: 0.0.0.0
 - Metric: 1
 - ▼ IP Address: 41.0.0.0, Metric: 1
 - Address Family: IP (2)
 - Route Tag: 0
 - IP Address: 41.0.0.0
 - Netmask: 255.0.0.0
 - Next Hop: 0.0.0.0
 - Metric: 1
 - ▼ IP Address: 50.0.0.0, Metric: 4
 - Address Family: IP (2)
 - Route Tag: 0
 - IP Address: 50.0.0.0
 - Netmask: 255.0.0.0
 - Next Hop: 0.0.0.0
 - Metric: 4

Imagen 4_b.6: Contenido de los mensajes de RESPONSE periódicos de R4

- c) Comprueba la tabla de encaminamiento del encaminador recién arrancado, así como las tablas de encaminamiento del resto de los encaminadores para ver cómo se van propagando las rutas. Explica el proceso de aprendizaje de rutas apoyándote en las capturas realizadas y en los mensajes de depuración.

RESPUESTA: Para mostrar las tablas de rutas de cada uno de los routers he utilizado la orden *show ip route*. En este caso , tal y como se puede ver en la imagen (4_b.5) R4 manda un mensaje REQUEST y un mensaje RESPONSE con la información de todas las redes a las que está directamente conectados (40.0.0.0 y 41.0.0.0) excepto la red 30.0.0.0, que será recibido por el router R3, quien aprenderá las nuevas rutas excepto la anterior ya que es la red a través de la cual está conectado a R4. Esto se debe a que R4 ,haciendo uso del mecanismo *Split Horizon*, al ser una ruta que ha aprendido a través de la interfaz que conecta con R3, no se lo envía en el mensaje RESPONSE.

Después de esto, R3 envía su vector de distancias completo a través de un mensaje RESPONSE (a excepción de las rutas 40.0.0.0 y 41.0.0.0 por lo explicado anteriormente y la ruta 30.0.0.0) que es recibido por el router R4 al estar suscrito al mismo grupo multicast de la dirección 224.0.0.9, quien anota las nuevas rutas (10.0.0.0, 11.0.0.0 y 20.0.0.0) en su tabla para poder acceder a ellas.

Como último paso del proceso, R4 enviará de manera periódica (cada 30 segundos) un mensaje *Trigger Update* con la información de las rutas aprendidas por las interfaces eth3 y eth4 (que son las que conectan R4 con R3 y R5 respectivamente) pero solo lo recibirá R3 ya que R5 tiene RIP desactivado por sus interfaces y la interfaz que conecta con PC2 quedo desactivada anteriormente para no trabajar con RIP.

El proceso y desarrollo del aprendizaje de rutas es análogo para el resto de routers.

```

C    50.0.0.0/8 is directly connected, GigabitEthernet4/0
R    20.0.0.0/8 [120/1] via 10.0.0.2, 00:00:17, GigabitEthernet3/0
R    40.0.0.0/8 [120/3] via 10.0.0.2, 00:00:17, GigabitEthernet3/0
C    10.0.0.0/8 is directly connected, GigabitEthernet3/0
R    41.0.0.0/8 [120/3] via 10.0.0.2, 00:00:17, GigabitEthernet3/0
C    11.0.0.0/8 is directly connected, FastEthernet0/0
R    30.0.0.0/8 [120/2] via 10.0.0.2, 00:00:17, GigabitEthernet3/0
R1#

```

Imagen 4_c.1: Tabla de rutas del router R1

```

R    50.0.0.0/8 [120/1] via 10.0.0.1, 00:00:16, GigabitEthernet3/0
C    20.0.0.0/8 is directly connected, GigabitEthernet4/0
R    40.0.0.0/8 [120/2] via 20.0.0.2, 00:00:15, GigabitEthernet4/0
C    10.0.0.0/8 is directly connected, GigabitEthernet3/0
R    41.0.0.0/8 [120/2] via 20.0.0.2, 00:00:15, GigabitEthernet4/0
R    11.0.0.0/8 [120/1] via 10.0.0.1, 00:00:16, GigabitEthernet3/0
R    30.0.0.0/8 [120/1] via 20.0.0.2, 00:00:16, GigabitEthernet4/0
R2#

```

Imagen 4_c.2: Tabla de rutas del router R2

```

R    50.0.0.0/8 [120/2] via 20.0.0.1, 00:00:03, GigabitEthernet3/0
C    20.0.0.0/8 is directly connected, GigabitEthernet3/0
R    40.0.0.0/8 [120/1] via 30.0.0.2, 00:00:27, GigabitEthernet4/0
R    10.0.0.0/8 [120/1] via 20.0.0.1, 00:00:03, GigabitEthernet3/0
R    41.0.0.0/8 [120/1] via 30.0.0.2, 00:00:27, GigabitEthernet4/0
R    11.0.0.0/8 [120/2] via 20.0.0.1, 00:00:03, GigabitEthernet3/0
C    30.0.0.0/8 is directly connected, GigabitEthernet4/0
R3#

```

Imagen 4_c.3: Tabla de rutas del router R3

```

R    50.0.0.0/8 [120/3] via 30.0.0.1, 00:00:16, GigabitEthernet3/0
R    20.0.0.0/8 [120/1] via 30.0.0.1, 00:00:16, GigabitEthernet3/0
C    40.0.0.0/8 is directly connected, GigabitEthernet4/0
R    10.0.0.0/8 [120/2] via 30.0.0.1, 00:00:16, GigabitEthernet3/0
C    41.0.0.0/8 is directly connected, FastEthernet0/0
R    11.0.0.0/8 [120/3] via 30.0.0.1, 00:00:16, GigabitEthernet3/0
C    30.0.0.0/8 is directly connected, GigabitEthernet3/0
R4#

```

Imagen 4_c.4: Tabla de rutas del router R4

R1#debug ip rip

RIP protocol debugging is on

R1#

*Mar 28 19:17:24.163: RIP: sending v2 update to 224.0.0.9 via GigabitEthernet3/0 (10.0.0.1)

*Mar 28 19:17:24.163: RIP: build update entries

*Mar 28 19:17:24.163: 11.0.0.0/8 via 0.0.0.0, metric 1, tag 0

*Mar 28 19:17:24.167: 50.0.0.0/8 via 0.0.0.0, metric 1, tag 0

R1#

*Mar 28 19:17:29.575: RIP: received v2 update from 10.0.0.2 on GigabitEthernet3/0

*Mar 28 19:17:29.575: 20.0.0.0/8 via 0.0.0.0 in 1 hops

*Mar 28 19:17:29.579: 30.0.0.0/8 via 0.0.0.0 in 2 hops

*Mar 28 19:17:29.579: 40.0.0.0/8 via 0.0.0.0 in 3 hops

*Mar 28 19:17:29.579: 41.0.0.0/8 via 0.0.0.0 in 3 hops

R1#

*Mar 28 19:17:32.215: RIP: sending v2 update to 224.0.0.9 via GigabitEthernet4/0 (50.0.0.2)

*Mar 28 19:17:32.215: RIP: build update entries

*Mar 28 19:17:32.215: 10.0.0.0/8 via 0.0.0.0, metric 1, tag 0

*Mar 28 19:17:32.215: 11.0.0.0/8 via 0.0.0.0, metric 1, tag 0

*Mar 28 19:17:32.219: 20.0.0.0/8 via 0.0.0.0, metric 2, tag 0

*Mar 28 19:17:32.219: 30.0.0.0/8 via 0.0.0.0, metric 3, tag 0

*Mar 28 19:17:32.219: 40.0.0.0/8 via 0.0.0.0, metric 4, tag 0

*Mar 28 19:17:32.223: 41.0.0.0/8 via 0.0.0.0, metric 4, tag 0

R1#

*Mar 28 19:17:52.311: RIP: sending v2 update to 224.0.0.9 via GigabitEthernet3/0 (10.0.0.1)

*Mar 28 19:17:52.311: RIP: build update entries

*Mar 28 19:17:52.311: 11.0.0.0/8 via 0.0.0.0, metric 1, tag 0

*Mar 28 19:17:52.315: 50.0.0.0/8 via 0.0.0.0, metric 1, tag 0

R1#

*Mar 28 19:17:59.151: RIP: received v2 update from 10.0.0.2 on GigabitEthernet3/0

*Mar 28 19:17:59.151: 20.0.0.0/8 via 0.0.0.0 in 1 hops

*Mar 28 19:17:59.155: 30.0.0.0/8 via 0.0.0.0 in 2 hops

*Mar 28 19:17:59.155: 40.0.0.0/8 via 0.0.0.0 in 3 hops

*Mar 28 19:17:59.155: 41.0.0.0/8 via 0.0.0.0 in 3 hops

R1#

```
*Mar 28 19:18:00.927: RIP: sending v2 update to 224.0.0.9 via GigabitEthernet4/0 (50.0.0.2)
*Mar 28 19:18:00.927: RIP: build update entries
*Mar 28 19:18:00.927: 10.0.0.0/8 via 0.0.0.0, metric 1, tag 0
*Mar 28 19:18:00.931: 11.0.0.0/8 via 0.0.0.0, metric 1, tag 0
*Mar 28 19:18:00.931: 20.0.0.0/8 via 0.0.0.0, metric 2, tag 0
*Mar 28 19:18:00.931: 30.0.0.0/8 via 0.0.0.0, metric 3, tag 0
*Mar 28 19:18:00.931: 40.0.0.0/8 via 0.0.0.0, metric 4, tag 0
*Mar 28 19:18:00.935: 41.0.0.0/8 via 0.0.0.0, metric 4, tag 0
```

- d) Comprueba la métrica de cada una de las rutas aprendidas. Detállalas a continuación.

RESPUESTA: Las redes a las que cada router está conectado directamente aparece con métrica 1, el resto de métricas representan el coste o número de saltos que tienen que hacer un datagrama para llegar a su destino. En este caso no es perceptible porque no se da el caso, pero las redes con métrica 16 representan un destino inalcanzable con coste infinito.

ROUTER 1:

```
*Mar 28 19:17:32.215: 10.0.0.0/8 via 0.0.0.0, metric 1, tag 0
*Mar 28 19:17:32.215: 11.0.0.0/8 via 0.0.0.0, metric 1, tag 0
*Mar 28 19:17:32.219: 20.0.0.0/8 via 0.0.0.0, metric 2, tag 0
*Mar 28 19:17:32.219: 30.0.0.0/8 via 0.0.0.0, metric 3, tag 0
*Mar 28 19:17:32.219: 40.0.0.0/8 via 0.0.0.0, metric 4, tag 0
*Mar 28 19:17:32.223: 41.0.0.0/8 via 0.0.0.0, metric 4, tag 0
```

ROUTER 2:

```
*Mar 28 19:16:04.751: 20.0.0.0/8 via 0.0.0.0, metric 1, tag 0
*Mar 28 19:16:04.755: 30.0.0.0/8 via 0.0.0.0, metric 2, tag 0
*Mar 28 19:16:04.755: 40.0.0.0/8 via 0.0.0.0, metric 3, tag 0
```

```
*Mar 28 19:16:04.755: 41.0.0.0/8 via 0.0.0.0, metric 3, tag 0
```

ROUTER 3:

```
*Mar 28 18:34:10.323: 30.0.0.0/8 via 0.0.0.0, metric 1, tag 0
```

```
*Mar 28 18:34:10.327: 40.0.0.0/8 via 0.0.0.0, metric 2, tag 0
```

```
*Mar 28 18:34:10.327: 41.0.0.0/8 via 0.0.0.0, metric 2, tag 0
```

```
*Mar 28 18:34:15.987: 10.0.0.0/8 via 0.0.0.0, metric 2, tag 0
```

```
*Mar 28 18:34:15.987: 11.0.0.0/8 via 0.0.0.0, metric 3, tag 0
```

```
*Mar 28 18:34:15.991: 20.0.0.0/8 via 0.0.0.0, metric 1, tag 0
```

```
*Mar 28 18:34:15.991: 50.0.0.0/8 via 0.0.0.0, metric 3, tag 0
```

ROUTER 4:

```
*Mar 28 19:03:32.427: 10.0.0.0/8 via 0.0.0.0, metric 3, tag 0
```

```
*Mar 28 19:03:32.427: 11.0.0.0/8 via 0.0.0.0, metric 4, tag 0
```

```
*Mar 28 19:03:32.431: 20.0.0.0/8 via 0.0.0.0, metric 2, tag 0
```

```
*Mar 28 19:03:32.431: 30.0.0.0/8 via 0.0.0.0, metric 1, tag 0
```

```
*Mar 28 19:03:32.431: 41.0.0.0/8 via 0.0.0.0, metric 1, tag 0
```

```
*Mar 28 19:03:32.435: 50.0.0.0/8 via 0.0.0.0, metric 4, tag 0
```

- e) ¿La implementación de RIP que utilizan los routers de Cisco está empleando el mecanismo *Split Horizon* o el mecanismo *Split Horizon + Poison Reverse*? ¿Cómo lo sabes?

RESPUESTA: Está utilizando *Split Horizon* porque se aplica este mecanismo , como explicamos en apartados anteriores, cuando un router manda por una interfaz en los mensajes **RESPONSE** las direcciones IP de todas las redes de su tabla de rutas excepto las rutan que ha aprendido por la interfaz de red por la que lo manda. Al contrario que el mecanismo que incluye *Poison Reverse*, en el cual el router sí anuncia las rutas que aprendido por esa interfaz, pero con coste 16 (infinito) indicando que no se puede alcanzar y que no se produzcan cuentas hasta el infinito , puesto que esa ruta la aprendido por la interfaz que la recibe el otro router.

5. Tras haber arrancado RIP en los encaminadores R1, R2, R3 y R4, PC1 y PC2 deberían tener conectividad IP. Compruébalo con las órdenes *ping* y *trace* (incluye aquí su salida).

RESPUESTA: Como todas las tablas de los routers están completas, cualquier comunicación entre los equipos PC1 y PC2 se realizará sin problema ya que los routers saben encaminar las direcciones IPs que reciben de ambos.

```
PC1> ping 41.0.0.10
84 bytes from 41.0.0.10 icmp_seq=1 ttl=60 time=86.768 ms
84 bytes from 41.0.0.10 icmp_seq=2 ttl=60 time=84.773 ms
84 bytes from 41.0.0.10 icmp_seq=3 ttl=60 time=84.772 ms
84 bytes from 41.0.0.10 icmp_seq=4 ttl=60 time=83.776 ms
84 bytes from 41.0.0.10 icmp_seq=5 ttl=60 time=83.777 ms
```

Imagen 5.1: Salida de la terminal del PC1 al hacer ping a PC2

```
PC2> ping 11.0.0.10
11.0.0.10 icmp_seq=1 timeout
84 bytes from 11.0.0.10 icmp_seq=2 ttl=60 time=81.781 ms
84 bytes from 11.0.0.10 icmp_seq=3 ttl=60 time=83.775 ms
84 bytes from 11.0.0.10 icmp_seq=4 ttl=60 time=83.776 ms
84 bytes from 11.0.0.10 icmp_seq=5 ttl=60 time=83.776 ms
```

Imagen 5.2: Salida de la terminal del PC2 al hacer ping a PC1

Deja lanzado el *ping* de PC1 a PC2 (*ping 40.0.0.10 -t*), y borra la tabla de rutas de R1 (*clear ip route **). ¿Se ha producido pérdida de paquetes? ¿Por qué? Comprueba lo que ha sucedido con las capturas de tráfico necesarias.

RESPUESTA: No, no se pierden paquetes porque en cuanto se borra la tabla de rutas de R1, manda varios mensajes de REQUEST a la dirección multicast 224.0.0.9 hasta que R2 le conteste con un mensaje RESPONSE para rellenar su tabla de rutas lo más rápido posible de manera que no se pierdan paquetes, mientras este proceso tiene lugar almacena los mensajes de ICMP recibidos durante unos segundos hasta que pueda volver a encaminarlos. A través de las capturas de wireshark , comparando los paquetes ICMP que entran en R1 y los que R1 deja pasar hasta R2 vemos que pasan todos y no se pierden ninguno.

4	69.635910	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0x8927, seq=1/256, ttl=64	(0
5	71.637561	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0x8b27, seq=2/512, ttl=64	(0
7	73.638214	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0x8d27, seq=3/768, ttl=64	(0
8	75.638868	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0x8f27, seq=4/1024, ttl=64	(0
9	77.640517	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0x9127, seq=5/1280, ttl=64	(0
10	79.641173	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0x9327, seq=6/1536, ttl=64	(0
11	81.641824	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0x9527, seq=7/1792, ttl=64	(0
13	83.643476	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0x9727, seq=8/2048, ttl=64	(0
14	85.645128	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0x9927, seq=9/2304, ttl=64	(0
15	87.646779	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0x9b27, seq=10/2560, ttl=64	(0
16	89.647433	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0x9d27, seq=11/2816, ttl=64	(0
17	91.649083	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0x9f27, seq=12/3072, ttl=64	(0
19	93.649737	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0xa127, seq=13/3328, ttl=64	(0
20	95.650391	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0xa327, seq=14/3584, ttl=64	(0
22	97.651044	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0xa527, seq=15/3840, ttl=64	(0
23	99.651698	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0xa727, seq=16/4096, ttl=64	(0
24	101.653349	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0xa927, seq=17/4352, ttl=64	(0
26	103.655000	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0xab27, seq=18/4608, ttl=64	(0
27	105.655653	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0xad27, seq=19/4864, ttl=64	(0
28	107.657315	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0xaf27, seq=20/5120, ttl=64	(0
29	109.657958	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0xb127, seq=21/5376, ttl=64	(0
30	111.658609	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0xb327, seq=22/5632, ttl=64	(0
32	113.660260	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0xb527, seq=23/5888, ttl=64	(0
33	115.661913	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0xb727, seq=24/6144, ttl=64	(0
34	117.663564	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0xb927, seq=25/6400, ttl=64	(0
35	119.665215	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0xbb27, seq=26/6656, ttl=64	(0
36	121.666866	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0xbd27, seq=27/6912, ttl=64	(0
38	123.668516	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0xbf27, seq=28/7168, ttl=64	(0
39	125.670168	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0xc127, seq=29/7424, ttl=64	(0
40	127.671819	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0xc327, seq=30/7680, ttl=64	(0
41	129.673470	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request	id=0xc527, seq=31/7936, ttl=64	(0

Imagen 5.3: Captura de wireshark de la interfaz eth0 de R1 (la que conecta con PC1)

5	44.934917	10.0.0.2	224.0.0.9	RIPv2	126	Response
6	45.187245	10.0.0.1	224.0.0.9	RIPv2	86	Response
9	69.684778	11.0.0.10	40.0.0.10	ICMP	98	Echo (ping) request id=0x8927, seq=1/256, ttl=63 (r
10	71.692414	11.0.0.10	40.0.0.10	ICMP	98	Echo (ping) request id=0x8b27, seq=2/512, ttl=63 (r
13	73.689879	11.0.0.10	40.0.0.10	ICMP	98	Echo (ping) request id=0x8d27, seq=3/768, ttl=63 (r
14	75.685741	11.0.0.10	40.0.0.10	ICMP	98	Echo (ping) request id=0x8f27, seq=4/1024, ttl=63 (r
15	77.693378	11.0.0.10	40.0.0.10	ICMP	98	Echo (ping) request id=0x9127, seq=5/1280, ttl=63 (r
16	78.417440	10.0.0.1	224.0.0.9	RIPv2	66	Request
17	78.417440	10.0.0.1	224.0.0.9	RIPv2	66	Request
18	78.428411	10.0.0.1	224.0.0.9	RIPv2	66	Request
19	78.428411	10.0.0.2	10.0.0.1	RIPv2	126	Response
20	78.428411	10.0.0.2	10.0.0.1	RIPv2	126	Response
21	78.439382	10.0.0.2	10.0.0.1	RIPv2	126	Response
22	79.690041	11.0.0.10	40.0.0.10	ICMP	98	Echo (ping) request id=0x9327, seq=6/1536, ttl=63 (r
23	80.315371	10.0.0.1	224.0.0.9	RIPv2	86	Response
24	81.686706	11.0.0.10	40.0.0.10	ICMP	98	Echo (ping) request id=0x9527, seq=7/1792, ttl=63 (r
26	83.693342	11.0.0.10	40.0.0.10	ICMP	98	Echo (ping) request id=0x9727, seq=8/2048, ttl=63 (r
27	85.690008	11.0.0.10	40.0.0.10	ICMP	98	Echo (ping) request id=0x9927, seq=9/2304, ttl=63 (r
28	87.697643	11.0.0.10	40.0.0.10	ICMP	98	Echo (ping) request id=0x9b27, seq=10/2560, ttl=63 (r
29	89.694308	11.0.0.10	40.0.0.10	ICMP	98	Echo (ping) request id=0x9d27, seq=11/2816, ttl=63 (r
30	91.701942	11.0.0.10	40.0.0.10	ICMP	98	Echo (ping) request id=0x9f27, seq=12/3072, ttl=63 (r
32	93.698606	11.0.0.10	40.0.0.10	ICMP	98	Echo (ping) request id=0xa127, seq=13/3328, ttl=63 (r
33	95.695269	11.0.0.10	40.0.0.10	ICMP	98	Echo (ping) request id=0xa327, seq=14/3584, ttl=63 (r
36	97.702904	11.0.0.10	40.0.0.10	ICMP	98	Echo (ping) request id=0xa527, seq=15/3840, ttl=63 (r
37	99.699570	11.0.0.10	40.0.0.10	ICMP	98	Echo (ping) request id=0xa727, seq=16/4096, ttl=63 (r
38	101.707205	11.0.0.10	40.0.0.10	ICMP	98	Echo (ping) request id=0xa927, seq=17/4352, ttl=63 (r
40	103.703869	11.0.0.10	40.0.0.10	ICMP	98	Echo (ping) request id=0xab27, seq=18/4608, ttl=63 (r
41	105.700531	11.0.0.10	40.0.0.10	ICMP	98	Echo (ping) request id=0xad27, seq=19/4864, ttl=63 (r
42	107.532637	10.0.0.1	224.0.0.9	RIPv2	86	Response
43	107.708168	11.0.0.10	40.0.0.10	ICMP	98	Echo (ping) request id=0xaf27, seq=20/5120, ttl=63 (r
44	109.704830	11.0.0.10	40.0.0.10	ICMP	98	Echo (ping) request id=0xb127, seq=21/5376, ttl=63 (r
45	111.713467	11.0.0.10	40.0.0.10	ICMP	98	Echo (ping) request id=0xb327, seq=22/5632, ttl=63 (r

Imagen 5.4: Captura de wireshark de la interfaz eth3 de R1 (la que conecta con R2)

6. A continuación, realiza los cambios necesarios para que la ruta seguida por los datagramas IP que envía PC1 a PC2 sea PC1 => R1 => R5 => R4 => PC2, y para que los que envía PC2 a PC1 sigan la ruta PC2 => R4 => R5 => R1 => PC1. Para realizar este apartado no podrás añadir o eliminar manualmente rutas en las tablas de encaminamiento. Describe las acciones realizadas.

RESPUESTA: Para conseguir que los datagramas IP vayan por la ruta requerida hemos activado el protocolo RIP por las dos interfaces de R5 con la siguiente configuración:

```
R5#enable
R5#config t
Enter configuration commands, one per line. End with CNTL/Z.
R5(config)#router rip
R5(config-router)#version 2
R5(config-router)#network 40.0.0.0
R5(config-router)#network 50.0.0.0
R5(config-router)#no auto-summary
R5(config-router)#exit
R5(config)#exit
R5#wr
*Mar 30 16:38:12.863: %SYS-5-CONFIG_I: Configured from console by console
R5#wr
```

Imagen 6.1: Configuración del router R5 para el protocolo RIP

De manera que cuando PC1 manda mensajes a PC2, la tabla de rutas R1 cambia debido a que cuando R5 activa el protocolo RIP en sus dos interfaces, después de enviar el consecuente mensaje REQUEST para rellenar su tabla de rutas, envía un mensaje RESPONSE de actualización de tablas o *Trigger Update* cuyo contenido es únicamente la ruta hacia una red nueva o un cambio en la métrica de una ruta ya existente que es recibido por R1, en este caso el contenido sería el de la subred 41.0.0.0 con un nuevo valor de métrica.

Es entonces cuando R1 , al recibir el mensaje, actualiza el valor de la métrica hacia la subred 41.0.0.0 , que es la subred en la que está alojado PC2, reduciendo el número de saltos ya que a través de R5 se ahorra mandar los mensajes por otros dos routers más (R2 y R3) siguiendo la ruta de mínimo coste para llegar al destino.

El mismo efecto es observado cuando se hace en la dirección opuesta y es PC2 el que envía los datagramas IP a PC1, ya que vemos que el router R4 actualiza su tabla de rutas con un nuevo valor de métrica con menor número de saltos para llegar hasta la subred 11.0.0.0 ,donde está alojado PC1, y que es recibido de la misma manera que la mencionada anteriormente, a través de un mensaje RESPONSE de actualización explícita de tablas o *Trigger Update*.

Mirando la tabla de encaminamiento de R1, observa y apunta el número de segundos que aproximadamente tarda en aprenderse la nueva ruta. ¿Por qué?

RESPUESTA: Como podemos observar en la imagen (6.2) las rutas que menos ha tardado en aprender R1 son las rutas que ha aprendido a través de la dirección 50.0.0.1 de la interfaz eth4 de R5, aquella ruta cuya métrica es menor o tiene menor número de saltos, a través de un mensaje *Trigger Update*.

En total 12 segundos frente a los 14 segundos que ha tardado en recibir los mensajes de actualización para las rutas 20.0.0.0 y la otra ruta de la dirección 30.0.0.0, que las recibe a través de un mensaje RESPONSE desde la ruta por las que atraviesa los router R2 y R3, la cual , como mencionamos anteriormente, descartó el router R1 debido a que suponía un mayor coste o mayor número de saltos para llegar a las subredes 30.0.0.0, 40.0.0.0 y 41.0.0.0.

De manera que R1 actualiza su tabla de rutas con un nuevo valor de metricas para las direcciones 40.0.0.0 y 41.0.0.0 (pasando como se puede comparar en las imágenes (4) y (5) de un valor de métrica 4 a un valor de 2) y un valor de métrica igual para la subred 30.0.0.0 (ya que cuesta lo mismo llegar hasta ella atravesando la ruta por el router R2 que la ruta por el router R5)

```
R1#show ip route rip
R    20.0.0.0/8 [120/1] via 10.0.0.2, 00:00:14, GigabitEthernet3/0
R    40.0.0.0/8 [120/1] via 50.0.0.1, 00:00:12, GigabitEthernet4/0
R    41.0.0.0/8 [120/2] via 50.0.0.1, 00:00:12, GigabitEthernet4/0
R    30.0.0.0/8 [120/2] via 50.0.0.1, 00:00:12, GigabitEthernet4/0
      [120/2] via 10.0.0.2, 00:00:14, GigabitEthernet3/0
```

Imagen 6.2: Tabla de rutas del router R5

57	464.776070	50.0.0.1	224.0.0.9	RIPv2	66 Request
63	482.734079	50.0.0.1	224.0.0.9	RIPv2	66 Response
69	513.791085	50.0.0.1	224.0.0.9	RIPv2	66 Response
73	528.577571	50.0.0.1	224.0.0.9	RIPv2	86 Response
76	540.532624	50.0.0.1	224.0.0.9	RIPv2	106 Response
81	568.229608	50.0.0.1	224.0.0.9	RIPv2	106 Response
88	597.858554	50.0.0.1	224.0.0.9	RIPv2	106 Response
94	627.322816	50.0.0.1	224.0.0.9	RIPv2	106 Response
101	657.137144	50.0.0.1	224.0.0.9	RIPv2	106 Response
107	686.290237	50.0.0.1	224.0.0.9	RIPv2	106 Response
112	714.662417	50.0.0.1	224.0.0.9	RIPv2	106 Response

Imagen 6.3: Captura de WireShark en la interfaz eth4 del router R5

```

Routing Information Protocol
  Command: Response (2)
  Version: RIPv2 (2)
  > IP Address: 30.0.0.0, Metric: 2
  > IP Address: 40.0.0.0, Metric: 1
  > IP Address: 41.0.0.0, Metric: 2

```

Imagen 6.4: Contenido del mensaje RESPONSE nº 76 (el primer Trigger Update de la secuencia de mensajes con un nuevo valor de métricas)

1	0.000000	50.0.0.2	224.0.0.9	RIPv2	86 Response
8	55.193507	50.0.0.2	224.0.0.9	RIPv2	166 Response
13	118.461434	50.0.0.2	224.0.0.9	RIPv2	166 Response
22	182.517311	50.0.0.2	224.0.0.9	RIPv2	166 Response


```

<
> Frame 1: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
> Ethernet II, Src: ca:01:21:3c:00:70 (ca:01:21:3c:00:70), Dst: IPv4mcast_09 (01:00:5e:00:00:09)
> Internet Protocol Version 4, Src: 50.0.0.2, Dst: 224.0.0.9
> User Datagram Protocol, Src Port: 520, Dst Port: 520
Routing Information Protocol
  Command: Response (2)
  Version: RIPv2 (2)
  > IP Address: 40.0.0.0, Metric: 4
  > IP Address: 41.0.0.0, Metric: 4

```

Imagen 6.5: Contenido del mensaje RESPONSE nº 1 (con un valor de métrica aún no actualizado)

Comprueba que se está utilizando dicha ruta a través de la orden *trace*.

RESPUESTA: Como podemos observar en la imagen (6.6), tal y como explicamos anteriormente, la ruta que toma el router R1 para encaminar cualquier datagrama IP hacia la dirección IP del equipo PC2 ha cambiado, y ahora es a través de la interfaz con la que R1 está conectada a R2 ya que el coste o número de saltos es menor.

```

PC1> trace 41.0.0.10
Trace to 41.0.0.10, 8 hops max, press Ctrl+C to stop
 1  11.0.0.1    1.995 ms  8.975 ms  9.973 ms
 2  50.0.0.1    30.918 ms  31.916 ms  30.916 ms
 3  40.0.0.1    53.856 ms  52.859 ms  53.856 ms
 4  *41.0.0.10  63.828 ms (ICMP type:3, code:3, Destination port unreachable)

```

Imagen 6.6: Salida del terminal del equipo PC1 al ejecutar la orden 'trace' hacia PC2

El mismo efecto ocurre si lo aplicamos en la dirección opuesta, desde PC2 hasta PC1 utilizando la orden *trace*, los datagramas son encaminados pasando por el router R5.

```

PC2> trace 11.0.0.10
Trace to 11.0.0.10, 8 hops max, press Ctrl+C to stop
 1  41.0.0.1    9.975 ms  8.976 ms  9.973 ms
 2  40.0.0.2    30.917 ms  31.915 ms  30.918 ms
 3  50.0.0.2    53.856 ms  53.856 ms  52.859 ms
 4  * * *
 5  *11.0.0.10  57.845 ms (ICMP type:3, code:3, Destination port unreachable)

```

Imagen 6.7: Salida del terminal del equipo PC2 al ejecutar la orden 'trace' hacia PC1

Comprueba las rutas y sus métricas en las tablas de encaminamiento de cada encaminador. Inclúyelas aquí. Comenta la salida.

RESPUESTA: Cada tabla de rutas de cada router muestra todas las entradas a las redes que están directamente conectadas y las redes que han aprendido a partir de alguna de sus interfaces conectadas al resto de routers vecinos.

En las tablas de rutas del router R1 se mantienen todas las entradas de puntos anteriores a las que estaba directamente conectadas y las que ha recibido de las rutas que atraviesa el router R2 (20.0.0.0). Los cambios que podemos observar son en las entradas de las redes que ha recibido de R5, que han actualizado el coste que supone llegar a ellas a uno menor que el que tenían antes encaminando los datagramas hacia dichas redes por la ruta que pasaba por R2 y R3, donde comparando con el tiempo que han recibido esas entradas con el resto de entradas que ya tenían se nota una mejoría significativa de tiempo debido al menor número de saltos que tiene que hacer el datagrama para llegar a su destino (4 segundos frente a 14 segundos):

40.0.0.0 con métrica 1

41.0.0.0 con métrica 2

30.0.0.0 con métrica 2 (la primera ruta)

```

C    50.0.0.0/8 is directly connected, GigabitEthernet4/0
R    20.0.0.0/8 [120/1] via 10.0.0.2, 00:00:14, GigabitEthernet3/0
R    40.0.0.0/8 [120/1] via 50.0.0.1, 00:00:04, GigabitEthernet4/0
C    10.0.0.0/8 is directly connected, GigabitEthernet3/0
R    41.0.0.0/8 [120/2] via 50.0.0.1, 00:00:04, GigabitEthernet4/0
C    11.0.0.0/8 is directly connected, FastEthernet0/0
R    30.0.0.0/8 [120/2] via 50.0.0.1, 00:00:04, GigabitEthernet4/0
      [120/2] via 10.0.0.2, 00:00:14, GigabitEthernet3/0
R1#

```

Imagen 6.8: Tabla de rutas del router R1

En las tablas de rutas de los routers R2 y R3 no se notan diferencias significativas tras haber activado el protocolo RIP en el router R5 puesto que no afecta al valor de la métrica de sus redes en sus tablas de rutas.

En cambio en el router R4 se observa el mismo cambio que en el Router R1 ya que se han visto modificadas aquellas redes que R4 recibía a través de la interfaz que conectaba con el router R3 y ahora recibe pasando por R5 con un coste menor. En concreto:

10.0.0.0 con métrica 2 (la primera ruta que tiene apuntada que corresponde con la que recibe desde el mensaje del router R5)

11.0.0.0 con métrica 2

50.0.0.0 con métrica 1

```

R    50.0.0.0/8 [120/1] via 10.0.0.1, 00:00:10, GigabitEthernet3/0
C    20.0.0.0/8 is directly connected, GigabitEthernet4/0
R    40.0.0.0/8 [120/2] via 20.0.0.2, 00:00:20, GigabitEthernet4/0
      [120/2] via 10.0.0.1, 00:00:10, GigabitEthernet3/0
C    10.0.0.0/8 is directly connected, GigabitEthernet3/0
R    41.0.0.0/8 [120/2] via 20.0.0.2, 00:00:20, GigabitEthernet4/0
R    11.0.0.0/8 [120/1] via 10.0.0.1, 00:00:10, GigabitEthernet3/0
R    30.0.0.0/8 [120/1] via 20.0.0.2, 00:00:21, GigabitEthernet4/0
R2#

```

Imagen 6.9: Tabla de rutas del router R2

```

R    50.0.0.0/8 [120/2] via 30.0.0.2, 00:00:26, GigabitEthernet4/0
      [120/2] via 20.0.0.1, 00:00:11, GigabitEthernet3/0
C    20.0.0.0/8 is directly connected, GigabitEthernet3/0
R    40.0.0.0/8 [120/1] via 30.0.0.2, 00:00:26, GigabitEthernet4/0
R    10.0.0.0/8 [120/1] via 20.0.0.1, 00:00:11, GigabitEthernet3/0
R    41.0.0.0/8 [120/1] via 30.0.0.2, 00:00:26, GigabitEthernet4/0
R    11.0.0.0/8 [120/2] via 20.0.0.1, 00:00:11, GigabitEthernet3/0
C    30.0.0.0/8 is directly connected, GigabitEthernet4/0
R3#

```

Imagen 6.10: Tabla de rutas del router R3

```

C    50.0.0.0/8 is directly connected, GigabitEthernet4/0
R    20.0.0.0/8 [120/2] via 50.0.0.2, 00:00:27, GigabitEthernet4/0
      [120/2] via 40.0.0.1, 00:00:08, GigabitEthernet3/0
C    40.0.0.0/8 is directly connected, GigabitEthernet3/0
R    10.0.0.0/8 [120/1] via 50.0.0.2, 00:00:27, GigabitEthernet4/0
R    41.0.0.0/8 [120/1] via 40.0.0.1, 00:00:08, GigabitEthernet3/0
R    11.0.0.0/8 [120/1] via 50.0.0.2, 00:00:27, GigabitEthernet4/0
R    30.0.0.0/8 [120/1] via 40.0.0.1, 00:00:08, GigabitEthernet3/0
R5#

```

Imagen 6.11: Tabla de rutas del router R4

```

R    50.0.0.0/8 [120/1] via 40.0.0.2, 00:00:11, GigabitEthernet4/0
R    20.0.0.0/8 [120/1] via 30.0.0.1, 00:00:11, GigabitEthernet3/0
C    40.0.0.0/8 is directly connected, GigabitEthernet4/0
R    10.0.0.0/8 [120/2] via 40.0.0.2, 00:00:11, GigabitEthernet4/0
      [120/2] via 30.0.0.1, 00:00:11, GigabitEthernet3/0
C    41.0.0.0/8 is directly connected, FastEthernet0/0
R    11.0.0.0/8 [120/2] via 40.0.0.2, 00:00:11, GigabitEthernet4/0
C    30.0.0.0/8 is directly connected, GigabitEthernet3/0
R4#

```

Imagen 6.12: Tabla de rutas del router R5

7. ¿Por qué ruta deberían ir los datagramas IP que envíe PC1 a la dirección 30.0.0.2? Justifica la respuesta. Comprueba tu respuesta utilizando `trace`. Con la misma red del escenario, y con los 5 encaminadores con RIP activado, ¿podrían haber seguido otra ruta los datagramas IP PC1 a la dirección 30.0.0.2? ¿Cómo actúan los routers CISCO cuando reciben una ruta con igual métrica que la que ya tienen en su tabla de rutas?

RESPUESTA: los datagramas IP deberían ir por cualquiera de las dos rutas posibles hasta la dirección 30.0.0.2, que como ya mencioné anteriormente, son a través de R5 o a través de R2 ya que todas las direcciones de la red 30.0.0.0 tienen el mismo coste o la misma métrica yendo por cualquiera de las dos rutas.

En nuestro caso, como se puede ver en la figura (7.1), al ejecutar el comando `trace` desde el equipo PC1 hasta la dirección 30.0.0.2, el router R1 decide tomar la ruta pasando a través del router R5 ,en vez de la ruta a través de R2. Aunque los datagramas podrían haber tomado otra ruta y haber sido encaminados a través de R2 de igual manera ya que , como dije anteriormente, conlleva el mismo coste que encaminar los paquetes pasando por el router R5.

```

PC1> trace 30.0.0.2
trace to 30.0.0.2, 8 hops max, press Ctrl+C to stop
 1  11.0.0.1   9.975 ms  8.976 ms  9.975 ms
 2  50.0.0.1  41.889 ms 52.859 ms 42.884 ms
 3  *40.0.0.1 53.856 ms (ICMP type:3, code:3, Destination port unreachable)

```

Imagen 7.1: Salida del terminal del equipo PC1 al ejecutar la orden 'trace' hacia la dirección 30.0.0.2

De igual manera, como ya mostré en la imagen (6.2) del anterior apartado, los routers de CISCO guardan dos o más rutas hacia una misma red si estas tienen igual métrica (como es el caso de la red 30.0.0.0 en R1 que guarda en su tabla de rutas las dos rutas para llegar hasta esa dirección) llevando a direccionarlas por una o por otra ruta sin ningún criterio definido.

1.4. Eliminación de rutas

El objetivo de este apartado es observar lo que ocurre cuando se interrumpe *RIP* en R5. Se estudiará, en particular, el comportamiento de los encaminadores R1 y R4.

1. Asegúrate de que los 5 encaminadores tienen arrancado *RIP*. Comprueba la ruta que están siguiendo los mensajes intercambiados entre PC1 y PC2 con `trace`. Incluye aquí la salida.

RESPUESTA:

```
PC1> trace 41.0.0.10
trace to 41.0.0.10, 8 hops max, press Ctrl+C to stop
 1  11.0.0.1    1.995 ms  8.975 ms  9.973 ms
 2  50.0.0.1    30.918 ms  31.916 ms  30.916 ms
 3  40.0.0.1    53.856 ms  52.859 ms  53.856 ms
 4  *41.0.0.10  63.828 ms (ICMP type:3, code:3, Destination port unreachable)
```

Imagen 1.1: Salida del terminal del equipo PC1 al ejecutar la orden 'trace' hacia PC2

```
PC2> trace 11.0.0.10
trace to 11.0.0.10, 8 hops max, press Ctrl+C to stop
 1  41.0.0.1    9.975 ms  8.976 ms  9.973 ms
 2  40.0.0.2    30.917 ms  31.915 ms  30.918 ms
 3  50.0.0.2    53.856 ms  53.856 ms  52.859 ms
 4  * * *
 5  *11.0.0.10  57.845 ms (ICMP type:3, code:3, Destination port unreachable)
```

Imagen 1.2: Salida del terminal del equipo PC2 al ejecutar la orden 'trace' hacia PC1

2. Deja en ejecución en PC1 un *ping* hacia PC2.
3. Para ver los mensajes RIP que envían R1 y R4, arranca *wireshark* en sus interfaces g3/0 y activa los mensajes de depuración.

```
R1#debug ip rip
```

```
RIP protocol debugging is on
```

R1#

*Mar 30 18:13:02.547: RIP: received v2 update from 50.0.0.1 on GigabitEthernet4/0

*Mar 30 18:13:02.547: 30.0.0.0/8 via 0.0.0.0 in 2 hops

*Mar 30 18:13:02.551: 40.0.0.0/8 via 0.0.0.0 in 1 hops

*Mar 30 18:13:02.551: 41.0.0.0/8 via 0.0.0.0 in 2 hops

R1#

*Mar 30 18:13:13.483: RIP: sending v2 update to 224.0.0.9 via GigabitEthernet4/0 (50.0.0.2)

*Mar 30 18:13:13.483: RIP: build update entries

*Mar 30 18:13:13.483: 10.0.0.0/8 via 0.0.0.0, metric 1, tag 0

*Mar 30 18:13:13.483: 11.0.0.0/8 via 0.0.0.0, metric 1, tag 0

*Mar 30 18:13:13.487: 20.0.0.0/8 via 0.0.0.0, metric 2, tag 0

R1#

*Mar 30 18:13:14.903: RIP: received v2 update from 10.0.0.2 on GigabitEthernet3/0

*Mar 30 18:13:14.903: 20.0.0.0/8 via 0.0.0.0 in 1 hops

*Mar 30 18:13:14.907: 30.0.0.0/8 via 0.0.0.0 in 2 hops

*Mar 30 18:13:14.907: 41.0.0.0/8 via 0.0.0.0 in 3 hops

*Mar 30 18:13:15.215: RIP: sending v2 update to 224.0.0.9 via GigabitEthernet3/0 (10.0.0.1)

*Mar 30 18:13:15.215: RIP: build update entries

*Mar 30 18:13:15.215: 11.0.0.0/8 via 0.0.0.0, metric 1, tag 0

*Mar 30 18:13:15.219: 40.0.0.0/8 via 0.0.0.0, metric 2, tag 0

*Mar 30 18:13:15.219: 41.0.0.0/8 via 0.0.0.0, metric 3, tag 0

*Mar 30 18:13:15.219: 50.0.0.0/8 via 0.0.0.0, metric 1, tag 0

R1#

*Mar 30 18:13:29.527: RIP: received v2 update from 50.0.0.1 on GigabitEthernet4/0

*Mar 30 18:13:29.527: 30.0.0.0/8 via 0.0.0.0 in 2 hops

*Mar 30 18:13:29.531: 40.0.0.0/8 via 0.0.0.0 in 1 hops

*Mar 30 18:13:29.531: 41.0.0.0/8 via 0.0.0.0 in 2 hops

R1#

*Mar 30 18:13:40.807: RIP: sending v2 update to 224.0.0.9 via GigabitEthernet3/0 (10.0.0.1)

*Mar 30 18:13:40.807: RIP: build update entries

*Mar 30 18:13:40.807: 11.0.0.0/8 via 0.0.0.0, metric 1, tag 0

*Mar 30 18:13:40.811: 40.0.0.0/8 via 0.0.0.0, metric 2, tag 0

*Mar 30 18:13:40.811: 41.0.0.0/8 via 0.0.0.0, metric 3, tag 0

*Mar 30 18:13:40.811: 50.0.0.0/8 via 0.0.0.0, metric 1, tag 0

R1#

*Mar 30 18:13:41.987: RIP: sending v2 update to 224.0.0.9 via GigabitEthernet4/0 (50.0.0.2)

*Mar 30 18:13:41.987: RIP: build update entries

*Mar 30 18:13:41.987: 10.0.0.0/8 via 0.0.0.0, metric 1, tag 0

*Mar 30 18:13:41.987: 11.0.0.0/8 via 0.0.0.0, metric 1, tag 0

*Mar 30 18:13:41.991: 20.0.0.0/8 via 0.0.0.0, metric 2, tag 0

*Mar 30 18:13:42.579: RIP: received v2 update from 10.0.0.2 on GigabitEthernet3/0

*Mar 30 18:13:42.579: 20.0.0.0/8 via 0.0.0.0 in 1 hops

*Mar 30 18:13:42.583: 30.0.0.0/8 via 0.0.0.0 in 2 hops

*Mar 30 18:13:42.583: 41.0.0.0/8 via 0.0.0.0 in 3 hops

R1#

*Mar 30 18:13:55.947: RIP: received v2 update from 50.0.0.1 on GigabitEthernet4/0

*Mar 30 18:13:55.947: 30.0.0.0/8 via 0.0.0.0 in 2 hops

*Mar 30 18:13:55.951: 40.0.0.0/8 via 0.0.0.0 in 1 hops

*Mar 30 18:13:55.951: 41.0.0.0/8 via 0.0.0.0 in 2 hops

R1#

*Mar 30 18:14:07.359: RIP: sending v2 update to 224.0.0.9 via GigabitEthernet3/0 (10.0.0.1)

*Mar 30 18:14:07.359: RIP: build update entries

*Mar 30 18:14:07.359: 11.0.0.0/8 via 0.0.0.0, metric 1, tag 0

*Mar 30 18:14:07.363: 40.0.0.0/8 via 0.0.0.0, metric 2, tag 0

*Mar 30 18:14:07.363: 41.0.0.0/8 via 0.0.0.0, metric 3, tag 0

*Mar 30 18:14:07.363: 50.0.0.0/8 via 0.0.0.0, metric 1, tag 0

R1#

*Mar 30 18:14:09.811: RIP: sending v2 update to 224.0.0.9 via GigabitEthernet4/0 (50.0.0.2)

*Mar 30 18:14:09.811: RIP: build update entries

*Mar 30 18:14:09.811: 10.0.0.0/8 via 0.0.0.0, metric 1, tag 0

*Mar 30 18:14:09.815: 11.0.0.0/8 via 0.0.0.0, metric 1, tag 0

*Mar 30 18:14:09.815: 20.0.0.0/8 via 0.0.0.0, metric 2, tag 0

R1#

*Mar 30 18:14:11.295: RIP: received v2 update from 10.0.0.2 on GigabitEthernet3/0

*Mar 30 18:14:11.295: 20.0.0.0/8 via 0.0.0.0 in 1 hops

*Mar 30 18:14:11.299: 30.0.0.0/8 via 0.0.0.0 in 2 hops

*Mar 30 18:14:11.299: 41.0.0.0/8 via 0.0.0.0 in 3 hops

R1#

*Mar 30 18:14:36.511: RIP: sending v2 update to 224.0.0.9 via GigabitEthernet4/0 (50.0.0.2)

*Mar 30 18:14:36.511: RIP: build update entries

*Mar 30 18:14:36.511: 10.0.0.0/8 via 0.0.0.0, metric 1, tag 0

*Mar 30 18:14:36.511: 11.0.0.0/8 via 0.0.0.0, metric 1, tag 0

*Mar 30 18:14:36.515: 20.0.0.0/8 via 0.0.0.0, metric 2, tag 0

*Mar 30 18:14:36.719: RIP: sending v2 update to 224.0.0.9 via GigabitEthernet3/0 (10.0.0.1)

*Mar 30 18:14:36.719: RIP: build update entries

*Mar 30 18:14:36.719: 11.0.0.0/8 via 0.0.0.0, metric 1, tag 0

*Mar 30 18:14:36.723: 40.0.0.0/8 via 0.0.0.0, metric 2, tag 0

*Mar 30 18:14:36.723: 41.0.0.0/8 via 0.0.0.0, metric 3, tag 0

R1#

*Mar 30 18:14:36.723: 50.0.0.0/8 via 0.0.0.0, metric 1, tag 0

R1#

*Mar 30 18:14:40.711: RIP: received v2 update from 10.0.0.2 on GigabitEthernet3/0

*Mar 30 18:14:40.711: 20.0.0.0/8 via 0.0.0.0 in 1 hops

*Mar 30 18:14:40.715: 30.0.0.0/8 via 0.0.0.0 in 2 hops

4. A continuación, interrumpe la ejecución de *RIP* en el encaminador R5 utilizando la orden *no router rip*. Podrás observar con la orden *show ip route* que ahora R5 no conoce rutas aprendidas por RIP. Tampoco exporta rutas hacia otros encaminadores.

RESPUESTA: Porque se ha desactivado el protocolo RIP por sus interfaces y por tanto han pasado más de 3 minutos, que es el tiempo que se le da al protocolo para el envío de mensajes periódicos de tipo *RESPONSE*, y la ruta hacia R5 queda obsoleta por lo que deja de recibir actualizaciones y únicamente conoce en su tabla de rutas, las dos redes a las que está directamente conectado a través de sus interfaces.

```
C    50.0.0.0/8 is directly connected, GigabitEthernet4/0
C    40.0.0.0/8 is directly connected, GigabitEthernet3/0
R5#show ip route rip
R5#
```

Imagen 4.1: Salida del terminal del router R5 al ejecutar la orden 'show ip route rip' y 'show ip route'

5. Observarás que el *ping* de PC1 a PC2 deja de funcionar durante un buen rato. Observa durante este periodo en el que no está funcionando RIP en R5, las entradas de las tablas de encaminamiento de R1 y de R4. Observa la evolución de la columna de tiempo de las distintas entradas. ¿Qué entradas no reinician la cuenta cada 30 segundos? ¿Por qué? Observa el valor de tiempo de esas entradas. Presta especial atención cuanto el valor de la columna de tiempo de algunas entradas de R1 y R4 se acerquen a 3 minutos ¿qué ocurrirá después?

RESPUESTA: Tras apagar el router R5, las entradas que no reinician la cuenta cada 30 segundos son las que R1 y R4 recibían a partir de la ruta que atravesaba este router, es decir:

Para R1: redes 30.0.0.0, 40.0.0.0 y 41.0.0.0

Para R2: redes 10.0.0.0, 11.0.0.0 y 20.0.0.0

Esto se debe a lo que comentamos en el apartado anterior, el protocolo RIP establece para cada uno de los routers que implementen o tengan activado este protocolo un tiempo de actualización de 30 segundos para recibir por cada una de sus interfaces

mensajes de tipo RESPONSE desde sus vecinos de manera periódica para que actualice sus tablas de rutas con las redes que aprendió por cada una de sus interfaces.

En cuanto al valor de tiempo de las entradas, como se puede observar en las imágenes siguientes, . Esto ocurre porque el protocolo RIP establece un tiempo máximo de 180 segundos (3 minutos) para recibir un mensaje de tipo RESPONSE por sus interfaces, procedentes de sus routers vecinos que también tienen activado el protocolo RIP. En caso de rebasar este límite, si un router no ha recibido ningún mensaje RESPONSE periódico de este tipo para actualizar su tabla de rutas, éste elimina la ruta por la que no recibe mensaje de su tabla y esa ruta queda desechada a la hora de encaminar paquetes.

En la siguiente figura, se muestran los tiempos de recepción de las redes que R1 recibía a través de la ruta de R5 (que son las que están marcadas en azul) y que superan los 30 segundos de refresco y están cerca de rebasar los 3 minutos, a partir de los cuales desearán esa ruta y la cambiarán por la que atraviesa los routers R2 y R3.

R1#show ip route

```
C 50.0.0.0/8 is directly connected, GigabitEthernet4/0
R 20.0.0.0/8 [120/1] via 10.0.0.2, 00:00:22, GigabitEthernet3/0
R 40.0.0.0/8 [120/1] via 50.0.0.1, 00:02:05, GigabitEthernet4/0
C 10.0.0.0/8 is directly connected, GigabitEthernet3/0
R 41.0.0.0/8 [120/2] via 50.0.0.1, 00:02:05, GigabitEthernet4/0
C 11.0.0.0/8 is directly connected, FastEthernet0/0
R 30.0.0.0/8 [120/2] via 50.0.0.1, 00:02:05, GigabitEthernet4/0
    [120/2] via 10.0.0.2, 00:00:22, GigabitEthernet3/0
```

Aquí ya se puede apreciar el cambio de ruta de la 50.0.0.1 que es la que conectaba R1 con el router R5 por la ruta de la dirección 10.0.0.2 , que es la dirección de la interfaz que conecta R1 con el router R2. Así como el cambio en el valor del coste para llegar a esas rutas que vuelve a aumentar con respecto a la anterior ruta.

R1#show ip route

```
C 50.0.0.0/8 is directly connected, GigabitEthernet4/0
```

```
R 20.0.0.0/8 [120/1] via 10.0.0.2, 00:00:14, GigabitEthernet3/0
R 40.0.0.0/8 [120/3] via 10.0.0.2, 00:00:14, GigabitEthernet3/0
C 10.0.0.0/8 is directly connected, GigabitEthernet3/0
R 41.0.0.0/8 [120/3] via 10.0.0.2, 00:00:14, GigabitEthernet3/0
C 11.0.0.0/8 is directly connected, FastEthernet0/0
R 30.0.0.0/8 [120/2] via 10.0.0.2, 00:00:14, GigabitEthernet3/0
```

ANOTACION: de manera análoga ocurre para el router R4.

Poco después el *ping* entre PC1 y PC2 habrá vuelto a funcionar. ¿Por qué? Interrúmpelo y, mirando los valores del *icmp_seq* apunta el número de segundos que aproximadamente ha estado sin funcionar el *ping* debido a que aún no se había olvidado la ruta a través de R5. Comprueba la ruta que están siguiendo los mensajes intercambiados entre PC1 y PC2 con *trace*. Incluye aquí la salida.

RESPUESTA: El *ping* ha vuelto a funcionar por lo que comentamos en apartados anteriores. Los routers R1 y R4 conectados directamente a R5 por una de sus interfaces respectivamente, tenían dos rutas para encaminar los datagramas IP que recibieran, una pasando por R5 y otra pasando por R2 y R3 y acabaron actualizando su tabla de rutas llevando los paquetes por la Ruta de R5 debido a un menor número de saltos con respecto a la otra ruta.

En este caso lo que ocurría es que, una vez desactivamos R5, se perdieron paquetes ICMP que se mandaban a través de la ruta que pasaba por el router R5 ya que, tanto R1 como R4, tenían como ruta por defecto para encaminar estos paquetes a través de R5. Y no sería hasta pasados 3 minutos (como ya explicamos en el apartado anterior) que R1 y R4 no sabrían que R5 había desactivado el protocolo RIP por sus interfaces, desechando la ruta que pasa a través de él y eliminando dicha ruta de las tablas de dichos routers. Es entonces, cuando podían retomar el envío de paquetes ICMP por la otra ruta que tenían apuntada, que pasaba por R2 y R3, en sus tablas de rutas a las redes que les llegaban a través de la ruta de R5.

```
PC1> ping 41.0.0.10 -t
41.0.0.10 icmp_seq=1 timeout
41.0.0.10 icmp_seq=2 timeout
```

Imagen 5.1: Salida del terminal del equipo PC1 al enviar el *ping* al PC2 (sin respuesta)

```

41.0.0.10 icmp_seq=121 timeout
41.0.0.10 icmp_seq=122 timeout
84 bytes from 41.0.0.10 icmp_seq=123 ttl=60 time=85.771 ms
84 bytes from 41.0.0.10 icmp_seq=124 ttl=60 time=105.720 ms
84 bytes from 41.0.0.10 icmp_seq=125 ttl=60 time=84.774 ms
84 bytes from 41.0.0.10 icmp_seq=126 ttl=60 time=84.774 ms
84 bytes from 41.0.0.10 icmp_seq=127 ttl=60 time=84.774 ms
84 bytes from 41.0.0.10 icmp_seq=128 ttl=60 time=83.777 ms
84 bytes from 41.0.0.10 icmp_seq=129 ttl=60 time=83.776 ms
84 bytes from 41.0.0.10 icmp_seq=130 ttl=60 time=83.777 ms
84 bytes from 41.0.0.10 icmp_seq=131 ttl=60 time=106.715 ms
84 bytes from 41.0.0.10 icmp_seq=132 ttl=60 time=104.720 ms

```

Imagen 5.2: Salida del terminal del equipo PC1 al enviar el ping al PC2 (con respuesta)

PC1> trace 41.0.0.10

trace to 41.0.0.10, 8 hops max, press Ctrl+C to stop

```

1  11.0.0.1  9.973 ms  9.964 ms  8.975 ms

2  10.0.0.2  31.913 ms  30.918 ms  31.915 ms

3  20.0.0.2  52.861 ms  53.855 ms  52.859 ms

4  30.0.0.2  97.738 ms  75.798 ms  74.800 ms

5  * * *

6  *41.0.0.10  78.790 ms (ICMP type:3, code:3, Destination port unreachable)

```

94 307.039513	10.0.0.1	224.0.0.9	RIPv2	86 Response
98 322.541088	10.0.0.1	224.0.0.9	RIPv2	106 Response
103 333.051002	10.0.0.2	224.0.0.9	RIPv2	126 Response
104 334.543014	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request id=0x29a8, seq=146/37376, ttl=63
106 336.539678	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request id=0x2ba8, seq=147/37632, ttl=63
107 338.536342	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request id=0x2da8, seq=148/37888, ttl=63
108 340.543978	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request id=0x2fa8, seq=149/38144, ttl=63
110 342.540643	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request id=0x31a8, seq=150/38400, ttl=63
111 344.548278	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request id=0x33a8, seq=151/38656, ttl=63
113 346.544942	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request id=0x35a8, seq=152/38912, ttl=63
114 348.541604	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request id=0x37a8, seq=153/39168, ttl=63
115 350.549241	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request id=0x39a8, seq=154/39424, ttl=63
117 352.534934	10.0.0.1	224.0.0.9	RIPv2	86 Response
118 352.545905	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request id=0x3ba8, seq=155/39680, ttl=63
119 354.553540	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request id=0x3da8, seq=156/39936, ttl=63
121 356.550204	11.0.0.10	40.0.0.10	ICMP	98 Echo (ping) request id=0x3fa8, seq=157/40192, ttl=63

Imagen 5.3: Captura de wireShark en los minutos en los que empezó a recibir la respuesta ICMP

Imagen 5.4: Contenido del mensaje nº 98

```

Routing Information Protocol
  Command: Response (2)
  Version: RIPv2 (2)
  > IP Address: 11.0.0.0, Metric: 1
  > IP Address: 40.0.0.0, Metric: 16
  > IP Address: 50.0.0.0, Metric: 1

```

Imagen 5.5: Contenido del mensaje nº 126

```

Routing Information Protocol
  Command: Response (2)
  Version: RIPv2 (2)
  > IP Address: 20.0.0.0, Metric: 1
  > IP Address: 30.0.0.0, Metric: 2
  > IP Address: 40.0.0.0, Metric: 3
  > IP Address: 41.0.0.0, Metric: 3

```

Como se puede observar en las imágenes anteriores (5.4 y 5.5) , la diferencia entre el contenido del mensaje RESPONSE nº 98 en la que una de las redes que le llegaban a R1 a través de la ruta de R5 (40.0.0.0) tiene métrica 16 antes de que funcione de nuevo el envío de mensajes ICMP indicando que es una red inalcanzable. Y el contenido del mensaje nº 126 donde R2 le manda un mensaje RESPONSE a R1 con la métrica nueva de cada una de las redes a las que vuelve a tener acceso para que actualice sus tabla de rutas.

6. Interrumpe las capturas. Analiza el tráfico capturado junto a los mensajes de depuración de R1 y R4. ¿Envían algún mensaje estos encaminadores en el momento en que el *tiempo* de algunas entradas llega a 3 minutos? ¿Por qué? ¿Qué mensajes de las capturas explican que estos encaminadores descubran las nuevas rutas? ¿Utiliza la implementación RIP de IOS triggered update?

RESPUESTA: Sí, los encaminadores envían mensajes cuando llegan a los 3 minutos como el mensaje indicado en la figura del apartado anterior (5.4) donde R1 ha desechado la ruta conectada al router R5 y ya no puede utilizarla para encaminar los paquetes en el que indica, a través de la métrica con valor 16, que las direcciones que antes recibía a través de la interfaz conectada al router R5 por la red 50.0.0.0 son inalcanzables ya que , como acabo de mencionar, ya no se puede utilizar. Una de estas redes es la red 40.0.0.0 y , aunque no lo hemos presentado a través de imágenes también se envían en otros paquetes el resto de redes que se recibían de R5 como las direcciones 41.0.0.0 y 30.0.0.0 a coste 16.

Cabe mencionar que de forma análoga sucede con R4 y las redes que recibía a través de R5 aunque nosotros no lo mostremos aquí explícitamente. El motivo de envío de estos mensajes de R1 y R4 es para indicar que la ruta a través de R5 está obsoleta y ya no se puede utilizar de manera que este mensaje se va difundiendo por el resto de routers vecinos entre ellos mandándolo a la dirección multicast 224.0.0.9, de la que ya hemos hablado anteriormente, para que estos borren esa ruta de sus tablas de rutas.

En lo referente a que mensajes explican que estos encaminadores descubran de nuevo las rutas está indicado en el apartado anterior en la imagen (5.5), donde se ve claramente que el valor de la métrica de una de las redes que antes tenían un valor inalcanzable ha cambiado. Esto se debe a que R2 ha comunicado a R1 y R3 a R4 respectivamente, pasados los 30 segundos a partir de los cuales tienen que enviar sus mensajes de refresco o actualización de tablas, a través de un mensaje RESPONSE la ruta para llegar a estas redes con una nueva métrica determinada pasando por ellos (R2 y R3).

De manera que volvemos a la misma configuración que teníamos al principio del informe, antes de activar RIP en R5, donde R1 y R4 encaminaban los datagramas IPs a través de las rutas dentro de sus tablas que pasaban por R2 y R3 con un coste mayor que con R5, en este caso con un valor de métrica 3 para 40.0.0.0 como se muestra en la imagen (5.5) del apartado anterior.

Como ya comentamos en anteriores apartados del informe, los routers CISCO utilizan los mensajes de actualización explícita o *Triggered Update*, para llevar a cabo la actualización de las tablas de rutas en los distintos routers que tienen activado el protocolo RIP cada vez que un router aprende una ruta hacia una red nueva o cuando cambia la métrica de una de sus rutas o incluso cuando una ruta queda obsoleta como es este caso. Esto lo podemos corroborar a través de los mensajes de depuración de los routers(para este ejemplo hemos escogido el de R1), donde se puede comprobar que manda mensajes de actualización a la dirección multicast 224.0.0.9.

```
*Mar 30 18:13:40.807: RIP: sending v2 update to 224.0.0.9 via GigabitEthernet3/0 (10.0.0.1)
*Mar 30 18:13:40.807: RIP: build update entries
*Mar 30 18:13:40.807: 11.0.0.0/8 via 0.0.0.0, metric 1, tag 0
*Mar 30 18:13:40.811: 40.0.0.0/8 via 0.0.0.0, metric 2, tag 0
*Mar 30 18:13:40.811: 41.0.0.0/8 via 0.0.0.0, metric 3, tag 0
*Mar 30 18:13:40.811: 50.0.0.0/8 via 0.0.0.0, metric 1, tag 0
```

7. Vuelve a configurar de nuevo *RIP* en R5. Observa cómo cambian las tablas de encaminamiento en R1 y R4. ¿Cuánto tiempo han tardado en aprender la nueva ruta? ¿Por qué? Comprueba de nuevo cuál es ahora la ruta que están siguiendo los mensajes intercambiados entre PC1 y PC2 con `trace`. Incluye aquí la salida.

RESPUESTA: Las tablas de rutas de R1 y R4 han cambiado , tal y como hemos mencionado en los anteriores apartados, a las rutas que recibían de R2 y R3 pero con una métrica menor. Tanto R1 como R4 han tardado un tiempo aproximado de 1.2 segundos desde que R5 mandó el mensaje de REQUEST y se envió el mensaje de RESPONSE periódico o *Trigger Update* de R5, como se puede observar en el mensaje nº 2 (que es el tiempo que ha tardado).

```

R4#show ip route rip
R    50.0.0.0/8 [120/1] via 40.0.0.2, 00:00:10, GigabitEthernet4/0
R    20.0.0.0/8 [120/1] via 30.0.0.1, 00:00:42, GigabitEthernet3/0
R    10.0.0.0/8 [120/2] via 40.0.0.2, 00:00:10, GigabitEthernet4/0
                        [120/2] via 30.0.0.1, 00:00:42, GigabitEthernet3/0
R    11.0.0.0/8 [120/2] via 40.0.0.2, 00:00:10, GigabitEthernet4/0

```

Imagen 7.1: Tabla de encaminamiento del router R4 (después de activar R5)

```

R1#show ip route rip
R    20.0.0.0/8 [120/1] via 10.0.0.2, 00:00:21, GigabitEthernet3/0
R    40.0.0.0/8 [120/1] via 50.0.0.1, 00:00:19, GigabitEthernet4/0
R    41.0.0.0/8 [120/2] via 50.0.0.1, 00:00:19, GigabitEthernet4/0
R    30.0.0.0/8 [120/2] via 50.0.0.1, 00:00:19, GigabitEthernet4/0
                        [120/2] via 10.0.0.2, 00:00:21, GigabitEthernet3/0

```

Imagen 7.2: Tabla de encaminamiento del router R1 (después de activar R5)

2	1.217744	50.0.0.1	224.0.0.9	RIPv2	86 Response
5	15.112614	50.0.0.1	224.0.0.9	RIPv2	106 Response
10	42.327887	50.0.0.1	224.0.0.9	RIPv2	106 Response
14	60.594073	50.0.0.2	224.0.0.9	RIPv2	106 Response
17	73.220330	50.0.0.1	224.0.0.9	RIPv2	106 Response
25	103.709853	50.0.0.1	224.0.0.9	RIPv2	106 Response
26	106.408640	50.0.0.2	224.0.0.9	RIPv2	106 Response
33	134.260212	50.0.0.1	224.0.0.9	RIPv2	106 Response
34	135.137867	50.0.0.2	224.0.0.9	RIPv2	106 Response
42	159.803951	50.0.0.1	224.0.0.9	RIPv2	106 Response
43	161.899352	50.0.0.2	224.0.0.9	RIPv2	106 Response
50	190.700386	50.0.0.1	224.0.0.9	RIPv2	106 Response
51	192.883552	50.0.0.2	224.0.0.9	RIPv2	106 Response
60	220.980467	50.0.0.1	224.0.0.9	RIPv2	106 Response
61	221.484120	50.0.0.2	224.0.0.9	RIPv2	106 Response
68	251.527835	50.0.0.1	224.0.0.9	RIPv2	106 Response

Imagen 7.3: Captura de WireShark en la interfaz eth3 del router R5 (NOTA: antes del response nº 2 aparece un mensaje de REQUEST en el que el Router R5 vuelve a pedir a R1 para rellenar sus tablas de rutas que están vacías aunque aquí no nos dimos cuenta de capturarlo)

```

Routing Information Protocol
  Command: Response (2)
  Version: RIPv2 (2)
  > IP Address: 30.0.0.0, Metric: 2
  > IP Address: 40.0.0.0, Metric: 1
  > IP Address: 41.0.0.0, Metric: 2

```

Imagen 7.4: Contenido del mensaje nº 2 RESPONSE

1	0.000000	40.0.0.1	224.0.0.9	RIPv2	106 Response
4	12.797799	40.0.0.2	224.0.0.9	RIPv2	106 Response
11	38.444265	40.0.0.1	224.0.0.9	RIPv2	106 Response
13	39.453566	40.0.0.2	224.0.0.9	RIPv2	106 Response
19	65.189790	40.0.0.1	224.0.0.9	RIPv2	106 Response
21	69.391564	40.0.0.2	224.0.0.9	RIPv2	106 Response
28	93.301668	40.0.0.1	224.0.0.9	RIPv2	106 Response
29	96.065283	40.0.0.2	224.0.0.9	RIPv2	106 Response
37	124.211068	40.0.0.1	224.0.0.9	RIPv2	106 Response
38	125.955406	40.0.0.2	224.0.0.9	RIPv2	106 Response
45	150.931662	40.0.0.1	224.0.0.9	RIPv2	106 Response
47	155.361823	40.0.0.2	224.0.0.9	RIPv2	106 Response
54	177.767947	40.0.0.1	224.0.0.9	RIPv2	106 Response
55	181.727365	40.0.0.2	224.0.0.9	RIPv2	106 Response
62	204.760811	40.0.0.1	224.0.0.9	RIPv2	106 Response
64	211.067955	40.0.0.2	224.0.0.9	RIPv2	106 Response
69	232.343103	40.0.0.1	224.0.0.9	RIPv2	106 Response
73	239.746319	40.0.0.2	224.0.0.9	RIPv2	106 Response

Imagen 7.5: Captura de WireShark en la interfaz eth4 del router R5

```

Routing Information Protocol
  Command: Response (2)
  Version: RIPv2 (2)
  > IP Address: 10.0.0.0, Metric: 2
  > IP Address: 11.0.0.0, Metric: 2
  > IP Address: 50.0.0.0, Metric: 1

```

Imagen 7.6: Contenido del mensaje nº 13 RESPONSE

```

PC2> trace 11.0.0.1
trace to 11.0.0.1, 8 hops max, press Ctrl+C to stop
 1  41.0.0.1    8.974 ms  9.972 ms  9.972 ms
 2  40.0.0.2   31.914 ms 30.915 ms 31.915 ms
 3  *50.0.0.2  53.855 ms (ICMP type:3, code:3, Destination port unreachable)

```

Imagen 7.7: Salida de la terminal de PC2 al ejecutar la orden 'Trace'

```

PC1> trace 41.0.0.1
trace to 41.0.0.1, 8 hops max, press Ctrl+C to stop
 1  11.0.0.1    8.973 ms  9.975 ms  8.976 ms
 2  50.0.0.1   31.916 ms 30.917 ms 30.918 ms
 3  *40.0.0.1  53.856 ms (ICMP type:3, code:3, Destination port unreachable)

```

Imagen 7.8: Salida de la terminal de PC1 al ejecutar la orden 'Trace'

8. ¿Qué podemos concluir respecto a los tiempos de aprendizaje y eliminación de rutas?

RESPUESTA: la conclusión que sacamos con respecto a estos tiempos es que los tiempos de aprendizaje de rutas son muy rápidos, una vez arrancado el protocolo RIP en un router, o por lo que podemos apreciar son casi inmediatos dado que el router manda un mensaje REQUEST a la dirección multicast del protocolo nada más activarlo por sus interfaces para rellenar su tabla de rutas. Además del mensaje RESPONSE

periódico que manda cada 30 segundos a sus routers vecinos para mantener dichas rutas en sus tablas y que no expiren. Aunque el proceso de aprendizaje puede llegar a sobrecargar la red debido a la cantidad de tráfico que genera con los mensajes periódicos.

Por otro lado, en cuanto a la eliminación de las rutas, éste es un proceso más lento ya que un router tendría que esperar 3 minutos para poder eliminar una entrada de su tabla de rutas, debido a que ésta ya no fuese accesible o quedase obsoleta, con lo que en el peor de los casos se llegarían a perder paquetes , tal y como hemos visto en apartados anteriores, si los routers que encaminan los datagramas tuvieran una sola ruta para encaminar (que es la que se va a eliminar) y no dos como hemos visto con R1 y R4 (pudiendo encaminar los datagramas por la otra ruta alternativa a la que se va a eliminar).

2. Órdenes IOS

RIP	
Permite configurar el protocolo de encaminamiento dinámico RIP. Es necesario indicar la versión en la que trabajará RIP, las subredes por las que se propagarán las rutas, deshabilitar auto-summary y opcionalmente los timers propios de RIP.	
router rip	Activa RIP en nuestro router.
no router rip	Desactiva RIP (Si se desea volver a activar RIP hay que establecer de nuevo todos los parámetros).
versión <versión>	Indica la versión de RIP. 2.
timers basic <update> <timeout> <holddown> <flush>	Establece los timers de RIP. Los valores predeterminados son 30, 180, 180, 240 respectivamente.
network <subred>	Indica la/s subred/es por las que se distribuirán las direcciones aprendidas.
passive-interface <ifaz>	Indica las interfaces por las que NO se distribuirán rutas.
no auto-summary	Deshabilita la opción de auto-summary que está habilitada por defecto.

Monitorizando el funcionamiento de RIP	
debug ip rip	Habilita los mensajes de depuración.

no debug ip rip	Deshabilita los mensajes de depuración.
show ip protocols	Muestra detalles de los protocolos.

Tabla de rutas	
show ip route	Muestra la tabla de rutas
show ip route rip	Muestra las entradas de la tabla de rutas aprendidas con RIP.
clear ip route *	Borra la tabla de rutas