**Práctica 2: Direccionamiento y Enrutamiento**

***Autor****: Juan Rodríguez Suárez,* [*alu0101477596@ull.edu.es*](mailto:alu0101477596@ull.edu.es)*.*

***Curso:*** *Redes y Sistemas Distribuidos, 2º Grado en Ingeniería Informática.*

***Fecha:*** *5 de abril de 2023.*

**Índice**

[1. Diferencias entre enrutamiento estático y dinámico 3](#_Toc132248983)

[2. Enrutamiento estático 4](#_Toc132248984)

[2.1. Descripción de los pasos realizados 4](#_Toc132248985)

[3. Enrutamiento dinámico 6](#_Toc132248986)

[3.1. Descripción de los pasos realizados 6](#_Toc132248987)

[4. Resumen de los comandos y ficheros 8](#_Toc132248988)

[4.1. Comandos 8](#_Toc132248989)

[4.2. Ficheros de configuración 8](#_Toc132248990)

# Diferencias entre enrutamiento estático y dinámico

En el enrutamiento estático, se definen manualmente todas las tablas de enrutamiento y sus entradas además de asignar direcciones IP de forma manual en la red. En el dinámico, las tablas de enrutamiento se calculan cada cierto tiempo según un algoritmo de cálculo de distancias entre nodos de un grafo y las direcciones IP pueden ser asignadas dinámicamente según un servidor DHCP (aunque este último punto no se aborda en esta práctica).

Por tanto, la principal diferencia es que las tablas de enrutamiento en el estático se tienen que escribir a mano y en el dinámico se definen cada cierto tiempo de forma automática siguiendo un protocolo. La principal ventaja es que al ser las tablas estáticas, nadie externo puede modificarlas para redirigir el tráfico a otro sitio además de que el encargado de realizar la red sabe con certeza y seguridad en todo momento como son estas tablas y en caso de enviar un paquete a una red, por donde va a viajar siempre en contraposición de las tablas dinámicas donde van cambiando periódicamente y se requieren medidas de seguridad para aislarlas de inyecciones de paquetes RIP externos que puedan modificar estas tablas. Sin embargo, si en una red estática un enlace deja de funcionar, es muy probable que dejemos a una parte de la red completamente incomunicada con otra red hasta que el encargado lo solucione o bien cambiando la tabla o reparando el problema. Esto no es tan grave en las dinámicas ya que, si un enlace deja de ser accesible, los protocolos y algoritmo recalculan las rutas modificando las tablas de enrutamiento y corrigiendo el error en muy poco tiempo. Además, en cuanto la red crezca, se vuelve muy tedioso y propenso a errores el hecho de establecer manualmente todas las entradas a todas las diferentes redes si utilizamos en enrutamiento estático. Esto no pasa en el dinámico puesto que se envían mensajes periódicos donde se da información sobre las redes directamente conectadas de cada router y se actualizan las tablas de cada uno de forma automática.

# Enrutamiento estático

## Descripción de los pasos realizados

Lo primero de todo es hacer a ‘papel’ el reparto de direcciones y redes. Para ello, comenzamos con la red con más hosts (B, 127 hosts) y le asignamos una máscara que cubra ese número de hosts más tres direcciones dedicadas que son las dos primeras de la red (dirección de red y puerta de enlace) y la última (difusión). Este número debe ser potencia de 2, con lo que el más pequeño es 256 = 28, por tanto, la máscara es 32 – 8 = 24 y la dirección de red es la primera dirección que es 8.0.0.0. Repetiremos este proceso para cada subred (resolviendo primero las de mayor tamaño).

Lo siguiente que tenemos que hacer es pasar esta red al simulador. Una vez la tenemos montada, primero tenemos que asignar a cada PC su dirección IP correspondiente en su interfaz eth0. Para conseguir esto, tenemos que ir PC por PC editando el fichero /etc/network/interfaces y descomentar la línea *auto eth0* para activar la interfaz y sustituir que en vez de que utilice DHCP para obtener su dirección IP, utilice el método estático (poner *static*). En la siguiente línea ponemos *address IP/mask*, donde en IP ponemos la dirección IP asignada previamente al PC y la máscara de la red a la que pertenece. Con esto conseguimos asignarle una dirección IP a su interfaz eth0 para que sea reconocida en su red local. En la siguiente línea escribimos *gateway IP\_ROUTER*, donde en IP\_ROUTER va la dirección IP de la puerta de enlace predeterminada de la red. Una vez hemos realizado estos cambios, debemos activar esta configuración usando el comando *ifup eth0*. Para comprobar que todo ha ido como esperaba, usé el comando *ifconfig* que resume los datos de las diferentes interfaces del PC.

El siguiente paso es configurar los routers. Para ello, tenemos que acceder a la Shell de configuración del router Quagga mediante el comando *vtysh*. Una vez dentro, tecleamos *configure terminal* para entrar en el modo configuración y seleccionamos una interfaz a editar con *interface ethX* donde *ethX* es la interfaz del router que queremos editar. Para asignar una dirección a una interfaz, usamos el comando *ip address IP/mask*, donde en IP ponemos la dirección IP asignada a la interfaz y la máscara de la red a la que pertenece. Para hacer que esta configuración sea permanente ejecutamos *no shutdown* y también *link-detect* para hacer que se vincule al enlace. Salimos del modo de configuración con *exit* y muy importante guardar todos los cambios con *write*. Para comprobar que todo ha salido satisfactoriamente, se puede utilizar *show running-config* para ver todas las interfaces.

El último paso es completar las tablas de enrutamiento de los routers. Actualmente se han añadido automáticamente aquellas redes que están directamente conectadas al router. Por tanto, tenemos que configurar las redes indirectas. Para ello, volvemos a entrar en el modo configuración y empleamos el comando *ip route RED/mask gateway* donde *RED/mask* indica la dirección de la red indirecta a añadir y *gateway* es la dirección de la puerta de enlace a donde redirigir el paquete. Este proceso lo repetí en cada router de forma que para el router 1 configuré que las redes B, C, E fuesen por la puerta de enlace de eth0 del router 2. Para el router 2 hice que las redes A y C entrasen por la interfaz eth0 del router 1 y la eth0 del router 3 respectivamente. Por último, para el router 3 añadí entrada para que los paquetes dirigidos a las redes A, D y E fuesen reenviados por la eth1 del router 2. Para comprobar que todo se había realizado correctamente (habiendo hecho *write*) se ejecuta el comando *show ip route* dentro de la Shell de Quagga para ver las tablas de enrutamiento.

Una vez completados los tres pasos, comprobé que todos los PC tenían conexión a todos los otros PC utilizando *ping* y *traceroute*. Para el último apartado de la práctica *“Añada una ruta por defecto a los routers de forma que, si no existe una ruta más específica, el router se envíe hacia el QuaggaRouter-1.”* Basta con añadir en cada tabla de enrutamiento de los routers 2 y 3 una nueva entrada con la dirección 0.0.0.0/0 que indica una dirección que no se ha especificado en las otras entradas y como puerta de enlace poner el ‘camino’ hacia el router 1. En el router 1 no se pone porque podría provocar problemas dado que podría suponer un reenvío a sí mismo si una dirección no especificada llega al router 1. Así, para el router 2 se ejecuta: *ip route 0.0.0.0/0 8.0.1.97*, que reenvía los paquetes no especificados en la tabla hacia la interfaz 8.0.1.97 que se corresponde con la eth0 del router 1. Para el router 2 se ejecuta: *ip route 0.0.0.0/0 8.0.1.101*, que reenvía los paquetes no especificados en la tabla hacia la interfaz 8.0.1.101 que se corresponde con la eth1 del router 2 que a su vez los reenviará a la eth0 del router 1.

# Enrutamiento dinámico

## Descripción de los pasos realizados

Lo primero de todo es asignar las direcciones IP a las interfaces de los diferentes PC y routers modificando */etc/network/interfaces* para los PC y entrando en el modo configuración de los routers Quagga (*vtysh -> configure terminal -> interface ethX -> ip address IP/mask -> no shutdown -> link-detect -> write*). Lo anterior se comprueba fácilmente haciendo *ping* desde los distintos dispositivos.

Para activar ARP en los routers de la red privada para que se completen las tablas de enrutamiento de los routers (*configure terminal -> router rip -> version 2).* Ahora le decimos a cada router las redes donde publicar rutas con RIP (*network IP/mask*). Se debe proceder en todos los routers menos en el tercero donde debemos excluir la ruta al exterior porque no pertenece a la red de área local. Usando show ip route se puede observar que en cada entrada de RIP al principio hay una R indicándolo. Estas entradas indican redes que no están directamente conectadas y en la misma entrada existe la dirección de la interfaz del router por donde ir en caso de que se referencia a esa red. Esto es gracias a RIP que envía paquetes periódicos actualizando las rutas y la información que tiene de la red actual.

Si intentamos abrir la captura de eth0 del PC1 filtrando el tráfico RIP se observan paquetes de Request con origen el router 1 y destino un grupo de difusión en el que los diferentes dispositivos no están configurados. El switch, al recibir esta dirección de difusión, reenvía el paquete a todos los dispositivos conectados, generando un tráfico innecesario dado que se trata de una red con un único router en el que no es necesario la existencia de paquetes RIP porque no existen routers ‘más allá’. Además, existe una brecha de seguridad ya que alguien podría capturar los paquetes RIP y averiguar la topología de la red privada.

Para solucionar el tráfico innecesario y la brecha de seguridad, necesitamos declarar interfaces pasivas en las que los routers puedan recibir peticiones RIP pero no enviarlas. Así, editamos las interfaces que conectadas directamente a los hosts (*passive-interface ethX*). Una vez hecho esto, si empezamos a capturar el tráfico en los diferentes hosts, ya no se observan paquetes RIP, pero, sin embargo, se observa también que en el router 2 ya no hay ninguna entrada en la que la ruta de destino sea la interfaz eth1 del router 3 y en el router 3 no se ve ninguna entrada cuya ruta de destino sea la eth1 del router 2. Esto implica que ese enlace es invisible para RIP, ya que lo hemos declarado pasivo y no se van a enviar paquetes RIP para ver el estado de la red entre ambos routers. Por tanto, si cortamos el enlace entre el router 1 y el 3, el router 3 queda aislado de la red y no se puede comunicar ni con el router 1 ni sus hosts ni con el 2 aunque sí puede seguir accediendo a los hosts de su eth1 porque está directamente conectada y a la eth1 del router 2.

Para añadir el enlace entre ambos routers y que no envíen paquetes RIP, debemos decirle explícitamente a estos dos routers que esta ruta existe y es válida, y que se envíen paquetes RIP entre ellos solos, sin necesidad de enviarlo a un grupo de difusión. Para ello, usamos el comando *neighbor IP* donde IP es la dirección de la interfaz del otro router en la misma red.

En el siguiente paso, existe un problema de seguridad en caso de que alguien inyecte paquetes RIP en la red, alterando las tablas de enrutamiento de los routers y pudiendo redirigir el tráfico a donde el atacante quiera, siendo esto una grave falla de seguridad. Para solventarlo, necesitamos autenticar estos paquetes empleando un mecanismo criptográfico basado en llaves que provee RIPv2. Para activar el mecanismo en un router: *key chain kal -> key 1 -> key-string “clave”*, para crear la llave y luego seleccionamos las interfaces con *interface ethX* *-> ip rip authentication mode text -> ip rip authentication key-chain kal.* Si solo activamos este mecanismo en el router 1, desde el router 3 no hay ninguna entrada para la red 192.168.1.0/26 porque los paquetes RIP enviados al router 1 son rechazados por no tener la autenticación activada y tener la misma clave. Sin embargo, sí aparece la ruta 192.168.1.132/30 porque eligió la ruta desde el router 2 que tampoco tiene la autenticación activada.

Por último, queremos establecer una ruta por defecto en todos los routers para redirigir el tráfico al router 3 y que este lo mande al 4 (Internet). Para lograr esto, primero debemos añadir una ruta por defecto estática desde el router 3 al 4 con *ip route 0.0.0.0/0 80.10.20.66*. Ahora en el router 4 añadimos una ruta sumarizada (que represente a toda la red privada interna) para dirigir los paquetes de esta red al router 3 mediante *ip route 192.168.1.0/24 80.10.20.65*. Se ha elegido la máscara 24 porque abarca todas las posibles direcciones de la red privada (aunque sobren algunas). Ahora ya solo que propagar la información de que el router 3 es el router por defecto con *router rip -> default-information originate* y se propagará por todos los routers de la red, añadiendo una entrada 0.0.0.0/0 que lleva al router 3. Ahora si comprobamos, se observa que hay conectividad entre el PC-1 y el PC-4. Si usamos *traceroute*, se observa que pasa por la interfaz eth2 del router 1, la eth0 del router 3, la eth0 del router 4 y finalmente la eth0 del PC-4. Si eliminamos el enlace entre el router 1 y 3, *traceroute* indica que ahora pasa por la interfaz eth0 del router 2 adicionalmente para llegar a la eth1 del router 3

# Resumen de los comandos y ficheros

## Comandos

-*ifconfig (interfaz)* --- Muestra información de la interfaz de red ‘interfaz’.

-*ifconfig (interfaz) (up/down)* --- Activa o desactiva la interfaz de red ‘interfaz’.

-*ip addr* --- Muestra información de las interfaces de red.

-*ip addr add (IP/mask) dev (interfaz)* --- Añade una dirección a la interfaz seleccionada.

-*ip link set eth0 (up/down)* --- Activa o desactiva la interfaz de red ‘interfaz’.

-*ping (IP)* --- Prueba la conexión entre el host local y la IP destino.

-*if(up/down) (interfaz)* --- Activa o desactiva los cambios del fichero de configuración de una interfaz.

-*vtysh* --- En un router Quagga, entra en una Shell de configuración.

-*write* --- Guarda toda la información modificada. (Importante)

-*show ip route* --- Muestra la tabla de enrutamiento.

-*show running-config* --- Muestra información de las interfaces actuales

-*sh interface (interfaz)* --- Muestra información de la interfaz ‘interfaz’.

-*configure terminal* --- Entrar en modo configuración.

-*interface ethX* --- Selecciona la interfaz a configurar.

-*ip address IP/mask* --- Asigna la IP correspondiente.

-*no shutdown* --- No desactivar la interfaz.

-*link-detect* --- Activa la configuración anterior.

-*ip rip authentication mode text*

*-ip rip authentication key-chain kal*

-*ip route RED/mask Gateway* --- Añade una entrada a la tabla enrutamiento.

-*router rip* --- Entra en la configuración de RIP.

*-version 2* --- Activar RIPv2 en el router.

-*passive-interface eth2* --- Declara la interfaz como pasiva (solo recibe).

-*neighbor IP* --- Establece una comunicación directa entre dos routers sin necesidad de que estén enviando paquetes multicast.

-*default-information originate* --- Establece el router como fuente de información por defecto y propaga dicha información al resto de routers, añadiendo en cada uno de ellos una ruta por defecto hacia él.

-*key chain kal* --- Entra en el menú de autenticación.

-*key N* --- Selecciona una llave.

-*key-string CLAVE* --- Elige una clave.

-*route (-n)* --- Permite cambiar las tablas de enrutamiento de un PC. Con -n es para mostrarla.

-*ip route* --- Lo mismo que el anterior.

## Ficheros de configuración

-/etc/network/interfaces --- Se activan al iniciarse el PC las interfaces configuradas

auto ethX

iface ethX inet **static**

address IP/mask

gateway IP\_ROUTER