Práctica 6. Border Gateway Protocol (BGP) y redundancia del primer salto

Objetivos de aprendizaje

- Entender el funcionamiento de BGP.
- Conocer los comandos para activar eBGP e iBGP y las diferencias entre ambos.
- Conocer los tipos de sistemas autónomos.
- Ser capaz de configurar BGP en un sistema autónomo multihomed.
- Manejo básico de mapas de rutas.

Introducción

Sistemas autónomos

Según el [RFC 1930], un sistema autónomo se define como Un grupo conectado de uno o más prefijos IP promovidos por uno o más operadores de red con una política de enrutamiento **única** y **claramente definida**.

Identificadores de sistemas autónomos

Cada sistema autónomo se identifica mediante un número entero asignado por la IANA (Internet Assigned Numbers Authority). Hasta 2007, estos identificadores eran de 16 bits, lo que permitía hasta un máximo de 65535 sistemas autónomos. A partir de este momento los identificadores se extendieron a 32 bits. En este caso, se consideran dos notaciones "asplain", que consiste en expresar el identificador como un número entero en notación decimal, y "asdot" en el que se separa en dos enteros de 16 bits mediante un punto `x.y`.

Los identificadores de sistemas autónomos se asignan por los registros regionales de internet (RIR, Regional Internet Registers), cada uno de los cuales se encarga de una zona geográfica distinta, tal y como se muestra en la figura 1:

- AFRINIC: África
- APNIC: Asia/Pacífico
- ARIN: Canada, USA e Islas del Caribe.
- LACNIC: América Latina e Islas del Caribe.
- RIPE NCC: Europa, Oriente Medio y Asia Central.

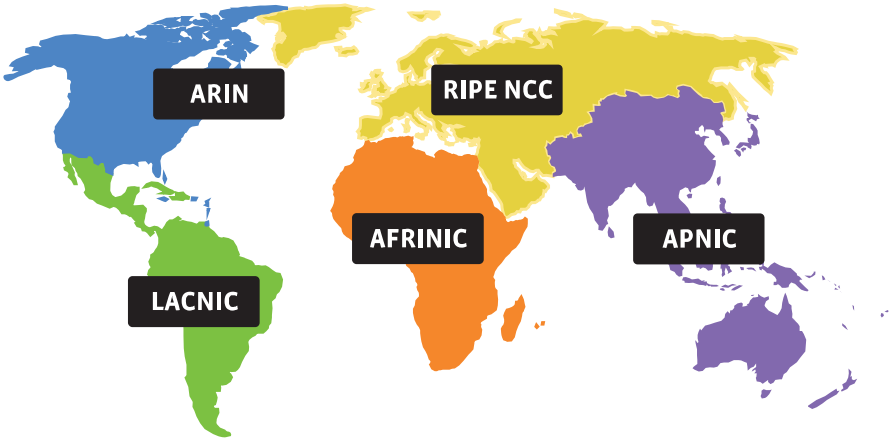

Figura 1: Mapa de las zonas gestionadas por cada RIR. Figura tomada de IANA.

Tipos de sistemas autónomos

- *Multihomed*: Son los que tienen conexiones con más de un sistema autónomo. Esto permite seguir conectado con Internet en el caso de fallar alguna conexión. Al contrario que los AS de tránsito no permiten el tráfico a través del sistema autónomo.
- *Stub*: Es un sistema autónomo que está conectado únicamente con otro sistema autónomo.
- Tránsito: Permite la interconexión entre otros sistemas autónomos a través de si mismo.
- *Internet Exchange Point* (IXP): Es una infraestructura física a través de la cual los proveedores de servicio o de contenido intercambian tráfico entre sus sistemas autónomos.

Enrutamiento entre sistemas autónomos: BGP (Border Gateway Protocol)

BGP utiliza TCP como protocolo de transporte, en el puerto 179. Dos routers BGP forman una conexión TCP entre ellos. Las parejas de routers con una conexión entre ellos se denominan peers. BGP contempla distintos tipos de mensajes que son intercambiados entre dos peers:

- OPEN: Se utiliza para el establecimiento de una sesión BGP una vez haya sido establecida la conexión TCP. Se suelen negociar ciertos parámetros que caracterizan a esa sesión. Por ejemplo, los peers se ponen de acuerdo en la versión de BGP que se va a utilizar.
- UPDATE: Es un mensaje de actualización que contiene los anuncios de nuevos prefijos. Se generarán mensajes de actualización cada vez que se determine una nueva ruta óptima para cierto destino o haya una modificación en alguna existente.
- KEEPALIVE: Cuando la sesión BGP está activa se envía periódicamente un mensaje para mantener viva la conexión.
- NOTIFICATION: Se envía al cerrar una sesión BGP. Esto sucede cuando ocurre algún error que requiera el cierre de la misma.

Atributos BGP

- ORIGIN: Identifica el mecanismo por el cual se anunció el prefijo IP por primera vez. Se puede especificar como IGP (0), EGP (1) o INCOMPLETE (2).

- AS-PATH: El atributo AS-PATH almacena una secuencia de números de AS que identifican la ruta de los AS por los que ha pasado el anuncio. Cada vez que un router frontera propaga una ruta añade a este atributo su número de AS constituyendo así la lista de los AS. La lista no se modifica si no se sale del sistema autónomo. Este atributo es el que luego permite la selección de rutas óptimas.
- NEXT-HOP: Identifica la dirección IP del router correspondiente al siguiente salto hacia el destino. Se modifica cuando se anuncia una ruta fuera del sistema autónomo o cuando se desea redirigir tráfico a otro interlocutor. La información contenida en este campo sirve para incluir los prefijos IP contenidos en el anuncio en la tabla de enrutamiento.
- LOCAL-PREF: Representa el grado de preferencia que el operador de red tiene por una determinada ruta dentro del sistema autónomo. El valor más alto indica una preferencia mayor. Por defecto, tiene el valor 100. El valor de este atributo es local al AS.
- Otros: Existen algunos atributos más que no se van a tratar aquí.

El proceso de enrutamiento: iBGP y eBGP

En el proceso BGP hay que distinguir dos casos:

- eBGP: Se produce cuando dos routers fronterizos de sistema autónomo intercambian información. En este caso, se toma el anuncio BGP del primer sistema autónomo y se fija el atributo NEXT-HOP a la dirección IP del router fronterizo que va a emitir el anuncio. Además se añade a la lista AS-PATH el número de sistema autónomo desde el que se emite el anuncio. Finalmente, se envía el anuncio al peer correspondiente del otro AS.
- iBGP: Se produce cuando dos routers pertenecientes al mismo AS intercambian información BGP. En este caso no se produce ninguna modificación sobre los atributos salvo que se indique explícitamente en la configuración.

De esta manera, cuando un anuncio llega a un AS , por un lado, se produce el peering iBGP entre los routers del mismo AS y, por otro, se redistribuye la información proveniente de BGP a los protocolos de pasarela interior.

Topología

La topología de red con la que se va a trabajar en esta práctica es la que se muestra en la figura 2. Se trata de la simulación de un AS *multihomed*, el AS 100, para el que se pretende tener conexión a Internet redundante a través de dos ISP distintos. Un ISP es el AS 200 y el segundo es el AS 300. Ambos ISP deben publicar una serie de prefijos de red mediante BGP y además se encuentran conectados a otros sistemas autónomos, que en este caso se simulan mediante el AS 400. En el AS 400 hay un servidor de internet con dirección IP 60.0.0.2, que se utilizará para probar el funcionamiento del esquema.
Hay que tener en cuenta que la red interna del AS 100 tiene dos partes:

- Una parte pública (coloreada en amarillo) cuyos prefijos deben publicarse en Internet y cuyos hosts tienen asignadas direcciones IP públicas.
- Una parte privada (coloreada en malva) cuyos host se direccionan con direcciones privadas y que no son enrutables en internet. Para obtener alcanzabilidad hacia Internet, debería realizarse un NAT (Network Address Translation) en el `QuaggaRouter-1`, pero este aspecto se resolverá en otra práctica.

Las redes de la parte privada deberían ser conocidas por el protocolo de enrutamiento de pasarela interior, pero nunca deben quedar publicadas hacia internet (fuera del AS). En los routers `QuaggaRouter-1` y `QuaggaRouter-3`, se encuentra activado el protocolo RIP. Revise las tablas de enrutamiento de estos router para asegurar que son correctas antes de comenzar la práctica. Recuerde que no deben aparecer entradas correspondientes a redes externas al AS100, exceptuando las redes de interconexión directamente conectadas con los router del ISP.

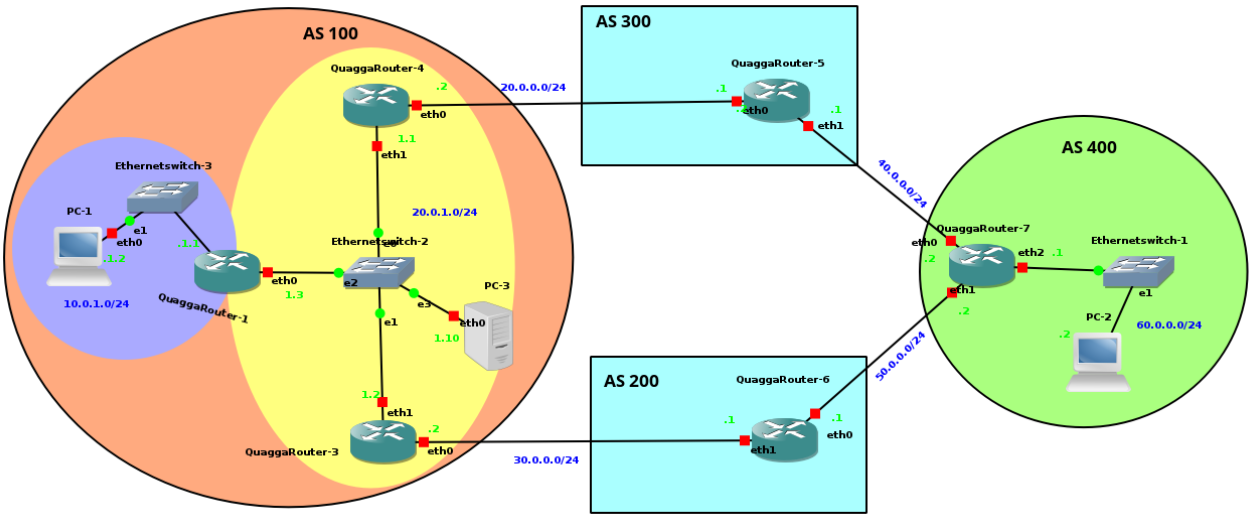

Figura 2: Esquema de red

Paso 1. Activar las sesiones BGP entre los peers

Para establecer una conexión entre dos peers se utiliza la siguiente secuencia de comandos:

```
router bgp <autonomous-system-number>
neighbor <ip-address> remote-as <as-number>
```

donde `<autonomous-system-number>` corresponde al número de sistema autónomo al que pertenece, `<ip-address>` es la direccion IP del peer, que es el router con el que se va a establecer la conexión, y `<as-number>` es el número del sistema autónomo del peer. De esta forma, en el caso de los vínculos externos eBGP `<autonomous-system-number>` y `<as-number>` serán distintos, mientras que para establecer una sesión iBGP ambos números corresponderán al número de sistema autónomo al que pertenece el router. Tenga en cuenta que las relaciones son simétricas por lo que deberá realizar esta secuencia de comandos en ambos peers para establecer una conexión.

Active una sesión eBGP entre los siguientes peers:

- `QuaggaRouter-4` y `QuaggaRouter-5`
- `QuaggaRouter-3` y `QuaggaRouter-6`
- `QuaggaRouter-5` y `QuaggaRouter-7`
- `QuaggaRouter-6` y `QuaggaRouter-7`

Igualmente deberá activar una sesión iBGP entre `QuaggaRouter-3` y `QuaggaRouter-4`. Generalmente, el atributo NEXT-HOP se fija sólo cuando el anuncio pasa de un sistema autónomo a otro. Esto puede suponer un problema cuando se establece una sesión BGP. Veámoslo con un ejemplo: Cuando un anuncio que proviene de `QuaggaRouter-6` llega a `QuaggaRouter-3`, NEXT-HOP se fija a la dirección IP de `QuaggaRouter-6`, que en este caso es 30.0.0.1. Ahora este mensaje se propaga hacia `QuaggaRouter-4` mediante la sesión iBGP. Por defecto, no se va a modificar el mensaje por ser iBGP. El problema surge en `QuaggaRouter-4` que toma las rutas anunciadas e intenta integrarlas en la tabla de enrutamiento. Sin embargo, no existe ninguna forma de llegar a 30.0.0.1, porque no es un destino conocido por los protocolos de pasarela interior. Por tanto, no se pueden integrar las rutas aprendidas en la tabla de enrutamiento. La solución a este problema es indicar que el valor del atributo NEXT-HOP se fije a la dirección IP del router que propaga el mensaje por iBGP, añadiendo el siguiente a los anteriores (sólo para sesiones iBGP):

```
neighbor <ip-address> next-hop-self
```

Paso 2. Comprobar que se han establecido las relaciones de peering entre los routers

Para comprobar los vecinos BGP de un nodo se utiliza el siguiente comando:

```
show bgp neighbors
```

Por ejemplo, en `QuaggaRouter-3` la salida debería ser similar a esta:

```
BGP neighbor is 20.0.1.1, remote AS 100, local AS 100, internal link
  BGP version 4, remote router ID 20.0.1.1
  BGP state = Established, up for 00:04:59
  Last read 00:00:59, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    4 Byte AS: advertised and received
    Route refresh: advertised and received(old & new)
    Address family IPv4 Unicast: advertised and received
  Message statistics:
    Inq depth is 0
    Outq depth is 0
                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                0          0
    Keepalives:             6          5
    Route Refresh:          0          0
    Capability:             0          0
    Total:                  7          6
  Minimum time between advertisement runs is 5 seconds

 For address family: IPv4 Unicast
  Community attribute sent to this neighbor(both)

  0 accepted prefixes

  Connections established 1; dropped 0
  Last reset never
Local host: 20.0.1.2, Local port: 50295
Foreign host: 20.0.1.1, Foreign port: 179
Nexthop: 20.0.1.2
Nexthop global: fe80::296:cbff:feab:be01
Nexthop local: ::
BGP connection: non shared network
Read thread: on  Write thread: off

BGP neighbor is 30.0.0.1, remote AS 200, local AS 100, external link
  BGP version 4, remote router ID 50.0.0.1
  BGP state = Established, up for 00:03:45
  Last read 00:00:45, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    4 Byte AS: advertised and received
    Route refresh: advertised and received(old & new)
    Address family IPv4 Unicast: advertised and received
  Message statistics:
    Inq depth is 0
```

```
    Outq depth is 0            Sent       Rcvd
    Opens:                  1          0
    Notifications:          0          0
    Updates:                0          0
    Keepalives:             5          4
    Route Refresh:          0          0
    Capability:             0          0
    Total:                  6          4
  Minimum time between advertisement runs is 30 seconds

 For address family: IPv4 Unicast
  Community attribute sent to this neighbor(both)
  0 accepted prefixes

  Connections established 1; dropped 0
  Last reset never
Local host: 30.0.0.2, Local port: 179
Foreign host: 30.0.0.1, Foreign port: 41824
Nexthop: 30.0.0.2
Nexthop global: fe80::296:cbff:feab:be00
Nexthop local: ::
BGP connection: non shared network
Read thread: on  Write thread: off
```

Observe como para el 20.0.1.1 la relación establecida es iBGP (`internal link`) mientras que para 30.0.0.1 es eBGP (`external link`)

Compruebe que todas las relaciones de vecindad establecidas son correctas.

Paso 3. Añadir anuncios de prefijo a los routers

Por el momento, BGP no hace nada, porque no hemos indicado ningún anuncio de prefijo en las configuraciones de los routers. Vamos a comenzar anunciando la red 60.0.0.0/24. Para ello introduzca el siguiente comando en el `QuaggaRouter-7`:

```
router bgp 400
network 60.0.0.0/24
```

Compruebe los prefijos conocidos por el router con el comando:

```
Debian# show bgp ipv4 unicast
BGP table version is 0, local router ID is 60.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
              i internal, r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 60.0.0.0/24      0.0.0.0                  0         32768 i

Total number of prefixes 1
```

En el `QuaggaRouter-7`, BGP conoce el prefijo 60.0.0.0/24. Haga la comprobación en el `QuaggaRouter-3` y el `QuaggaRouter-4`.

En `QuaggaRouter-4` la salida debería ser similar a la siguiente:

```
Debian# show bgp ipv4 unicast
```

```
BGP table version is 0, local router ID is 20.0.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
              i internal, r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
* i60.0.0.0/24      20.0.1.2                 100      0 200 400 i
*>                  20.0.0.1                          0 300 400 i

Total number of prefixes 1
```

Fíjese como el prefijo 60.0.0.0/24 se conoce y además es alcanzable por dos caminos, que es lo que se pretende para tener una conexión a internet redundante.

Revise ahora las tablas de enrutamiento de `QuaggaRouter-3` y `QuaggaRouter-4`. Observe que la entrada correspondiente al prefijo anunciado por el AS 400 ya está en la tabla de enrutamiento.

En `QuaggaRouter-3` la salida sería similar a la siguiente:

```
Debian# 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

R>* 10.0.1.0/24 [120/2] via 20.0.1.3, eth1, 00:31:39
C>* 20.0.1.0/24 is directly connected, eth1
C>* 30.0.0.0/24 is directly connected, eth0
B>* 60.0.0.0/24 [20/0] via 30.0.0.1, eth0, 00:04:53
C>* 127.0.0.0/8 is directly connected, lo
```

Paso 4. Propagar los prefijos internos del AS 100

En el paso 2 añadimos los prefijos de red del AS 400, pero no se hizo nada para el AS 100, por lo que los routers externos al AS no conocen este prefijo. Hay que tener en cuenta, que aquí sólo se va a publicar el prefijo correspondiente a la parte pública de la red, es decir las redes 20.0.1.0/24 y 20.0.0.0/24. Esto lo haremos mediante una ruta sumarizada, que corresponde al bloque de direcciones IP asignado al AS 100, que consideraremos que es el 20.0.0.0/16. Para anunciar este prefijo debemos repetir el proceso descrito en el paso 2 en ambos routers BGP del AS100.

Compruebe que el prefijo se ha propagado correctamente hacia todos los routers BGP. Por ejemplo, en `QuaggaRouter-5` tendríamos:

```
Debian# show bgp ipv4 unicast
BGP table version is 0, local router ID is 40.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
              i internal, r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  20.0.0.0/16      40.0.0.2                               0 400 200 100 i
*>                  20.0.0.2                 0             0 100 i
*> 60.0.0.0/24      40.0.0.2                 0             0 400 i

Total number of prefixes 2
```

En este router, para llegar al prefijo 20.0.0.0/16 existen dos caminos alternativos.

sds

Compruebe si existe conectividad entre PC-3 y PC-2. En el estado actual de la configuración la salida de traceroute en el PC-3 debería ser similar a la siguiente:

```
 traceroute to 60.0.0.2 (60.0.0.2), 30 hops max, 60 byte packets
 1  20.0.1.1 (20.0.1.1)  0.720 ms  0.579 ms  0.539 ms
 2  20.0.0.1 (20.0.0.1)  1.203 ms  2.187 ms  2.302 ms
 3  50.0.0.2 (50.0.0.2)  1.977 ms  1.852 ms  1.793 ms
 4  60.0.0.2 (60.0.0.2)  2.980 ms  3.273 ms  3.174 ms
root@Debian:~#
```

Paso 5. Las tablas de enrutamiento de internet

En el estado actual de la configuración los routers fronterizos de un AS contienen una versión resumida de la tabla de enrutamiento de Internet, es decir, todos los prefijos agregados anunciados por los demás sistemas autónomos. Esto supone el manejo de una tabla de enrutamiento con un número elevado de entradas (del orden del millón). En el siguiente enlace puede comprobar el número de prefijos y sistemas autónomos actualmente activos en Internet: `http://www.cidr-report.org/as2.0`. Fíjese en la tabla *Aggregation summary* y compruebe el número de redes y de prefijos agregados.

Obviamente, no es deseable propagar esta información hacia los routers internos del AS. Deben tomarse medidas adecuadas, puesto que para el manejo de un número tan elevado de entradas el hardware debe estar adecuadamente dimensionado, lo que supone un coste significativo.

Paso 6. Propagar una ruta por defecto hacia los routers frontera

Para evitar la propagación de la tabla de enrutamiento de Internet hacia los routers internos del AS (en este caso sólo el `QuaggaRouter-1`), lo que haremos será establecer uno de los dos routers fronterizos como salida por defecto dentro del protocolo de enrutamiento de pasarela interior. En `QuaggaRouter-4` ejecute el siguiente comando:

```
router rip
default-information originate
```

Compruebe como en el `QuaggaRouter-1` aparece una ruta por defecto que apunta hacia el `QuaggaRouter-4`. Ahora haga un ping desde el `QuaggaRouter-1` hacia el PC-2 y observe que existe conexión al servidor remoto en internet (no sucede lo mismo para el PC-1 porque no hemos configurado NAT en `QuaggaRouter-1`).

Con este procedimiento se evita la redistribución de toda la tabla de enrutamiento de internet (representada por el prefijo 60.0.0.0/24) hacia el `QuaggaRouter-1`.

Paso 7. Comprobar el funcionamiento de la salida redundante a Internet

.

- Haga un traceroute desde el `QuaggaRouter-1` hacia el PC-2 y compruebe el recorrido de los paquetes.
- A continuación, vamos a desactivar la interfaz `eth1` del `QuaggaRouter-5`.
- Repita ahora el traceroute y compare la salida con la anterior.

La salida de ambos `traceroute` se muestra a continuación:

```
 Debian# traceroute 60.0.0.2
traceroute to 60.0.0.2 (60.0.0.2), 30 hops max, 60 byte packets
 1  20.0.1.1 (20.0.1.1)  0.805 ms  0.619 ms  0.576 ms
 2  20.0.0.1 (20.0.0.1)  0.792 ms  0.753 ms  2.555 ms
 3  50.0.0.2 (50.0.0.2)  2.466 ms  2.401 ms  2.361 ms
 4  60.0.0.2 (60.0.0.2)  2.111 ms  2.208 ms  2.388 ms
```

```
Debian# traceroute 60.0.0.2
traceroute to 60.0.0.2 (60.0.0.2), 30 hops max, 60 byte packets
 1  20.0.1.1 (20.0.1.1)  0.613 ms  0.556 ms  0.605 ms
 2  20.0.1.2 (20.0.1.2)  1.033 ms  0.955 ms  0.836 ms
 3  30.0.0.1 (30.0.0.1)  2.542 ms  2.502 ms  2.459 ms
 4  50.0.0.2 (50.0.0.2)  2.405 ms  2.284 ms  4.396 ms
 5  60.0.0.2 (60.0.0.2)  5.224 ms  5.172 ms  5.132 ms
Debian#
```

Observe que cuando el as 400 no es alcanzable a través del `QuaggaRouter-5` se llega a través de un camino alternativo, que tiene un salto más. ¿Por qué hay un salto más? Lo deseable sería que directamente se reenviaran los paquetes hacia el `QuaggaRouter-3` y no pasaran por el `QuaggaRouter-4`. La solución a esto no la proporciona BGP, sino el protocolo VRRP (Virtual Router Redundancy Protocol). Por ello, en esta parte de la práctica nos conformaremos con esta solución.

Paso 8. Ingeniería de tráfico: Evitar que el AS100 se convierta en un AS de tránsito (políticas de enrutamiento)

El propósito de tener dos conexiones a otros sistemas autónomos en AS 100 era tener salida redundante a Internet por si uno de los ISP falla. Esto lo hemos resuelto, como hemos podido comprobar en el paso anterior.

Sin embargo, existe otro problema que debemos solucionar. Vaya al `QuaggaRouter-5` (mantenga la interfaz eth1 apagada) y ejecute un traceroute hacia 60.0.0.2. Observe como existe un camino a través del AS 100 hacia el AS 400 y por tanto el tráfico llega perfectamente hacia el servidor del AS 400. **El AS 100 se ha convertido en un sistema autónomo de tránsito**, es decir, por nuestro sistema autónomo pasa tráfico cuyo emisor y receptor finales no pertenecen al mismo. Esto implica los costes correspondientes en ancho de banda, que no queremos pagar.

El problema está en que estamos publicando a través de `QuaggaRouter-4` las rutas que ha aprendido `QuaggaRouter-3` desde el exterior. Igualmente, sucede a la inversa. En el AS 300 no debería conocerse que existe un camino a través del AS 100 hacia el AS 400 e igualmente en el AS 200 no debería conocerse que existe un camino hacia el AS 400 a través del AS 100 (si la interfaz eth1 del `QuaggaRouter-5` estuviera activa).

Debemos establecer una política de enrutamiento:

- Tráfico de salida:
  - Debe preferir la ruta por defecto a través de `QuaggaRouter-4`.
  - Si el enlace con el AS300 falla, entonces debemos recurrir a la segunda salida a través de `QuaggaRouter-3`.
- Tráfico de entrada:
  - Si uno de los dos ISP falla deberá devolver el tŕafico a través del otro.

La herramienta que se utiliza para establecer esta política de enrutamiento son los mapas de rutas.

Vamos a centrarnos primero en el tŕafico de salida. Para ello, hay que tener en cuenta que el tráfico de salida será reenviado mediante las rutas que se inyectan desde fuera del sistema autónomo. Los mapas de rutas permiten filtrar determinadas rutas en base a ciertos criterios, como los prefijos o los atributos BGP y aplicarles ciertas acciones (modificando atributos). Un mapa de rutas es una lista de patrones y acciones, que se ejecutan secuencialmente hasta que se casa con el patrón. Entonces se ejecuta la acción y finaliza el proceso. El mapa de rutas tiene un nombre y se le asocia una acción (`permit | deny`). Vamos a crear un mapa de rutas para modificar el atributo de preferencia local de las rutas que vienen a través de `QuaggaRouter-4`. En `QuaggaRouter-4` tecleamos:

```
(congig)# route-map AS300-entrada permit 10
(config-route-map)# set local-preference 200
(exit)#
```

Esto último hace que a todas las rutas filtradas por el mapa de rutas se les fije el atributo de preferencia local (dentro del sistema autónomo) a 200. El valor por defecto de la preferencia local es 100, por lo que las rutas que se aprendan desde `QuaggaRouter-3` tienen menos prioridad que las aprendidas a través del `QuaggaRouter-4`. Fíjese que aquí no se ha aplicado ningún patrón sobre las rutas. Para ello se debería haber añadido una sentencia `match` que permite especificar patrones en base a distintos criterios:

```
 Debian(config-route-map)# match ?
  as-path       Match BGP AS path list
  community     Match BGP community list
  extcommunity  Match BGP/VPN extended community list
  interface     match first hop interface of route
  ip            IP information
  ipv6          IPv6 information
  metric        Match metric of route
  origin        BGP origin code
  peer          Match peer address
  probability   Match portion of routes defined by percentage value
  tag           Match tag of route
Debian(config-route-map)# match
```

En este caso concreto no vamos a aplicar ningún criterio porque queremos priorizar siempre el AS300.

Ahora vamos a aplicar el mapa de rutas a los mensajes BGP entrantes desde `QuaggaRouter-5`.

```
(config)# router bgp 100
(config-router)# neighbor 20.0.0.1 route-map AS300-entrada in
```

Active de nuevo la interfaz `eth1` de `QuaggaRouter-5` y ahora vaya a `QuaggaRouter-3`. Vamos a comprobar el prefijo 60.0.0.0/24.

```
Debian# sh ip bgp 60.0.0.0/24
BGP routing table entry for 60.0.0.0/24
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  30.0.0.1
  300 400
    20.0.1.1 from 20.0.1.1 (20.0.1.1)
      Origin IGP, localpref 200, valid, internal, best
      Last update: Wed Mar  7 18:53:13 2018

  200 400
    30.0.0.1 from 30.0.0.1 (50.0.0.1)
      Origin IGP, localpref 100, valid, external
      Last update: Wed Mar  7 18:23:58 2018
```

Observe como hay dos rutas alternativas pero hay una en la que `localpref` tiene valor 200 y es más prioritaria.

En `QuaggaRouter-3`, ejecute un traceroute a 60.0.0.2 y observe el camino que se ha elegido.

En este momento, vamos a ocuparnos del tráfico de salida. Para ello, debemos filtrar los prefijos que se propagan desde AS100 hacia AS 200 y AS 300. Vamos a volver a establecer un mapa de rutas pero en este caso para las rutas salientes. Si filtramos las rutas en función del atributo AS-PATH, podremos seleccionar aquellas rutas que se hayan generado en el AS 100. La expresión regular `^$` se aplica antes de añadir el AS 100 al PATH, por lo que debemos dejar pasar únicamente los PATH vacíos. Luego se le añadirá el 100 cuando se reenvíe el mensaje de actualización al AS vecino. En el `QuaggaRouter-4` ejecutamos los siguientes comandos:

```
(config)# ip as-path access-list 1 permit ^$

(config)# route-map AS300-salida permit 10
(config-route-map)# match as-path 1
```

La primera línea genera una lista de acceso para el AS-PATH con identificador 1, que permite pasar los anuncios cuyo AS-PATH contenga un 100 y sólo un 100, que es el identificador del AS. Seguidamente, se crea el mapa de rutas con nombre `AS300-salida` permitiendo los anuncios que encajan con el criterio definido en la lista de acceso 1. Finalmente, debemos aplicar el mapa de rutas recién creado al tráfico de salida:

```
(config-router)# router bgp 100
(config-router)# neighbor 20.0.0.1 route-map AS300-salida out
```

> **NOTA IMPORTANTE**: Estas políticas de filtrado no son de aplicación inmediata. Una vez configuradas, debemos forzar su aplicacción enviando mensajes de UPDATE a los vecinos. Para ello, se utiliza el comando:
>
> `Debian# clear ip bgp * soft`
>
> El efecto es que BGP envia mensajes de UPDATE completos hacia todos los vecinos, tanto internos como externos al AS (*), mediante una política de soft-reset, que no interrumpe el servicio (si usamos hard-resets se reinicia el proceso BGP).

Ahora observe el efecto en el `QuaggaRouter-5`. Comprobamos el prefijo 60.0.0.0/24, que sólo es alcanzable por un único camino. aunque existe un camino alternativo:

```
Debian# sh ip bgp 60.0.0.0/24

BGP routing table entry for 60.0.0.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  20.0.0.2
  400
    40.0.0.2 from 40.0.0.2 (60.0.0.1)
      Origin IGP, metric 0, localpref 100, valid, external, best
      Last update: Wed Mar  7 18:52:52 2018
```

Si comprobamos el prefijo 20.0.0.0/26, sí que se consideran los dos caminos alternativos:

```
Debian# sh ip bgp 20.0.0.0/16

BGP routing table entry for 20.0.0.0/16
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  40.0.0.2
  400 200 100
    40.0.0.2 from 40.0.0.2 (60.0.0.1)
      Origin IGP, localpref 100, valid, external
      Last update: Wed Mar  7 18:52:52 2018

  100
    20.0.0.2 from 20.0.0.2 (20.0.1.1)
      Origin IGP, metric 0, localpref 100, valid, external, best
      Last update: Wed Mar  7 18:15:37 2018
```

Apliquemos el mismo principio al `QuaggaRouter-3`:

```
(config)# ip as-path access-list 1 permit ^$
```

```
(config)# route-map AS400-salida permit 10
(config-route-map)# match as-path 1
(config-route-map)# exit
(config-router)# router bgp 100
(config-router)# neighbor 30.0.0.1 route-map AS400-salida out
```

Revise el efecto en el `QuaggaRouter-3`.

Finalmente, vamos a comprobar qué sucede cuando paramos nuevamente la interfaz `eth1` del `QuaggaRouter-5`. Haga ping desde `QuaggaRouter-5` hacia 60.0.0.2. En esta ocasión ya no hay conectividad.

Compruebe la conectividad desde `QuaggaRouter-1` hacia 60.0.0.2. Verá que en este caso sí funciona.

> Es posible que en un primer momento esto no funcione. Si es así, obsever la tabla de enrutamiento de `QuaggaRouter-7` y verá que el prefijo 20.0.0.0/16 sigue apuntando hacia 40.0.0.1. Esto se debe a que no se ha producido aún la actualización de las tablas de enrutamiento, porque no ha pasado el *hold time* de 180 segundos. BGP espera este tiempo para considerar que un *peer* está efectivamente caído y considerar definitivamente rota la relación de vecindad.

Ya no somos un sistema autónomo de tránsito

Comentario final

En esta práctica hemos visto el manejo básico de BGP. Queda pendiente la optimización de las tablas de enrutamiento de los routers fronterizos de nuestro AS 100. En este momento manejan toda la tabla de enrutamiento que nos pueden suministrar los dos AS vecinos. Podría plantearse sustituirla por rutas por defecto propagadas desde estos vecinos con el fin de reducir los recursos de hardware necesarios. Por otra parte, aunque tengamos salida dual a Internet, el esquema sigue sufriendo del problema comentado en el paso 6, porque los routers internos no pueden redirigir el tráfico hacia la puerta de enlace más conveniente. Actualmente, sólo hay una ruta por defecto. Esto debe resolverse mediante otros mecanismos distintos de BGP, como ya se ha comentado. Con la configuración actual sólo queda resuelto el fallo de un enlace, pero no quedaría resuelta una avería del propio `QuaggaRouter-4`.

Referencias

[RFC 1930] J. Hawkinson y T. Bates. *Guidelines for creation, selection, and registration of an Autonomous System (AS)*. RFC 1930 (Best Current Practice). Updated by RFCs 6996, 7300. Internet Engineering Task Force, mar. de 1996. URL: `http://www.ietf.org/rfc/rfc1930.txt`.