Práctica 1.5. RIP y BGP

**Objetivos**

En esta práctica se afianzan los conceptos elementales del encaminamiento. En particular, se estudia un protocolo de encaminamiento interior y otro exterior: RIP (*Routing Information Protocol*) y BGP (*Border Gateway Protocol*).

Existen muchas implementaciones de los protocolos de encaminamiento. En esta práctica vamos a utilizar Quagga, que actualmente implementa RIP (versiones 1 y 2), RIPng, OSPF, OSPFv3, IS-IS y BGP. Quagga está estructurado en diferentes servicios (uno para cada protocolo) controlados por un servicio central (Zebra) que hace de interfaz entre la tabla de reenvío del *kernel* y las tabla de encaminamiento de cada protocolo.

Todos los ficheros de configuración han de almacenarse en el directorio /etc/quagga. La sintaxis de estos ficheros es sencilla y está disponible en <http://quagga.net>. Revisar especialmente la correspondiente a RIP y BGP en <https://www.quagga.net/docs/quagga.html>. Además, en /usr/share/doc/quagga-0.99.22.4 hay ficheros de ejemplo.

|  | Activar el **portapapeles bidireccional** (menú Dispositivos) en las máquinas virtuales.  Usar la opción de Virtualbox (menú Ver) para realizar **capturas de pantalla**.  La **contraseña** del usuario cursoredes es cursoredes. |
| --- | --- |

**Contenidos**

[Parte I. Protocolo interior: RIP](#_sa3p99nlg3w8)

[Preparación del entorno](#_xbnt2qq4f9bk)

[Configuración del protocolo RIP](#_tc4o0itw6ni)

[Parte II. Protocolo exterior: BGP](#_y1ewi83cdwp7)

[Preparación del entorno](#_14a3ftqman5y)

[Configuración del protocolo BGP](#_cp4o2e2rj36j)

# Parte I. Protocolo interior: RIP

## Preparación del entorno

Configuraremos la topología de red que se muestra en la siguiente figura, donde cada encaminador (Router1…Router4) tiene tres interfaces, cada uno conectado a una red diferente.:



Al igual que en prácticas anteriores, usaremos la herramienta vtopol 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  eth1  eth2 | 172.16.0.0/16  172.19.0.0/16  192.168.0.0/24 | 172.16.0.1  172.19.0.1  192.168.0.1 |
| Router2 | eth0  eth1  eth2 | 172.16.0.0/16  172.17.0.0/16  192.168.1.0/24 | 172.16.0.2  172.17.0.2  192.168.1.2 |
| Router3 | eth0  eth1  eth2 | 172.18.0.0/16  172.17.0.0/16  192.168.2.0/24 | 172.18.0.3  172.17.0.3  192.168.2.3 |
| Router4 | eth0  eth1  eth2 | 172.18.0.0/16  172.19.0.0/16  192.168.3.0/24 | 172.18.0.4  172.19.0.4  192.168.3.4 |

Configurar todos los encaminadores según la figura y tabla anterior. Además, activar el reenvío de paquetes IPv4 igual que en la práctica 1.1. Después, comprobar:

* Que los encaminadores adyacentes son alcanzables, por ejemplo, Router1 puede hacer *ping* a Router2 y Router4.
* Que la tabla de reenvío de cada encaminador es la correcta e incluye una entrada para cada una de las tres redes a las que está conectado.

## Configuración del protocolo RIP

***Ejercicio 1****.* Configurar RIP en todos los encaminadores para que intercambien información:

* Crear un fichero ripd.conf en /etc/quagga con el contenido que se muestra a continuación.
* Iniciar el servicio RIP (y Zebra) con service ripd start.

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

| *# Activar el encaminamiento por RIP*  router rip  *# Definir la versión del protocolo que se usará*  version 2  *# Habilitar información de encaminamiento en redes asociadas a los interfaces*  network eth0  network eth1  network eth2 |
| --- |

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

| *Copia los comandos usados y su salida.*  ***ROUTER 1:***  ***sudo vtysh -c "show ip rip"***  *Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP*  *Sub-codes:*  *(n) - normal, (s) - static, (d) - default, (r) - redistribute,*  *(i) - interface*  *Network Next Hop Metric From Tag Time*  *C(i) 172.16.0.0/16 0.0.0.0 1 self 0*  *R(n) 172.17.0.0/16 172.16.0.2 2 172.16.0.2 0 02:44*  *R(n) 172.18.0.0/16 172.19.0.4 2 172.19.0.4 0 02:59*  *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 02:44*  *R(n) 192.168.2.0/24 172.16.0.2 3 172.16.0.2 0 02:44*  *R(n) 192.168.3.0/24 172.19.0.4 2 172.19.0.4 0 02:59*  ***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, 00:02:44*  *R>\* 172.18.0.0/16 [120/2] via 172.19.0.4, eth1, 00:01:28*  *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, 00:02:44*  *R>\* 192.168.2.0/24 [120/3] via 172.16.0.2, eth0, 00:02:06*  *R>\* 192.168.3.0/24 [120/2] via 172.19.0.4, eth1, 00:01:28*  ***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*  ***ROUTER 2:***  ***sudo vtysh -c "show ip rip"***  *Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP*  *Sub-codes:*  *(n) - normal, (s) - static, (d) - default, (r) - redistribute,*  *(i) - interface*  *Network Next Hop Metric From Tag Time*  *C(i) 172.16.0.0/16 0.0.0.0 1 self 0*  *C(i) 172.17.0.0/16 0.0.0.0 1 self 0*  *R(n) 172.18.0.0/16 172.17.0.3 2 172.17.0.3 0 02:28*  *R(n) 172.19.0.0/16 172.16.0.1 2 172.16.0.1 0 02:37*  *R(n) 192.168.0.0/24 172.16.0.1 2 172.16.0.1 0 02:37*  *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 02:28*  *R(n) 192.168.3.0/24 172.16.0.1 3 172.16.0.1 0 02:37*  ***sudo vtysh -c "show ip route"***  *Codes: K - kernel route, C - connected, S - static, R - RIP,*  *O - OSPF, I - IS-IS, B - BGP, A - Babel,*  *> - selected route, \* - FIB route*  *C>\* 127.0.0.0/8 is directly connected, lo*  *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, 00:04:34*  *R>\* 172.19.0.0/16 [120/2] via 172.16.0.1, eth0, 00:05:13*  *R>\* 192.168.0.0/24 [120/2] via 172.16.0.1, eth0, 00:05:13*  *C>\* 192.168.1.0/24 is directly connected, eth2*  *R>\* 192.168.2.0/24 [120/2] via 172.17.0.3, eth1, 00:04:34*  *R>\* 192.168.3.0/24 [120/3] via 172.16.0.1, eth0, 00:03:56*  ***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 172.16.0.1 dev eth0 proto zebra metric 3*  ***ROUTER 3:***  ***sudo vtysh -c "show ip rip"***  *Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP*  *Sub-codes:*  *(n) - normal, (s) - static, (d) - default, (r) - redistribute,*  *(i) - interface*  *Network Next Hop Metric From Tag Time*  *R(n) 172.16.0.0/16 172.17.0.2 2 172.17.0.2 0 02:54*  *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 02:36*  *R(n) 192.168.0.0/24 172.17.0.2 3 172.17.0.2 0 02:54*  *R(n) 192.168.1.0/24 172.17.0.2 2 172.17.0.2 0 02:54*  *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 02:36*  ***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*  *R>\* 172.16.0.0/16 [120/2] via 172.17.0.2, eth1, 00:05:40*  *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, 00:05:00*  *R>\* 192.168.0.0/24 [120/3] via 172.17.0.2, eth1, 00:05:40*  *R>\* 192.168.1.0/24 [120/2] via 172.17.0.2, eth1, 00:05:40*  *C>\* 192.168.2.0/24 is directly connected, eth2*  *R>\* 192.168.3.0/24 [120/2] via 172.18.0.4, eth0, 00:05:00*  ***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 172.17.0.2 dev eth1 proto zebra metric 3*  *192.168.1.0/24 via 172.17.0.2 dev eth1 proto zebra metric 2*  *192.168.2.0/24 dev eth2 proto kernel scope link src 192.168.2.3*  *192.168.3.0/24 via 172.18.0.4 dev eth0 proto zebra metric 2*  ***ROUTER 4:***  ***sudo vtysh -c "show ip rip"***  *Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP*  *Sub-codes:*  *(n) - normal, (s) - static, (d) - default, (r) - redistribute,*  *(i) - interface*  *Network Next Hop Metric From Tag Time*  *R(n) 172.16.0.0/16 172.19.0.1 2 172.19.0.1 0 02:34*  *R(n) 172.17.0.0/16 172.18.0.3 2 172.18.0.3 0 02:35*  *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 02:34*  *R(n) 192.168.1.0/24 172.19.0.1 3 172.19.0.1 0 02:34*  *R(n) 192.168.2.0/24 172.18.0.3 2 172.18.0.3 0 02:35*  *C(i) 192.168.3.0/24 0.0.0.0 1 self 0*  ***sudo vtysh -c "show ip route"***  *Codes: K - kernel route, C - connected, S - static, R - RIP,*  *O - OSPF, I - IS-IS, B - BGP, A - Babel,*  *> - selected route, \* - FIB route*  *C>\* 127.0.0.0/8 is directly connected, lo*  *R>\* 172.16.0.0/16 [120/2] via 172.19.0.1, eth1, 00:05:54*  *R>\* 172.17.0.0/16 [120/2] via 172.18.0.3, eth0, 00:05:54*  *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, 00:05:54*  *R>\* 192.168.1.0/24 [120/3] via 172.19.0.1, eth1, 00:05:54*  *R>\* 192.168.2.0/24 [120/2] via 172.18.0.3, eth0, 00:05:54*  *C>\* 192.168.3.0/24 is directly connected, eth2*  ***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 172.19.0.1 dev eth1 proto zebra metric 3*  *192.168.2.0/24 via 172.18.0.3 dev eth0 proto zebra metric 2*  *192.168.3.0/24 dev eth2 proto kernel scope link src 192.168.3.4* |
| --- |

***Ejercicio 3****.* Con la herramienta wireshark, estudiar los mensajes RIP intercambiados, en particular:

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

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

***Ejercicio 4****.* Eliminar el enlace entre Router1 y Router4 (por ejemplo, desactivando el interfaz eth1 en Router4). Comprobar que Router1 deja de recibir los anuncios de Router4 y que, pasados aproximadamente 3 minutos (valor de *timeout* por defecto para las rutas), ha reajustado su tabla.

| *Copia los comandos usados y su salida.*  ***Comando sudo vtysh -c "show ip rip" en Router1***  ***Antes del time-out, la métrica es 2 y el siguiente salto está en 172.19.0.4***  *odes: 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 02:41*  *R(n) 172.18.0.0/16 172.19.0.4 2 172.19.0.4 0 00:03*  *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 02:41*  *R(n) 192.168.2.0/24 172.16.0.2 3 172.16.0.2 0 02:41*  *R(n) 192.168.3.0/24 172.19.0.4 2 172.19.0.4 0 00:03*  ***Justo después del time-out, la métrica se pone por defecto a 16, ya que es inalcanzable***  *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 02:37*  *R(n) 172.18.0.0/16 172.19.0.4 16 172.19.0.4 0 01:59*  *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 02:37*  *R(n) 192.168.2.0/24 172.16.0.2 3 172.16.0.2 0 02:37*  *R(n) 192.168.3.0/24 172.19.0.4 16 172.19.0.4 0 01:59*  ***Más tarde, se ha anunciado una nueva ruta que pasa por 172.16.0.2 que pertenece al Router 2, esta ruta tiene una métrica de 4 saltos***  *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 02:38*  *R(n) 172.18.0.0/16 172.16.0.2 3 172.16.0.2 0 02:38*  *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) 192.168.3.0/24 172.16.0.2 4 172.16.0.2 0 02:54* |
| --- |

***Ejercicio 5 (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 vtopol 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  eth1 | 2001:db8:200:1::/64  2001:db8:200:2::/64 | 2001:db8:200:1::2  2001:db8:200:2::2 |
| Router3 | eth0 | 2001:db8:200:2::/64 | 2001:db8:200:2::3 |

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

## Configuración del protocolo BGP

***Ejercicio 6****.* Consultar la documentación de las clases de teoría para determinar el tipo de AS (*stub*, *multihomed* o *transit* ) y los prefijos de red que debe anunciar. Recordar que el prefijo global de encaminamiento es de 48 bits y que los prefijos anunciados deben agregarse al máximo.

| **Número de AS** | **Tipo** | **Prefijos agregados** |
| --- | --- | --- |
| AS100 | Stub | 2001:db8:100::/47 |
| AS200 | Transit | - |
| AS300 | Stub | 2001:db8:300::/47 |

***Ejercicio 7****.* Configurar BGP en los encaminadores para que intercambien información:

* Crear un fichero bgpd.conf en /etc/quagga usando como referencia el que se muestra a continuación.
* Iniciar el servicio BGP (y Zebra) con service bgpd start.

Por ejemplo, el contenido del fichero /etc/quagga/bgpd.conf de Router1 en el AS 100 sería:

| *# Activar el encaminamiento BGP en el AS 100*  router bgp 100  *# Establecer el identificador de encaminador BGP*  bgp router-id 0.0.0.1  *# Añadir el encaminador BGP vecino en el AS 200*  neighbor 2001:db8:200:1::2 remote-as 200  *# Empezar a trabajar con direcciones IPv6*  address-family ipv6  *# Anunciar un prefijo de red agregado*  network 2001:db8:100::/47  *# Activar IPv6 en el encaminador BGP vecino*  neighbor 2001:db8:200:1::2 activate  *# Dejar de trabajar con direcciones IPv6*  exit-address-family |
| --- |

***Ejercicio 8.*** Consultar la tabla de encaminamiento de BGP y de Zebra en cada encaminador con el comando vtysh (sudo vtysh -c "show ipv6 bgp" y sudo vtysh -c "show ipv6 route"). Comprobar también la tabla de reenvío de IPv6 con el comando ip (ip -6 route).

| *Copia los comandos usados y su salida.*  ***ROUTER 1:***  ***sudo vtysh -c "show ipv6 bgp"***  *BGP table version is 0, local router ID is 0.0.0.1*  *Status codes: s suppressed, d damped, h history, \* valid, > best, i - internal,*  *r RIB-failure, S Stale, R Removed*  *Origin codes: i - IGP, e - EGP, ? - incomplete*  *Network Next Hop Metric LocPrf Weight Path*  *\*> 2001:db8:100::/47*  *:: 0 32768 i*  *\*> 2001:db8:300::/47*  *2001:db8:200:1::2*  *0 200 300 i*  *Total number of prefixes 2*  ***sudo vtysh -c "show ipv6 route"***  *Codes: K - kernel route, C - connected, S - static, R - RIPng,*  *O - OSPFv6, I - IS-IS, B - BGP, A - Babel,*  *> - selected route, \* - FIB route*  *C>\* ::1/128 is directly connected, lo*  *C>\* 2001:db8:200:1::/64 is directly connected, eth0*  *B>\* 2001:db8:300::/47 [20/0] via fe80::a00:27ff:feca:e9d4, eth0, 00:00:12*  *C>\* fe80::/64 is directly connected, eth0*  ***ip -6 route***  *unreachable ::/96 dev lo metric 1024 error -113 pref medium*  *unreachable ::ffff:0.0.0.0/96 dev lo metric 1024 error -113 pref medium*  *2001:db8:200:1::/64 dev eth0 proto kernel metric 256 pref medium*  *2001:db8:300::/47 via fe80::a00:27ff:feca:e9d4 dev eth0 proto zebra metric 1024 pref medium*  *unreachable 2002:a00::/24 dev lo metric 1024 error -113 pref medium*  *unreachable 2002:7f00::/24 dev lo metric 1024 error -113 pref medium*  *unreachable 2002:a9fe::/32 dev lo metric 1024 error -113 pref medium*  *unreachable 2002:ac10::/28 dev lo metric 1024 error -113 pref medium*  *unreachable 2002:c0a8::/32 dev lo metric 1024 error -113 pref medium*  *unreachable 2002:e000::/19 dev lo metric 1024 error -113 pref medium*  *unreachable 3ffe:ffff::/32 dev lo metric 1024 error -113 pref medium*  *fe80::/64 dev eth0 proto kernel metric 256 pref medium*  ***ROUTER 2:***  ***sudo vtysh -c "show ipv6 bgp"***  *BGP table version is 0, local router ID is 0.0.0.2*  *Status codes: s suppressed, d damped, h history, \* valid, > best, i - internal,*  *r RIB-failure, S Stale, R Removed*  *Origin codes: i - IGP, e - EGP, ? - incomplete*  *Network Next Hop Metric LocPrf Weight Path*  *\*> 2001:db8:100::/47*  *2001:db8:200:1::1*  *0 0 100 i*  *\*> 2001:db8:300::/47*  *2001:db8:200:2::3*  *0 0 300 i*  *Total number of prefixes 2*  ***sudo vtysh -c "show ipv6 route"***  *Codes: K - kernel route, C - connected, S - static, R - RIPng,*  *O - OSPFv6, I - IS-IS, B - BGP, A - Babel,*  *> - selected route, \* - FIB route*  *C>\* ::1/128 is directly connected, lo*  *B>\* 2001:db8:100::/47 [20/0] via fe80::a00:27ff:fe4a:cf3d, eth0, 00:01:31*  *C>\* 2001:db8:200:1::/64 is directly connected, eth0*  *C>\* 2001:db8:200:2::/64 is directly connected, eth1*  *B>\* 2001:db8:300::/47 [20/0] via fe80::a00:27ff:fe51:f314, eth1, 00:01:26*  *C \* fe80::/64 is directly connected, eth1*  *C>\* fe80::/64 is directly connected, eth0*  ***ip -6 route***  *unreachable ::/96 dev lo metric 1024 error -113 pref medium*  *unreachable ::ffff:0.0.0.0/96 dev lo metric 1024 error -113 pref medium*  *2001:db8:100::/47 via fe80::a00:27ff:fe4a:cf3d dev eth0 proto zebra metric 1024 pref medium*  *2001:db8:200:1::/64 dev eth0 proto kernel metric 256 pref medium*  *2001:db8:200:2::/64 dev eth1 proto kernel metric 256 pref medium*  *2001:db8:300::/47 via fe80::a00:27ff:fe51:f314 dev eth1 proto zebra metric 1024 pref medium*  *unreachable 2002:a00::/24 dev lo metric 1024 error -113 pref medium*  *unreachable 2002:7f00::/24 dev lo metric 1024 error -113 pref medium*  *unreachable 2002:a9fe::/32 dev lo metric 1024 error -113 pref medium*  *unreachable 2002:ac10::/28 dev lo metric 1024 error -113 pref medium*  *unreachable 2002:c0a8::/32 dev lo metric 1024 error -113 pref medium*  *unreachable 2002:e000::/19 dev lo metric 1024 error -113 pref medium*  *unreachable 3ffe:ffff::/32 dev lo metric 1024 error -113 pref medium*  *fe80::/64 dev eth0 proto kernel metric 256 pref medium*  *fe80::/64 dev eth1 proto kernel metric 256 pref medium*  ***ROUTER 3:***  ***sudo vtysh -c "show ipv6 bgp"***  *BGP table version is 0, local router ID is 0.0.0.3*  *Status codes: s suppressed, d damped, h history, \* valid, > best, i - internal,*  *r RIB-failure, S Stale, R Removed*  *Origin codes: i - IGP, e - EGP, ? - incomplete*  *Network Next Hop Metric LocPrf Weight Path*  *\*> 2001:db8:100::/47*  *2001:db8:200:2::2*  *0 200 100 i*  *\*> 2001:db8:300::/47*  *:: 0 32768 i*  *Total number of prefixes 2*  ***sudo vtysh -c "show ipv6 route"***  *Codes: K - kernel route, C - connected, S - static, R - RIPng,*  *O - OSPFv6, I - IS-IS, B - BGP, A - Babel,*  *> - selected route, \* - FIB route*  *C>\* ::1/128 is directly connected, lo*  *B>\* 2001:db8:100::/47 [20/0] via fe80::a00:27ff:fe0a:6b2d, eth0, 00:02:03*  *C>\* 2001:db8:200:2::/64 is directly connected, eth0*  *C>\* fe80::/64 is directly connected, eth0*  ***ip -6 route***  *unreachable ::/96 dev lo metric 1024 error -113 pref medium*  *unreachable ::ffff:0.0.0.0/96 dev lo metric 1024 error -113 pref medium*  *2001:db8:100::/47 via fe80::a00:27ff:fe0a:6b2d dev eth0 proto zebra metric 1024 pref medium*  *2001:db8:200:2::/64 dev eth0 proto kernel metric 256 pref medium*  *unreachable 2002:a00::/24 dev lo metric 1024 error -113 pref medium*  *unreachable 2002:7f00::/24 dev lo metric 1024 error -113 pref medium*  *unreachable 2002:a9fe::/32 dev lo metric 1024 error -113 pref medium*  *unreachable 2002:ac10::/28 dev lo metric 1024 error -113 pref medium*  *unreachable 2002:c0a8::/32 dev lo metric 1024 error -113 pref medium*  *unreachable 2002:e000::/19 dev lo metric 1024 error -113 pref medium*  *unreachable 3ffe:ffff::/32 dev lo metric 1024 error -113 pref medium*  *fe80::/64 dev eth0 proto kernel metric 256 pref medium* |
| --- |

***Ejercicio 9****.* Con ayuda de la herramienta wireshark, estudiar los mensajes BGP intercambiados (OPEN, KEEPALIVE y UPDATE).

| *Copia una captura de pantalla de Wireshark con mensajes BGP mostrando el formato del mensaje UPDATE.* |
| --- |