

ANEXO C. CONFIGURACIÓN DE ENRUTADORES MULTICAST

CONFIGURACIÓN DE UN ENRUTADOR MULTICAST CON EL DEMONIO MROUTED

El sistema operativo Linux en sus diversas distribuciones cuenta con un demonio de enrutamiento llamado **mrouterd** y es una implementación del protocolo DVMRP que permite a una máquina actuar como un enrutador multicast. Este demonio también se puede obtener para diferentes sistemas Unix, como Solaris, FreeBSD, entre otros. Se puede encontrar información de su uso en la historia del Mbone en el capítulo 5 "Mbone – Backbone Multicast de Internet". En la práctica para la implementación del proyecto en la red de la Universidad se trabajó con el sistema operativo Linux en sus versiones Debian 3.0 y RedHat 9.0. Si se trabaja con una distribución Debian se tiene que recompilar el Kernel habilitando las opciones: CONFIG_IP_MULTICAST, CONFIG_IP_ROUTER, CONFIG_IP_MROUTE y CONFIG_NET_IPIP que son necesarias para que el equipo funcione como un enrutador multicast; si se trabaja con RedHat 9.0 o posterior estas opciones están configuradas por defecto en el sistema base.

El archivo de configuración de mrouterd se encuentra por lo general en */etc/mrouterd.conf* aunque se puede escoger otra ubicación.

El formato de este archivo no es indentado, es decir, no toma en cuenta los espacios en blanco ni los caracteres de terminación de línea. Las entradas **phyint**, **túnel** y **name** se pueden declarar mas de una vez. Los valores **boundary** y **altnet** se pueden especificar tantas veces como sea necesario.

Las siguientes entradas y sus opciones pueden ser utilizadas en el archivo *mrouterd.conf*:

- **phyint** *local_addr* [**disable**] [**metric** *m*] [**threshold** *t*] [**rate_limit** *b*] [**boundary** (*boundary_name*|*scoped_addr/mask_len*)] [**alnet network/mask_len**]

La entrada **phyint** se puede utilizar para deshabilitar el enrutamiento multicast en la interfaz física identificada por la dirección IP local *local_addr* o para asociarla con un valor de métrica o umbral que no está por defecto. La dirección IP local se puede reemplazar por el nombre de la interfaz (por ejemplo eth0). Si una interfaz está vinculada a varias subredes IP la opción **alnet** describirá cada subred adicional. Las entradas **phyint** deben estar declaradas antes que las entradas **tunnel**.

Las opciones para la entrada **phyint** y su significado es el siguiente:

local_addr: especifica una dirección local, usando una dirección IP o el nombre de la interfaz, como s0.

disable: deshabilita el enrutamiento multicast en la interfaz física identificada con *local_addr*.

metric *m*: especifica el costo asociado a la interfaz o túnel dado para mandar datagramas. Esta opción se puede usar para afectar la elección de rutas. El valor por defecto de *m* es 1. La métrica debe mantenerse lo mas pequeña posible, por que **mrouted** no puede enrutar a lo largo de caminos que sumen una métrica mayor a 31.

threshold *t*: especifica el TTL IP mínimo que se requiere para que un datagrama multicast sea enviado por la interfaz o túnel indicado. Esta opción controla el alcance de los datagramas multicast (el TTL de los paquetes reenviados solamente se compara con el valor de *threshold* pero no se decrementa por este valor). El valor por defecto de *t* es 1. En general, todos los demonios **mrouted** conectados a un túnel o subred particular deben usar la misma métrica y umbral para ese túnel o subred.

rate_limit *b*: especifica el ancho de banda en Kbps que se tiene asignado para el tráfico multicast. El valor por defecto es 500 Kbps en tuneles y 0 (ilimitado) en interfaces físicas.

boundary *boundary_name*|*scoped_addr/mask_len*: configura una interfaz como un límite administrativo para la dirección definida. Los paquetes pertenecientes a esta dirección no se reenvían por esa interfaz. La opción

boundary acepta tanto el nombre del limite como la dirección definida con su máscara de red. El *boundary_name* es el nombre asignado a un limite a través de la entrada **name**. El valor *scoped_addr* es una dirección multicasat. El valor *mask_len* es la longitud de la máscara de red.

alnet network/mask_len: especifica una subred o red adicional vinculada a la interfaz física descrita en la entrada **phyint**. El valor *mask_len* es la longitud de la máscara de red.

- **tunnel local_addr remote_addr [metric m] [threshold t] [rate_limit b] [boundary (boundary_name|scoped_addr/mask_len)] [alnet network/mask_len]**

La entrada **tunnel** se usa para establecer un túnel entre la dirección IP local (*local_addr*) y la dirección IP remota (*remote_addr*) y para asociarla con un valor de métrica o umbral que no está por defecto. La dirección IP local se puede reemplazar por el nombre de la interfaz (por ejemplo eth1). La dirección IP remota se puede reemplazar por el nombre del host si y sólo si tiene asociada una sola dirección IP. El túnel debe ser configurado en los archivos **mrouterd.conf** de ambos enrutadores para poder ser utilizado. La descripción de las opciones usadas con la entrada **tunnel** es la misma de la entrada **phyint**.

- **cache_lifetime ct**

La entrada **cache_lifetime** determina la cantidad de tiempo que una ruta multicast guardada en cache estará presente en el kernel antes de expirar. El valor de *ct* esta dado en segundos y debe estar entre 300 (5 minutos) y 86400 (un día). El valor por defecto es 300 segundos.

- **pruning state**

La entrada **pruning** habilita a **mrouterd** para actuar como un enrutador *non-pruning*. El valor de *state* puede ser **on** o **off**. Se debe configurar el enrutador como *non-pruning* sólo para propósitos de pruebas. El valor por defecto es **on** el cual habilita la función de podado.

- **name boundary_name scoped_addr/mask_len**

la entrada **name** permite asignar nombres a los limites para hacer mas fácil la configuración. La opción **boundary** en las entradas **phyint** y **tunnel** acepta el nombre del limite o la dirección definida. *boundary_name* es el nombre que usted

que se desea dar al límite. El valor *scoped_addr* es una dirección multicast. El valor *mask_len* es la longitud de la máscara de red.

MROUTED identifica cada interfaz física y cada túnel como *vif* (virtual interface) y asigna el valor de 0 para la primera interfaz física y termina con el último túnel definido en el fichero *mouted.conf*. A continuación está el archivo de configuración que se utilizó en la red Unicauca.

El demonio *mouted* se puede lanzar en una terminal siguiendo la siguiente sintaxis:

```
mouted [-c config_file] [-d [debug_level]] [-p]
```

- La opción *-c* indica que la cadena *config_file* es el archivo de configuración que el demonio utilizará, por defecto se utiliza el */etc/mouted.conf*.
- La opción *-d* indica el nivel de depuración (debug level) que se mostrará en pantalla. El valor *debug_level* puede tomar valores de 0 a 3, el valor por defecto es 0. Si esta opción no es habilitada el demonio correrá como un proceso normal en background.
- La opción obliga al protocolo DVMRP a actuar en modo de no podado, por lo que no es recomendada.

Una vez lanzado el demonio *mouted* por lo general escribe su *pid* en el archivo */var/run/mouted.pid* o en */etc/mouted.pid*.

El demonio *mouted* soporta las siguientes señales del sistema operativo:

- HUP: reinicia el demonio forzándolo a que lea nuevamente el archivo de configuración.
- INT: termina su ejecución de una manera adecuada mandando mensajes de abandono (goodbye message) a los enrutadores vecinos.
- TERM: igual que la señal INT.
- USR1: vuelca las tablas de enrutamiento interno al fichero */var/tmp/mouted.dump*.
- USR2: vuelca las tablas de cache interno al fichero */var/tmp/mouted.cache*.

- QUIT: vuelca las tablas de enrutamiento interno a la salida de error estándar del sistema o *stderr* (solo si está en un nivel de depuración diferente de 0).

La forma adecuada de enviar estas señales a través de una consola es la siguiente:

```
kill -signal pid
```

Donde *-signal* es una de las señales descritas anteriormente y *pid* el es identificador de proceso del demonio.

Una salida típica de la señal USR1 seria la siguiente:

```
Virtual Interface Table
Vif  Local-Address      Metric  Thresh  Flags
  0   36.2.0.8          subnet: 36.2        1       1     querier
      groups: 224.0.2.1
      224.0.0.4
      pkts in: 3456
      pkts out: 2322323

  1   36.11.0.1         subnet: 36.11       1       1     querier
      groups: 224.0.2.1
      224.0.1.0
      224.0.0.4
      pkts in: 345
      pkts out: 3456

  2   36.2.0.8          tunnel: 36.8.0.77   3       1
      peers: 36.8.0.77 (2.2)
      boundaries: 239.0.1
      : 239.1.2
      pkts in: 34545433
      pkts out: 234342

  3   36.2.0.8          tunnel: 36.6.8.23   3       16

Multicast Routing Table (1136 entries)
Origin-Subnet  From-Gateway  Metric  Tmr  In-Vif  Out-Vifs
36.2           36.8.0.77     1       45   0       1* 2 3*
36.8           36.8.0.77     4       15   2       0* 1* 3*
36.11          36.8.0.77     1       20   1       0* 2 3*
.
.
.
```

Existen cuatro *vifs* que conectan dos subnets y dos túneles. La *vif* 3 es un túnel que no esta en uso, la *vif* 0 y la *vif* 1 conectan a subredes que tienen algunos grupos

presentes, los túneles nunca tienen grupos unidos. En este ejemplo, el demonio `mrouterd` es el responsable de mandar periódicamente mensajes de petición de pertenencia a grupo por las interfaces `vif 0` y `vif 1` como lo indica la bandera "querier". La lista de límites muestra el alcance de las direcciones en cada interfaz como también se puede observar el registro de los paquetes entrantes y salientes. Asociado a cada subred desde la cual un datagrama multicast puede ser originado están: la dirección IP del enrutador del salto anterior (a menos que la red este directamente conectada), la métrica de la ruta inversa hacia el origen, el periodo de tiempo desde que se recibió la ultima actualización de esa red, la `vif` que lleva hacia el origen y una lista de las `vifs` salientes. Un `*` indica que el enrutador es un enrutador leaf del árbol broadcast para esa `vif`.

Un ejemplo de la salida de la señal `USR2` se puede ver a continuación:

```
Multicast Routing Cache Table (147 entries)
  Origin          Mcast-group      CTmr  Age  Ptmr  IVif  Forwvifs
  13.2.116/22     224.2.127.255   3m    2m   -    0    1
>13.2.116.19
>13.2.116.196
  138.96.48/21   224.2.127.255   5m    2m   -    0    1
>138.96.48.108
  128.9.160/20   224.2.127.255   3m    2m   -    0    1
>128.9.160.45
  198.106.194/24 224.2.135.190   9m    28s  9m   0P
>198.106.194.22
```

Cada entrada se caracteriza por el número de la subred de origen, su máscara y el grupo multicast destino. Cada línea adicional que comience por un ">" hace referencia a cada fuente de la subred (puede haber mas de una). Adicional a esta información se presentan los siguientes campos:

- *Age*, es el tiempo desde que cada entrada fue creada en el cache.
- *CTmr*, indica el tiempo de vida de cada entrada, cuando este valor llega a cero la entrada es borrada del cache.
- *Forwvifs*, muestra las interfaces por las cuales se debe reenviar el tráfico que pertenezca al par (S, G). Una *p* representa que no se reenvía ningún datagrama por esa interfaz. Cuando no hay ninguna interfaz quiere decir que se está en una subred leaf y que no hay miembros presentes para ese (S, G). Una *b* advierte que esta interfaz es un limite para esta dirección y los paquetes no serán reenviados.

- *Ivif*, muestra la *vif* entrante para el tráfico multicast para ese origen. Si no existe ningún miembro para este grupo multicast en los enlaces del enrutador del árbol broadcast, se envía un mensaje de podado al enrutador upstream y se representa por una *p* al lado de la *vif*.
- *Ptmr*, es la cantidad de tiempo antes de que el mensaje de podado expire, si hay un punto significa que ningún mensaje ha sido enviado.

El archivo final `mrouterd.conf`, en el cual se especifica la implementación del túnel entre la Universidad del Cauca y RETINA, se muestra a continuación:

```
#Se definen las direcciones multicast para uso local
name LOCAL 239.255.0.0/16

#La interfaz eth1 es de la red interna (172.16.0.0/16) y por eso se
#deshabilita
phyint eth1 disable

#A continuación se indica el túnel con RETINA
#    dir.Unicauca dir.RETINA
tunnel 200.30.71.32 200.10.202.1 metric 1 threshold 16 rate_limit 500
boundary LOCAL #<-- REPLACE
```

CONFIGURACIÓN DE UN ENRUTADOR CISCO

Habilitar un enrutador Cisco en el manejo de tráfico multicast es una tarea bastante sencilla para el administrador de una red, se deben utilizar dos comandos globales como se ilustra a continuación.

- **ip multicast-routing [distributed]**

Este comando habilita el reenvío IP multicast; si se deshabilita los paquetes IP direccionados a un grupo multicast del cual el enrutador no es miembro serán descartados. Por defecto esta opción está deshabilitada. Este comando fue introducido en la versión IOS 10.2. La palabra opcional **distributed** habilita la conmutación rápida distribuida del enrutador. Esta opción es soportada por los enrutadores Cisco de la serie 7500 y 12000 y fue introducida en la versión del IOS 11.1(20)

- **ip multicast multipath**

Si existen varias interfaces por las que se pueda alcanzar una misma red con igual costo en la tabla de enrutamiento unicast, se emplea este comando para que se puedan utilizar estas interfaces en el RPF. La carga compartida es sólo para el par (S, G). Este comando es soportado por la versión del IOS 12.0, 12.0S y 12.0T.

Para configurar el protocolo de enrutamiento multicast PIM en las interfaces, se utilizan dos comandos de configuración de interfaz.

- **ip pim sparse-dense-mode**

Este comando habilita el protocolo de enrutamiento multicast PIM en modo denso y en modo disperso en una interfaz. El enrutador detectará si el vecino en esa interfaz utiliza un protocolo de modo denso o modo disperso.

- **ip mroute-cache [distributed]**

Este comando configura la conmutación rápida multicast. Por defecto esta opción está habilitada en todas las interfaces; fue introducida en la versión del IOS 11.0.

La opción **distributed** permite que la interfaz realice una conmutación rápida distribuida en los paquetes de entrada. Este comando se aplica a las interfaces físicas y no a las subinterfaces. Se debe configurar con anterioridad el comando global **ip route-cache distributed** en la serie 7500 de Cisco (no es necesario en la serie 12000 Gigabit Switch Router). Esta opción se introdujo en la versión IOS 11.1(20).

Los enrutadores Cisco que utilizan el protocolo de enrutamiento multicast PIM pueden interoperar con enrutadores multicast que trabajen con el protocolo DVMRP.

Los enrutadores PIM descubren dinámicamente los enrutadores multicast DVMRP, cuando lo hacen envían periódicamente mensajes de reporte DVMRP anunciando las fuentes unicast que se pueden alcanzar en el dominio PIM. El enrutador reenvía los paquetes multicast que han sido enviados por el enrutador DVMRP y también reenvía los paquetes multicast a los enrutadores DVMRP.

Es necesario utilizar la versión 3.8 del mrouterd, la cual implementa una versión non-pruning de DVMRP, cuando los enrutadores Cisco están conectados directamente a los enrutadores DVMRP o interoperan con enrutadores DVMRP a través de un túnel. Los mensajes DVMRP enviados por el software IOS de Cisco pueden causar daños en las tablas de enrutamiento de las versiones antiguas del mrouterd y de sus vecinos. Cualquier enrutador conectado al Mbone debe tener una lista de acceso para limitar el número de rutas unicast que se anuncian vía DVMRP.

Para que un enrutador Cisco pueda interoperar con un enrutador DVMRP se debe utilizar el siguiente comando.

- **ip dvmrp metric *metric* [list *access-list*] [*protocol process-id*]
[route-map *map-name*]**

Configura la métrica asociada a un grupo de destinos para los reportes DVMRP.

metric, tiene un valor entre 0 (ruta no anunciada) y 32 (ruta inalcanzable).

access-list, es el nombre o el número de la lista de acceso, si este argumento se especifica sólo los destinos multicast que concuerden con la lista de acceso serán reportados con la métrica configurada.

protocol, el nombre del protocolo unicast como *bgp*, *eigrp*, *igrp*, *isis*, *ospf*, *rip*, *static* o *dvmrp*; si se especificas estos argumentos sólo las rutas aprendidas por el protocolo de enrutamiento señalado se difundirán en los mensajes de reporte DVMRP.

process-id, es el número del identificador de proceso del protocolo de enrutamiento unicast.

route-map *map-name*, es el nombre del mapa de ruta. Solo los destinos que coincidan con el "route map" se reportan con la métrica configurada. Las rutas unicast son filtradas por el mapa de ruta antes de ser introducidas en los mensajes DVMRP.

Configuración de un túnel DVMRP en un enrutador Cisco

El software IOS de Cisco soporta túneles DVMRP hacia el Mbone. Se puede configurar un túnel DVMRP en un enrutador si en el otro extremo se está ejecutando DVMRP, de esta manera el software envía y recibe paquetes multicast a través del túnel; esto permite que dominios PIM se puedan conectar a un enrutador DVMRP en el caso donde todos los enrutadores en el camino no soporten enrutamiento multicast.

Cuando un enrutador Cisco corre DVMRP sobre un túnel anuncia las fuentes en los mensajes de reporte DVMRP igual a como lo hace en las redes reales. Además, el software guarda en cache los mensajes de reporte DVMRP que recibe y los utiliza en el RPF. Este comportamiento permite al software reenviar paquetes multicast recibidos por un túnel.

Para configurar un túnel se utilizan los siguientes comandos:

- **interface tunnel** *number*

Especifica una interfaz tipo túnel en el modo de configuración global y coloca al enrutador en el modo de configuración de interfaz. *number* es el número del túnel.

- **ip sap listen**
Habilita al software de Cisco para escuchar anuncios de sesión de directorio; para deshabilitar esta opción se antepone la palabra **no** al comando.
- **tunnel source** *ip-address*
Fija la dirección de la fuente del túnel. Esta es la dirección IP de la interfaz del enrutador.
- **tunnel destination** *ip-address*
Fija la dirección de destino del túnel. Esta es la dirección IP del enrutador mrouterd.
- **tunnel mode** *dvmrp*
Lo configura como un túnel DVMRP
- **ip unnumbered** *type number*
Configura la interfaz como no numerada. Es una interfaz que no tiene una dirección ip asignada.
- **ip pim** *sparse-dense-mode*
Configura el protocolo PIM en modo denso y disperso en el túnel.

Los comandos utilizados para la configuración del enrutador Cisco 2600 de la red Retina de Argentina con el cual se implementó el túnel con la red Unicauca son los siguientes.

```
# en el modo configure #
    ip multicast-routing
    ip multicast multipath

# los siguientes comandos en cada interfaz del enrutador #
    interface eth0
        ip pim sparse-dense-mode
        ip mroute-cache
        ip dvmrp metric 1

# lo siguiente en el túnel #
    interface tunnel 0
        ip unnumbered ethernet 0
        ip pim sparse-dense-mode
        ip sap listen
        tunnel source ethernet 0
```

```
tunnel destination 200.30.71.32  
tunnel mode dvmrp
```