

02 Laboratorio de routing

CC50P, Taller de Redes de Datos

Profesor: Sebastián Castro A.

Versión: 2006-2

Introducción

Para asegurar la conectividad IP no basta asegurar que las señales viajen de un lado a otro, los datagramas IP deben ser capaces de encontrar el camino a su destino. El proceso que sigue un datagrama IP, saltando de nodo en nodo hasta llegar a su destino, se conoce como **forwarding IP**. Un elemento esencial para el proceso es la “tabla de rutas”, que indica los diferentes caminos que puede tomar el datagrama.

La construcción de una tabla de rutas puede seguir dos estrategias: estática o dinámica. En el caso de una construcción estática, donde personal humano configura los caminos, adolece de falta de respuesta en caso de eventos que afecten la conectividad: adicción, falla o cambio en las condiciones de un enlace. Por ello, para asegurar la correcta operación de la red IP, se utiliza una estrategia dinámica.

La estrategia dinámica introduce un nuevo concepto: protocolos de routing. Éstos son ejecutados por los routers y proveen mecanismos para intercambiar información entre ellos acerca de la topología de la red. Con ellos, se pueden calcular caminos entre los diferentes componentes de la red y construir la tabla de forwarding de cada router.

Los protocolos de routing se dividen entre internos y externos. Ejemplos de protocolos internos son RIP, OSPF y IS-IS. Protocolos externos son BGP y EIGRP.

Herramientas a utilizar

Linux

Usando un computador con Linux y el software adecuado, vamos a simular un router que nos permita realizar la experiencia. Todos los comandos deberán ser ejecutados como root.

Comandos Necesarios

- **ifconfig**: permite configurar y chequear el estado de las interfaces de red, en particular la interfaz eth0 es normalmente la interfaz real de red existente, algunos ejemplos:
 - *ifconfig -a*, despliega el estado y configuración de todas las interfaces existentes para el sistema.
 - *ifconfig eth0 inet 192.168.88.3 netmask 255.255.255.240 broadcast 192.168.88.15 mtu 1496 up*, configura la interfaz eth0 con el numero ip 192.168.88.3, define la mtu en 1496 y activa la interfaz si esta estuviera desactivada.
 - *ifconfig eth0:0 inet 192.168.89.4 netmask 255.255.255.240 up*, crea la interfaz virtual eth0:0 y le asigna la dirección IP 192.168.89.4.
- **vconfig**: permite configurar y chequear el estado de las vlan agregadas al sistema, una vez que se agrega una vlan al sistema esta aparece como interfaz de red. Los comandos típicos son:
 - *vconfig add eth0 37*, agrega la interfaz “eth0” a la vlan con TAG 37, creando la interfaz virtual “eth0.37”. Dicha interfaz queda “administrativamente desactivada”

al momento de definirla. Para activarla, deberá asignarle una IP y ponerla en funcionamiento usando `ifconfig`.

- `tcpdump`: permite mirar los paquetes que llegan o salen por la interfaz de red seleccionada, por ejemplo: `tcpdump -qn -i eth0`
- `modprobe`: permite agregar en forma dinámica módulos al sistema operativo mediante lo cual este adquiere nuevas funcionalidades, por ejemplo: “`modprobe 8021q`”, agrega la capacidad al sistema de manejar vlan.
- `sysctl`: permite leer o cambiar el valor de una variable del kernel. Lo utilizaremos para habilitar el IP forwarding a nivel del kernel (usando `sysctl -w net/ipv4/ip_forward=1`)

Software de routing

Utilizaremos para la experiencia un software llamado “quagga”, sucesor del software “zebra”. Este programa está compuesto de implementación para todos los protocolos de routing estándar (RIP, RIPng, OSPF, OSPFng, BGP) y cada protocolo es operado por un “demonio” diferente. Además, existe otro programa integrante del paquete, llamado “zebra” (por razones históricas). Éste toma la información de routing aportada por todos los otros demonios, calcula la tabla de forwarding y la instala en la tabla del kernel.

Cada programa de routing tiene un puerto en la IP local (localhost) donde activa una interfaz de texto para aceptar los comandos. Así, el puerto 2601 es el puerto de administración de zebra; el 2602 para RIP; el 2604 para ospf; el 2605 para bgp. Para poder entrar a la ventana de administración, deberá ejecutar `telnet localhost puerto`, donde *puerto* es el número de su interés. El programa “zebra” tiene además algunos comandos útiles para depurar la construcción de la tabla de forwarding.

- `show ip route`, muestra la tabla de rutas IP, clasificando las entradas por tipos como Kernel, Static, Ospf, Rip, Bgp, etc.
- `ip route <red>< ip_gateway>`, agrega una ruta estática al router, que después se puede usar para comunicarla a otros routers. Ejemplo, `ip route 172.30.10.0/24 172.31.1.1` agrega una ruta estática a la red 172.30.10.0/24 a través del router 172.31.1.1.

Todas las interfaces tienen el común dos modos de operación: operador y administrador. Los comandos que ejecuten para configurar “zebra” o los demonios de routing deberán ser ejecutados como administrador. Para habilitar dicho modo dentro de la interfaz, usen el comando “enable” o “ena”.

Dentro de los demonios de routing y trabajando en modo administrador, antes de cambiar cualquier configuración, deberán ejecutar “configure terminal” para habilitar el ingreso de cambios.

La clave de ingreso para todas las interfaces de configuración asociadas a quagga será “taller”.

Configuración de OSPF

El software a utilizar tiene algunas limitaciones con respecto a la especificación de CISCO. Por tanto, las instrucciones de configuración de OSPF vista en clases no sirven bien para nuestro caso. Por tanto, las instrucciones para configurar OSPF en quagga son:

```
router ospf
  ospf router-id ID
  network red_interface_a_usar area id_area
```

Donde *ID* es un identificador del router; *red_interface_a_usar* es la combinación red/máscara de la interfaz a usar para hablar OSPF. ¡NO ES LA IP DEL VECINO OSPF!. Sea muy cuidadoso de hacer coincidir el número de red y la máscara usada en la configuración de la interfaz con el valor especificado aquí. Sea muy cuidadoso con el número de broadcast especificado en la interfaz a usar para OSPF. El *id_area* identifica un área de OSPF. Entre cada par de vecinos deberán tener un área diferente.

Recordatorio de máscaras

Durante el transcurso del laboratorio, necesitará recordar algunas máscaras de red.
/24 es 255.255.255.0, /30 es 255.255.255.252.

Infraestructura de trabajo

Para realizar las diferentes experiencias de este laboratorio, contaremos con un switch “Cisco Catalyst 2650”, 8 computadores tipo PC corriendo Linux Knoppix, 1 notebook con Linux, conectado al switch mediante un cable serial (para tener acceso a la consola de administración). Cada alumno estará a cargo de un computador, que se configurará para operar como router de algún tipo. Quedarán dos computadores libres que serán los “clientes” de la red a construir, y que usarán para verificar la conectividad IP a través de los routers a configurar.

Experiencia 1: Instalación

Objetivo: Instalar el software para administración de VLAN y el software de routing en los computadores. El software para VLAN debe ir en los 11 computadores, el de routing sólo en los 8 seleccionados. Instalar archivos de configuración base para los routers.

Datos:

- Red 192.168.0.0/25 (máscara 255.255.255.128)
- Dirección 192.168.0.126 será la dirección IP del servidor del profesor.

Procedimiento

1. Configure la interfaz eth0 de su computador, utilizando una dirección del rango especificado. Para evitar conflictos, utilice su número de grupo como último número de la dirección IP.
2. Verifique conectividad con el profesor, ejecutando “ping 192.168.0.126”
3. Descargue el archivo con las herramientas para VLAN en Linux, mediante wget. El archivo está en el computador del profesor, en la URL <http://192.168.0.126/vlan.tar.gz>. Guarde el archivo en el directorio “/ramdisk”.
4. Descargue el archivo con la herramienta de routing, mediante “wget”. El archivo está en el computador del profesor, en la URL <http://192.168.0.126/quagga.tar.gz>. Guarde el archivo en el directorio “/ramdisk”.
5. Descargue el archivo con los archivos de configuración para los programas de routing. El archivo está en el computador del profesor, en la URL <http://192.168.0.126/conf-quagga.tar>. Guarde el archivo en el directorio “/ramdisk”.
6. Descargue el shell script para compilar quagga desde la URL <http://192.168.0.126/make-quagga.sh> y guárdelo en el directorio “/ramdisk”.
7. Descomprima el archivo *vlan.tar.gz*, y compílelo usando “make”.

8. Descomprima el archivo *quagga.tar.gz*, prepare la compilación, compílelo e instálelo usando el shell script *make-quagga.sh*.
9. Descomprima el archivo *conf-quagga.tar* dentro del directorio “*/ramdisk/quagga/etc*”.
10. Desconfigure la interfaz de red *eth0* (asignándole la IP 10.0.0.1, máscara 255.255.255.255), para evitar interferencias posteriores.

Experiencia 2: Router OSPF y Host

Datos

- VLAN 1X entre los vecinos OSPF.
- Red IP entre routers OSPF: 172.18.X.0/30
- Area X entre vecinos OSPF
- VLAN 3X entre el router OSPF y los hosts clientes.
- Red IP entre router/host: 192.168.X.64/30.

Instrucciones

1. Configure ambas VLAN asociadas en el switch. Recuerde que las puertas que use deberán ser definidas como “trunk”.
2. Cargue el módulo “8021q” en Linux, usando “modprobe”, para agregar el soporte VLAN.
3. Cree las interfaces VLAN en el Linux, usando vconfig.
4. Asigne direcciones IP a las interfaces recién creadas, según instrucciones.
5. Verifique conectividad entre host/router (Ud. Mismo, pues tiene ambos computadores) y entre router/router (con el apoyo del compañero que tenga asignado dicho router) usando ping.
6. Habilite forwarding IP usando “sysctl”.
7. Inicie “zebra” ejecutando “./zebra -d” en el directorio */ramdisk/quagga/sbin*.
8. Inicie “ospf” ejecutando “./ospfd -d” en el directorio */ramdisk/quagga/sbin*.
9. Verifique establecimiento de la sesión RIP con el otro router usando las instrucciones dadas en clase (**show ip ospf neighbor**).
10. Verifique recepción de las rutas anunciadas por el vecino OSPF, usando “zebra” (**show ip route**).

Experiencia 4: Router OSPF/BGP

Datos

- VLAN 1X entre vecinos OSPF.
- VLAN 4 entre vecinos BGP.
- Red IP entre routers OSPF: 172.18.X.0/30
- Area X entre vecinos OSPF.
- Red IP entre routers BGP: 172.17.10.0/30
- ASN 65000 anuncia red 10.10.0.0/24
- ASN 65001 anuncia red 10.20.0.0/24

Instrucciones

1. Configure ambas VLAN asociadas en el switch. Recuerde que las puertas que use deberán ser definidas como “trunk”.
2. Cargue el módulo “8021q” en Linux, usando “modprobe”, para agregar el soporte VLAN.
3. Cree las interfaces VLAN en el Linux, usando vconfig.

4. Asigne direcciones IP a las interfaces recién creadas, según instrucciones.
5. Verifique conectividad entre router/router (con el apoyo del compañero que tenga asignado dicho router) usando ping.
6. Habilite forwarding IP usando “sysctl”.
7. Inicie “zebra” ejecutando “./zebra -d” en el directorio /ramdisk/quagga/sbin.
8. Inicie “ospf” ejecutando “./ospfd -d” en el directorio /ramdisk/quagga/sbin.
9. Inicie “bgp” ejecutando “./bgpd -d” en el directorio /ramdisk/quagga/bin
10. Verifique establecimiento de las sesiones OSPF con los otros routers usando las instrucciones dadas en clase.
11. Verifique establecimiento de la sesión BGP con el vecino, usando las instrucciones dadas en clases.
12. Verifique recepción de las rutas anunciadas por el vecino BGP y los vecinos OSPF, usando “zebra”.

Experiencia 5: Verificación de la configuración

Realizadas todas las instrucciones, los computadores hosts de la red *deberían* ser capaces de comunicarse a nivel IP entre sí. Se verificará correcta configuración de los routers mediante ping entre los hosts.