Práctica 4 – Enrutamiento con RIP

1- Objetivos

En esta práctica vamos a ver cómo configurar el protocolo de enrutamiento RIP (RIP versión 1 y versión 2) en los routers Cisco y en PCs con sistema operativo GNU/Linux empleando el software Quagga.

2- Trabajo previo

Se recomienda consultar un poco la documentación de configuración para cada caso (IOS y Quagga).

En primer lugar trabajaremos con routers Cisco. Se puede encontrar el caso de configuración en IOS (aunque no es exactamente la misma versión que hay en el laboratorio) en documentos colgados en la web de la asignatura.

En la segunda parte trabajaremos con Quagga. Quagga es un paquete software que provee varios programas que implementan diversos protocolos de encaminamiento para redes IP. Se basa en una estructura modular en la que cada protocolo de encaminamiento es implementado por un proceso independiente y existe un proceso general que unifica las rutas entre todos ellos para introducirlas en la tabla de reenvío del router.

El programa de control general de la tabla de rutas se llama zebra y de hecho Quagga es una evolución del software de GNU Zebra, cuyo desarrollo ha quedado descontinuado.

Cada uno de los protocolos de encaminamiento está implementado en un programa diferente y así nos encontraremos con ripd (para RIP), ospfd (para OSPF) bgpd (para BGP), isisd (para IS-IS), así como las versiones de varios de estos protocolos para IPv6.

Puede obtener más información sobre este software en su web http://www.nongnu.org/quagga

La documentación se encuentra en:

http://www.nongnu.org/quagga/docs/docs-info.html

Es interesante conocer la arquitectura general del software (sección 1), los comandos básicos (sección 3), los de zebra (sección 4) y los de RIP (sección 5).

3- Configurando RIPv1 en un escenario Classful con routers Cisco

Recuerde que RIPv1 está pensado para trabajar en un entorno classful. En ese caso, todas las redes son A, B o C y no hay subredes en las mismas. RIP serviría para anunciar ese tipo de redes. Comenzaremos por un escenario classful.

- Dispongan una topología como la de la Figura 1. En la LAN1 empleen el espacio de direcciones `192.168.1.0/24`. En la LAN2 empleen `192.168.2.0/24` y en la LAN3 `192.168.3.0/24`
- Asignen dirección IP a los interfaces de router2 y a PC A y configuren el router por defecto de PC A (debería ser el interfaz de router2 en su LAN)
- Hagan lo mismo con router3 y PC C en la otra LAN extremo
- Configuren PC B para que tenga dirección IP de la LAN 3 y el router por defecto que quieran (router2 o router3)
- Ahora, si intentan enviar un paquete IP desde PC A a PC C o viceversa verán que no funciona. Los routers tienen sólo rutas a las redes a las que están conectados (no hemos

configurado ninguna ruta estática, por ejemplo ninguna ruta por defecto). No las modifiquen. ¿Qué tipo de error reciben? ¿Se debe a algún mensaje de algún protocolo?

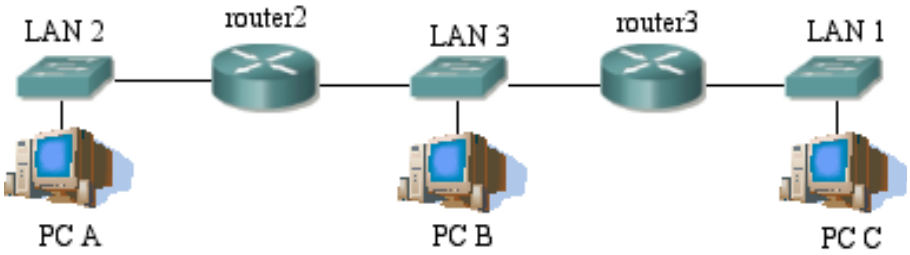

Figura 1 - Topología con 3 redes

A- Activando el proceso de RIP

Como paso previo vamos a desactivar el mecanismo llamado split horizon que evita las cuentas a infinito en ciertas situaciones. Lo vamos a desactivar para ver primero el caso más simple de RIP. Se puede activar/desactivar en cada interfaz de forma independiente.

- Vayan al modo configuración de cada interfaz y ejecuten el comando:

```
Router(config-if)# no ip split-horizon
```

- Entren en el modo global de configuración de uno de los routers (de momento sólo de uno). A partir de ahí entren en el modo de configuración de RIP:

```
Router(config)# router rip
```

Este comando también ha activado un proceso de RIP. Ahora, debemos decirle en qué redes (classful) queremos que emplee RIP. Para ello, se especifican con el siguiente comando:

```
Router(config-router)# network direcciondelared
```

- Ejecuten ese comando para la red `192.168.3.0`

- Coloquen un `tcpdump` en PC B, podrán ver el paquete de RIP que envía el router. ¿Cada cuánto tiempo lo envía? Verán que sólo anuncia la red para la que hemos ejecutado el comando `network`. ¿Manda algo a la otra red a la que está conectado? ¿A qué dirección IP manda los paquetes? ¿Qué protocolo de transporte emplean? ¿Y qué puertos? Si queremos que anuncie también la otra red debemos ejecutar el comando también para ella. Háganlo.

- Ahora, en la LAN3 vean el paquete de RIP que envía ese router. ¿Qué redes anuncia? ¿Y qué anuncia por la otra LAN a la que está conectado?

Pueden obtener información sobre la configuración del proceso de RIP con el comando:

```
Router> show ip protocols
```

En el resultado de ese comando identifiquen: cada cuánto se envían las actualizaciones de RIP, cuánto falta para que se envíe la próxima y en qué redes se ha activado.

- Consulten ahora la tabla de rutas del otro router. ¿Ha cambiado algo desde que activamos RIP en el primero? Recuerden que sólo hemos activado el proceso de RIP en uno de los routers. Eso quiere decir que aunque este router pueda ver los paquetes enviados por el otro no tiene ningún programa que los procese.

En los routers Cisco se pueden activar funciones de depuración para que nos digan lo que está haciendo ante ciertos paquetes o eventos. No es recomendable tenerlas activas mas que cuando se busquen problemas de configuración o de red porque ralentizan el funcionamiento del router. Se activan desde modo privilegiado con el comando `debug`.

- En el router que tiene RIP activo ejecuten:

```
Router# debug ip rip
Router# debug ip routing
```

La primera de ellas hará que el router nos indique cuándo envía o recibe paquetes de RIP. Con la segunda nos notificará cuando se produzcan cambios en la tabla de rutas. Se desactivan sin más que poner `no` delante de cada una de ellas. Generalmente hace falta conectarse al router por el puerto de consola para ver los mensajes de depuración.

- Activen ahora un proceso de RIP en el otro router. Configúrenlo para las dos redes a las que está conectado

- Vean los paquetes de RIP de los dos routers ahora en la LAN3 y de cada uno de ellos en la LAN1 y la LAN2

- Vean cómo cambian las tablas de rutas de cada uno. Verán en cada uno la ruta indirecta a la red que está sólo conectada al otro router. El campo de métrica en los mensajes de RIP tiene un valor válido entre 1 y 15, reservándose 16 para marcar el destino como inalcanzable. Así, las redes directamente conectadas se anuncian con una métrica de 1 y las que están a un router de distancia con una métrica de 2. ¿Cómo marca en la tabla de rutas el Cisco IOS las que ha aprendido con RIP?

- Vean que cada ruta aprendida con RIP tiene además un contador del tiempo que ha transcurrido desde la última vez que se recibió una actualización que anunciaba esta ruta.

- Prueben ahora a hacer ping entre PC A y PC C.

B- Reaccionando a la desconexión de redes

- Desactiven ahora uno de los routers del conmutador de la LAN3. Por ejemplo desconecten el router router2 del conmutador.

router2 desactiva automáticamente su interfaz. router3 no sabe que el router router2 se ha desconectado de la red. Sigue pensando que puede llegar a la LAN2 través de él. Transcurrido cierto tiempo, router3 marca esa red como posiblemente inalcanzable y empieza a mandar los paquetes de RIP con métrica 16 en esa ruta. ¿Cúanto es ese tiempo? En la salida del comando `show ip protocols` aparece marcado como *"invalid after"*. Un tiempo después el router borra la ruta de su tabla.

El proceso en router2 ha sido diferente. El router detecta que ya no está conectado al conmutador y desactiva ese interfaz. Al desactivarlo elimina todas las rutas que iban por ese interfaz.

- Reconecten router2 al conmutador y vean cómo los routers aprenden de nuevo las rutas.

- Desactiven ahora una de las redes de los extremos. Por ejemplo, desconecten el cable del router router3 al conmutador de la LAN1

Verán que router3 desactiva inmediatamente ese interfaz y desaparece la ruta a esa red de su tabla de rutas. Si tienen activado el debug de RIP verán que en el siguiente paquete de RIP que manda sigue anunciando la ruta a la LAN1 pero ahora con métrica 16. Al recibir el otro router este paquete, como tenía apuntado que llegaba a esa red a través de ese router cambia la métrica de su ruta aunque el nuevo valor es mayor que el que él conocía. Pero dado que el valor de métrica que le llega es 16 elimina la ruta de su tabla de rutas. En las siguientes actualizaciones enviadas por router2 aún se incluye la ruta a esa red ahora inalcanzable, pero se envía indicando una métrica de 16.

- Reconecten el cable y vean cómo se propaga la información de que esa red vuelve a estar accesible.

C- Timers

Hemos visto varios timers en funcionamiento. Por un lado el tiempo entre cada actualización de RIP que envía el router. Por otro lado el tiempo hasta que marca como inalcanzable una red por no recibir actualizaciones y finalmente el tiempo hasta que la borra. Podemos cambiar estos valores desde el modo de configuración del proceso de RIP con el comando `timers`.

- Pruébenlo. El timer de *Holddown* pueden ignorarlo para esta topología (por ejemplo ponerlo a 0).

D- Split-Horizon

Este mecanismo viene activado por defecto y evita que se produzcan ciclos en ciertas topologías. Lo que se hace es, en la tabla de rutas que se envía por un interfaz no enviar las rutas a todas las redes a las que el router llega por ese interfaz. Veamos un ejemplo.

- Reactiven *split-horizon* en los interfaces de los routers que están conectados a la LAN3.

A partir de ahí verán que los paquetes que por ejemplo envía router3 a esa red no incluyen la ruta a la LAN3 ni a la LAN2. Dado que para alcanzar esas redes él envía los paquetes por ese interfaz, o sea, ha aprendido las rutas por él, es lógico pensar que los routers que estén en la LAN3 no necesitan que él les comunique de nuevo esa información.

4- RIPv1 en un escenario de Subnetting

El primer esquema de hacer subredes, extendido en la época en que se seguía haciendo una asignación de redes con clase, se basaba en hacer todas las subredes de una red con clase de igual tamaño (lo que se conoce como FLSM o *Fixed Length Subnet Masks*). Esto sería el equivalente a emplear la misma máscara en todas esas subredes o reservar un número de bits del identificador del host de la red para identificar a la subred.

- Manteniendo la topología de la figura 1 y *split horizon* desactivado haga una asignación de direcciones FLSM. Puede partir por ejemplo de la red B 172.16.0.0/16, que es parte del rango privado 172.16.0.0/12. Haga tres subredes, todas con la misma máscara y configúrelas en los equipos de la red.

- Mantenga los routers sin rutas estáticas y active RIP para esa red. No hace falta hacer el comando para cada subred pues lo que activamos es RIP para toda la red clase B.

- Observen los anuncios que hacen los routers a cada una de las LANs.

En los mensajes de RIPv1 no hay máscaras. En un escenario de subnetting se anuncian las direcciones de red de esas subredes. En un escenario classful, esas direcciones serían direcciones de hosts de dentro de la red con clase. Por ejemplo, si una de sus subredes es la 172.16.64.0/20 se anunciará como dirección de red 172.16.64.0 pero eso es la dirección de un host en un escenario classful; es la dirección de un host de dentro de 172.16.0.0/16. Sin embargo, teniendo configuradas las subredes con máscaras, el router, al recibir un anuncio por un interfaz, los anuncios de rutas a subredes de dentro de la red en la cuál él tiene un interfaz se toman con la máscara del interfaz por el cual le ha llegado el anuncio. Es decir, uno cualquiera de esos routers recibe el anuncio a redes de dentro de la 172.16.0.0/16 por un interfaz que él tiene en esa red y por lo tanto, asumiendo subnetting clásico, es correcto que tome la máscara del interfaz por el cual le ha llegado el anuncio, que es el interfaz (o uno de los interfaces) que tiene en esa red.

Punto de control: Muestre al profesor de prácticas este último escenario, que sabe configurar RIP y que entiende cómo funciona

5- RIPv1, RIPv2 y VLSM

RIPv1 no funciona correctamente en el caso de que dentro de una red con clase se hagan subredes empleando máscaras de diferente tamaño. Esto es debido a que no anuncia la máscara y tomar la del interfaz por el cual le llega el anuncio puede ser correcto o no, según el caso.

Partiendo del escenario del apartado anterior modificaremos la máscara para la LAN 3. Por ejemplo ampliando la máscara.

- Modifique la máscara solo para el interfaz en la LAN 3 de router2

Ahora, router3 sigue teniendo la misma máscara en ambos interfaces y sigue anunciando las redes, pero router2 tiene máscara diferente en cada uno. Vea cómo aprende ahora router2 la ruta a la LAN 1.

- Modifique la versión de RIP a la versión 2 en ambos routers. Para ello entre en el modo de configuración del proceso de RIP y ejecute el comando:

```
Router(config-router)# version 2
```

- Ahora vean cómo se actualizan las tablas de rutas de los routers. Vean con un `tcpdump` los paquetes de RIP. ¿A qué dirección se envían los paquetes de RIP? ¿De qué tipo es esta dirección?

6- Introducción a Quagga Routing Software

La arquitectura de este software requiere tener siempre corriendo el programa zebra, dado que es a través de él que el resto de programas puede introducir rutas en la tabla del router.

Cada uno de estos programas tiene su propio fichero de configuración. En esta práctica, para no modificar los ficheros globales del sistema, lanzaremos siempre los programas especificando en los argumentos el fichero de configuración que deben utilizar.

En el laboratorio puede encontrar estos programas en el directorio /usr/lib/quagga

Dado que en esta práctica nos limitaremos a configurar RIP necesitaremos tener corriendo zebra y ripd. Para mayor comodidad los lanzaremos capturando el terminal, es decir, no como *daemon,* para ver más cómodamente sus mensajes y poder detenerlos con Ctrl+C.

- Busque con qué opción se le indica al programa zebra y a ripd el fichero que debe emplear como fichero de configuración (en la documentación en la web, en la página del manual o con la opción –h que enumera las opciones del programa).

Una vez lanzados estos programas podemos comunicarnos con ellos para hacer configuraciones adicionales a la del fichero de arranque o para consultar su estado y el de las tablas de rutas que calculan. Para ello, estos programas pueden crear un socket TCP y esperar conexiones a él. Podremos emplear el programa telnet para conectar con el programa y acceder a su configuración. Solo necesitaremos saber en qué puerto están aceptando conexiones. Estos puertos se pueden especificar para cada programa.

- Busque la opción de línea de comandos que sirve para indicar el puerto TCP en el que debe aceptar conexiones uno de estos demonios de cara a recibir comandos de configuración.
- Cree un fichero mínimo de configuración para el programa zebra. Puede crearlo en su propio HOME pues va a indicarle el path al lanzar el programa. Lo mínimo que necesitará es indicar en dicho fichero la clave para acceder a la configuración del demonio. Eso sería un fichero que simplemente contuviera:

```
password aquilaclave
```

- El programa puede guardar en un fichero el identificador del proceso (PID) en el que se está ejecutando, para ayudar a localizar el proceso y terminarlo con una señal (un kill). Para poder crear ese fichero necesita permisos de escritura sobre dicho fichero el usuario con el que correrá el programa. Aunque lo lanzamos como root con *sudo*, el programa se cambia a otro usuario para "rebajar" sus permisos, así que si ese usuario no tiene permiso para modificar ese fichero va a tener problemas. Por defecto el fichero es /var/run/quagga/zebra.pid. Si por algún motivo el fichero no existe o no tiene permisos para escribir puede resolverlo sin más que creando un fichero con permisos de escritura para todo el mundo e indicando su ruta al programa con la opción –i. También puede pedirle al programa que no se cambie de usuario de forma que no pierda los permisos para escribir en ciertos directorios, esto se hace indicando con –u que mantenga la ejecución como root (o le puede decir que lo haga como otro usuario cualquiera siempre que ese tenga permiso).
- Lance zebra indicando el fichero de configuración, el del PID si hace falta y un puerto (uno no reservado) para atender a las conexiones para configuración (recuerde que necesitará lanzarlo con sudo para que tenga permisos para hacer su trabajo).
- Pruebe a conectarse al mismo con el programa telnet (que permite especificar el puerto al que conectarse mediante un argumento)

La interfaz de configuración de zebra y los programas que implementan los protocolos de enrutamiento recuerdan bastante a la CLI de Cisco IOS. Por ejemplo puede preguntar por los comandos o las opciones disponibles con '?'. Dispone de dos modos de funcionamiento a la hora de la gestión de los demonios. Por un lado el modo no privilegiado, que es al que se accede por defecto al conectarse al mismo y que solo permite "ver" pero no hacer cambios, y por otro el modo privilegiado, tras ejecutar el comando "enable", que permite hacer cambios. Si no ha especificado la contraseña para el modo privilegiado en el fichero de configuración no tendrá ninguna.

- Explore los comandos de la interfaz de gestión de zebra en modo no privilegiado. Por ejemplo averigüe cómo ver el contenido de la tabla de rutas del equipo o la lista de interfaces de red.
- Explore los comandos de zebra en modo privilegiado (*enable*). Averigüe cómo ver la configuración actual del demonio y cómo añadir y eliminar una ruta estática.

Ahora lanzaremos ripd. Necesitará un fichero de configuración similar (con la password), indicárselo en los argumentos así como el puerto en el que aceptar conexiones para gestión, probablemente un fichero para el PID también con la opción -i y lanzarlo con sudo.

Recuerde que zebra debe seguir corriendo todo el tiempo.

- Lance ripd. Acceda a su interfaz de configuración y explore los comandos, en modo normal y privilegiado. Busque cómo activarlo para una red, cómo configurar los timers, cómo ver las rutas que ha aprendido el programa y cómo depurar su funcionamiento (comando debug).

7- Prueba de interoperabilidad

- Configure la topología de la Figura 2. Emplee subredes de 192.168.0.0/24 con máscaras de diferente longitud para cada subred y configure RIP versión 2 en los routers Cisco y en el PC A con Quagga.
- Vea en PC C los anuncios de ambos routers, tanto con *Split-horizon* activado como desactivado
- Configure zebra y ripd en PC B para que aprenda las rutas anunciadas por PC A y router3 aunque él no anuncie nada. Intente conseguir que el PC B no tenga ruta por defecto configurada sino que aprenda las rutas por RIP de forma que su siguiente salto sea el PC A para ir a la subred B y a la C pero sea el router3 para ir a la subred D.

Pruébelo moviendo el PC C a cada una de esas subredes y calculando la ruta con traceroute o ping.

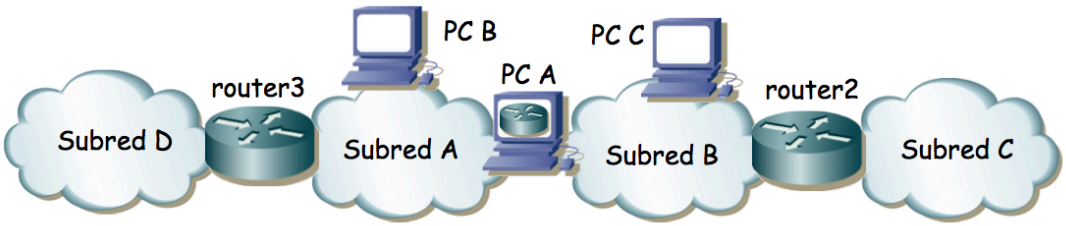

Figura 2 - Topología para prueba de interoperabilidad

Punto de control: Muestre al profesor de prácticas este último escenario y que sabe configurar RIP con Quagga

8- Conectándose a la red del Laboratorio

A continuación veremos una topología más grande con varios routers RIP y rutas alternativas.

- Configure la topología de la Figura 3, donde PC A y PC B así como los routers 1-3 corren demonios de RIP haciendo los anuncios en versión 2. Emplee la red 10.3.48+armario.0/24 para sus subredes y en el interfaz LAN de router1 la dirección 10.3.17.16+armario (máscara de 20 bits).
- Inspeccione la tabla de rutas del router de acceso (desde su PC-SC mediante telnet, password invitado) y, al final del apartado, también la información de configuración RIP del mismo (show ip protocols); le ayudará a completar su aprendizaje sobre RIP. Consulte también su caché ARP.
- Desactive *split-horizon* en los routers e intente provocar un bucle y cuenta a infinito.

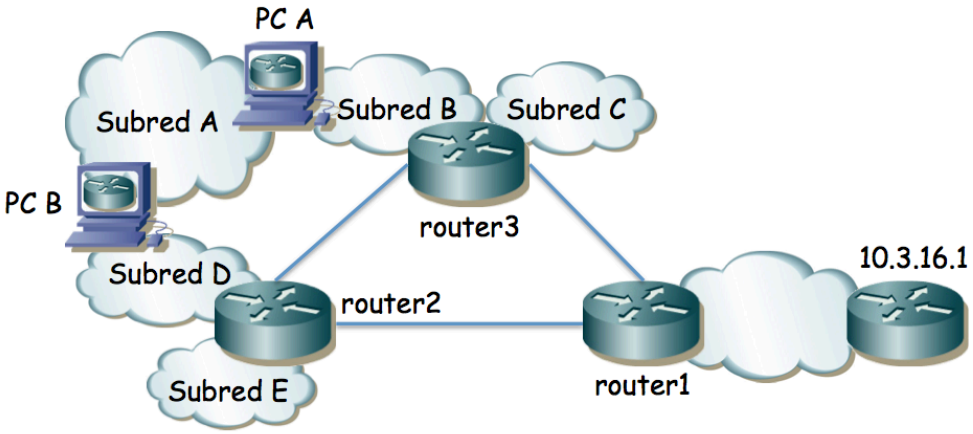

Figura 3 - Dominio RIP conectado al laboratorio

Punto de control: Muestre al profesor de prácticas la configuración de red que ha hecho (comandos Quagga e IOS) para este apartado así como de si ha conseguido una cuenta a infinito y en su caso cómo y demuéstrelo con una traza.