Definición del Protocolo BGP

El Border Gateway Protocol (BGP) es un protocolo de enrutamiento utilizado para intercambiar información de rutas entre sistemas autónomos (AS) en redes de gran escala, como Internet. Su flexibilidad y escalabilidad lo hacen fundamental en la conectividad global, permitiendo la selección óptima de trayectorias basadas en políticas definidas por el administrador de la red.

BGP Multihoming

El BGP multihoming es una estrategia utilizada para mejorar la redundancia y la disponibilidad de una red mediante la conexión a múltiples proveedores de servicios de Internet (ISP). Existen diferentes enfoques para implementar el multihoming:

Multihoming con múltiples ISPs: Permite mantener la conectividad en caso de falla de un proveedor.

Multihoming con balanceo de carga: Distribuye el tráfico entre múltiples enlaces según criterios específicos.

Multihoming con preferencia de rutas: Se establecen políticas de preferencia para priorizar ciertos enlaces sobre otros.

El objetivo principal del multihoming es mejorar la resiliencia de la red y optimizar el rendimiento del tráfico de datos.

Filtrado de Rutas en BGP

El filtrado de rutas en BGP es una técnica clave para controlar qué rutas se anuncian y se reciben, asegurando que solo se propaguen rutas válidas y seguras. Se pueden emplear diversos métodos para el filtrado:

1. Listas de Acceso (ACLs): Permiten el filtrado de rutas basado en direcciones IP o prefijos específicos.

2. Listas de Prefijos (Prefix Lists): Proporcionan una manera más eficiente de especificar qué prefijos de red pueden ser anunciados o recibidos.

3. Mapas de Rutas (Route Maps): Son estructuras más avanzadas que permiten aplicar políticas de filtrado, manipulación de atributos y modificación de rutas según criterios específicos.

Cada uno de estos métodos permite una administración granular del tráfico en BGP y es fundamental en redes que requieren un control detallado de sus rutas.

Tipos de Comunidades BGP

Las comunidades BGP son etiquetas opcionales utilizadas para clasificar y controlar el comportamiento de las rutas en una red. Algunos de los tipos más comunes incluyen:

Comunidades Bien Conocidas (Well-Known Communities):

no-export: Impide que una ruta sea anunciada fuera del AS.

no-advertise: Evita que una ruta sea propagada a cualquier vecino BGP.

local-as: Restringe la propagación de la ruta dentro del mismo AS.

Comunidades Personalizadas (Custom Communities):

Se utilizan para definir políticas específicas dentro de una red, como el ajuste de preferencias en función de reglas internas de enrutamiento.

Las comunidades BGP permiten establecer políticas de enrutamiento avanzadas y mejorar la administración de tráfico en redes de gran escala.

Mecanismos de Selección de Trayectorias en BGP Avanzado

BGP utiliza varios criterios para la selección de la mejor ruta, aplicando un proceso de toma de decisiones basado en los siguientes factores:

1. Preferencia de Origen (Weight y Local Preference): Se prefiere la ruta con el peso más alto o mayor Local Preference.

2. Menor Número de Saltos AS-Path: Se elige la ruta con el menor número de sistemas autónomos en el camino.

3. Origen de la Ruta: Las rutas aprendidas internamente (IGP) tienen prioridad sobre las externas (EGP) o las inyectadas estáticamente.

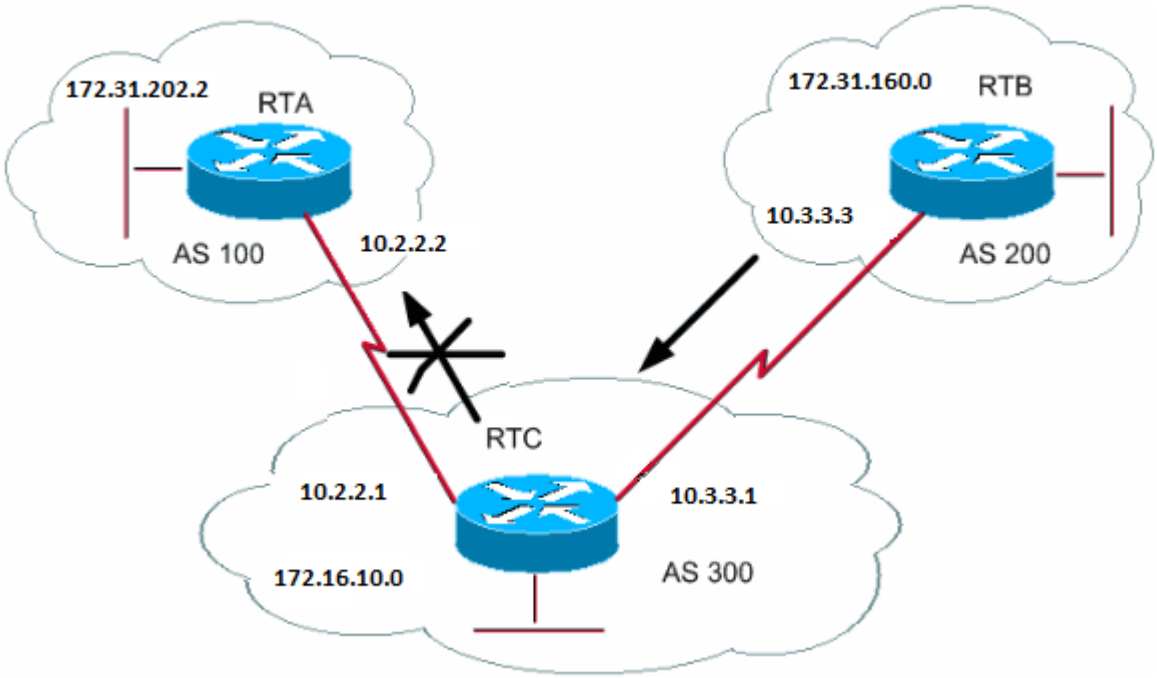
4. Menor MED (Multi-Exit Discriminator): Se prefiere la ruta con el menor valor MED en caso de múltiples puntos de entrada a un AS.

5. Menor ID de Router BGP: Si todas las demás métricas son iguales, se selecciona la ruta del router con la ID más baja.

Caso Práctico de BGP 3
BGP Filter (Filtro de BGP)

Diversos métodos de filtro le permiten controlar el envío y la recepción de las actualizaciones de BGP. Puede filtrar las actualizaciones de BGP con la información de ruta como base, o con la información de trayectoria o las comunidades como base. Todos los métodos alcanzan los mismos resultados. La opción de un método sobre otro método depende de la configuración de red específica.

Route Filter (Filtro de ruta)



Para restringir la información de ruteo que el router detecta o anuncia, puede filtrar BGP con el uso de actualizaciones de ruteo para o de un vecino en particular. Usted define una lista de acceso y aplica la lista de acceso a las actualizaciones para o de un vecino. Ejecute este comando en el modo de configuración del router:

neighbor {ip-address | peer-group-name} distribute-list access-list-number {in | out}

En este ejemplo, el RTB origina la red 172.31.160.0 y envía la actualización al RTC. Si el RTC desea detener la propagación de las actualizaciones al AS100, usted debe definir una lista de acceso para filtrar esas actualizaciones y aplicar la lista de acceso durante la comunicación con el RTA:

```
RTC#
router bgp 300
 network 172.16.10.0
 neighbor 10.3.3.3 remote-as 200
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.2.2.2 distribute-list 1 out

access-list 1 deny 172.31.160.0 0.0.255.255

access-list 1 permit 0.0.0.0 255.255.255.255
```

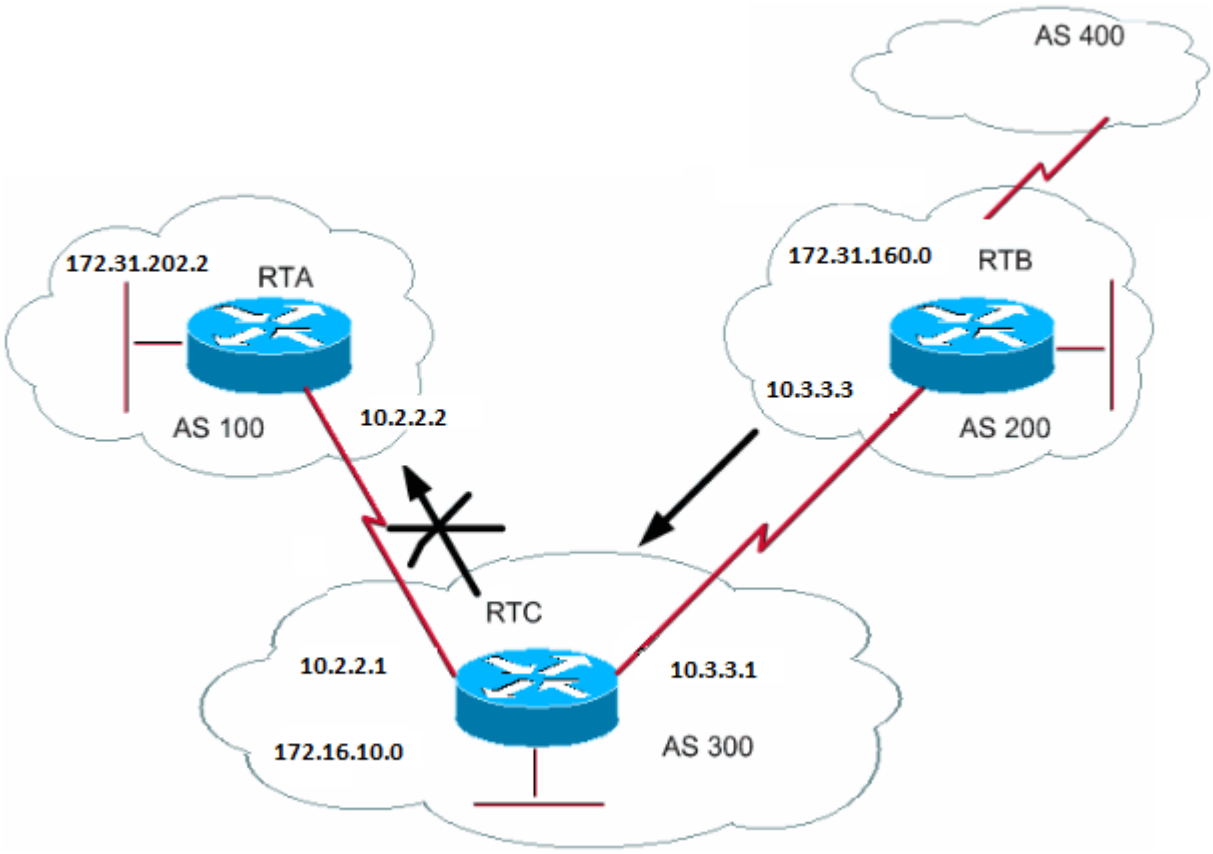
Filter out all routing updates about 160.10.x.x.

El uso de las listas de acceso es un poco difícil cuando usted trata superedes que pueden causar algunos conflictos.
Suponga que, en el ejemplo de esta sección, el RTB tiene diferentes subredes de 160.10.x.x. Su meta es filtrar las actualizaciones y anunciar solamente 192.168.160.0/8.

***Nota**:* La anotación /8 significa que usa 8 bits de máscara de subred, que comienza a la izquierda de la dirección IP. Esta dirección es equivalente a 192.168.160.0 255.0.0.0.
El comando access-list 1 permit 192.168.160.0 0.255.255.25 5permite 192.168.160.0/8, 192.168.160.0/9, etc. Para restringir la actualización a solamente 192.168.160.0/8, debe utilizar una lista de acceso extendida de este formato:
access-list 101 permit ip 192.168.160.0 0.255.255.255 255.0.0.0 0.0.0.0.
Esta lista permite 192.168.160.0/8 solamente.

Filtro de ruta

También puede filtrar rutas.



Usted puede especificar una lista de acceso en las actualizaciones entrantes y salientes con el uso de la información de trayectorias de AS de BGP. En el diagrama de esta sección, puede bloquear las actualizaciones por 172.31.160.0 para que no vayan al AS100. Para bloquear las actualizaciones, defina una lista de acceso en el RTC que prevenga la transmisión al AS100 de cualquier actualización que se haya originado desde el AS200.

Ejecute estos comandos:

ip as-path access-list access-list-number {permit | deny} as-regular-expression

neighbor {ip-address | peer-group-name} filter-list access-list-number {in | out}

Este ejemplo detiene el envío de actualizaciones del RTC por 172.31.160.0 al RTA:

```
RTC#
router bgp 300
 neighbor 10.3.3.3 remote-as 200
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.2.2.2 filter-list 1 out
```

The 1 is the access list number below.

```
ip as-path access-list 1 deny ^200$
ip as-path access-list 1 permit .*
```

El access-list 1comando de este ejemplo fuerza la negación de cualquier actualización con información de trayectoria que comienza con 200 y termina con 200. El dato ^200$ en el comando es una "expresión regular", donde ^ significa "comienza con" y $ significa "termina con". Dado que RTB envía actualizaciones de 172.31.160.0 con información de ruta que comienza con 200 y termina con 200, las actualizaciones coinciden con la lista de acceso. La lista de acceso niega estas actualizaciones.

`.*`es otra expresión regular en la que `.` significa “cualquier caracter” y el `*`significa “la repetición de ese caracter”. Entonces `.*`representa cualquier información de ruta, que es necesaria para permitir la transmisión de todas las otras actualizaciones.

¿Qué sucede si, en lugar de usar `^200$`, utiliza`^200`? Con un AS400, como en el diagrama de esta sección, las actualizaciones que el AS400 origina tienen información de trayectoria de la forma (200, 400). En esta información de trayectoria, 200 es el primer dato y 400 es el último dato. Estas actualizaciones coinciden con la lista de acceso `^200` porque la información de la ruta comienza con 200. La lista de acceso previene la transmisión de estas actualizaciones al RTA, que no es el requisito.

Para verificar si ha implementado la expresión normal correcta, ejecute el comando **show ip bgp regexpregular-expression** . Este comando muestra todas las trayectorias que han coincidido con la configuración de expresión regular.

Expresión Regular de AS

En esta sección, se explica la creación de una expresión regular.

Una expresión regular es un patrón que debe coincidir con una cadena de entrada. Cuando usted crea una expresión regular, especifica una cadena con la que debe coincidir la entrada. En el caso de BGP, usted especifica una cadena que está compuesta de información de trayectoria con la que debe coincidir una entrada.

En el ejemplo de la sección **Filtro de ruta** , especificó la cadena`^200$`. Deseaba que la información de ruta que incorporan las actualizaciones coincidiera con la cadena para tomar una decisión.

Una expresión regular consta de:

- **Rango**
  Un rango es una secuencia de caracteres dentro de los corchetes de apertura y cierre. Un ejemplo es [abcd].
- **Átomo**
  Un átomo es un único carácter. A continuación, se incluyen algunos ejemplos:

  `.`

  - `.`coincide con cualquier único caracter.

  `^`

  - `^` coincide con el comienzo de la cadena de entrada.

  `$`

  - $ coincide con el final de la cadena de entrada.

  `\`

  - `\`coincide con el caracter.

  `_`

  - _coincide con una coma (`,`), llave izquierda (`{`), llave derecha (`}`), el inicio de la cadena de entrada, el final de la cadena de entrada o un espacio.
- **Pieza**
  Una pieza es uno de estos símbolos, que sigue a un átomo:

  `*`

  - * coincide con 0 o más secuencias del átomo.

  `+`

  - + coincide con 1 o más secuencias del átomo.

  `?`

  - `?`coincide con el átomo o con la cadena nula.
- **Sucursal**
  Una ramificación es 0 o más partes concatenadas.

Aquí hay algunos ejemplos de expresiones regulares:

`a*`

- Esta expresión indica cualquier repetición de la letra "a", que incluye ninguna.

```
a+
```

- Esta expresión indica que por lo menos una repetición de la letra "a" debe estar presente.

```
ab?a
```

- Esta expresión coincide con "aa" o "aba".

```
_100_
```

- Esta expresión significa vía el AS100.

```
_100$
```

- Esta expresión indica un origen del AS100.

```
^100 .*
```

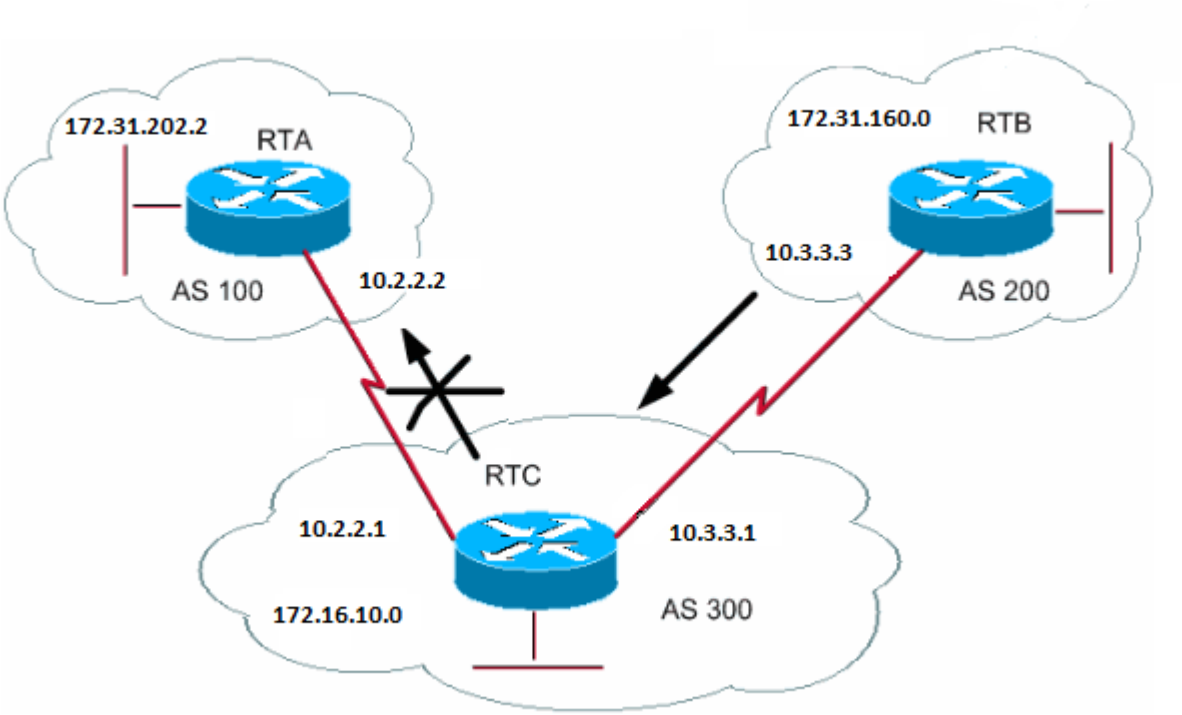
- Esta expresión indica la transmisión del AS100.

```
^$
```

- Esta expresión indica el origen desde este AS.

Filtrado de comunidad BGP

En este documento, se ha cubierto el filtrado de rutas y el filtrado de trayectorias de AS. Otro método es el filtrado de comunidades. En la sección Atributo de la comunidad se analiza la comunidad, y en esta sección se proporcionan algunos ejemplos de cómo usar la comunidad.



En este ejemplo, usted desea que el RTB configure el atributo de comunidad en las rutas BGP que el RTB anuncia como que el RTC no propaga estas rutas a los peers externos. Utilice el atributo de no-exportcomunidad.

```
RTB#
router bgp 200
 network 172.31.160.0
 neighbor 10.3.3.1 remote-as 300
 neighbor 10.3.3.1 send-community
 neighbor 10.3.3.1 route-map setcommunity out

route-map setcommunity
 match ip address 1
 set community no-export

access-list 1 permit 0.0.0.0 255.255.255.255
```

Nota:
Este ejemplo utiliza el route-map setcommunity comando para establecer la comunidad en no-export.

Nota:
El **neighbor send-community** comando es necesario para enviar este atributo al RTC.
Cuando el RTC obtiene las actualizaciones con el atributo NO_EXPORT, el RTC no propaga las actualizaciones al peer externo RTA.
En este ejemplo, el RTB ha establecido el atributo de comunidad en**100 200 additive** . Esta acción agrega el valor 100 200 a cualquier valor de comunidad actual antes de la transmisión a RTC.

```
RTB#
router bgp 200
 network 172.31.160.0
 neighbor 10.3.3.1 remote-as 300
 neighbor 10.3.3.1 send-community
 neighbor 10.3.3.1 route-map setcommunity out

route-map setcommunity
 match ip address 2
 set community 100 200 additive

access-list 2 permit 0.0.0.0 255.255.255.255
```

Una lista de comunidades es un grupo de comunidades que usted utiliza en una cláusula match de un mapa de ruta. La lista de comunidades le permite filtrar o configurar atributos con diferentes listas de números de comunidad como base.
ip community-list <community-list-number> {permit | deny} <community-number>

Por ejemplo, puede definir este mapa de ruta, match-on-community:

```
route-map match-on-community
 match community 10
```

The community list number is 10.

```
 set weight 20
 ip community-list 10 permit 200 300
```

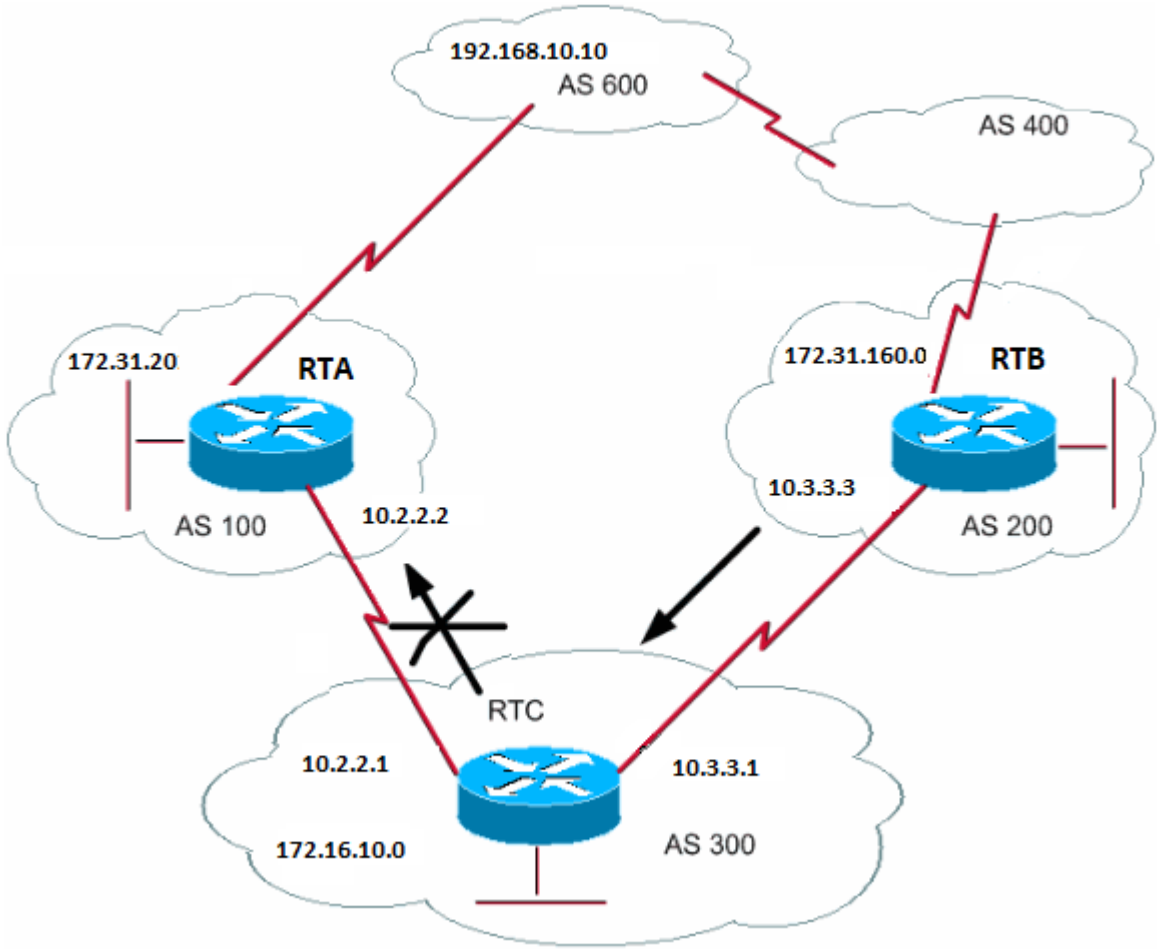
The community number is 200 300.

Puede utilizar la lista de comunidades para filtrar o configurar ciertos parámetros, como peso y métrica, en determinadas actualizaciones con el valor de comunidad como base. En el segundo ejemplo de esta sección, el RTB envió las actualizaciones al RTC con una comunidad de 100 200. Si el RTC desea configurar el peso con esos valores como base, usted puede hacer lo siguiente:

```
RTC#
router bgp 300
 neighbor 10.3.3.3 remote-as 200
 neighbor 10.3.3.3 route-map check-community in

route-map check-community permit 10
 match community 1
 set weight 20

route-map check-community permit 20
 match community 2 exact
 set weight 10

route-map check-community permit 30
 match community 3

ip community-list 1 permit 100
ip community-list 2 permit 200
ip community-list 3 permit internet
```

En este ejemplo, cualquier ruta que tenga 100 en el atributo de comunidad coincide con la lista 1. El peso de esta ruta está configurado en 20. Cualquier ruta que tenga solamente 200 como comunidad coincide con la lista 2 y tiene un peso de 20. La palabra clave exact establece que la comunidad está compuesta de 200 solamente y nada más. La última lista de comunidades está aquí para garantizar que las otras actualizaciones no se descarten. Recuerde que cualquier cosa que no coincida, se descarta de forma predeterminada. La palabra clave internet indica todas las rutas porque todas las rutas son miembros de la comunidad de Internet.

Mapas de Ruta y Vecinos BGP



Usted puede utilizar el comando neighbor junto con mapas de ruta para filtrar o configurar parámetros en las actualizaciones entrantes y salientes.

Los mapas de ruta asociados con la declaración neighbor no tienen ningún efecto en las actualizaciones entrantes cuando usted realiza coincidencias según la dirección IP:
neighbor <ip-address> route-map <route-map-name>

Suponga que, en el diagrama de esta sección, usted desea que el RTC detecte de AS200 redes que sean locales para el AS200 y nada más. También desea configurar el peso en las rutas aceptadas en 20. Utilice una combinación de las listas de acceso neighbor y as-path:

```
RTC#
router bgp 300
 network 172.16.10.0
 neighbor 10.3.3.3 remote-as 200
 neighbor 10.3.3.3 route-map stamp in

route-map stamp
 match as-path 1
 set weight 20

ip as-path access-list 1 permit ^200$
```

Toda actualización que se origina desde el AS200 tiene información de trayectoria que comienza con 200 y termina con 200. Se permiten estas actualizaciones. Se descarta cualquier otra actualización.

Suponga que usted desea:

- Una aceptación de las actualizaciones que se originen desde el AS200 y tengan un peso de 20.
- El descarte de las actualizaciones que se originen desde el AS400.
- Un peso de 10 para las otras actualizaciones.

```
RTC#
router bgp 300
 network 172.16.10.0
 neighbor 10.3.3.3 remote-as 200
 neighbor 10.3.3.3 route-map stamp in

route-map stamp permit 10
 match as-path 1
 set weight 20

route-map stamp permit 20
 match as-path 2
 set weight 10

ip as-path access-list 1 permit ^200$
ip as-path access-list 2 permit ^200 600 .*
```

Esta declaración configura un peso de 20 para las actualizaciones que son locales para el AS200. Esta sentencia también define un peso de 10 para las actualizaciones que están detrás de AS400, y descarta las actualizaciones que proceden de AS400.

El uso del comando set as-path prepend

En algunas situaciones, usted debe manipular la información de trayectoria para manipular el proceso de decisión de BGP. El comando que usted utiliza con un mapa de ruta es:

set as-path prepend **<as-path#> <as-path#>**

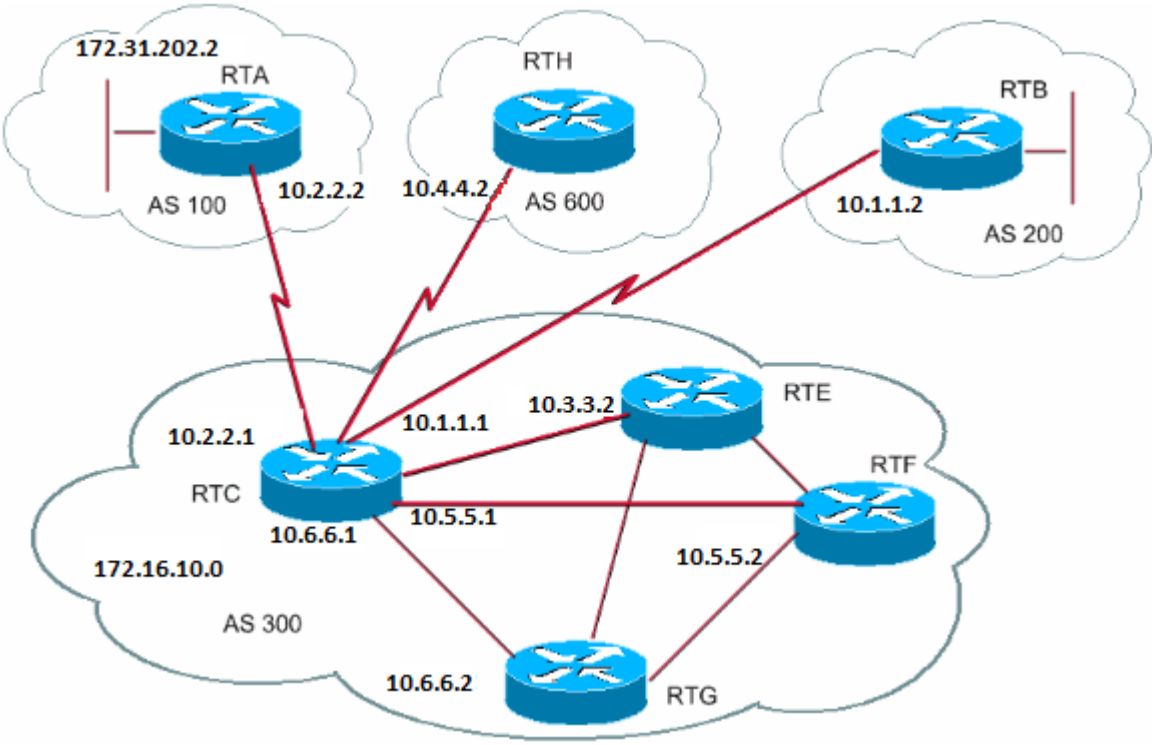
Suponga que, en el diagrama de la sección Vecinos BGP y mapas de rutas, el RTC anuncia su propia red 172.16.10.0 a dos AS diferentes, AS100 y AS200. Cuando la información se propaga al AS600, los routers en el AS600 tienen información sobre la posibilidad de alcance de la red por 172.16.10.0 vía dos rutas diferentes. La primera ruta es vía el AS100 con la trayectoria (100, 300) y segunda es vía el AS400 con la trayectoria (400, 200, 300). Si todos los demás atributos son los mismos, el AS600 selecciona la trayectoria más corta y elige la ruta vía el AS100.

El AS300 obtiene todo el tráfico vía el AS100. Si desea influir sobre esta decisión del extremo de AS300, puede hacer que la trayectoria a través del AS100 parezca más larga que la trayectoria que pasa a través del AS400. Puede hacer esto si antepone números de AS a la información de la ruta actual que se anuncia en AS100. Una práctica común es repetir su propio número de AS de esta manera:

```
RTC#
router bgp 300
 network 172.16.10.0
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.2.2.2 route-map SETPATH out

route-map SETPATH
 set as-path prepend 300 300
```

Debido a esta configuración, AS600 recibe actualizaciones de 172.16.10.0 por AS100 con información de trayecto de: (100, 300, 300, 300). Esta información de trayectoria es más larga que (400, 200, 300) que el AS600 recibió del AS400.

Grupos de Pares BGP



Un grupo de peers BGP es un grupo de vecinos BGP con las mismas políticas de actualización. Los mapas de ruta, las listas de distribución y las listas de filtros típicamente configuran políticas de actualización. No se definen las mismas políticas para cada vecino, sino que se define un nombre de grupo de pares y se asigna estas políticas al grupo de pares.

Los miembros del grupo de peers heredan todas las opciones de configuración del grupo de peers. Usted también puede configurar que los miembros invaliden estas opciones si las opciones no afectan las actualizaciones salientes. Solo puede invalidar opciones que se configuren en las actualizaciones entrantes.

Para definir un grupo de peers, ejecute este comando:
neighbor peer-group-name peer-group

Este ejemplo aplica grupos de peers a vecinos BGP internos y externos:

```
RTC#
router bgp 300
 neighbor internalmap peer-group
 neighbor internalmap remote-as 300
 neighbor internalmap route-map SETMETRIC out
 neighbor internalmap filter-list 1 out
 neighbor internalmap filter-list 2 in
 neighbor 10.5.5.2 peer-group internalmap
 neighbor 10.6.6.2 peer-group internalmap
 neighbor 10.3.3.2 peer-group internalmap
 neighbor 10.3.3.2 filter-list 3 in
```

Esta configuración define un grupo de peers con el nombre internalmap. La configuración define algunas políticas para el grupo, como un mapa de ruta SETMETRIC para configurar la métrica en 5 y dos listas de filtros diferentes, 1 y 2. La configuración aplica el grupo de peers a todos los vecinos internos, RTE, RTF y RTG. Además, la configuración define una lista de filtros separada 3 para el vecino RTE. Esta lista de filtros invalida la lista de filtros 2 dentro del grupo de peers.

Nota:
Sólo puede anular opciones que afectan a las actualizaciones de entrada.
Ahora, observe cómo puede utilizar los grupos de peers con los vecinos externos. Con el mismo diagrama de esta sección, usted configura el RTC con un grupo de peers externalmap y aplica el grupo de peers a los vecinos externos.

```
RTC#
router bgp 300
 neighbor externalmap peer-group
 neighbor externalmap route-map SETMETRIC
 neighbor externalmap filter-list 1 out
 neighbor externalmap filter-list 2 in
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.2.2.2 peer-group externalmap
 neighbor 10.4.4.2 remote-as 600
 neighbor 10.4.4.2 peer-group externalmap
 neighbor 10.1.1.2 remote-as 200
 neighbor 10.1.1.2 peer-group externalmap
 neighbor 10.1.1.2 filter-list 3 in
```

Fuente:

“Examinar los casos de estudio del protocolo Border Gateway”. Cisco. Accedido el 20 de febrero de 2025. [En línea]. Disponible: https://www.cisco.com/c/es_mx/support/docs/ip/border-gateway-protocol-bgp/26634-bgp-toc.html