5. Fundamentos de BGP

BGP es el protocolo de encaminamiento EGP más utilizado en Internet. La versión 1 de este protocolo (RFC 1105) apareció en 1989 para sustituir a EGP. Posteriormente, salieron nuevas versiones como la versión 2 en 1990 (RFC 1163) y la versión 3 en 1991 (RFC 1267). Finalmente apareció la versión 4 (RFC 1771 y RFC 4271) que proporciona soporte para CIDR (*Classless Interdomain Routing*).

BGP es un protocolo que funciona sobre TCP por el puerto 179. BGP permite el encaminamiento de los paquetes IP que se intercambian entre los distintos AS. Para ello, es necesario el intercambio de prefijos de rutas entre los diferentes AS de forma dinámica, lo cual se lleva a cabo mediante el establecimiento de sesiones BGP inter-AS sobre conexiones TCP. Este tipo de operación proporciona comunicación fiable y esconde todos los detalles de la red por la que se pasa.

Debido a que en cada AS se utiliza un protocolo IGP con una definición distinta para el coste de los enlaces, es imposible encontrar el camino más corto hacia cada destino. Por ello, una vez se han aplicado las restricciones sobre las rutas, BGP utiliza un algoritmo similar al tipo vector de distancia, llamado *path-vector*, para seleccionar aquellas rutas que impliquen el mínimo número de AS a atravesar.

Las tablas de encaminamiento de BGP almacenan rutas para alcanzar redes (indicadas mediante prefijos). Las rutas están formadas por una secuencia de números de sistemas autónomos que se deben seguir para alcanzar el prefijo indicado. El último número de AS de la ruta se corresponde con la organización que tiene registrado el prefijo, es decir, el AS donde se encuentra el destino. El principal motivo para almacenar la ruta completa es la detección y eliminación de bucles (*loops*) para evitar que los paquetes se envíen de forma infinita pasando varias veces por un mismo AS.

5.1. Sesiones BGP

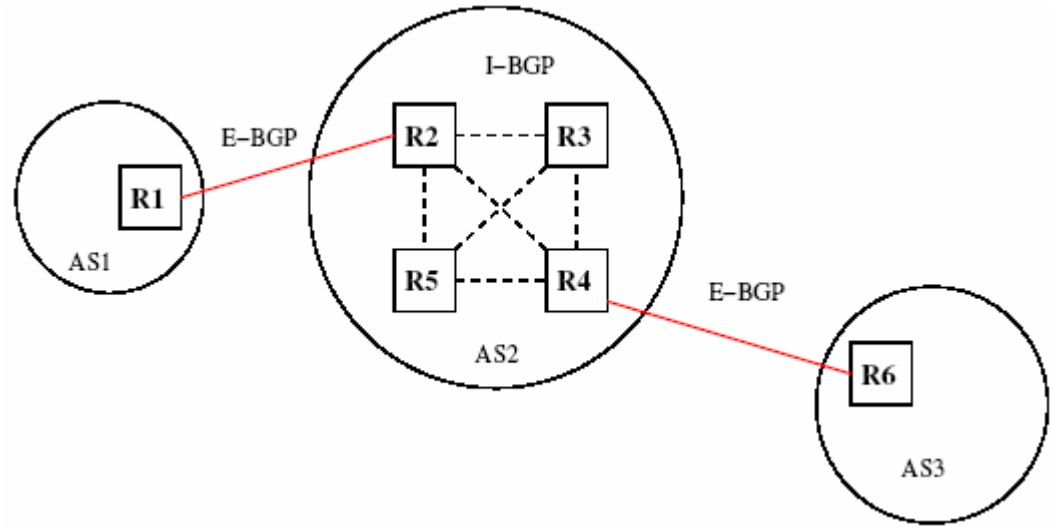
En una sesión BGP participan sólo dos routers (*peers*). En cualquier momento una red puede tener muchas sesiones BGP concurrentes y también una misma pasarela puede participar en muchas sesiones BGP. En la sesión BGP se lleva a cabo el proceso denominado *peering*, que consiste en que un AS informa a otro sobre las redes que puede alcanzar a partir de éste.

Además de las sesiones inter-AS, los routers de borde de un mismo AS deben intercambiar también informaciones BGP para conocer las mismas rutas externas e internas. Para ello se utiliza el protocolo I-BGP, definido en la versión 4 de BGP, que utiliza el mismo tipo de mensajes que E-BGP, el cual es el protocolo utilizado en las sesiones BGP entre dos pasarelas de dos AS distintos. Según la especificación de BGP-4, existe una diferencia a la hora de reanunciar rutas en E-BGP y en I-BGP. En E-BGP, los prefijos que aprende un router de un vecino pueden ser anunciados a otro vecino mediante I-BGP y viceversa, pero un prefijo aprendido de un vecino mediante I-BGP no puede reanunciarse a otro vecino por I-BGP. Esta regla de limitación para reanunciar

prefijos entre routers vecinos mediante I-BGP sirve para evitar bucles (*loops*) dentro de un AS.

Debido a que no se pueden reanunciar prefijos entre routers I-BGP, es necesario que exista conectividad entre todos los routers vecinos que se comuniquen mediante I-BGP dentro de un mismo AS, por lo que se utiliza un mallado total entre éstos (*full-mesh*). Esta malla es realmente virtual en la práctica ya que se implementa de una forma independiente a la conectividad física. Por ello, otra diferencia es que en I-BGP los vecinos no tienen que estar obligatoriamente conectados de forma directa como en el caso de E-BGP.

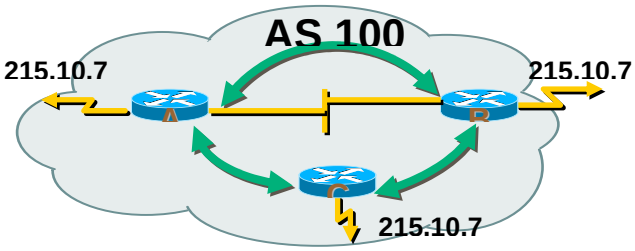
La conectividad entre los routers de borde que intercambian sus informaciones BGP mediante I-BGP en un mismo AS vendrá asegurada por el protocolo IGP utilizado. Si un router de borde no es capaz de alcanzar una ruta de su propio AS, la cual le ha sido anunciada por un vecino interno, esta ruta no será propagada a los vecinos BGP internos o externos.



En el ejemplo anterior se producen las siguientes comunicaciones inter e intra-AS:

- R1 anuncia rutas para prefijos de AS1.
- R2 anuncia rutas para prefijos de AS2.
- R2 aprenderá rutas para prefijos de AS3 vía una sesión I-BGP con R4. R4 aprendió estas rutas de R6 vía una sesión E-BGP.
- R4 anuncia rutas a R6 para prefijos de AS2 y AS1.

En el caso de E-BGP, la forma de prevenir bucles es mediante el atributo *AS-PATH*. Este atributo se incluye en cada ruta anunciada y sólo se modifica en los anuncios E-BGP. En el ejemplo anterior, si AS2 aprende una ruta de AS1, mientras esta ruta se transmite por la malla I-BGP de AS2, tendrá un atributo AS-PATH con valor AS1. A continuación, si AS2 anuncia esta ruta a AS3, la ruta que R6 aprende tendrá un atributo AS-PATH con valor AS2 AS1.



Como ejemplo, para la figura anterior, la configuración de la interfaz virtual que utiliza la pasarela A para las sesiones I-BGP podría ser la siguiente:

```
interface loopback 0
ip address 215.10.7.1 255.255.255.255
router bgpd
      network 220.220.1.0
      neighbor 215.10.7.2 remote-as 100
      neighbor 215.10.7.2 update-source loopback0
      neighbor 215.10.7.3 remote-as 100
      neighbor 215.10.7.3 update-source loopback0
```

5.2. Funcionamiento del proceso BGP

Cuando un router anuncia un prefijo a uno de sus vecinos BGP, esa información es considerada válida hasta que el primer router explícitamente anuncia que la información ya no es válida o hasta que la sesión BGP se pierde. Esto significa que BGP no requiere que la información de routing se refresque periódicamente. De este modo, en un principio existirá un alto flujo de mensajes cuando se establece la sesión BGP, pero transcurrido un tiempo de estabilización los routers sólo necesitarán informar de los cambios que han ocurrido. Por ejemplo, en un AS tipo *backbone* el intercambio es del orden de 50.000 prefijos inicialmente.

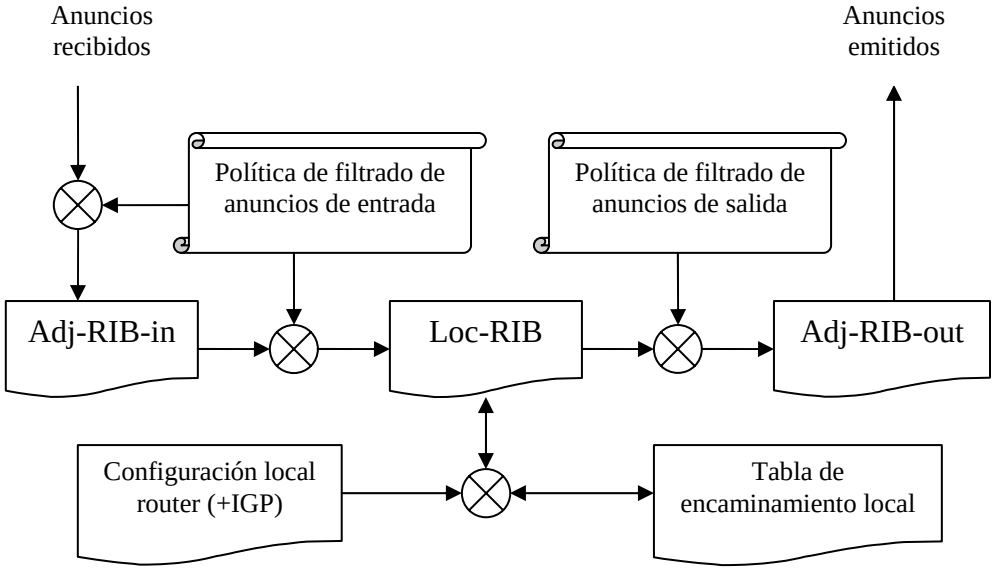
Para almacenar información de encaminamiento, el protocolo BGP necesita un conjunto de tablas de datos denominadas RIBs (*Routing Information Bases*). Éstas son las siguientes:

- **Adj-RIB-in:** En esta tabla se almacenan prefijos aprendidos de un vecino particular. Hay tantas tablas de este tipo como pares BGP.

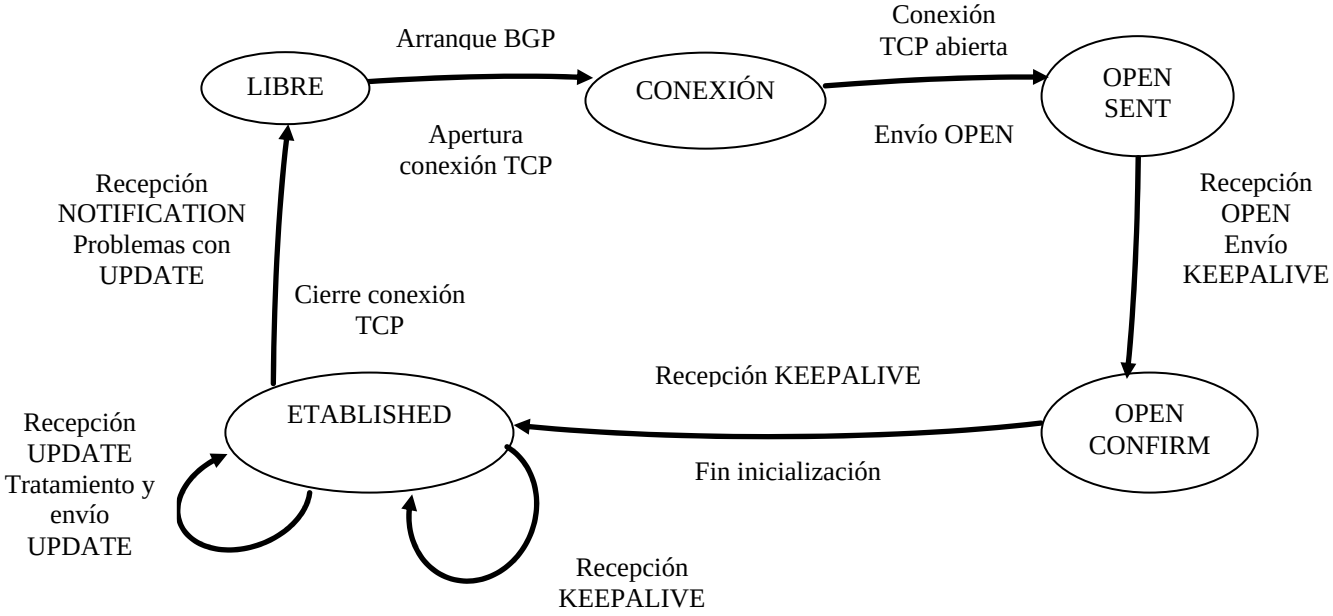
- **Loc-RIB:** Almacena las mejores rutas seleccionadas (prefijos + longitud máscara) que conoce el proceso BGP bien porque las ha obtenido de la tabla de encaminamiento (comandos `network`, `agrégate-address` y `redistribute`), o bien porque se han aprendido por BGP (I-BGP o E-BGP), tras pasar los filtros de entrada. Estas rutas pueden ser anunciadas si la política de encaminamiento a la salida lo permite. Hay sólo uno por cada sistema autónomo.

- **Adj-RIB-out:** Almacena prefijos para ser anunciados a otros vecinos. Esta tabla se construye a partir de las informaciones de la tabla *Loc-RIB* que han sido filtrados y cuyos atributos han sido modificados según configuración. Se tiene una tabla de este tipo por cada par BGP.

A continuación se muestra el esquema funcional del proceso BGP:



El proceso BGP consiste en un autómata de 6 estados con 13 eventos posibles. La interacción con otros procesos BGP se lleva a cabo intercambiando mensajes. Los mensajes intercambiados en una sesión BGP sirven para informar sobre el conocimiento de nuevas rutas activas, para suprimir rutas que ya no están activas, para indicar la viabilidad actual de la conexión o para informar sobre la existencia de condiciones inusuales en la conexión TCP. El siguiente esquema muestra los estados y los mensajes del proceso BGP:



Los estados posibles son los siguientes:

- Estado libre.
- En conexión: Uno de los extremos intenta una conexión TCP.

- Activo: Cuando uno de los extremos no puede establecer conexión y lo reintenta periódicamente.

- *OpenSent*: Un extremo envía un mensaje de identificación.

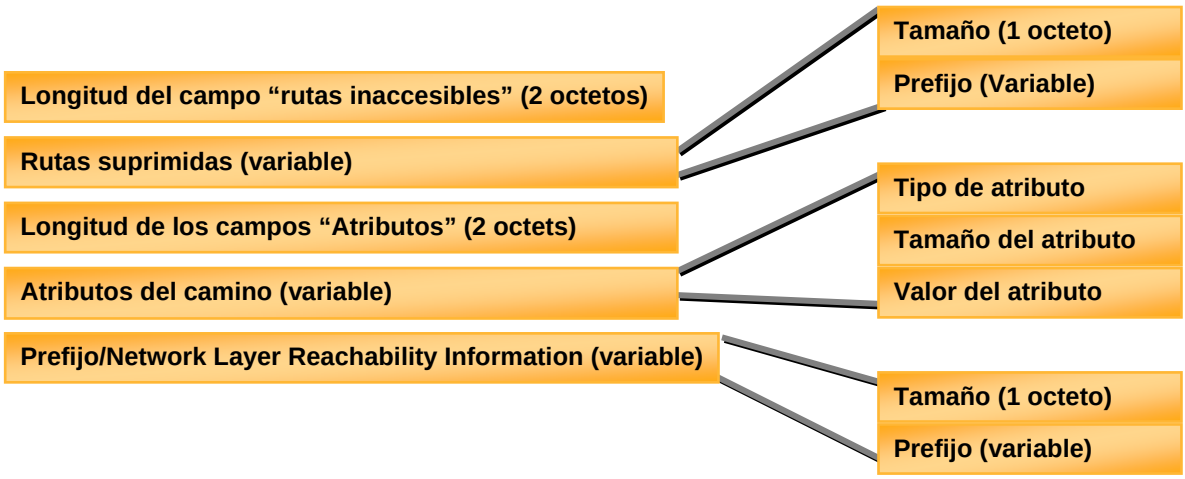
- *OpenConfirm*: Se recibe respuesta al mensaje de identificación.

- *Established*: Se aceptan las identificaciones. De aquí en adelante, la sesión se considera completamente activa.

5.3. Mensajes BGP

El tamaño de los mensajes puede variar entre 19 y 4096 octetos y éstos pueden enviarse de forma segura mediante la función de *hash* MD5. La cabecera es común a todos los mensajes y está formada por un marcador (16 octetos) que contiene información de sincronización y de seguridad, un campo longitud (2 octetos) que indica la longitud total del mensaje y un campo tipo (1 octeto) que indica el tipo del mensaje.

La siguiente figura muestra el formato general de los mensajes BGP:



Existen 4 tipos de mensajes:

- OPEN: Este mensaje es el primero que se envía tras el establecimiento de la conexión TCP. Su función es la de informar a los vecinos sobre la versión del protocolo BGP utilizado, el número de AS y el número identificador del proceso BGP. Además, este mensaje incluye un valor de tiempo durante el cual se va a mantener la sesión (90 segundos normalmente). Si se indica el valor 0 significa que la sesión no va a tener límite de duración. Una vez que se envía este mensaje, el proceso BGP se queda en espera de recibir un mensaje KEEPALIVE.

- KEEPALIVE: Este mensaje sirve como confirmación a un mensaje OPEN. Si el tiempo que se estableció para la duración de la sesión es limitado, es necesario que los procesos BGP envíen este mensaje cada cierto tiempo (30 segundos normalmente) para indicar que se mantiene la sesión. De este modo, en el caso de que no haya modificación de la tabla de encaminamiento, los routers BGP sólo intercambian este tipo de mensaje de forma periódica, lo cual genera un tráfico de unos 5bits/s en el nivel BGP (cada mensaje tiene un tamaño mínimo de unos 19 octetos).

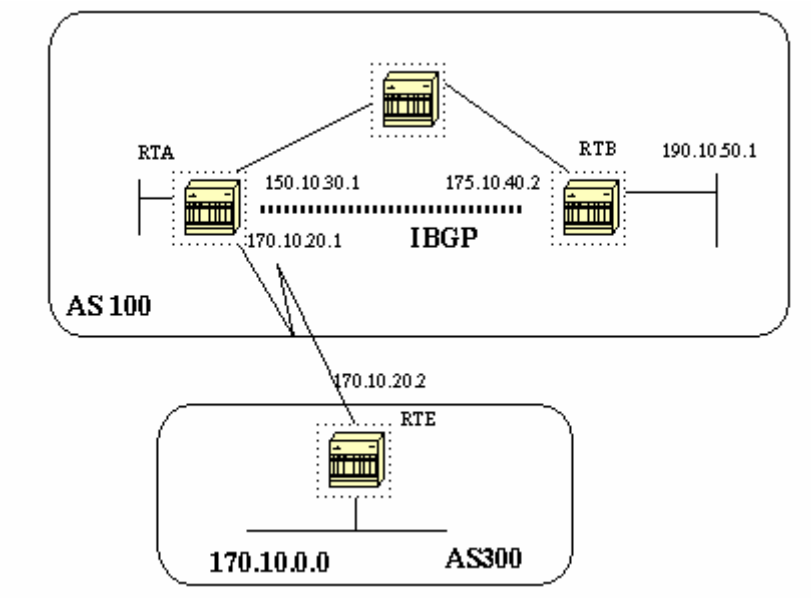
- NOTIFICATION: Este mensaje sirve para cerrar la sesión BGP, cerrando también la conexión TCP. Además, se envía un código para indicar si hubo errores, como por ejemplo la recepción de un mensaje incorrecto, un problema del proceso BGP o la ausencia de mensajes KEEPALIVE durante 90 segundos (*hello time*). La consecuencia del cierre de la sesión BGP es la anulación de todas las rutas aprendidas en dicha sesión.

- UPDATE: Este mensaje sirve para intercambiar las informaciones de encaminamiento como las rutas a eliminar, el conjunto de atributos de cada ruta, las informaciones sobre los prefijos de redes accesibles (red y longitud de la máscara) o NLRI (*Network Layer Reachability Information*) y la longitud de cada ruta. Este mensaje se envía sólo cuando existe algún cambio y su recepción produce la activación del proceso BGP, que se encargará entonces de modificar convenientemente las tablas RIB y de emitir a su vez un mensaje UPDATE hacia los otros vecinos.

5.4. Atributos BGP

Dentro del mensaje UPDATE se distinguen una serie de atributos que indican una serie de informaciones adicionales asociadas al prefijo de la ruta. Estos atributos se codifican en forma de tripleta con los campos TIPO, LONGITUD y VALOR y son utilizados principalmente para elegir la mejor ruta hacia un destino y también para aplicar reglas de filtrado a los mensajes BGP recibidos y anunciados (política de encaminamiento). Los atributos obligatorios son los siguientes:

- ORIGIN: Indica la forma por la que se ha aprendido la ruta: *i* si la ruta ha sido aprendida por un protocolo IGP (ruta interior al AS del router origen que se ha configurado con comando `network` o `redistribute`), *e* si se ha aprendido por EGP (ruta exterior al AS), o *?* (*INCOMPLETE*) en el caso de que el origen sea desconocido o que se haya aprendido de una forma distinta (normalmente por redistribución en BGP de una ruta estática). La función de este atributo es también la selección de rutas, dando prioridad según los valores en el siguiente orden: IGP < EGP <INCOMPLETE.



La configuración para el ejemplo de la figura anterior sería la siguiente:

```
RTA#
router bgp 100
neighbor 190.10.50.1 remote-as 100
neighbor 170.10.20.2 remote-as 300
network 150.10.0.0
redistribute static
ip route 190.10.0.0 255.255.0.0 null0

RTB#
router bgp 100
neighbor 150.10.30.1 remote-as 100
network 190.10.50.0
RTE#
router bgp 300
neighbor 170.10.20.1 remote-as 100
network 170.10.0.0
```

En este ejemplo se tienen una serie de rutas en las tablas de encaminamiento de los distintos routers con su valor correspondiente del atributo ORIGIN:

- RTA alcanzará la red `170.10.0.0` vía 300i (lo que indica que el próximo AS para llegar al destino es el 300 y que el origen de la ruta es IGP, lo cual quiere decir que el router BGP del AS 300 que envió la ruta a RTA la aprendió de otro router interno por IGP).

- RTA también alcanzará la ruta `190.10.50.0` vía i (lo cual quiere decir que dicha entrada pertenece al mismo AS que RTA y que ha sido aprendida por IGP).

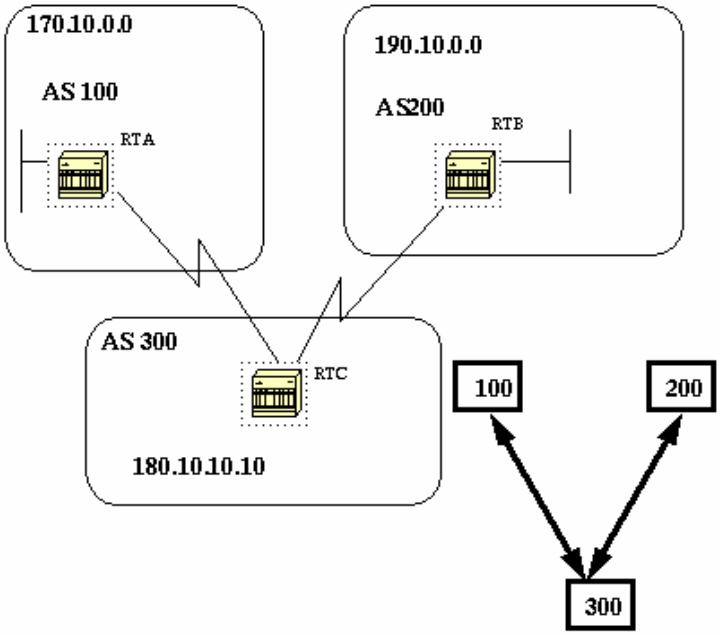
- RTE alcanzará el destino `150.10.0.0` vía 100i (la próxima AS es 100 y su origen es IGP).

- RTE alcanzará también la red `190.10.0.0` vía 100? (la próxima AS para llegar al destino es 100 y el origen es incompleto, es decir, proveniente de una ruta estática añadida en el router origen y redistribuida por BGP).

- AS-PATH (no modificado en anuncios I-BGP): Cada AS añade su número ASN en este atributo para cada una de las rutas que aprende antes de reenviarlas. Así, este atributo contiene una lista con los números de los AS que el anuncio de ruta ha atravesado para llegar al destino. Los números de ASN de este atributo pueden estar ordenados o no (según se indique en AS_SET). Los números no ordenados resultan de la agregación realizada por un nodo el cual añade todos los ASN asociados a las rutas que han sido agregadas.

  Otra función de este atributo, además de indicar el camino de los AS a seguir para llegar al destino (algoritmo *Path Vector*), es también servir para la detección de bucles (un AS ignora un anuncio de ruta si éste ya contiene su propio ASN) y para el filtrado de rutas según las políticas de encaminamiento.

  A continuación se muestra un ejemplo sobre el atributo AS_PATH:

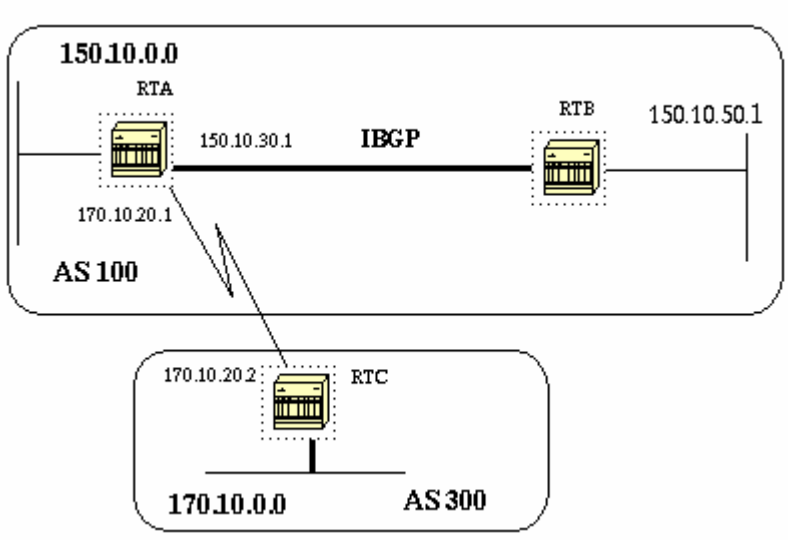


  En este ejemplo, el router RTB (AS 200) anuncia la ruta `190.10.0.0` y, cuando esta ruta atraviesa el AS 300, RTC añade su ASN al atributo AS_PATH de dicha ruta antes de reenviarla. De esta manera, cuando la ruta `190.10.0.0` llega al router RTA, el atributo AS_PATH contendrá dos ASN: primero el ASN 200 y después el ASN 300. Así, desde el punto de vista del router RTA, el camino para alcanzar la ruta `190.10.0.0` es (300, 200).

  Del mimo modo, los paquetes enviados por el router RTB tendrán que tomar el camino (300, 100) para llegar a destino `170.10.0.0`, y los paquetes de RTC se enviarán por el camino (200) para alcanzar la red `190.10.0.0`, así como por el camino (100) para llegar a la red `170.10.0.0`.

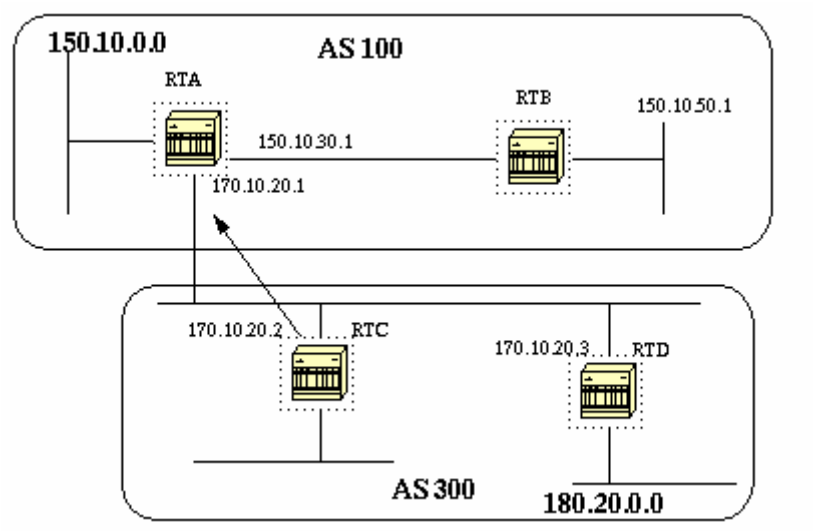
- NEXT-HOP (no modificado en anuncios I-BGP): Cuando un nodo BGP anuncia un prefijo a otro nodo BGP indica en este atributo la dirección del nodo siguiente para llegar al destino. Este atributo es útil en el caso de que el siguiente nodo no utilice BGP. Así, en el caso de que un nodo BGP A anuncie a otro nodo BGP B una ruta cuyo nodo siguiente es C, si B recibe un paquete cuyo destino es la ruta aprendida, B lo envía directamente a C. El valor de este atributo en una sesión E-BGP es normalmente el de una dirección local conocida gracias a un protocolo IGP. Este atributo permite concebir la topología BGP de forma independiente a la topología física de la red (ya que en un AS multiconectado el nodo siguiente BGP no tiene por qué ser el próximo nodo físicamente).

  A continuación se muestra un ejemplo sobre el atributo NEXT-HOP:



En la figura anterior, el router RTC anunciará la ruta `170.10.0.0` a RTA con la dirección `170.10.20.2` como NEXT_HOP. Por su parte, RTA anunciará a RTC la ruta `150.10.0.0` con un valor de NEXT_HOP igual a `170.10.20.1`. Además, RTA anunciará a su vecino RTB mediante I-BGP la ruta `170.10.0.0` aprendida a través de RTC, manteniendo como NEXT_HOP la dirección `170.10.20.2` y no `150.10.30.1` (el atributo NEXT_HOP no se modifica por I_BGP). En este caso, RTB deberá saber cómo llegar a 170.10.20.2, lo cual puede aprenderlo por redistribución de rutas BGP en IGP.

En el caso de que se tenga una red multiacceso como Ethernet, el comportamiento del atributo NEXT_HOP es algo diferente. La siguiente figura muestra un ejemplo relacionado con este caso:

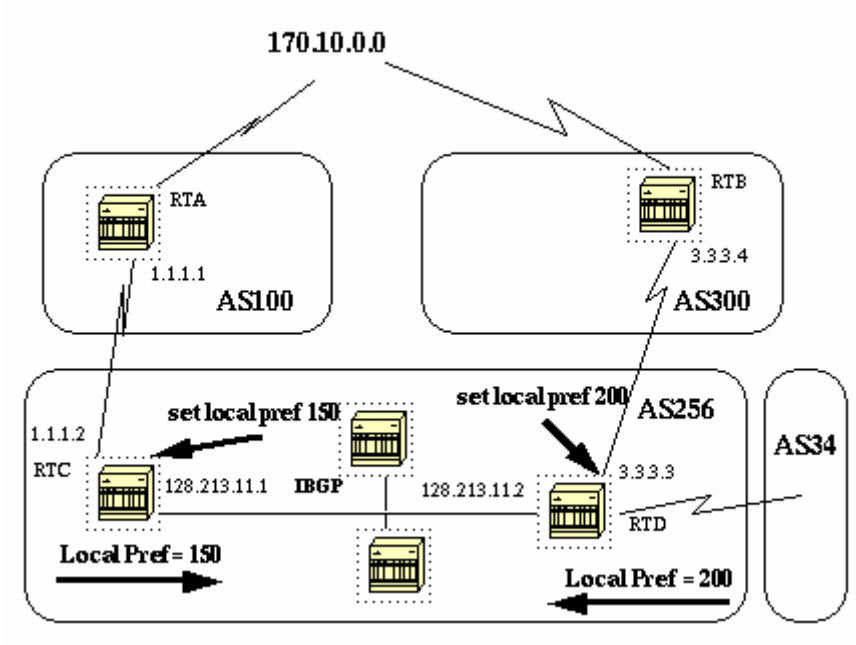


En este ejemplo RTC y RTD pertenecen ambos al mismo AS y se intercambian información de encaminamiento mediante un protocolo IGP. Además, el router RTC tiene establecida una sesión BGP con RTA. Debido a que RTA, RTC y RTD se encuentran en la misma red multiacceso, cuando RTC anuncia la red `180.20.0.0` a RTA, la dirección del NEXT_HOP indicada no es la propia dirección de RTC (`170.10.20.2`), sino la dirección a través de la que RTC alcanza dicho destino (`170.10.20.3`).

Por otro lado, existen una serie de atributos que están reconocidos por la especificación pero que no son obligatorios:

- LOCAL_PREF (sólo anunciado en I-BGP): Este atributo es un parámetro local a un AS y sirve para ponderar la prioridad de las rutas que se anuncian internamente en el AS mediante I-BGP (no se utiliza en anuncios E-BGP). De esta manera, se puede configurar la preferencia de las rutas anunciadas hacia el interior desde el exterior que provengan de una pasarela de borde sobre las que provienen de otra. Este atributo se tiene en cuenta antes que el atributo AS_PATH a la hora de seleccionar la mejor ruta hacia un destino (se elige la ruta con mayor preferencia).

En el siguiente ejemplo se puede ver cómo un router BGP del AS 256 aprende rutas externas a través de los routers RTC y RTD mediante I-BGP. En el caso de que se reciba la misma ruta por RTC y por RTD, por ejemplo la ruta `170.10.0.0`, el primer criterio para la selección de la mejor ruta será el atributo LOCAL_PREF. De este modo, para las rutas con el mismo destino se elegirá la ruta con un mayor valor de este atributo.



En este ejemplo se configura RTC para que asocie el valor LOCAL_PREF=150 a las rutas que le llegan del exterior del AS y que anuncia con I-BGP y, por otro lado, se configura RTD para que les asocie el valor LOCAL_PREF=200. La ruta que se elegirá en el caso de que se reciban varias con el mismo destino será la anunciada por RTD.

Este atributo se intercambia entre los routers que pertenecen a un mismo AS y su valor no se modifica por el protocolo I-BGP, sino que se modificará justo después de que la ruta haya sido aprendida por E-BGP y justo antes de que sea anunciada en el interior. El valor por defecto de este atributo en los routers Cisco es de 100.

El atributo LOCAL_PREF se puede configurar mediante un *route map* o mediante el comando `bgp default local-preference <value>`, como se muestra en la siguiente configuración correspondiente al ejemplo anterior:

```
RTC#
router bgp 256
neighbor 1.1.1.1 remote-as 100
neighbor 128.213.11.2 remote-as 256
bgp default local-preference 150

RTD#
router bgp 256
neighbor 3.3.3.4 remote-as 300
neighbor 128.213.11.1 remote-as 256
bgp default local-preference 200
```

En la configuración anterior, RTC asignará el valor 150 al atributo LOCAL_PREF de todas las rutas que anuncie por I-BGP. A su vez, RTD establece un LOCAL_PREF de 200 para todas las redes que anuncia hacia el interior. De este modo, la ruta `170.10.0.0` recibida en el AS 256 tendrá un valor mayor de este atributo cuando proviene del AS 300 que cuando lo hace del AS 100, y todo el tráfico hacia dicha red tendrá el router RTD como punto de salida del AS 256.

Por otra parte, los *route maps* (ver apartados 7.2 y 11.5) permiten una mayor flexibilidad a la hora de modificar el valor del atributo LOCAL_PREF. En la configuración anterior, todas las rutas recibidas por el router RTD eran modificadas con un valor de LOCAL_PREF igual a 200, lo cual no es necesario cuando las rutas proceden del AS 34. Por ello, utilizando un *route map* se puede especificar qué rutas serán modificadas:

```
RTD#
router bgp 256
neighbor 3.3.3.4 remote-as 300
neighbor 3.3.3.4 setlocalin in
neighbor 128.213.11.1 remote-as 256
....
ip as-path 7 permit ^300$
...
route-map setlocalin permit 10
match as-path 7
set local-preference 400
route-map setlocalin permit 20
set local-preference 150
```

Con esta configuración en el router RTD, cualquier ruta que provenga del AS 300 será modificada con un valor de LOCAL_PREF igual a 200, mientras que el resto de rutas recibidas (como las que provienen del AS 34) serán modificadas con un LOCAL_PREF igual a 150.

- ATOMIC_AGGREGATE: Este atributo indica que la ruta correspondiente se ha obtenido mediante agregación de otras rutas más precisas.

Además de los atributos anteriores, existen otros atributos opcionales que no están recogidos por la norma:

- METRIC o MED (*Multi-Exit-Discriminator*): Este atributo se utiliza en el caso en que se tengan dos AS multiconectados (con varios routers pasarela conectados entre sí). Su función es servir para seleccionar una ruta cuando se reciben por E-BGP varias rutas iguales anunciadas desde el mismo AS por varios enlaces. Así, al configurar los routers pasarela en el AS que envía los anuncios, se da privilegios a un enlace respecto a otro para anunciar una ruta si se configura un valor del atributo MED más bajo para esa ruta.

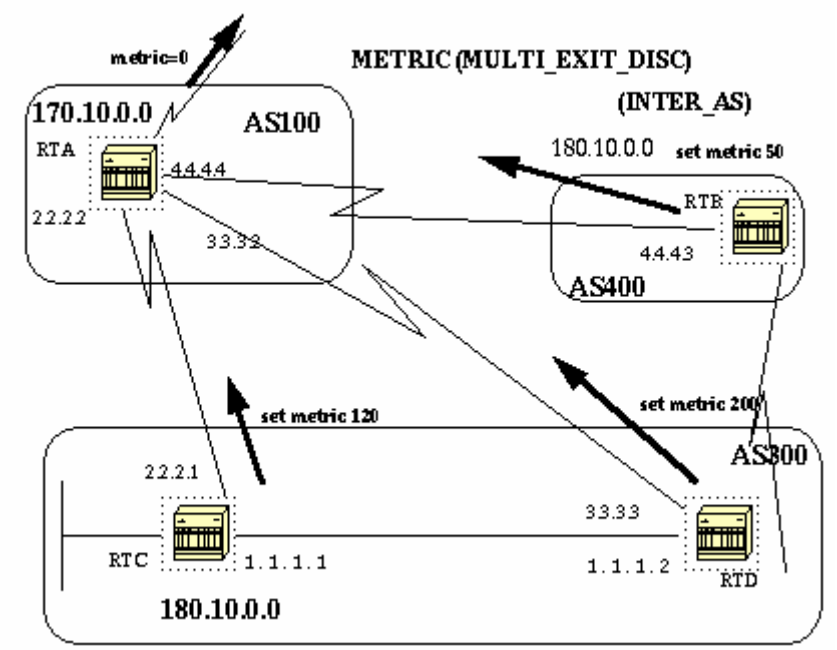
  El atributo MED es intercambiado mediante E-BGP entre dos routers externos pertenecientes a dos AS distintas. El valor de este atributo no se traspasa de un AS a otro, de forma que cuando una ruta recibida por un AS se envía a otro AS tercero se modifica el valor del MED a 0 (por defecto).

  El valor que se suele asignar en los routers pasarela para el MED de las rutas cuando van a ser anunciadas al exterior es el de la métrica IGP que tengan esas rutas al llegar a las pasarelas. Este valor es numérico y varía entre 0 y 0xFFFFFFFF.

  A menos que se especifique lo contrario, un router comparará el valor del atributo MED en las rutas recibidas de otro AS en el caso de que dichas rutas

hayan sido anunciadas por vecinos que se encuentran ambos en dicho AS. Por ello, si se quiere comparar el valor de este atributo también para el caso de rutas anunciadas por vecinos pertenecientes a diferentes AS, será necesario configurar el comando `bgp always-compare-med` en el router.

La siguiente figura muestra un ejemplo de uso de este atributo:



En el diagrama anterior, el AS 100 obtiene información de la red `180.10.0.0` a través de tres routers diferentes: RTC, RTD (ambos pertenecientes al AS 300) y RTB (que pertenece al AS 400). Se va a suponer que se ha configurado un valor del atributo MED diferente para los anuncios que provienen de cada router (120 para los anuncios de RTC, 200 para los de RTD y 50 para los de RTB).

Dado que por defecto un router sólo compara el valor del atributo MED para las rutas anunciadas por routers que pertenecen al mismo AS, RTA sólo comparará los valores del MED de las rutas de RTC con las de RTD, tomando RTC como siguiente salto para los destinos coincidentes (ya que 120 es menor MED que 200).

Debido a que RTB se encuentra en un AS diferente que los otros dos routers, RTA no podrá comparar el valor 50 del MED de las rutas anunciadas por RTB, por lo que tendrán que utilizarse otros atributos para la elección de la mejor ruta en este caso. Sin embargo, se puede configurar RTA para forzar a que compare los valores de los MED de las rutas recibidas independientemente del AS del que provengan mediante el comando bgp always-compare-med. Las siguientes líneas muestran los comandos necesarios para llevar a cabo la configuración del ejemplo anterior:

```
RTA#
router bgp 100
```

```
neighbor 2.2.2.1 remote-as 300
neighbor 3.3.3.3 remote-as 300
neighbor 4.4.4.3 remote-as 400
bgp always-compare-med
....
RTC#
router bgp 300
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-map setmetricout out
neighbor 1.1.1.2 remote-as 300
route-map setmetricout permit 10
set metric 120

RTD#
router bgp 300
neighbor 3.3.3.2 remote-as 100
neighbor 3.3.3.2 route-map setmetricout out
neighbor 1.1.1.1 remote-as 300
route-map setmetricout permit 10
set metric 200

RTB#
router bgp 400
neighbor 4.4.4.4 remote-as 100
neighbor 4.4.4.4 route-map setmetricout out
route-map setmetricout permit 10
set metric 50
```

Otra manera de configurar el valor del atributo MED es mediante el comando `default-metric` *`number`*, en el caso de que se estén redistribuyendo rutas en BGP. De este modo, el router RTB se podría haber configurado para que redistribuyese la red `180.10.0.0` en BGP hacia el AS 100 con un valor del atributo MED igual a 50, lo cual sería de la siguiente manera:

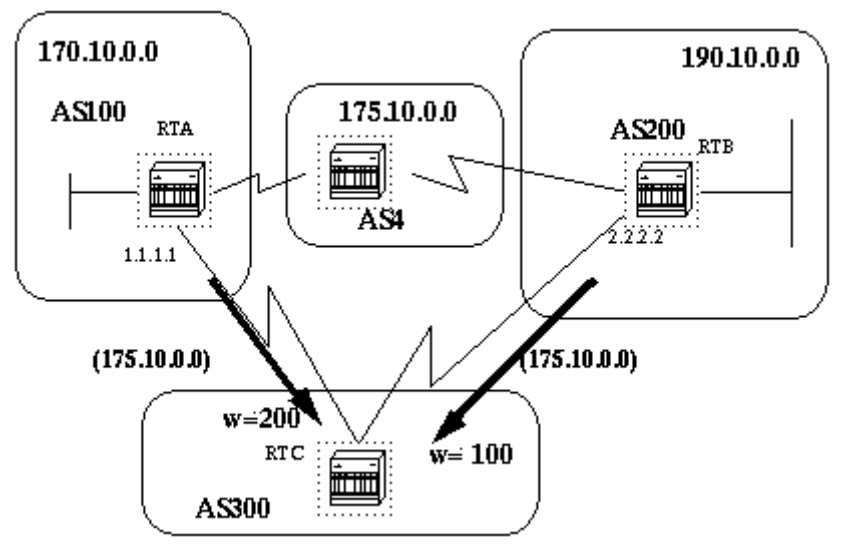
```
RTB#
router bgp 400
redistribute static
default-metric 50

ip route 180.10.0.0 255.255.0.0 null 0
```

- WEIGHT (propietario CISCO): Se utiliza como primer criterio de selección para obtener la mejor ruta cuando se tienen varias rutas hacia el mismo destino. Este atributo es asignado localmente en el router, de modo que no tiene sentido anuncirlo a otros routers. Su valor puede variar entre 0 y 65535 y por defecto vale 32768 para las rutas cuyo origen es el propio router y con valor 0 para el resto de rutas. Se elegirá la ruta con el valor mayor de este atributo.

En el siguiente ejemplo, el router RTA aprende la red `175.10.0.0` del AS 4 para a continuación anunciarla al router RTC. Del mismo modo, el router RTB aprende la red `175.10.0.0` del AS 4 y también la anunciará al router RTC. Así, RTC dispone de dos caminos para alcanzar la red `175.10.0.0`, por lo que tendrá que decidirse por uno de ellos. Suponiendo que en el router RTC se configura que todos los anuncios provenientes de RTA tengan un valor del atributo WEIGHT mayor que en el caso de los anuncios de RTB, se forzará así a RTC a utilizar RTA como NEXT_HOP para llegar a la red `175.10.0.0`.

La configuración del atributo WEIGHT se puede llevar a cabo mediante tres maneras distintas:



- Mediante el comando *neighbor*:

```
neighbor {ip-address|peer-group} weight weight
```

La configuración del atributo WEIGHT en el ejemplo anterior sería de la siguiente manera utilizando el comando *neighbor*:

```
RTC#
router bgp 300
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 weight 200
neighbor 2.2.2.2 remote-as 200
neighbor 2.2.2.2 weight 100
```

- Utilizando una lista de acceso basada en *as-path* (ver apartados 7.1.3 y 11.6):

```
ip as-path access-list access-list-number {permit|deny} as-regular-expression
neighbor ip-address filter-list access-list-number weight weight
```

La configuración del ejemplo anterior quedaría de la siguiente forma utilizando listas de acceso:

```
RTC#
router bgp 300
neighbor 1.1.1.1 remote-as 100
```

```
neighbor 1.1.1.1 filter-list 5 weight 200
neighbor 2.2.2.2 remote-as 200
neighbor 2.2.2.2 filter-list 6 weight 100
...
ip as-path access-list 5 permit ^100$
ip as-path access-list 6 permit ^200$
```

- Usando *route-maps* (ver apartados 7.2 y 11.3). El ejemplo anterior tendría la siguiente configuración usando *route maps*:

```
RTC#
router bgp 300
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 route-map setweightin in
neighbor 2.2.2.2 remote-as 200
neighbor 2.2.2.2 route-map setweightin in
...
ip as-path access-list 5 permit ^100$
...
route-map setweightin permit 10
match as-path 5
set weight 200

route-map setweightin permit 20
set weight 100
```

La lista de acceso definida en el *route map* se aplicará a todos los anuncios de entrada que provengan del AS 100, de manera que se les modificará el valor del atributo WEIGHT a 200. Por defecto, al resto de anuncios que provengan de otro AS se les asignará el valor 100 para el atributo WEIGHT.

- COMMUNITY: Este atributo opcional permite agrupar los destinos en comunidades de destino (grupos de routers con unas mismas propiedades) para ayudar a escalar la aplicación de decisiones de encaminamiento (aceptar una ruta, preferir una ruta ante otra, redistribuir una ruta, etc). Cada destino puede ser miembro de varias comunidades.

  El atributo COMMUNITY consiste en un valor de 23 bits en el cual los 16 bits más significativos son el indicador del AS, mientras que los 16 bits menos significativos son definidos por el administrador del AS. Su valor puede indicar si una ruta no es anunciable a los vecinos del grupo (*no-export*), si no es anunciable a ningún vecino BGP (*no-advertise*) o si no es anunciable vía E-BGP (*no-export-subconfed*).

  Para establecer la comunidad a la que pertenece un destino se puede utilizar la directiva *set community* en un *route-map*. En este último caso se pueden indicar las siguientes opciones:

  - *<1-4294967295>*: Número de comunidad

  - *aa:nn*: Número de comunidad en formato *aa:nn*.

- *additive*: Se añade a una comunidad existente.
- *local-AS*: No enviar a los peers EBGP (*well-known community*).
- *no-advertise*: No enviar a ningún peer (*well-known community*).
- *no-export*: No exportar fuera del AS (*well-known community*)
- *none*: No atributo de comunidad.

A continuación se muestran dos ejemplos de configuración de este atributo mediante *route maps*:

```
route-map communitymap
match ip address 1
set community no-advertise
--
route-map setcommunity
match as-path 1
set community 200 additive
```

La opción *additive* permite añadir el destino a la comunidad 200. Si no se hubiese indicado esta opción, la nueva comunidad sustituiría las comunidades existentes que hayan sido configuradas para ese destino (un destino puede formar parte de varias comunidades).

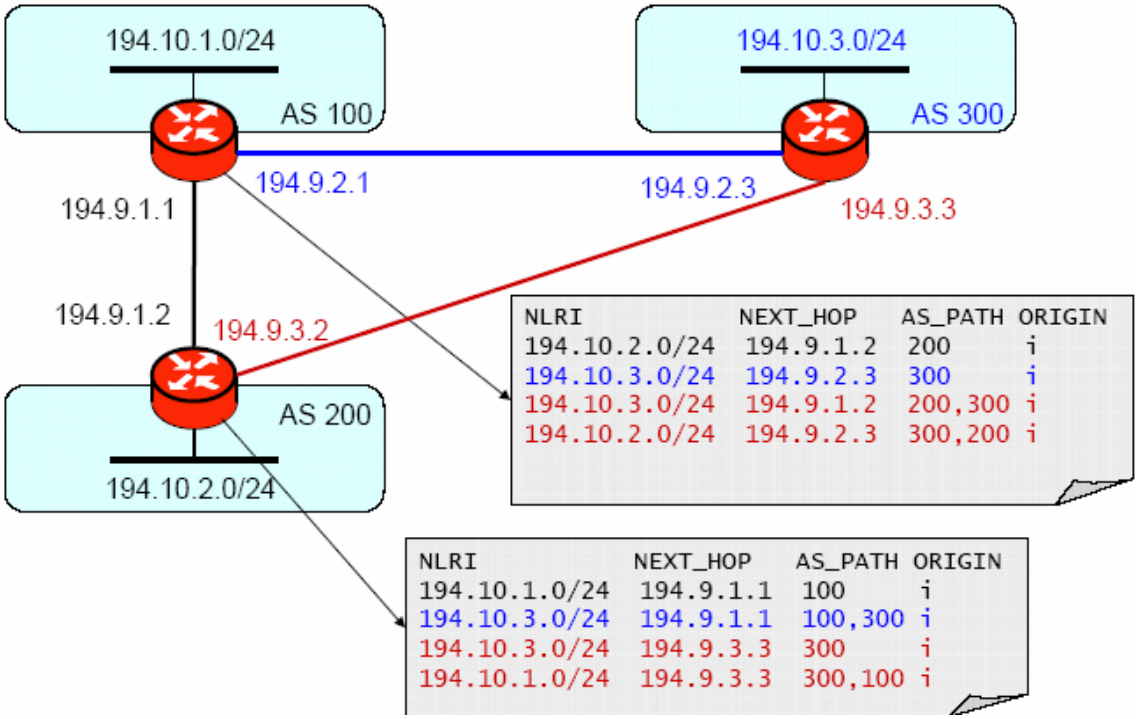
Aunque se modifique el atributo COMMUNITY de una ruta, éste no será anunciado por defecto a los vecinos BGP. Para ello, será necesario utilizar el comando `neighbor {peer-address | peer-group-name} send-community`.

Así, como ejemplo de uso del *route map* anterior se podría tener la siguiente configuración en un router:

```
RT#
router bgp 100
neighbor 3.3.3.3 remote-as 300
neighbor 3.3.3.3 send-community
neighbor 3.3.3.3 route-map setcommunity out
```

- AGGREGATOR: Indica el AS que ha formado la ruta agregada y la dirección IP del router en el que se realizó dicha agregación.
- Otros atributos: ORIGINATOR_ID (*router ID* del vecino I-BGP que refleja rutas del cliente *route reflector* a no clientes), CLUSTER_LIST (cadena de *ORIGINATOR_IDs* a través de los cuales la ruta ha pasado y cuya función servir para el chequeo de relaciones circulares), MP_REACH_NLRI y MP_UNREACH_NLRI.

En la siguiente figura se muestran las tablas Adj-RIB-in de dos routers de borde de dos AS en las que se puede ver el valor de los diferentes atributos BGP:



En las tablas se indican las rutas alcanzables (NLRI) y sus atributos obligatorios: el siguiente nodo para llegar a éstas (NEXT_HOP), el camino de ASNs necesario a seguir para llegar a la ruta (AS_PATH) y el origen por el que se aprendió la ruta (*i* en este caso debido a que la ruta proviene del interior del AS que la anuncia y fue aprendida mediante un protocolo IGP).

Para este ejemplo anterior, la configuración de los routers pasarela que establecen la sesión sería la siguiente:

Configuración de la pasarela del AS100:

```
router bgp 100
neighbor 194.9.1.2 remote-as 200
neighbor 194.9.2.3 remote-as 300
network 194.10.1.0 mask 255.255.255.0
```

Configuración de la pasarela del AS200:

```
router bgp 200
neighbor 194.9.1.1 remote-as 100
neighbor 194.9.3.3 remote-as 300
network 194.10.2.0 mask 255.255.255.0
```

Al haberse configurado un número de AS distinto para cada pasarela, se utilizará el protocolo E-BGP en la sesión BGP entre ambos. El comando `network` hace que se rellene la tabla *Local-RIB* a partir de la tabla de encaminamiento.

En un *route-map* (conjunto de reglas de encaminamiento) se pueden especificar los valores de los atributos mediante la directiva *set* con las siguientes opciones:

- *as-path*: Añade una cadena de AS para el atributo AS-PATH.
- *community*: Atributo de comunidad.
- *local-preference*: Atributo de preferencia local de BGP.
- *metric*: Valor de la métrica para el protocolo de encaminamiento.
- *origin*: Código de origen BGP.
- *weight*: Peso BGP para la tabla de encaminamiento.
- *ip next-hop { A.B.C.D | peer-address }*: Salto siguiente para llegar al destino.

5.5. El proceso de decisión

Cada vez que se recibe un anuncio de ruta, el proceso BGP se encarga de calcular el grado de preferencia de cada ruta aprendida, elegir las mejores rutas para guardarlas en la tabla RIB-Loc, y elegir las rutas que van a ser anunciadas. Para ello, el proceso BGP aplica un tratamiento a las informaciones de encaminamiento basado en una serie de criterios técnicos (supresión de bucles, optimización del camino,…) y administrativos (aplicación de la política de encaminamiento del AS).

Por otro lado, las rutas BGP deben estar sincronizadas, es decir, estar en la tabla de encaminamiento de todos los routers de un mismo AS. De esta manera, BGP no anunciará una ruta mediante E-BGP a un vecino externo antes de que todos los routers del AS la hayan aprendido mediante IGP.

Otro aspecto a verificar antes de validar una ruta es que el nodo indicado en el atributo NEXT-HOP sea alcanzable, lo cual puede comprobarse consultando la tabla de encaminamiento.

Cuando se selecciona una ruta como la mejor para llegar a un destino, ésta se guarda en la tabla de encaminamiento del router y, posteriormente, se propaga a los vecinos BGP. A la hora de elegir entre dos rutas, el proceso BGP tiene en cuenta los siguientes criterios:

1º Si el NEXT_HOP (siguiente salto) es inaccesible no se considera la ruta.
2º Mayor WEIGHT: Se elige la ruta con el valor más grande de este atributo. Este criterio es específico para los routers Cisco y se aplica localmente en cada router.

3º Mayor LOCAL_PREF (anunciado por I-BGP): Se elige la ruta con el valor más grande de este atributo. Esto se aplica a todos los routers del AS.

4º En el caso de que se tenga el mismo valor de LOCAL_PREF, se elige una ruta originada por el propio router (configurada mediante comandos `redistribute`, `aggregate` o `network`) antes que una aprendida a través de un vecino.

5º Más corto AS_PATH: Se elige la ruta con el mínimo número de ASNs en este atributo.

6º Menor ORIGIN: Se elige la ruta según el modo en que se aprendió (IGP < EGP < INCOMPLETE).

7º Menor MED: En el caso de que se tenga el mismo origen para la ruta, en el caso de las rutas que provienen de un mismo AS se elige la ruta con el mínimo valor de este atributo (se puede configurar también para comparar este atributo en rutas de diferentes AS).

8º Se elige una ruta aprendida por E-BGP antes que una aprendida por I-BGP.

9º Menor IGP METRIC al NEXT-HOP: Se elige la ruta con el NEXT-HOP más próximo, es decir, aquélla para la cual es necesario pasar por el vecino más próximo localmente (vecino no BGP, sino IGP). Este vecino más próximo vendrá indicado por la métrica IGP, de manera que sea la salida más próxima del AS.

10º Ruta hacia el router BGP con el *Router-ID* (dirección IP) más pequeño.

Otro parámetro que se define es la distancia administrativa. Esta distancia no se aplica al algoritmo para la selección de las rutas BGP, pero sí cuando se aplican las rutas aprendidas BGP a la tabla de encaminamiento del sistema. Este concepto de distancia administrativa sirve para tener un parámetro común para las rutas independientemente de la forma en que se hayan aprendido. Así, las rutas cuya distancia sea menor serán instaladas en la tabla de encaminamiento del sistema. En BGP, a las rutas locales (que provienen del propio router) se les asigna por defecto una distancia de 200, al igual que para las rutas adquiridas mediante I-BGP, mientras que el valor por defecto para las rutas E-BGP es 20.