Casos Prácticos de BGP

Contenido

Introducción

Este documento contiene cinco casos prácticos del Border Gateway Protocol (BGP).

prerrequisitos

Requisitos

No hay requisitos específicos para este documento.

Componentes Utilizados

Este documento no tiene restricciones específicas en cuanto a versiones de software y de hardware.

Convenciones

Consulte Convenciones de Consejos TécnicosCisco para obtener más información sobre las convenciones del documento.

Caso Práctico de BGP 1

El BGP, que define RFC 1771, le permite crear ruteo de interdominios libre de loops entre sistemas autónomos (AS). Un AS es un conjunto de routers bajo una sola administración técnica. Los routers de un AS pueden utilizar varios protocolos Interior Gateway Protocol (IGP) para intercambiar información de ruteo dentro del AS. Los routers pueden utilizar un protocolo de gateway exterior para rutear paquetes fuera del AS.

¿Cómo Funciona el BGP?

El BGP utiliza TCP como protocolo de transporte, en el puerto 179. Dos routers BGP forman una conexión TCP entre ellos. Estos routers son routers de peer. Los routers de peer intercambian mensajes para abrir y confirmar los parámetros de conexión.

Los routers BGP intercambian información sobre la posibilidad de alcance de la red. Esta información es principalmente una indicación de las trayectorias completas que una ruta debe tomar para llegar a la red de destino. Las trayectorias son números de AS BGP. Esta información ayuda con la construcción de un gráfico de los AS que son libres de loops. En el gráfico, también se muestra dónde aplicar las políticas de ruteo para hacer cumplir algunas restricciones en el comportamiento de ruteo.

Los dos routers que forman una conexión TCP para intercambiar información de ruteo BGP son "peers" o "vecinos". Los peers BGP intercambian inicialmente las tablas de ruteo BGP completas. Después de este intercambio, los peers envían actualizaciones graduales como los cambios de tabla de ruteo. El BGP guarda un número de versión de la tabla de BGP. El número de versión es el mismo para todos los peers BGP. El número de versión cambia cada vez que BGP actualiza la tabla con cambios de información de ruteo. El envío de paquetes keepalive garantiza que se mantenga activa la conexión entre los peers BGP. Los paquetes de notificación se envían en

respuesta a errores o condiciones especiales.

iBGP y eBGP

Si un AS tiene varios altavoces BGP, el AS puede funcionar como servicio de tránsito para otros AS. Como se muestra en el diagrama de esta sección, el AS200 es un AS de tránsito para AS100 y AS300.

Para enviar la información a AS externos, se debe garantizar la posibilidad de alcance de la red. Para garantizar la posibilidad de alcance de la red, se llevan a cabo estos procesos:

Peering de iBGP entre los routers dentro de un AS

Redistribución de la información sobre BGP a los IGP que se ejecutan en el AS

Cuando el BGP se ejecuta entre routers que pertenecen a dos AS diferentes, esto se llama BGP externo (eBGP). Cuando el BGP se ejecuta entre routers en el mismo AS, esto se llama BGP interno (iBGP).

Habilitación del Ruteo BGP

Realice estos pasos para habilitar y configurar el BGP.

Suponga que desea tener dos routers, RTA y RTB, con comunicación vía BGP. En el primer ejemplo, el RTA y el RTB están en AS diferentes. En el segundo ejemplo, ambos routers pertenecen al mismo AS.

Defina el proceso de router y el número de AS al que pertenecen los routers.

Ejecute este comando para habilitar el BGP en un router:

router bgp *autonomous-system* RTA# router bgp 100 RTB# router bgp 200
Estas declaraciones indican que el RTA ejecuta BGP y pertenece a AS100. El RTB ejecuta BGP y pertenece a AS200.

Defina los vecinos BGP.

La formación de vecinos BGP indica los routers que intentarán comunicarse vía BGP. En la sección Formación de Vecinos BGP, se explica este proceso.

Formación de Vecinos BGP

Dos routers BGP se convierten en vecinos después de que los routers establezcan una conexión TCP entre ellos. La conexión TCP es esencial para que los dos routers de peer comiencen el intercambio de las actualizaciones de ruteo.

Una vez que la conexión TCP esté activa, los routers enviarán mensajes de apertura para intercambiar valores. Los valores que intercambian los routers incluyen el número de AS, la

versión de BGP que ejecutan los routers, el ID de router BGP y el tiempo de espera de keepalive. Después de la confirmación y la aceptación de estos valores, ocurre el establecimiento de la conexión de vecinos. Cualquier estado diferente a Established es una indicación de que los dos routers no se convirtieron en vecinos y de que los routers no pueden intercambiar las actualizaciones de BGP.

Ejecute este **comando neighbor** para establecer una conexión TCP:

```
neighbor ip-address remote-as number
```

El dato ***number*** en el comando es el número de AS del router al que desea realizar una conexión con BGP. El dato ***ip-address*** es la dirección de salto siguiente con conexión directa para eBGP. Para iBGP, el dato ***ip-address*** es cualquier dirección IP en el otro router.

Las dos direcciones IP que utiliza en el **comando neighbor** de los routers de peer *deben* poder alcanzarse entre sí. Una manera de verificar la posibilidad de alcance es un ping extendido entre las dos direcciones IP. El ping extendido fuerza al router que hace el ping a utilizar como origen la dirección IP que especifica el **comando neighbor**. El router debe utilizar esta dirección en lugar de la dirección IP de la interfaz de la cual pasa el paquete.

Si hay algún cambio de la configuración de BGP, *debe* restablecer la conexión de vecinos para permitir que los nuevos parámetros entren en vigencia.

clear ip bgp address

Nota: El dato ***address*** es la dirección de vecino.

clear ip bgp *

Este comando borra todas las conexiones de vecinos.

De forma predeterminada, las sesiones de BGP comienzan con el uso de la versión 4 de BGP y negocian de forma descendente las versiones anteriores, en caso de ser necesario. Usted puede prevenir las negociaciones y forzar la versión de BGP que los routers utilizan para comunicarse con un vecino. Ejecute este comando en el modo de configuración de router:

```
neighbor {ip address | peer-group-name} version value
```

Este es un ejemplo de la **configuración del comando neighbor**:

```
RTA#
router bgp 100
neighbor 129.213.1.1 remote-as 200

RTB#
router bgp 200
neighbor 129.213.1.2 remote-as 100
neighbor 175.220.1.2 remote-as 200

RTC#
router bgp 200
neighbor 175.220.212.1 remote-as 200
```

En este ejemplo, el RTA y el RTB ejecutan eBGP. El RTB y el RTC ejecutan iBGP. El número de AS remoto apunta a un AS interno o externo, que indica iBGP o eBGP. Además, los peers eBGP tienen conexión directa, pero los peers iBGP no tienen conexión directa. No es necesario que los routers iBGP tengan conexión directa. Pero, debe haber algún IGP que se ejecute y permita que los dos vecinos se alcancen entre sí.

En esta sección, se proporciona un ejemplo de la información que muestra el comando show ip bgp neighbors.

Nota: Preste especial atención al estado de BGP. Cualquier estado diferente a Established indica que los peers no están activos.

Nota: También, observe estos elementos:

La versión de BGP, que es 4.

El ID de router remoto.

Este número es la dirección IP más alta en el router o la interfaz Loopback más alta, si existiera.

La versión de tabla.

La versión de tabla proporciona el estado de la tabla. Cada vez que se recibe información nueva, la tabla aumenta la versión. Una versión que se incrementa continuamente indica que hay alguna inestabilidad de ruta que causa la actualización continua de las rutas.

```
# show ip bgp neighbors BGP neighbor is 129.213.1.1, remote AS 200, external link BGP version 4,
remote router ID 175.220.12.1 BGP state = Established, table version = 3, up for 0:10:59 Last
read 0:00:29, hold time is 180, keepalive interval is 60 seconds Minimum time between
advertisement runs is 30 seconds Received 2828 messages, 0 notifications, 0 in queue Sent 2826
messages, 0 notifications, 0 in queue Connections established 11; dropped 10
```

Interfaces Loopback y BGP

El uso de una interfaz Loopback para definir vecinos es común con iBGP, pero no con eBGP. Normalmente, usted utiliza la interfaz Loopback para asegurarse de que la dirección IP del vecino permanece activa y sea independiente del hardware que funciona correctamente. En el caso de eBGP, los routers de peer con frecuencia tienen conexión directa, y no se aplica Loopback.

Si utiliza la dirección IP en una interfaz Loopback en el **comando neighbor**, necesitará cierta configuración adicional en el router vecino. El router vecino necesita informar al BGP sobre el uso de una interfaz Loopback en lugar de una interfaz física para iniciar la conexión TCP de vecinos BGP. Para indicar una interfaz Loopback, ejecute este comando:

neighbor *ip-address* **update-source** *interface*

En este ejemplo, se ilustra el uso de este comando:

```
RTA#
     router bgp 100
     neighbor 190.225.11.1 remote-as 100
```

```
     neighbor 190.225.11.1 update-source loopback 1
RTB#
     router bgp 100
     neighbor 150.212.1.1 remote-as 100
```

En este ejemplo, el RTA y el RTB ejecutan iBGP dentro de AS100. En el **comando neighbor**, el RTB utiliza la interfaz Loopback del RTA, 150.212.1.1. En este caso, el RTA debe forzar al BGP a utilizar la dirección IP Loopback como origen en la conexión de vecinos TCP. Para forzar esta acción, el RTA agrega **update-source a *interface-type interface-number*** para que el comando sea **neighbor 190.225.11.1 update-source loopback 1**. Esta declaración fuerza al BGP a utilizar la dirección IP de la interfaz Loopback cuando BGP se comunica con el vecino 190.225.11.1.

Nota: El RTA ha utilizado la dirección IP de interfaz física del RTB, 190.225.11.1, como vecino. El uso de esta dirección IP es el motivo por el cual el RTB no necesita ninguna configuración especial. Consulte Configuración de Ejemplo de iBGP y eBGP Con o Sin una Interfaz Loopback para obtener una configuración de ejemplo de una situación de red completa.

Multisalto eBGP

En algunos casos, un router de Cisco puede ejecutar eBGP con un router externo que no permita la conexión directa de los dos peers externos. Para lograr la conexión, usted puede utilizar el multisalto eBGP. El multisalto eBGP permite una conexión de vecinos entre dos peers externos que no tengan conexión directa. El multisalto está disponible solo para eBGP, no para iBGP. En este ejemplo, se ilustra el multisalto eBGP:

```
RTA#
router bgp 100
neighbor 180.225.11.1 remote-as 300
neighbor 180.225.11.1 ebgp-multihop
RTB#
router bgp 300
neighbor 129.213.1.2 remote-as 100
```

El RTA indica un vecino externo que no tiene conexión directa. El RTA necesita indicar su uso del **comando neighbor ebgp-multihop**. Por otra parte, el RTB indica un vecino que tiene conexión directa, que es 129.213.1.2. Debido a esta conexión directa, el RTB no necesita el **comando neighbor ebgp-multihop**. Usted también debe configurar un ruteo estático o IGP para permitir que los vecinos sin conexión se alcancen entre sí.

En el ejemplo de la sección Multisalto eBGP (Balanceo de Carga), se muestra cómo alcanzar el balanceo de carga con BGP en un caso donde tiene eBGP en líneas paralelas.

Multisalto eBGP (Balanceo de Carga)

```
RTA#
int loopback 0
ip address 150.10.1.1 255.255.255.0
router bgp 100
neighbor 160.10.1.1 remote-as 200
neighbor 160.10.1.1 ebgp-multihop
neighbor 160.10.1.1 update-source loopback 0
network 150.10.0.0


ip route 160.10.0.0 255.255.0.0 1.1.1.2
```

```
ip route 160.10.0.0 255.255.0.0 2.2.2.2
RTB#
int loopback 0
ip address 160.10.1.1 255.255.255.0
router bgp 200
neighbor 150.10.1.1 remote-as 100
neighbor 150.10.1.1 update-source loopback 0
neighbor 150.10.1.1 ebgp-multihop
network 160.10.0.0


ip route 150.10.0.0 255.255.0.0 1.1.1.1
ip route 150.10.0.0 255.255.0.0 2.2.2.1
```

En este ejemplo, se ilustra el uso de las interfaces Loopback, **update-source** y **ebgp-multihop**. El ejemplo es una solución temporal para alcanzar el balanceo de carga entre dos altavoces eBGP en líneas seriales paralelas. En situaciones normales, BGP selecciona una de las líneas en la que se enviarán los paquetes y no sucede el balanceo de carga. Con la introducción de las interfaces Loopback, el salto siguiente para eBGP es la interfaz Loopback. Usted utiliza rutas estáticas, o IGP, para introducir dos trayectorias de costos equivalentes para alcanzar el destino. El RTA tiene dos opciones para alcanzar el salto siguiente 160.10.1.1: una trayectoria vía 1.1.1.2 y la otra trayectoria vía 2.2.2.2. El RTB tiene las mismas opciones.

Mapas de Ruta

Se utilizan mucho los mapas de ruta con BGP. En el contexto de BGP, el mapa de ruta es un método para controlar y modificar la información de ruteo. El control y la modificación de la información de ruteo ocurre a través de la definición de condiciones para la redistribución de rutas de un protocolo de ruteo a otro. O bien, el control de la información de ruteo puede ocurrir en la inserción dentro y fuera de BGP. El formato del mapa de ruta es el siguiente:

route-map *map-tag* **[[permit | deny] | [***sequence-number***]]**

La etiqueta de mapa es simplemente un nombre que usted le coloca al mapa de ruta. Puede definir varias instancias del mismo mapa de ruta, o bien la misma etiqueta de nombre. El número de secuencia es simplemente una indicación de la posición que un nuevo mapa de ruta tendrá en la lista de mapas de ruta que usted ya ha configurado con el mismo nombre.

En este ejemplo, hay dos instancias del mapa de ruta definido, con el nombre MYMAP. La primera instancia tiene un número de secuencia de 10 y la segunda tiene un número de secuencia de 20.

route-map MYMAP permit 10 (El primer conjunto de condiciones va aquí).

route-map MYMAP permit 20 (El segundo conjunto de condiciones va aquí).

Cuando usted aplica el mapa de ruta MYMAP a las rutas entrantes o salientes, el primer conjunto de condiciones se aplica vía la instancia 10. Si el primer conjunto de condiciones no se cumple, usted pasa a una instancia más alta del mapa de ruta.

Comandos de Configuración match y set

Cada mapa de ruta está compuesto de una lista de comandos de configuración **match** y **set**. El comando match especifica un **criterio de coincidencia** y el comando set especifica una acción de **configuración** si se cumple el criterio que hace cumplir el **comando match**.

Por ejemplo, puede definir un mapa de ruta que verifique las actualizaciones salientes. Si hay una coincidencia para la dirección IP 1.1.1.1, la métrica para esa actualización se configura en 5. Estos comandos ilustran el ejemplo:

```
match ip address 1.1.1.1 set metric 5
```

Ahora, si se cumple el criterio de coincidencia y usted tiene un **permiso**, hay una redistribución o un control de las rutas, como lo especifica la acción de configuración. Usted sale de la lista.

Si se cumple el criterio de coincidencia y usted tiene una **negación**, no hay redistribución ni control de la ruta. Usted sale de la lista.

Si no se cumple el criterio de coincidencia y usted tiene **un permiso o una negación**, se verifica la siguiente instancia del mapa de ruta. Por ejemplo, se verifica la instancia 20. Esta verificación de la siguiente instancia continúa hasta que usted salga de la lista o termine todas las instancias del mapa de ruta. Si usted termina la lista sin una coincidencia, la ruta **no se acepta ni se reenvía**.

En las versiones anteriores al Cisco IOS Software, versión 11.2, cuando usted utiliza mapas de ruta para filtrar actualizaciones de BGP en lugar de redistribuirlas entre los protocolos, *no puede* filtrar en el entrante si utiliza un **comando match** en la dirección IP. Un filtro en el saliente es aceptable. El Cisco IOS Software, versión 11.2, y las versiones posteriores no tienen esta restricción.

Los comandos relacionados para **match** son:

- **match as-path**
- **match community**
- **match clns**
- **match interface**
- **match ip address**
- **match ip next-hop**
- **match ip route-source**
- **match metric**
- **match route-type**

- **match tag**

Los comandos relacionados para **conjunto** son:

- **set as-path**
- **set clns**
- **set automatic-tag**
- **set community**
- **set interface**
- **set default interface**
- **set ip default next-hop**
- **set level**
- **set local-preference**
- **set metric**
- **set metric-type**
- **set next-hop**
- **set origin**
- **set tag**
- **set weight**

Observe algunos ejemplos de mapa de ruta:

Ejemplo 1

Suponga que el RTA y el RTB ejecutan el Routing Information Protocol (RIP), y que el RTA y el RTC ejecutan el BGP. El RTA obtiene actualizaciones vía BGP y redistribuye las actualizaciones a RIP. Suponga que el RTA desea redistribuir a rutas RTB por 170.10.0.0 con una métrica de 2 y

a todas las otras rutas con una métrica de 5. En este caso, usted puede utilizar esta configuración:

```
RTA#
router rip
network 3.0.0.0
network 2.0.0.0
network 150.10.0.0
passive-interface Serial0
redistribute bgp 100 route-map SETMETRIC

router bgp 100
neighbor 2.2.2.3 remote-as 300
network 150.10.0.0

route-map SETMETRIC permit 10
match ip-address 1
set metric 2

route-map SETMETRIC permit 20
set metric 5

access-list 1 permit 170.10.0.0 0.0.255.255
```

En este ejemplo, si una ruta coincide con la dirección IP 170.10.0.0, la ruta tiene una métrica de 2. Luego, usted sale de la lista de mapa de ruta. Si no hay una coincidencia, continúa hacia abajo por la lista de mapa de ruta, lo que indica la configuración de todo lo demás en la métrica 5.

Nota: Siempre hágase esta pregunta: "¿Qué sucede con las rutas que no coinciden con ninguna de las declaraciones match?" Esas rutas se descartan, de forma predeterminada.

'Ejemplo 2'

Suponga que, en el ejemplo 1, usted no desea que el AS100 acepte actualizaciones por 170.10.0.0. No puede aplicar mapas de ruta en el entrante cuando coincide con una dirección IP como base. Por lo tanto, debe utilizar un mapa de ruta saliente en RTC:

```
RTC#

router bgp 300
network 170.10.0.0
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-map STOPUPDATES out

route-map STOPUPDATES permit 10
match ip address 1

access-list 1 deny 170.10.0.0 0.0.255.255
access-list 1 permit 0.0.0.0 255.255.255.255
```

Ahora que usted está más familiarizado con cómo comenzar el BGP y cómo definir un vecino, observe cómo comenzar el intercambio de información de la red.

Hay varias formas de enviar la información de la red con el uso de BGP. En estas secciones, se analizan los métodos uno por uno:

Comando network

El formato del **comando network** es:

network *network-number* **[mask** *network-mask***]**

El **comando network controla** las redes que se originan desde este cuadro. Este concepto es diferente a la configuración familiar con el Interior Gateway Routing Protocol (IGRP) y RIP. Con este comando, usted no intenta ejecutar BGP en una interfaz determinada. En su lugar, intenta indicarle a BGP qué redes debe originar desde este cuadro. El comando utiliza una parte de la máscara porque la versión 4 de BGP (BGP4) puede manejar subredes y superedes. Se acepta un máximo de 200 entradas del **comando network**.

El **comando network** funciona si el router conoce la red que usted intenta anunciar, ya sea conectada, estática o detectada dinámicamente.

Un ejemplo del **comando network** es:

```
RTA#
router bgp 1
network 192.213.0.0 mask 255.255.0.0
ip route 192.213.0.0 255.255.0.0 null 0
```

Este ejemplo indica que el Router A genera una entrada de red para 192.213.0.0/16. El dato /16 indica que usted utiliza una supered de la dirección clase C y que anuncia los primeros dos octetos, o los primeros 16 bits.

Nota: Usted necesita la ruta estática para que el router genere 192.213.0.0 porque la ruta estática coloca una entrada coincidente en la tabla de ruteo.

Redistribución

El **comando network** es una manera de anunciar sus redes vía BGP. Otra manera es redistribuir su IGP en BGP. Su IGP puede ser IGRP, Open Shortest Path First (OSPF), RIP, Enhanced Interior Gateway Routing Protocol (EIGRP) u otro protocolo. Esta redistribución puede parecer complicada porque usted ahora vacía todas sus rutas internas en BGP; algunas de estas rutas se pueden haber detectado vía BGP y no es necesario que las envíe otra vez. Aplique el filtrado con cuidado para asegurarse de enviar a las rutas de solo Internet que desea anunciar y no a todas las rutas que usted tiene. Aquí tiene un ejemplo:

El RTA anuncia 129.213.1.0 y el RTC anuncia 175.220.0.0. Observe la configuración de RTC:

Si usted ejecuta el **comando network**, tiene:

```
RTC#
router eigrp 10
network 175.220.0.0
```

```
redistribute bgp 200
default-metric 1000 100 250 100 1500

router bgp 200
neighbor 1.1.1.1 remote-as 300
network 175.220.0.0 mask 255.255.0.0
!--- This limits the networks that your AS originates to 175.220.0.0.
```

Si usted utiliza la redistribución en su lugar, tiene:

```
RTC#
router eigrp 10
network 175.220.0.0
redistribute bgp 200
default-metric 1000 100 250 100 1500

router bgp 200
neighbor 1.1.1.1 remote-as 300
redistribute eigrp 10
!--- EIGRP injects 129.213.1.0 again into BGP.
```

Esta redistribución causa el origen de 129.213.1.0 por su AS. Usted no es el origen de 129.213.1.0; AS100 es el origen. Por lo tanto, usted debe utilizar filtros para prevenir que el origen salga de esa red por su AS. La configuración correcta es:

```
RTC#
router eigrp 10
network 175.220.0.0
redistribute bgp 200
default-metric 1000 100 250 100 1500

router bgp 200
neighbor 1.1.1.1 remote-as 300
neighbor 1.1.1.1 distribute-list 1 out
redistribute eigrp 10

access-list 1 permit 175.220.0.0 0.0.255.255
```

Utilice el **comando access-list** para controlar las redes que se originan desde AS200.

La redistribución de OSPF en BGP es levemente diferente a la redistribución para otros IGP. La simple ejecución de **redistribute ospf 1** en **router bgp** no funciona. Las palabras clave específicas como **internal**, **external** y **nssa-external** son necesarias para redistribuir las rutas respectivas. Consulte Comprensión de la Redistribución de Rutas OSPF en BGP para obtener más detalles.

Rutas Estáticas y Redistribución

Siempre puede utilizar rutas estáticas para originar una red o una subred. La única diferencia es que BGP considera estas rutas para tener un origen que esté incompleto, o sea desconocido. Usted puede lograr el mismo resultado que se logra en el ejemplo de la sección Redistribución con esto:

```
RTC#
router eigrp 10
network 175.220.0.0
redistribute bgp 200
default-metric 1000 100 250 100 1500
```

```
router bgp 200
neighbor 1.1.1.1 remote-as 300
redistribute static
...
ip route 175.220.0.0 255.255.255.0 null0
....
```

La interfaz **null0** significa ignorar el paquete. Por lo tanto, si usted obtiene el paquete y hay una coincidencia más específica que 175.220.0.0, que existe, el router envía el paquete a la coincidencia específica. De lo contrario, el router ignora el paquete. Este método es una buena manera de anunciar una supered.

En este documento, se ha analizado cómo puede utilizar diferentes métodos para originar rutas fuera de su AS. Recuerde que estas rutas se generan además de otras rutas BGP que BGP detecte vía vecinos, ya sean internos o externos. BGP pasa información que detecta de un peer a otros peers. La diferencia es que las rutas que se generan del **comando network**, de la redistribución o de la estática indican a su AS como el origen de estas redes.

La redistribución es siempre el método de inserción de BGP en IGP.

Aquí tiene un ejemplo:

```
RTA#
router bgp 100
neighbor 150.10.20.2 remote-as 300
network 150.10.0.0

RTB#
router bgp 200
neighbor 160.10.20.2 remote-as 300
network 160.10.0.0

RTC#
router bgp 300
neighbor 150.10.20.1 remote-as 100
neighbor 160.10.20.1 remote-as 200
network 170.10.00
```

Nota: Usted no necesita la red 150.10.0.0 o la red 160.10.0.0 en RTC, a menos que desee que RTC genere estas redes así como que pase estas redes a medida que lleguen de AS100 y AS200. Una vez más, la diferencia es que el **comando network** agrega un anuncio adicional para estas mismas redes, lo que indica que AS300 es también un origen para estas rutas.

Nota: Recuerde que BGP no acepta actualizaciones que se hayan originado desde su propio AS. Esta negación garantiza una topología de interdominios libre de loops.

Por ejemplo, suponga que el AS200, del ejemplo de esta sección, tiene una conexión BGP directa al AS100. El RTA genera una ruta 150.10.0.0 y envía la ruta a AS300. Luego, el RTC pasa esta ruta a AS200 y guarda el origen como AS100. El RTB pasa 150.10.0.0 a AS100 con el origen todavía en AS100. El RTA nota que la actualización se ha originado desde su propio AS e ignora la actualización.

iBGP

Usted utiliza iBGP si un AS desea actuar como un sistema de tránsito a otros AS. ¿Es verdad que

usted puede obtener el mismo resultado al detectar vía eBGP, redistribuir en IGP y después redistribuir otra vez en otro AS? Sí, pero iBGP ofrece mayor flexibilidad y maneras más eficaces de intercambiar información dentro de un AS. Por ejemplo, iBGP proporciona formas de controlar el mejor punto de salida fuera del AS con el uso de preferencia local. En la sección Atributo de Preferencia Local, se proporciona más información sobre preferencia local.

```
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

RTB#
router bgp 100
neighbor 150.10.30.1 remote-as 100
neighbor 175.10.40.1 remote-as 400
network 190.10.50.0

RTC#
router bgp 400
neighbor 175.10.40.2 remote-as 100
network 175.10.0.0
```

Nota: Recuerde que cuando un altavoz BGP recibe una actualización de otros altavoces BGP en su propio AS (iBGP), el altavoz BGP que recibe la actualización no redistribuye esa información a otros altavoces BGP en su propio AS. El altavoz BGP que recibe la actualización redistribuye la información a otros altavoces BGP fuera de su AS. Por lo tanto, mantenga una malla completa entre los altavoces iBGP dentro de un AS.

En el diagrama de esta sección, el RTA y el RTB ejecutan iBGP. El RTA y el RTD también ejecutan iBGP. Las actualizaciones de BGP que vienen del RTB al RTA se transmiten al RTE, que está fuera del AS. Las actualizaciones no se transmiten al RTD, que está dentro del AS. Por lo tanto, realice un peering de iBGP entre el RTB y el RTD para no interrumpir el flujo de las actualizaciones.

El Algoritmo de Decisión de BGP

Después de que BGP reciba actualizaciones sobre diferentes destinos de diferentes sistemas autónomos, el protocolo deberá elegir las trayectorias para alcanzar un destino específico. BGP elige solo una única trayectoria para alcanzar un destino específico.

BGP basa la decisión en diferentes **atributos**, como salto siguiente, pesos administrativos, preferencia local, origen de ruta, longitud de trayectoria, código de origen, métrica y otros atributos.

BGP siempre propaga la mejor trayectoria a los vecinos. Si desea obtener más información, consulte Algoritmo de Selección de la Mejor Trayectoria de BGP.

En la sección Caso Práctico de BGP 2, se explican estos atributos y su uso.

Caso Práctico de BGP 2

Atributo AS_PATH

Siempre que una actualización de ruta pase a través de un AS, el número de AS se antepone a esa actualización. El atributo AS_PATH es en verdad la lista de números de AS que una ruta ha atravesado para alcanzar un destino. Un AS_SET es un conjunto matemático ordenado {} de todos los AS que se han atravesado. En la sección Ejemplo de CIDR 2 (as-set) de este documento, se proporciona un ejemplo de AS_SET.

En el ejemplo de esta sección, el RTB anuncia la red 190.10.0.0 en AS200. Cuando esa ruta atraviesa AS300, el RTC agrega su propio número de AS a la red. Por lo tanto, cuando 190.10.0.0 alcanza el RTA, la red tiene dos números de AS adjuntos: primero 200 y luego 300. Para el RTA, la trayectoria para alcanzar 190.10.0.0 es (300, 200).

El mismo proceso se aplica a 170.10.0.0 y a 180.10.0.0. El RTB tiene que tomar la trayectoria (300, 100); el RTB atraviesa AS300 y luego AS100 para alcanzar 170.10.0.0. El RTC tiene que atravesar la trayectoria (200) para alcanzar 190.10.0.0 y la trayectoria (100) para alcanzar 170.10.0.0.

Atributo de Origen

El origen es un atributo obligatorio que define el origen de la información de trayectoria. El atributo de origen puede suponer tres valores:

IGP — La Información de alcance de la capa de red (NLRI) es interior al a partir de las creaciones. Esto sucede normalmente cuando usted ejecuta el **comando bgp network**. Una i en la tabla de BGP indica IGP.

EGP: el NLRI se detecta vía Exterior Gateway Protocol (EGP). Una e en la tabla de BGP indica EGP.

INCOMPLETE: el NLRI es desconocido o se detecta vía algún otro medio. INCOMPLETE ocurre generalmente cuando usted redistribuye las rutas de otros protocolos de ruteo en BGP y el origen de la ruta está incompleto. Un ? en la tabla de BGP indica INCOMPLETE.

```
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
```
El RTA alcanza 170.10.0.0 vía 300 i. "300 i" significa que la siguiente trayectoria de AS es 300 y que el origen de la ruta es IGP. El RTA también alcanza 190.10.50.0 vía i. "i" significa que la entrada está en el mismo AS y que el origen es IGP. El RTE alcanza 150.10.0.0 vía 100 i. "100 i" significa que el siguiente AS es 100 y que el origen es IGP. El RTE también alcanza 190.10.0.0 vía 100 ?. "100 ?" significa que el siguiente AS es 100 y que el origen está incompleto y viene de una ruta estática.

Atributo de Salto Siguiente de BGP

El atributo de salto siguiente de BGP es la dirección IP de salto siguiente que se utiliza para alcanzar un destino determinado.

Para eBGP, el salto siguiente es siempre la dirección IP del vecino que especifica el **comando neighbor**. En el ejemplo de esta sección, el RTC anuncia 170.10.0.0 al RTA con un salto siguiente de 170.10.20.2. El RTA anuncia 150.10.0.0 al RTC con un salto siguiente de 170.10.20.1. Para iBGP, el protocolo establece que el salto siguiente que eBGP anuncie se debería llevar en iBGP. Debido a esta regla, el RTA anuncia 170.10.0.0 a su peer iBGP, RTB, con un salto siguiente de 170.10.20.2. Por lo tanto, según el RTB, el salto siguiente para alcanzar 170.10.0.0 es 170.10.20.2 y *no* 150.10.30.1.

Asegúrese de que el RTB pueda alcanzar 170.10.20.2 vía IGP. De no ser así, el RTB descarta los paquetes con el destino de 170.10.0.0 porque la dirección de salto siguiente es inaccesible. Por ejemplo, si el RTB ejecuta iGRP, usted también puede ejecutar iGRP en el RTA, en la red 170.10.0.0. Desea que iGRP esté pasivo en el link al RTC para que solo se intercambie BGP.

```
RTA#
router bgp 100
neighbor 170.10.20.2 remote-as 300
neighbor 150.10.50.1 remote-as 100
network 150.10.0.0
RTB#
router bgp 100
neighbor 150.10.30.1 remote-as 100
RTC#
router bgp 300
neighbor 170.10.20.1 remote-as 100
network 170.10.0.0
```
Nota: El RTC anuncia 170.10.0.0 al RTA con un salto siguiente igual a 170.10.20.2.

Nota: El RTA anuncia 170.10.0.0 al RTB con un salto siguiente igual a 170.10.20.2. El salto siguiente de eBGP se lleva en iBGP.

Tenga especial cuidado cuando trate redes multiacceso y redes multiacceso sin broadcast (NBMA). En las secciones Salto Siguiente de BGP (Redes Multiacceso) y Salto Siguiente de BGP (NBMA), se proporcionan más detalles.

Salto Siguiente de BGP (Redes Multiacceso)

En este ejemplo, se muestra cómo se comporta el salto siguiente en una red multiacceso como Ethernet.

Suponga que el RTC y el RTD en AS300 ejecutan OSPF. El RTC ejecuta BGP con el RTA. El RTC puede alcanzar la red 180.20.0.0 vía 170.10.20.3. Cuando el RTC envía una actualización

de BGP al RTA con respecto a 180.20.0.0, el RTC utiliza a 170.10.20.3 como salto siguiente. El RTC no utiliza su propia dirección IP, 170.10.20.2. El RTC utiliza esta dirección porque la red entre el RTA, el RTC y el RTD es una red multiacceso. El uso de RTD del RTA como salto siguiente para alcanzar 180.20.0.0 es más sensato que el salto adicional vía el RTC.

Nota: El RTC anuncia 180.20.0.0 al RTA con un salto siguiente 170.10.20.3.

Si el medio común al RTA, al RTC y al RTD no es una red multiacceso, sino NBMA, ocurren otras complicaciones.

Salto Siguiente de BGP (NBMA)

El medio común aparece como una nube en el diagrama. Si el medio común es un Frame Relay o cualquier nube NBMA, el comportamiento exacto es como si usted tuviera una conexión vía Ethernet. El RTC anuncia 180.20.0.0 al RTA con un salto siguiente de 170.10.20.3.

El problema es que el RTA no tiene un circuito virtual permanente (PVC) directo al RTD y no puede alcanzar el salto siguiente. En este caso, el ruteo falla.

El **comando next-hop-self** soluciona esta situación.

Comando next-hop-self

Para situaciones con el salto siguiente, como en el ejemplo de Salto Siguiente de BGP (NBMA), usted puede utilizar el **comando next-hop-self**. La sintaxis es la siguiente:

```
neighbor {ip-address | peer-group-name} next-hop-self
```

El **comando next-hop-self** le permite forzar al BGP a utilizar una dirección IP específica como salto siguiente.

Para el ejemplo de Salto Siguiente de BGP (NBMA), esta configuración soluciona el problema:

```
RTC#
router bgp 300
neighbor 170.10.20.1 remote-as 100
neighbor 170.10.20.1 next-hop-self
```

El RTC anuncia 180.20.0.0 con un salto siguiente igual a 170.10.20.2.

Puerta Trasera de BGP

En este diagrama, el RTA y el RTC ejecutan eBGP. El RTB y el RTC ejecutan eBGP. El RTA y el RTB ejecutan un tipo de IGP, ya sea RIP, IGRP u otro protocolo. Por definición, las actualizaciones de eBGP tienen una distancia de 20, que es menor que las distancias de IGP. Las distancias predeterminadas son:

- 120 para RIP
- 100 para IGRP
- 90 para EIGRP

110 para OSPF

El RTA recibe las actualizaciones por 160.10.0.0 vía dos protocolos de ruteo:

eBGP con una distancia de 20

IGP con una distancia que es mayor que 20

De forma predeterminada, BGP tiene estas distancias:

Distancia externa: 20

Distancia interna: 200

Distancia local: 200

Pero usted puede utilizar el **comando distance** para cambiar las distancias predeterminadas:

distance bgp *external-distance internal-distance local-distance*

El RTA selecciona eBGP vía el RTC debido a la distancia más corta.

Si usted desea que RTA detecte 160.10.0.0 vía RTB (IGP), tendrá dos opciones:

Cambiar la distancia externa de eBGP o la distancia de IGP.

Nota: No se recomienda este cambio.

Utilizar la puerta trasera de BGP.

La puerta trasera de BGP hace que la ruta de IGP sea la preferida.

Ejecute el comando network address backdoor.

La red configurada es la red que usted desea alcanzar vía IGP. Para BGP, esta red obtiene el mismo tratamiento que una red asignada localmente, excepto que las actualizaciones de BGP no anuncian esta red.

```
RTA#
router eigrp 10

network 150.10.0.0

router bgp 100
neighbor 2.2.2.1 remote-as 300
network 160.10.0.0 backdoor
```

La red 160.10.0.0 se trata como entrada local, pero no se anuncia como entrada de red normal.

El RTA detecta 160.10.0.0 del RTB vía EIGRP con una distancia de 90. El RTA también detecta la dirección del RTC vía eBGP con una distancia de 20. El eBGP es normalmente la preferencia, pero debido al **comando network backdoor**, el EIGRP es la preferencia.

Sincronización

Antes de tratar el tema de sincronización, observe esta situación. El RTC en AS300 envía las actualizaciones por 170.10.0.0. El RTA y el RTB ejecutan iBGP; por lo tanto, el RTB obtiene la actualización y puede alcanzar 170.10.0.0 vía el salto siguiente 2.2.2.1. Recuerde que el salto siguiente se lleva vía iBGP. Para alcanzar el salto siguiente, el RTB debe enviar el tráfico al RTE.

Suponga que el RTA no tiene la red redistribuida 170.10.0.0 en IGP. En este punto, el RTE no tiene idea de que 170.10.0.0 existe.

Si el RTB comienza a anunciar al AS400 que el RTB puede alcanzar 170.10.0.0, el tráfico que viene del RTD al RTB con destino 170.10.0.0 fluye y se descarta en el RTE.

La sincronización establece que, si su AS pasa el tráfico de otro AS a un tercer AS, el BGP no debería anunciar una ruta antes de que todos los routers en su AS hayan detectado la ruta vía IGP. El BGP esperará hasta que el IGP haya propagado la ruta dentro del AS. Luego, el BGP anuncia de la ruta a los peers externos.

En el ejemplo de esta sección, el RTB espera para obtener datos por 170.10.0.0 vía IGP. Luego, el RTB comienza a enviar la actualización al RTD. Usted puede hacer que el RTB piense que el IGP ha propagado la información al agregar una ruta estática en RTB que apunte a 170.10.0.0. Asegúrese de que los otros routers pueden alcanzar 170.10.0.0.

Inhabilitación de la Sincronización

En algunos casos, usted no necesita la sincronización. Si no pasa el tráfico de un AS diferente a través de su AS, puede inhabilitar la sincronización. También puede inhabilitar la sincronización si todos los routers en su AS ejecutan BGP. La inhabilitación de esta función puede permitirle llevar menos rutas en su IGP y permitir que el BGP converja más rápidamente.

La inhabilitación de la sincronización no es automática. Si todos sus routers en el AS ejecutan BGP y usted no ejecuta IGP en absoluto, el router no tiene ninguna manera de saberlo. El router espera indefinidamente una actualización de IGP por una ruta determinada antes de enviar la ruta a los peers externos. Usted debe inhabilitar la sincronización manualmente en este caso para que el ruteo pueda funcionar correctamente:

```
router bgp 100
no synchronization
```

Nota: Asegúrese de ejecutar el **comando clear ip bgp address** para restablecer la sesión.

```
RTB#
router bgp 100
network 150.10.0.0
neighbor 1.1.1.2 remote-as 400
neighbor 3.3.3.3 remote-as 100
no synchronization
!--- RTB puts 170.10.0.0 in its IP routing table and advertises the network !--- to RTD, even if
RTB does not have an IGP path to 170.10.0.0. RTD# router bgp 400 neighbor 1.1.1.1 remote-as 100
network 175.10.0.0 RTA# router bgp 100 network 150.10.0.0 neighbor 3.3.3.4 remote-as 100
```

Atributo de Peso

El atributo de peso es un atributo definido por Cisco. Este atributo utiliza el peso para seleccionar una mejor trayectoria. El peso se asigna localmente al router. El valor solo tiene sentido para el router específico. El valor no se propaga ni se lleva a través de ninguna de las actualizaciones de ruta. Un peso puede ser un número del 0 al 65.535. Las trayectorias que el router origina tienen un peso de 32.768 de forma predeterminada y las demás trayectorias tienen un peso de 0.

Las rutas con un valor de peso más alto tienen preferencia cuando existen varias rutas hacia el mismo destino. Observe el ejemplo de esta sección. El RTA ha detectado la red 175.10.0.0 del AS4. El RTA propaga la actualización al RTC. El RTB también ha detectado la red 175.10.0.0 del AS4. El RTB propaga la actualización al RTC. El RTC ahora tiene dos maneras de alcanzar 175.10.0.0 y tiene que decidir qué opción elegirá. Si usted configura el peso de las actualizaciones en el RTC que vienen del RTA de modo que el peso sea mayor que el peso de las actualizaciones que vienen del RTB, fuerza al RTC a utilizar el RTA como salto siguiente para alcanzar 175.10.0.0. Varios métodos logran este peso configurado:

Utilice el **comando neighbor**.

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

Utilice listas de acceso AS_PATH.

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

Utilice mapas de ruta.

```
RTC#
router bgp 300
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 weight 200
!--- The route to 175.10.0.0 from RTA has a 200 weight. neighbor 2.2.2.2 remote-as 200
neighbor 2.2.2.2 weight 100 !--- The route to 175.10.0.0 from RTB has a 100 weight.
```

El RTA, que tiene un valor de peso más alto, tiene preferencia como salto siguiente.

Usted puede alcanzar el mismo resultado con listas de filtros y AS_PATH IP.

```
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$
!--- This only permits path 100. ip as-path access-list 6 permit ^200$ ...
```

También puede alcanzar el mismo resultado con el uso de mapas de ruta.

```
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
!--- Anything that applies to access list 5, such as packets from AS100, has weight 200. route-
map setweightin permit 20 set weight 100 !--- Anything else has weight 100.
```

Nota: Usted puede modificar la ponderación para preferir el trayecto BGP del MPLS VPN con la trayectoria IGP como respaldo.

Nota: Para más información, refiera a este documento de la comunidad del soporte de Cisco que describa cómo configurar al router para tener un trayecto preferido en primario y las condiciones de error y para rerrutearlo en la recuperación del trayecto principal: Preferir el trayecto BGP del MPLS VPN con el respaldo IGP

Atributo de Preferencia Local

La preferencia local es una indicación para el AS sobre qué trayectoria tiene preferencia para salir del AS a fin de alcanzar una red determinada. Una trayectoria con una preferencia local más alta se prefiere más. El valor predeterminado por preferencia local es 100.

A diferencia del atributo de peso, que es importante solo para el router local, la preferencia local es un atributo que los routers intercambian en el mismo AS.

Usted configura la preferencia local con la ejecución del **comando bgp default local-preference value**. También puede configurar la preferencia local con los mapas de ruta, como se muestra en el ejemplo de esta sección:

Nota: Es necesario realizar un restablecimiento por software (es decir, borrar el proceso de BGP en el router) para que se consideren los cambios. Para borrar el proceso de BGP, utilice el comando **clear ip bgp** **[soft][in/out] donde el dato soft** indica un restablecimiento por software sin interrumpir la sesión y el dato **[in/out]** especifica la configuración de entrante o saliente. Si el dato **in/out** no se especifica, se restablecen las sesiones tanto entrantes como salientes.

El **comando bgp default local-preference** configura la preferencia local en las actualizaciones fuera del router que van a peers en el mismo AS. En el diagrama de esta sección, el AS256 recibe actualizaciones por 170.10.0.0 de dos lados diferentes de la organización. La preferencia local le ayuda a determinar la manera de salir del AS256 para alcanzar esa red. Suponga que el RTD es la preferencia de punto de salida. Esta configuración configura la preferencia local para las actualizaciones que vienen del AS300 en 200 y para las actualizaciones que vienen del AS100 en 150:

```
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 esta configuración, el RTC configura la preferencia local de todas las actualizaciones en 150. El RTD configura la preferencia local de todas las actualizaciones en 200. Hay un intercambio de preferencia local dentro del AS256. Por lo tanto, el RTC y el RTD se dan cuenta de que la red 170.10.0.0 tiene una preferencia local más alta cuando las actualizaciones vienen del AS300 que cuando vienen del AS100. Todo el tráfico en el AS256 que tiene a esa red como destino transmite con el RTD como punto de salida.

El uso de mapas de ruta proporciona mayor flexibilidad. En el ejemplo de esta sección, todas las actualizaciones que el RTD recibe se etiquetan con la preferencia local de 200 cuando las actualizaciones alcanzan el RTD. Las actualizaciones que vienen del AS34 también se etiquetan con la preferencia local de 200. Esta etiqueta puede ser innecesaria. Por esta razón, usted puede utilizar mapas de ruta para especificar las actualizaciones específicas que se deben etiquetar con una preferencia local específica. Aquí tiene un ejemplo:

```
RTD#
router bgp 256
neighbor 3.3.3.4 remote-as 300
neighbor 3.3.3.4 route-map setlocalin in
neighbor 128.213.11.1 remote-as 256
....
ip as-path access-list 7 permit ^300$
...

route-map setlocalin permit 10
match as-path 7
set local-preference 200

route-map setlocalin permit 20
set local-preference 150
```

Con esta configuración, cualquier actualización que venga del AS300 tendrá una preferencia local de 200. Todas las demás actualizaciones, como actualizaciones que vengan del AS34, tendrán un valor de 150.

Atributo de Métrica

El atributo de métrica también tiene el nombre MULTI_EXIT_DISCRIMINATOR, MED (BGP4) o INTER_AS (BGP3). Este atributo es una sugerencia para los vecinos externos sobre la preferencia de trayectoria en un AS. El atributo proporciona una forma dinámica de influir en otro AS sobre la manera de alcanzar una ruta determinada cuando hay varios puntos de entrada en ese AS. Un valor de métrica más bajo se prefiere más.

A diferencia de la preferencia local, la métrica se intercambia entre los AS. Una métrica se lleva en un AS, pero no sale del AS. Cuando una actualización ingresa en el AS con una métrica determinada, esa métrica se utiliza para tomar decisiones dentro del AS. Cuando la misma actualización pasa a un tercer AS, esa métrica regresa a 0. En el diagrama de esta sección, se muestra la configuración de la métrica. El valor de métrica predeterminado es 0.

A menos que un router reciba otras instrucciones, el router compara las métricas para las trayectorias de los vecinos en el mismo AS. Para que el router compare las métricas de vecinos que vienen de diferentes AS, debe ejecutar el comando de configuración espacial bgp always-compare-med en el router.

Nota: Hay dos comandos de configuración de BGP que pueden influir sobre la selección de trayectoria basada en discriminador de salidas múltiples (MED). Los comandos son bgp deterministic-med y bgp always-compare-med. Una ejecución del **comando bgp deterministic-med** garantiza la comparación de la variable MED en la opción de ruta cuando diferentes peers anuncian en el mismo AS. Una ejecución del **comando bgp always-compare-med** garantiza la comparación de MED para las trayectorias de vecinos en diferentes AS. El **comando bgp always-compare-med** es útil cuando varios proveedores de servicios o varias empresas acuerdan el uso de una política uniforme para configurar MED. Consulte Cómo Difiere el Comando bgp deterministic-med del Comando bgp always-compare-med para comprender cómo estos comandos influyen sobre la selección de trayectoria de BGP.

En el diagrama de esta sección, el AS100 obtiene información por la red 180.10.0.0 vía tres routers diferentes: RTC, RTD y RTB. El RTC y el RTD están en el AS300 y el RTB está en el AS400.

En este ejemplo, la comparación de la Como-trayectoria en el RTA del comando bgp bestpath as-path ignore se ignora. Se configura para forzar el BGP para caer encendido al atributo siguiente para la comparación de la ruta (en este caso métrica o MED). Si se omite el comando, el BGP instalará la ruta 180.10.0.0 del router RTC como ése tiene la Como-trayectoria más corta.

Suponga que usted ha configurado la métrica que viene del RTC en 120, la métrica que viene del RTD en 200 y la métrica que viene del RTB en 50. De forma predeterminada, un router compara las métricas que vienen de los vecinos en el mismo AS. Por lo tanto, el RTA puede comparar solamente la métrica que viene del RTC con la métrica que viene del RTD. El RTA elige el RTC como el mejor salto siguiente porque 120 es menor que 200. Cuando el RTA obtiene una actualización del RTB con una métrica de 50, el RTA no puede comparar la métrica con 120 porque el RTC y el RTB están en AS diferentes. El RTA debe elegir según algunos otros atributos.

Para forzar al RTA a comparar las métricas, usted debe ejecutar el **comando bgp always-compare-med** en el RTA. Estas configuraciones ilustran este proceso:

```
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 bestpath as-path ignore
   ....

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
```

Con estas configuraciones, el RTA selecciona RTC como salto siguiente, con la consideración del hecho de que los demás atributos son los mismos. Para incluir RTB en la comparación de métricas, usted debe configurar el RTA de esta manera:

```
RTA#
router bgp 100
neighbor 2.2.21 remote-as 300
neighbor 3.3.3.3 remote-as 300
neighbor 4.4.4.3 remote-as 400
bgp always-compare-med
```

En este caso, el RTA selecciona RTB como el mejor salto siguiente para alcanzar la red 180.10.0.0.

Usted también puede configurar la métrica durante la redistribución de rutas en BGP si ejecuta el **comando default-metric number**.

Suponga que, en el ejemplo de esta sección, el RTB inserta una red vía estática en el AS100. Esta es la configuración:

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

ip route 180.10.0.0 255.255.0.0 null 0
!--- This causes RTB to send out 180.10.0.0 with a metric of 50.
```

Atributo de Comunidad

El atributo de comunidad es un atributo opcional transitivo en el rango de 0 a 4.294.967.200. El atributo de comunidad es una manera de agrupar destinos en una comunidad determinada y de aplicar decisiones de ruteo según esas comunidades. Las decisiones de ruteo son aceptar, preferir y redistribuir, entre otras.

Usted puede utilizar mapas de ruta para configurar atributos de comunidad. El comando de configuración de mapa de ruta tiene esta sintaxis:

```
set community community-number [additive] [well-known-community]
```

Algunas comunidades conocidas predefinidas para su uso en este comando son:

no-export: para no anunciar a peers eBGP. Conserva esta ruta dentro de un AS.

no-advertise: para no anunciar esta ruta a ningún peer, interno ni externo.

internet: para anunciar esta ruta a la comunidad de Internet. Todo router pertenece a esta comunidad.

local-as: para usar en situaciones de confederación para prevenir la transmisión de paquetes fuera del AS local.

Aquí hay dos ejemplos de mapas de ruta que configuran la comunidad:

- ```
  route-map communitymap
  match ip address 1
  set community no-advertise
  ```

  o

- ```
  route-map setcommunity
  match as-path 1
  set community 200 additive
  ```

Si usted no configura la **palabra clave additive**, 200 reemplazará toda comunidad anterior que ya salga. Si usted utiliza la palabra clave additive, ocurre una adición de 200 a la comunidad. Incluso si usted configura el atributo de comunidad, este atributo no se transmite a los vecinos de forma predeterminada. Para enviar el atributo a un vecino, debe utilizar este comando:

neighbor {*ip-address* **|** *peer-group-name***} send-community**

Aquí tiene un ejemplo:

```
RTA#
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
```

En el Cisco IOS Software, versión 12.0 y posteriores, puede configurar comunidades en tres formatos diferentes: decimal, hexadecimal y AA: NN. De forma predeterminada, el Cisco IOS Software utiliza el formato decimal más antiguo. Para configurar y mostrar en el formato AA: NN, ejecute el **comando de configuración global ip bgp-community new-format**. La primera parte de AA: NN representa el número de AS y la segunda parte representa un número de 2 bytes.

Aquí tiene un ejemplo:

Sin el comando ip bgp-community new-format en la configuración global, una ejecución del comando show ip bgp 6.0.0.0 muestra el valor del atributo de comunidad en el formato decimal.
En este ejemplo, el valor del atributo de comunidad aparece como 6553620.

```
Router# show ip bgp 6.0.0.0 BGP routing table entry for 6.0.0.0/8, version 7 Paths: (1
available, best #1, table Default-IP-Routing-Table) Not advertised to any peer 1 10.10.10.1 from
10.10.10.1 (200.200.200.1) Origin IGP, metric 0, localpref 100, valid, external, best
Community: 6553620
```

Ahora, ejecute el **comando ip bgp-community new-format** de forma global en este router.

```
Router# configure terminal Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# ip bgp-community new-format Router(config)# exit
```

Con el **comando de configuración global ip bgp-community new-format**, el valor de comunidad se muestra en el formato AA: NN. El valor aparece como 100:20 en el resultado del **comando show ip bgp 6.0.0.0** en este ejemplo:

```
Router# show ip bgp 6.0.0.0 BGP routing table entry for 6.0.0.0/8, version 9 Paths: (1
available, best #1, table Default-IP-Routing-Table) Not advertised to any peer 1 10.10.10.1 from
10.10.10.1 (200.200.200.1) Origin IGP, metric 0, localpref 100, valid, external, best
Community: 100:20
```

Caso Práctico de BGP 3

Filtrado 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.

Filtrado de Rutas

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 160.10.0.0 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 170.10.0.0
neighbor 3.3.3.3 remote-as 200
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 distribute-list 1 out

access-list 1 deny 160.10.0.0 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 160.0.0.0/8.

Nota: La anotación /8 significa que usted utiliza 8 bits de máscara de subred, que comienzan del extremo izquierdo de la dirección IP. Esta dirección es equivalente a 160.0.0.0 255.0.0.0.

El comando access-list 1 permit 160.0.0.0 0.255.255.255 permite 160.0.0.0/8, 160.0.0.0/9 y así sucesivamente. Para restringir la actualización a solamente 160.0.0.0/8, debe utilizar una lista de acceso extendida de este formato:

```
access-list 101 permit ip 160.0.0.0 0.255.255.255 255.0.0.0 0.0.0.0.
```

Esta lista permite 160.0.0.0/8 solamente.

Consulte Cómo Bloquear Una o Más Redes de un Peer BGP para obtener configuraciones de ejemplo de cómo filtrar redes de peers BGP. El método utiliza el **comando distribute-list** con las listas de control de acceso (ACL) estándar y extendidas, así como el filtrado de listas de prefijos.

Filtrado de Trayectorias

Otro tipo de filtrado es filtrado de trayectorias.

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 160.10.0.0 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 160.10.0.0 al RTA:

```
RTC#
router bgp 300
neighbor 3.3.3.3 remote-as 200
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 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 **comando access-list 1** en este ejemplo fuerza la negación de cualquier actualización con información de trayectoria que comience con 200 y termine con 200. El dato ^200$ en el comando es una "expresión regular", donde ^ significa "comienza con" y $ significa "termina con". Dado que el RTB envía actualizaciones por 160.10.0.0 con información de trayectoria 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 donde . significa "cualquier carácter" y * significa "la repetición de ese carácter". Por lo tanto, . * representa cualquier información de trayectoria, que sea necesaria para permitir la transmisión del resto de las actualizaciones.

¿Qué sucede si, en vez del uso de ^200$, usted 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 trayectoria comienza con 200. La lista de acceso previene la transmisión de estas actualizaciones al RTA, que no es el requisito.

Para verificar si usted ha implementado la expresión regular correcta, ejecute el comando show ip bgp regexp regular-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 Filtrado de Trayectorias, usted especificó la cadena ^200$. Quería que la información de trayectoria que viene dentro de 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 carácter único.

^

^ coincide con el comienzo de la cadena de entrada.

$

$ coincide con el final de la cadena de entrada.

\

\ coincide con el carácter.

-

_ coincide con una coma (,), la llave de apertura ({), la llave de cierre (}), el comienzo de la cadena de entrada, el final de la cadena de entrada o un espacio.

Pedazo

Una parte 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 una cadena nula.

Bifurcación

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.

Consulte Uso de Expresiones Regulares en BGP para obtener configuraciones de ejemplo del filtrado de expresiones regulares.

Filtrado de Comunidades de 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 Comunidad, se analiza la comunidad y, en esta sección, se proporcionan algunos ejemplos de cómo utilizar una 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 comunidad no-export**.

```
RTB#
router bgp 200
network 160.10.0.0
neighbor 3.3.3.1 remote-as 300
neighbor 3.3.3.1 send-community
neighbor 3.3.3.1 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 **comando route-map setcommunity** para configurar la comunidad en **no-export**.

Nota: El **comando neighbor send-community** 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 configurado el atributo de comunidad en **100 200 additive**. Esta acción agrega el valor 100 200 a cualquier valor de comunidad existente antes de la transmisión al RTC.

```
RTB#
   router bgp 200
   network 160.10.0.0
   neighbor 3.3.3.1 remote-as 300
   neighbor 3.3.3.1 send-community
   neighbor 3.3.3.1 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 3.3.3.3 remote-as 200
neighbor 3.3.3.3 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.

Consulte Uso de Valores de Comunidad de BGP para Controlar las Políticas de Ruteo en una Red de Proveedor de Flujo Ascendente para obtener más información.

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 170.10.0.0
   neighbor 3.3.3.3 remote-as 200
   neighbor 3.3.3.3 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 170.10.0.0
neighbor 3.3.3.3 remote-as 200
neighbor 3.3.3.3 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. La declaración también configura un peso de 10 para las actualizaciones que están detrás del AS400 y descarta las actualizaciones que vienen del 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 Mapas de Rutas y Vecinos BGP, el RTC anuncia su propia red 170.10.0.0 a dos AS diferentes, AS100 y AS200. Cuando la información se propaga al AS600, el Routers en el AS600 tiene información sobre el alcance de la red sobre 170.10.0.0 vía dos diversas rutas. 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 trayectoria existente que se anuncia al AS100. Una práctica común es repetir su propio número de AS de esta manera:

```
RTC#
router bgp 300
network 170.10.0.0
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-map SETPATH out

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

Debido a esta configuración, el AS600 recibe las actualizaciones por 170.10.0.0 vía el AS100 con la información de trayectoria 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. Usted no define las mismas políticas para cada vecino por separado; en su lugar, define un nombre de grupo de peers y asigna estas políticas al grupo de peers.

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 5.5.5.2 peer-group internalmap
   neighbor 5.6.6.2 peer-group internalmap
   neighbor 3.3.3.2 peer-group internalmap
   neighbor 3.3.3.2 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: Usted solo puede invalidar opciones que afecten las actualizaciones entrantes.

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 2.2.2.2 remote-as 100
   neighbor 2.2.2.2 peer-group externalmap
   neighbor 4.4.4.2 remote-as 600
   neighbor 4.4.4.2 peer-group externalmap
   neighbor 1.1.1.2 remote-as 200
   neighbor 1.1.1.2 peer-group externalmap
   neighbor 1.1.1.2 filter-list 3 in
```

Nota: En estas configuraciones, usted define las declaraciones **remote-as** fuera del grupo de peers porque debe definir AS externos diferentes. Además, invalida las actualizaciones entrantes del vecino 1.1.1.2 con la asignación de la lista de filtros 3.

Para obtener más información sobre los grupos de peers, consulte Grupos de Peers BGP.

Nota: En el Cisco IOS Software, versión 12.0(24)S, Cisco introdujo la función de grupos de peers de actualizaciones dinámicas de BGP. Esta función también está disponible en las versiones posteriores del Cisco IOS Software. La función introduce un nuevo algoritmo que calcula dinámicamente y optimiza los grupos de actualizaciones de vecinos que compartan las mismas políticas de salida. Estos vecinos pueden compartir los mismos mensajes de actualización. En las versiones anteriores del Cisco IOS Software, el grupo de mensajes de actualización de BGP se basaba en configuraciones de grupo de peers. Este método para agrupar las actualizaciones limitaba las políticas de salida y las configuraciones de sesión específicas. La función de grupos de peers de actualizaciones dinámicas de BGP separa la réplica del grupo de actualizaciones de la configuración de grupo de peers. Esta separación mejora el tiempo de convergencia y la flexibilidad de la configuración de vecinos. Consulte Grupos de Peers de Actualizaciones Dinámicas de BGP para obtener más detalles.

Caso Práctico de BGP 4

CIDR y Direcciones Agregadas

Una de las mejoras principales del BGP4 sobre BGP3 es el ruteo de interdominios sin clase (CIDR). El CIDR, o las superedes, es una nueva manera de observar las direcciones IP. Con el CIDR, no hay noción de las clases, como clase A, B o C. Por ejemplo, la red 192.213.0.0 era una vez una red clase C ilegal. Ahora, la red es una supered legal, 192.213.0.0/16. El dato "16" representa el número de bits en la máscara de subred, si cuenta del extremo izquierdo de la dirección IP. Esta representación es similar a 192.213.0.0 255.255.0.0.

Usted utiliza los agregados para minimizar el tamaño de las tablas de ruteo. La agregación es el proceso que combina las características de varias rutas diferentes de tal manera que es posible el anuncio de una sola ruta. En este ejemplo, el RTB genera la red 160.10.0.0. Usted configura el RTC para propagar una supered de esa ruta 160.0.0.0 al RTA:

```
RTB#
router bgp 200
neighbor 3.3.3.1 remote-as 300
network 160.10.0.0

#RTC
router bgp 300
neighbor 3.3.3.3 remote-as 200
neighbor 2.2.2.2 remote-as 100
network 170.10.0.0
aggregate-address 160.0.0.0 255.0.0.0
```

El RTC propaga la dirección agregada 160.0.0.0 al RTA.

Comandos de Agregado

Hay una amplia gama de comandos de agregado. Debe comprender cómo funciona cada uno para obtener el comportamiento de agregación que desea.

El primer comando es el que está en el ejemplo de la sección CIDR y Direcciones Agregadas:

aggregate-address *address-mask*

Este comando anuncia la ruta de prefijo y todas las rutas más específicas. El comando aggregate-address 160.0.0.0 propaga una red adicional 160.0.0.0, pero no previene la propagación de 160.10.0.0 al RTA. El resultado es la propagación de las redes 160.0.0.0 y 160.10.0.0 al RTA, que es el anuncio tanto de la ruta de prefijo como de la ruta más específica.

Nota: Usted no puede agregar una dirección si no tiene una ruta más específica de esa dirección en la tabla de ruteo BGP.

Por ejemplo, el RTB no puede generar un agregado para 160.0.0.0 si el RTB no tiene una entrada más específica de 160.0.0.0 en la tabla de BGP. Es posible una inserción de la ruta más específica en la tabla de BGP. La inserción de la ruta puede ocurrir vía:

- Actualizaciones entrantes de otros AS
- Redistribución de un IGP o estática en BGP
- El **comando network**, por ejemplo, **network 160.10.0.0**

Si usted desea que el RTC propague la red 160.0.0.0 solamente y **no** la ruta más específica, ejecute este comando:

aggregate-address *address mask* **summary-only**
Este comando anuncia el prefijo solamente. El comando omite todas las rutas más específicas.

El comando aggregate 160.0.0.0 255.0.0.0 summary-only propaga la red 160.0.0.0 y omite la ruta más específica 160.10.0.0.

Nota: Si usted agrega una red que insertó en su BGP vía la **declaración network**, la entrada de red siempre se inserta en las actualizaciones de BGP. Esta inserción ocurre aunque usted utilice el **comando aggregate summary-only**. En el ejemplo de la sección Ejemplo de CIDR 1, se analiza esta situación.

aggregate-address *address-mask* **as-set**
Este comando anuncia el prefijo y las rutas más específicas. Pero el comando incluye la información **as-set** en la información de trayectoria de las actualizaciones de ruteo.

aggregate 129.0.0.0 255.0.0.0 as-set
En la sección Ejemplo de CIDR 2 (as-set), se analiza este comando.

Si usted desea omitir las rutas más específicas cuando realiza la agregación, defina un mapa de ruta y aplique el mapa de ruta a los agregados. La acción le permite elegir qué rutas más específicas se omitirán.

aggregate-address *address-mask* **suppress-map** *map-name*
Este comando anuncia el prefijo y las rutas más específicas. Pero el comando omite el anuncio con una base de mapa de ruta. Suponga que, con el diagrama de la sección CIDR y Direcciones Agregadas, usted desea agregar 160.0.0.0, omitir la ruta más específica 160.20.0.0 y permitir la propagación de 160.10.0.0. Utilice este mapa de ruta:

```
route-map CHECK permit 10
match ip address 1

access-list 1 permit 160.20.0.0 0.0.255.255
access-list 1 deny 0.0.0.0 255.255.255.255
```

Por definición de **suppress-map**, hay una omisión de las actualizaciones de todos los paquetes que permite la lista de acceso.

Por lo tanto, aplique el mapa de ruta a la **declaración aggregate**.

```
RTC#
router bgp 300
neighbor 3.3.3.3 remote-as 200
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 remote-as 100
network 170.10.0.0
aggregate-address 160.0.0.0 255.0.0.0 suppress-map CHECK
```

Esta es otra variación:

aggregate-address *address-mask* **attribute-map** *map-name*
Este comando le permite configurar los atributos, como métrica, a la hora del envío de los agregados. Para configurar el origen de los agregados al IGP, aplique este mapa de ruta al **comando aggregate attribute-map**:

```
route-map SETMETRIC
set origin igp

aggregate-address 160.0.0.0 255.0.0.0 attribute-map SETORIGIN
```

Para obtener más información, consulte Comprensión de la Agregación de Rutas en BGP.

Ejemplo de CIDR 1

Petición: Permitir que el RTB anuncie el prefijo 160.0.0.0 y omita todas las rutas más específicas. El problema con esta solicitud es que la red 160.10.0.0 es local para el AS200, lo que significa que el AS200 es el creador de 160.10.0.0. Usted no puede hacer que el RTB genere un prefijo para 160.0.0.0 sin la generación de una entrada para 160.10.0.0, incluso si utiliza el **comando aggregate summary-only**. El RTB genera ambas redes porque el RTB es el creador de 160.10.0.0. Hay dos soluciones para este problema.

La primera solución es utilizar una ruta estática y redistribuirla en BGP. El resultado es que el RTB anuncia el agregado con un origen de incompleto (?).

```
RTB#
router bgp 200
neighbor 3.3.3.1 remote-as 300
redistribute static
!--- This generates an update for 160.0.0.0 !--- with the origin path as "incomplete". ip route
160.0.0.0 255.0.0.0 null0
```

En la segunda solución, además de la ruta estática, usted agrega una entrada para el **comando network**. Esta entrada tiene el mismo efecto, excepto que la entrada configura el origen de la actualización en IGP.

```
RTB#
router bgp 200
network 160.0.0.0 mask 255.0.0.0
!--- This entry marks the update with origin IGP. neighbor 3.3.3.1 remote-as 300 redistribute
static ip route 160.0.0.0 255.0.0.0 null0
```

Ejemplo de CIDR 2 (as-set)

Usted utiliza la declaración **as-set** en la agregación para reducir el tamaño de la información de trayectoria. Con **as-set**, el número de AS se enumera solamente una vez, sin importar cuántas veces apareció el número de AS en las diversas trayectorias que se agregaron. Usted utiliza el **comando aggregate as-set** en las situaciones en que la agregación de la información causa la pérdida de información con respecto al atributo de trayectoria. En este ejemplo, el RTC obtiene actualizaciones por 160.20.0.0 del RTA y actualizaciones por 160.10.0.0 del RTB. Suponga que el RTC desea agregar la red 160.0.0.0/8 y enviar la red al RTD. El RTD no conoce el origen de esa ruta. Si usted agrega la **declaración aggregate as-set**, fuerza al RTC a generar la información de trayectoria en la forma de un conjunto {}. Ese conjunto incluye toda la información de trayectoria, independientemente de qué trayectoria vino primero.

```
RTB#
```

```
router bgp 200
network 160.10.0.0
neighbor 3.3.3.1 remote-as 300

RTA#
router bgp 100
network 160.20.0.0
neighbor 2.2.2.1 remote-as 300
```

Caso 1:

El RTC no tiene una declaración **as-set**. El RTC envía una actualización 160.0.0.0/8 al RTD con la información de trayectoria (300), como si la ruta se originara desde el AS300.

```
RTC#
router bgp 300
neighbor 3.3.3.3 remote-as 200
neighbor 2.2.2.2 remote-as 100
neighbor 4.4.4.4 remote-as 400
aggregate 160.0.0.0 255.0.0.0 summary-only
!--- This command causes RTC to send RTD updates about 160.0.0.0/8 !--- with no indication that
160.0.0.0 actually comes from two different ASs. !--- This may create loops if RTD has an entry
back into AS100 or AS200.
```

Caso 2:

```
RTC#
   router bgp 300
   neighbor 3.3.3.3 remote-as 200
   neighbor 2.2.2.2 remote-as 100
   neighbor 4.4.4.4 remote-as 400
   aggregate 160.0.0.0 255.0.0.0 summary-only
   aggregate 160.0.0.0 255.0.0.0 as-set
!--- This command causes RTC to send RTD updates about 160.0.0.0/8 !--- with an indication that
160.0.0.0 belongs to a set {100 200}.
```

Los dos temas siguientes, Confederación de BGP y Reflectores de Ruta, son para proveedores de servicios de Internet (ISP) que deseen mayor control de la explosión del peering de iBGP dentro de sus AS.

Confederación de BGP

La implementación de la confederación de BGP reduce la malla de iBGP dentro de un AS. El truco es dividir un AS en varios AS y asignar el grupo entero a una sola confederación. Cada AS independiente tiene una malla completa de iBGP y tiene conexiones a otros AS dentro de la confederación. Aunque estos AS tienen peers eBGP a los AS dentro de la confederación, los AS intercambian el ruteo como si utilizaran iBGP. De esta manera, la confederación preserva el salto siguiente, la métrica y la información de preferencia local. Para el mundo exterior, la confederación parece un solo AS.

Para configurar una confederación de BGP, ejecute este comando:

bgp confederation identifier *autonomous-system*

El identificador de confederación es el número de AS del grupo de la confederación.

La ejecución este comando realiza peering entre varios AS dentro de la confederación:

bgp confederation peers autonomous-system [*autonomous-system***]**
Este es un ejemplo de una confederación:

Suponga que usted tiene un AS500 que está compuesto de nueve altavoces BGP. También hay otros altavoces que no son BGP, pero usted solo está interesado en los altavoces BGP que tienen conexiones eBGP a otros AS. Si usted desea crear una malla completa de iBGP dentro del AS500, necesita nueve conexiones de peers para cada router. Necesita ocho peers iBGP y un peer eBGP a AS externos.

Si utiliza una confederación, puede dividir el AS500 en varios AS: AS50, AS60 y AS70. Usted le da al AS un identificador de confederación de 500. El mundo exterior verá solo un AS, AS500. Para cada AS (AS50, AS60 y AS70), defina una malla completa de peers iBGP y defina la lista de peers de confederación con el **comando bgp confederation peers**.

Esta es una configuración de ejemplo de los routers RTC, RTD y RTA:

Nota: El RTA no tiene conocimiento del AS50, AS60 ni AS70. El RTA solo tiene conocimiento del AS500.

```
RTC#
router bgp 50
bgp confederation identifier 500
bgp confederation peers 60 70
neighbor 128.213.10.1 remote-as 50 (IBGP connection within AS50)
neighbor 128.213.20.1 remote-as 50 (IBGP connection within AS50)
neighbor 129.210.11.1 remote-as 60 (BGP connection with confederation peer 60)
neighbor 135.212.14.1 remote-as 70 (BGP connection with confederation peer 70)
neighbor 5.5.5.5 remote-as 100 (EBGP connection to external AS100)

RTD#
router bgp 60
bgp confederation identifier 500
bgp confederation peers 50 70
neighbor 129.210.30.2 remote-as 60 (IBGP connection within AS60)
neighbor 128.213.30.1 remote-as 50(BGP connection with confederation peer 50)
neighbor 135.212.14.1 remote-as 70 (BGP connection with confederation peer 70)
neighbor 6.6.6.6 remote-as 600 (EBGP connection to external AS600)

RTA#
   router bgp 100
   neighbor 5.5.5.4 remote-as 500 (EBGP connection to confederation 500)
```

Reflectores de Ruta

Otra solución para la explosión del peering de iBGP dentro de un AS es el uso de reflectores de ruta (RR). Como se muestra en la sección iBGP, un altavoz BGP no anunciará una ruta que este detecte vía otro altavoz iBGP a un tercer altavoz iBGP. Usted puede disminuir esta restricción un poco y proporcionar control adicional, que permite que un router anuncie, o refleje, las rutas detectadas iBGP a otros altavoces iBGP. Esta reflexión de rutas reduce el número de peers iBGP dentro de un AS.

En los casos normales, mantenga una malla completa de iBGP entre el RTA, el RTB y el RTC dentro del AS100. Si usted utiliza el concepto de RR, el RTC se puede elegir como un RR. De esta manera, el RTC tiene un peering de iBGP parcial con el RTA y el RTB. El peering entre el

RTA y el RTB no es necesario porque el RTC es un RR para las actualizaciones que vienen del RTA y del RTB.

neighbor route-reflector-client

El router con este comando es el RR y los vecinos a los que apunta el comando son los clientes de ese RR. En el ejemplo, la configuración de RTC tiene el **comando neighbor route-reflector-client** que apunta a las direcciones IP de RTA y RTB. La combinación del RR y de los clientes es un "clúster". En este ejemplo, el RTA, el RTB y el RTC forman un clúster con un solo RR dentro del AS100.

Los otros peers iBGP del RR que no son clientes son "no clientes".

Un AS puede tener más de un RR. En esta situación, un RR trata a los otros RR simplemente como cualquier otro altavoz iBGP. Los otros RR pueden pertenecer al mismo clúster (grupo de clientes) o a otros clústeres. En una configuración simple, usted puede dividir el AS en varios clústeres. Configura cada RR con los otros RR como peers no clientes en una topología completamente mallada. Los clientes no deben hacer peer con altavoces iBGP fuera del clúster de clientes.

Considere este diagrama. El RTA, el RTB y el RTC forman un solo clúster. El RTC es el RR. Para el RTC, el RTA y el RTB son clientes y cualquier otra cosa es un no cliente. Recuerde que el **comando neighbor route-reflector-client** apunta a los clientes de un RR. El mismo RTD es el RR para los clientes RTE y RTF. El RTG es un RR en un tercer clúster.

Nota: El RTD, el RTC y el RTG tienen una malla completa, pero los routers dentro de un clúster no. Cuando un RR recibe una ruta, el RR rutea como se muestra en esta lista. Sin embargo, esta actividad depende del tipo de peer:

Rutea de un peer no cliente: refleja a todos los clientes dentro del clúster.

Rutea de un peer cliente: refleja a todos los peers no clientes y también a los peers clientes.

Rutea de un peer eBGP: envía la actualización a todos los peers clientes y no clientes.

Esta es la configuración de BGP relativa de los routers RTC, RTD y RTB:

```
RTC#

router bgp 100
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-reflector-client
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 route-reflector-client
neighbor 7.7.7.7 remote-as 100
neighbor 4.4.4.4 remote-as 100
neighbor 8.8.8.8 remote-as 200


RTB#
```

```
router bgp 100
neighbor 3.3.3.3 remote-as 100
neighbor 12.12.12.12 remote-as 300


RTD#

router bgp 100
neighbor 6.6.6.6 remote-as 100
neighbor 6.6.6.6 route-reflector-client
neighbor 5.5.5.5 remote-as 100
neighbor 5.5.5.5 route-reflector-client
neighbor 7.7.7.7 remote-as 100
neighbor 3.3.3.3 remote-as 100
```

Debido a que hay una reflexión de las rutas detectadas iBGP, puede haber un loop de información de ruteo. El esquema de RR tiene algunos métodos para evitar este loop:

originator-id: este es un atributo de BGP opcional no transitivo que tiene 4 bytes. Un RR crea este atributo. El atributo lleva el ID de router (RID) del creador de la ruta en el AS local. Si, debido a una mala configuración, la información de ruteo regresa al creador, se ignora la información.

cluster-list: la sección Varios RR Dentro de un Clúster cubre la lista de clústeres.

Varios RR Dentro de un Clúster

Generalmente, un clúster de clientes tiene un solo RR. En este caso, el ID de router del RR identifica el clúster. Para aumentar la redundancia y evitar puntos únicos de falla, un clúster puede tener más de un RR. Usted debe configurar todos los RR en el mismo clúster con un ID de clúster de 4 bytes, de modo que un RR pueda reconocer las actualizaciones de los RR en el mismo clúster.

Una lista de clústeres es una secuencia de ID de clúster que la ruta ha pasado. Cuando un RR refleja una ruta de los clientes de RR a los no clientes fuera del clúster, el RR agrega el ID de clúster local a la lista de clústeres. Si esta actualización tiene una lista de clústeres vacía, el RR crea una. Con este atributo, un RR puede identificar si la información de ruteo tiene un loop de regreso al mismo clúster debido a una mala configuración. Si el ID de clúster local se encuentra en la lista de clústeres, se ignora el anuncio.

En el diagrama de esta sección, el RTD, el RTE, el RTF y el RTH pertenecen a un clúster. El RTD y el RTH son RR para el mismo clúster.

Nota: Hay redundancia porque el RTH tiene peering completamente mallado con todos los RR. Si el RTD baja, el RTH toma el lugar del RTD.

Esta es la configuración del RTH, RTD, RTF y RTC:

```
RTH#

router bgp 100
neighbor 4.4.4.4 remote-as 100
```

```
neighbor 5.5.5.5 remote-as 100
neighbor 5.5.5.5 route-reflector-client
neighbor 6.6.6.6 remote-as 100
neighbor 6.6.6.6 route-reflector-client
neighbor 7.7.7.7 remote-as 100
neighbor 3.3.3.3 remote-as 100
neighbor 9.9.9.9 remote-as 300
bgp cluster-id 10


RTD#

router bgp 100
neighbor 10.10.10.10 remote-as 100
neighbor 5.5.5.5 remote-as 100
neighbor 5.5.5.5 route-reflector-client
neighbor 6.6.6.6 remote-as 100
neighbor 6.6.6.6 route-reflector-client
neighbor 7.7.7.7 remote-as 100
neighbor 3.3.3.3 remote-as 100
neighbor 11.11.11.11 remote-as 400
bgp cluster-id 10


RTF#

router bgp 100
neighbor 10.10.10.10 remote-as 100
neighbor 4.4.4.4 remote-as 100
neighbor 13.13.13.13 remote-as 500


RTC#

router bgp 100
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 route-reflector-client
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-reflector-client
neighbor 4.4.4.4 remote-as 100
neighbor 7.7.7.7 remote-as 100
neighbor 10.10.10.10 remote-as 100
neighbor 8.8.8.8 remote-as 200
```

Nota: Usted no necesita el comando bgp cluster-id para el RTC porque existe solo un RR en ese clúster.

Nota importante: Esta configuración no utiliza grupos de peers. No utilice grupos de peers si los clientes dentro de un clúster no tienen peers iBGP directos entre sí y los clientes intercambian actualizaciones a través del RR. Si usted configura grupos de peers, una posible retirada al origen de una ruta en el RR se transmite a todos los clientes dentro del clúster. Esta transmisión puede causar problemas.

El subcomando de router bgp client-to-client reflection se habilita de forma predeterminada en el

RR. Si usted desactiva la reflexión cliente a cliente de BGP en el RR y realiza peering de BGP redundante entre los clientes, puede utilizar con seguridad los grupos de peers. Refiera a las limitaciones de los grupos de peer para más información.

RR y Altavoces BGP Convencionales

Un AS puede tener altavoces BGP que no comprendan el concepto de RR. En este documento, a estos routers se los llama altavoces BGP convencionales. El esquema de RR permite que dichos altavoces BGP convencionales coexistan. Estos routers pueden ser miembros de un grupo de clientes o un grupo de no clientes. La existencia de estos routers permite una migración fácil y gradual del modelo actual de iBGP al modelo de RR. Usted puede comenzar a crear clústeres si configura un único router como RR y hace que los otros RR y clientes de RR sean peers iBGP normales. Luego, puede crear más clústeres gradualmente.

En este diagrama, el RTD, el RTE y el RTF tienen el concepto de reflexión de ruta. El RTC, el RTA y el RTB son routers "convencionales". Usted no puede configurar estos routers como RR. Puede crear una malla normal de iBGP entre estos routers y el RTD. Después, cuando usted esté listo para una actualización, puede hacer que el RTC sea un RR con los clientes RTA y RTB. Los clientes no tienen que comprender el esquema de reflexión de ruta; solamente los RR requieren la actualización.

Esta es la configuración del RTD y RTC:

```
RTD#

router bgp 100
neighbor 6.6.6.6 remote-as 100
neighbor 6.6.6.6 route-reflector-client
neighbor 5.5.5.5 remote-as 100
neighbor 5.5.5.5 route-reflector-client
neighbor 3.3.3.3 remote-as 100
neighbor 2.2.2.2 remote-as 100
neighbor 1.1.1.1 remote-as 100
neighbor 13.13.13.13 remote-as 300


RTC#

router bgp 100
neighbor 4.4.4.4 remote-as 100
neighbor 2.2.2.2 remote-as 100
neighbor 1.1.1.1 remote-as 100
neighbor 14.14.14.14 remote-as 400
```

Cuando usted esté listo para actualizar el RTC y hacer que el RTC sea un RR, quite la malla completa de iBGP y el RTA y el RTB se convertirán en clientes del RTC.

Cómo Evitar un Loop de la Información de Ruteo

Hasta ahora, en este documento, se han mencionado dos atributos que usted puede utilizar para prevenir los posibles loops de información: **originator-id** y **cluster-list**.

Otra manera de controlar los loops es colocar más restricciones en la cláusula **set** de los mapas de ruta salientes. La cláusula **set** para los mapas de ruta salientes no afecta las rutas que se

reflejan a los peers iBGP.

Usted también puede colocar más restricciones en **nexthop-self**, que es una opción de configuración por vecino. Cuando usted utiliza **nexthop-self** en los RR, la cláusula afecta solamente el salto siguiente de las rutas detectas eBGP porque el salto siguiente de las rutas reflejadas no se debe modificar.

Dampening de Inestabilidad de Ruta

En el Cisco IOS Software, versión 11.0, se introdujo el dampening de ruta. El dampening de ruta es un mecanismo para minimizar la inestabilidad de una ruta. También reduce la oscilación en la red. Usted define criterios para identificar rutas que tengan un mal comportamiento. Una ruta que tiene inestabilidad obtiene una penalización de 1000 por cada caso de inestabilidad. Ni bien la penalización acumulativa alcance un "limite de omisión" predefinido, ocurrirá la omisión del anuncio de la ruta. La penalización decae de manera exponencial según un "tiempo de media duración" preconfigurado. Una vez que la penalización haya decrecido por debajo de un "límite de reutilización" predefinido, ocurrirá la anulación de la omisión del anuncio de la ruta.

El dampening de ruta no se aplica a rutas que sean externas a un AS y se detecten vía iBGP. De esta manera, el dampening de ruta evita una penalización más alta para los peers iBGP por rutas externas al AS.

La penalización decae a una granularidad de 5 segundos. La anulación de la omisión de las rutas es a una granularidad de 10 segundos. El router conserva la información de dampening hasta que la penalización se convierta en menos que la mitad del "límite de reutilización". En ese punto, el router purga la información.

Inicialmente, el dampening está desactivado de forma predeterminada. Si hay una necesidad, se puede proporcionar una habilitación predeterminada a esta función en el futuro. Estos comandos controlan el dampening de la ruta:

bgp dampening: activa el dampening.

no bgp dampening: desactiva el dampening.

bgp dampening ***half-life-time***: cambia el tiempo de media duración.

Un comando que configura todos los parámetros al mismo tiempo es:

bgp dampening ***half-life-time reuse suppress maximum-suppress-time***.

Esta lista detalla la sintaxis:

half-life-time: el rango es de 1 a 45 minutos y el valor predeterminado actual es 15 minutos.

reuse-value: el rango es de 1 a 20.000 y el valor predeterminado es 750.

suppress-value: el rango es de 1 a 20.000 y el valor predeterminado es 2000.

max-suppress-time: esta es la duración máxima para la omisión de una ruta. El rango es de 1 a 255 minutos y el valor predeterminado es 4 veces el tiempo de duración media.

```
RTB#
hostname RTB

interface Serial0
 ip address 203.250.15.2 255.255.255.252

interface Serial1
 ip address 192.208.10.6 255.255.255.252

router bgp 100
 bgp dampening
 network 203.250.15.0
 neighbor 192.208.10.5 remote-as 300



RTD#
hostname RTD

interface Loopback0
ip address 192.208.10.174 255.255.255.192

interface Serial0/0
 ip address 192.208.10.5 255.255.255.252

router bgp 300
 network 192.208.10.0
 neighbor 192.208.10.6 remote-as 100
```

La configuración de RTB es para el dampening de ruta con los parámetros predeterminados. Si usted supone que el link de eBGP al RTD es estable, la tabla de BGP del RTB es similar a lo siguiente:

```
RTB# show ip bgp BGP table version is 24, local router ID is 203.250.15.2 Status codes: s
suppressed, d damped, h history, * valid, > best, i - internal Origin codes: i - IGP, e - EGP, ?
- incomplete Network Next Hop Metric LocPrf Weight Path *> 192.208.10.0 192.208.10.5 0 0 300 i
*> 203.250.15.0 0.0.0.0 0 32768 i
```

Para simular un caso de inestabilidad de ruta, ejecute el comando clear ip bgp 192.208.10.6 en el RTD. La tabla de BGP del RTB es similar a lo siguiente:

```
RTB# show ip bgp BGP table version is 24, local router ID is 203.250.15.2 Status codes: s
suppressed, d damped, h history, * valid, > best, i - internal Origin codes: i - IGP, e - EGP, ?
- incomplete Network Next Hop Metric LocPrf Weight Path h 192.208.10.0 192.208.10.5 0 0 300 i *>
203.250.15.0 0.0.0.0 0 32768 i
```

La entrada de BGP para 192.208.10.0 está en estado history. Esta colocación significa que usted no tiene una mejor trayectoria a la ruta, pero que la información sobre la inestabilidad de la ruta todavía existe.

```
RTB# show ip bgp 192.208.10.0 BGP routing table entry for 192.208.10.0 255.255.255.0, version 25
Paths: (1 available, no best path) 300 (history entry) 192.208.10.5 from 192.208.10.5
(192.208.10.174) Origin IGP, metric 0, external Dampinfo: penalty 910, flapped 1 times in
0:02:03
```

La ruta ha recibido una penalización por la inestabilidad, pero esta penalización todavía está por debajo del "límite de omisión". El valor predeterminado es 2000. La omisión de la ruta todavía no ha ocurrido. Si la ruta registra inestabilidad algunas veces más, usted verá lo siguiente:

```
RTB# show ip bgp BGP table version is 32, local router ID is 203.250.15.2 Status codes: s
suppressed, d damped, h history, * valid, > best, i - internal Origin codes: i - IGP, e - EGP, ?
- incomplete Network Next Hop Metric LocPrf Weight Path *d 192.208.10.0 192.208.10.5 0 0 300 i
*> 203.250.15.0 0.0.0.0 0 32768 i RTB# show ip bgp 192.208.10.0 BGP routing table entry for
192.208.10.0 255.255.255.0, version 32 Paths: (1 available, no best path) 300, (suppressed due
to dampening) 192.208.10.5 from 192.208.10.5 (192.208.10.174) Origin IGP, metric 0, valid,
external Dampinfo: penalty 2615, flapped 3 times in 0:05:18 , reuse in 0:27:00
```

Se ha registrado dampening para la ruta, o esta se ha omitido. La ruta se reutiliza cuando la penalización alcanza el "valor de reutilización". En este caso, el valor de reutilización es el valor predeterminado, 750. La información de dampening se purga cuando la penalización se convierte en menos que la mitad del límite de reutilización. En este caso, la purga ocurre cuando la penalización se convierte en 375 (750 / 2 = 375). Estos comandos muestran y borran información de estadísticas de inestabilidad:

show ip bgp flap-statistics: muestra estadísticas de inestabilidad para todas las trayectorias.

show ip bgp flap-statistics regexp ***regular-expression***: muestra estadísticas de inestabilidad para todas las trayectorias que coincidan con la expresión regular.

show ip bgp flap-statistics filter-list list: muestra estadísticas de inestabilidad para todas las trayectorias que pasen el filtro.

show ip bgp flap-statistics ***A.B.C.D m.m.m.m: muestra*** estadísticas de inestabilidad para una sola entrada.

show ip bgp flap-statistics ***A.B.C.D m.m.m.m longer-prefix*****: muestra** estadísticas de inestabilidad para más entradas específicas.

show ip bgp neighbor [dampened-routes] | [[flap-statistics]: muestra estadísticas de inestabilidad para todas las trayectorias de un vecino.

clear ip bgp flap-statistics: borra estadísticas de inestabilidad para todas las rutas.

clear ip bgp flap-statistics regexp ***regular-expression***: borra estadísticas de inestabilidad para todas las trayectorias que coincidan con la expresión regular.

clear ip bgp flap-statistics filter-list list: borra estadísticas de inestabilidad para todas las trayectorias que pasen el filtro.

clear ip bgp flap-statistics *A.B.C.D m.m.m.m: borra* estadísticas de inestabilidad para una sola entrada.

clear ip bgp *A.B.C.D flap-statistics*: **borra** estadísticas de inestabilidad para todas las trayectorias de un vecino.

Cómo BGP Selecciona una Trayectoria

Ahora que usted está familiarizado con los atributos de BGP y la terminología, consulte Algoritmo de Selección de la Mejor Trayectoria de BGP.

Caso Práctico de BGP 5

Ejemplo de Diseño Práctico

Esta sección contiene un ejemplo de diseño donde se muestran las tablas de ruteo y configuración como aparecen realmente las tablas en los routers de Cisco.

En esta sección, se muestra cómo crear esta configuración paso a paso y qué puede salir mal a lo largo del camino. Cada vez que usted tenga un AS que conecte a dos ISP vía eBGP, ejecute siempre iBGP dentro de su AS para tener mejor control de sus rutas. En este ejemplo, iBGP se ejecuta dentro del AS100 entre el RTA y el RTB, y OSPF se ejecuta como IGP. Suponga que usted se conecta a dos ISP, AS200 y AS300. Esta es la primera ejecución de las configuraciones para todos los routers:

Nota: Estas configuraciones no son las configuraciones finales.

```
RTA#
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 203.250.13.41 255.255.255.0

interface Ethernet0
ip address 203.250.14.1 255.255.255.0

interface Serial0
 ip address 128.213.63.1 255.255.255.252

router ospf 10
 network 203.250.0.0 0.0.255.255 area 0

router bgp 100
 network 203.250.13.0
 network 203.250.14.0
 neighbor 128.213.63.2 remote-as 200
 neighbor 203.250.15.2 remote-as 100
 neighbor 203.250.15.2 update-source Loopback0
```

```
RTF#
hostname RTF

ip subnet-zero

interface Ethernet0
 ip address 203.250.14.2 255.255.255.0

interface Serial1
 ip address 203.250.15.1 255.255.255.252

router ospf 10
 network 203.250.0.0 0.0.255.255 area 0

RTB#
hostname RTB

ip subnet-zero

interface Serial0
 ip address 203.250.15.2 255.255.255.252

interface Serial1
 ip address 192.208.10.6 255.255.255.252

router ospf 10
 network 203.250.0.0 0.0.255.255 area 0

router bgp 100
network 203.250.15.0
 neighbor 192.208.10.5 remote-as 300
 neighbor 203.250.13.41 remote-as 100

RTC#
hostname RTC

ip subnet-zero

interface Loopback0
 ip address 128.213.63.130 255.255.255.192

interface Serial2/0
 ip address 128.213.63.5 255.255.255.252
!
interface Serial2/1
 ip address 128.213.63.2 255.255.255.252

router bgp 200
 network 128.213.0.0
 neighbor 128.213.63.1 remote-as 100
 neighbor 128.213.63.6 remote-as 400

RTD#
```

```
hostname RTD

ip subnet-zero

interface Loopback0
ip address 192.208.10.174 255.255.255.192

interface Serial0/0
 ip address 192.208.10.5 255.255.255.252
!
interface Serial0/1
 ip address 192.208.10.2 255.255.255.252

router bgp 300
 network 192.208.10.0
 neighbor 192.208.10.1 remote-as 500
 neighbor 192.208.10.6 remote-as 100

RTE#
hostname RTE

ip subnet-zero

interface Loopback0
ip address 200.200.10.1 255.255.255.0

interface Serial0
 ip address 195.211.10.2 255.255.255.252

interface Serial1
 ip address 128.213.63.6 255.255.255.252
 clockrate 1000000

router bgp 400
 network 200.200.10.0
 neighbor 128.213.63.5 remote-as 200
 neighbor 195.211.10.1 remote-as 500

RTG#
hostname RTG

ip subnet-zero

interface Loopback0
 ip address 195.211.10.174 255.255.255.192

interface Serial0
 ip address 192.208.10.1 255.255.255.252

interface Serial1
 ip address 195.211.10.1 255.255.255.252

router bgp 500
```

```
 network 195.211.10.0
 neighbor 192.208.10.2 remote-as 300
 neighbor 195.211.10.2 remote-as 400
```

Siempre utilice el **comando network** o redistribuya entradas estáticas en BGP para anunciar redes. Este método es mejor que una redistribución de IGP en BGP. Este ejemplo utiliza el **comando network** para insertar redes en BGP.

Aquí, usted comienza con la interfaz s1 en el apagado de RTB, como si no existiera el link entre el RTB y el RTD. Esta es la tabla de BGP del RTB:

```
RTB# show ip bgp BGP table version is 4, local router ID is 203.250.15.2 Status codes: s
suppressed, d damped, h history, * valid, > best, i - internal Origin codes: i - IGP, e - EGP, ?
- incomplete Network Next Hop Metric LocPrf Weight Path *i128.213.0.0 128.213.63.2 0 100 0 200 i
*i192.208.10.0 128.213.63.2 100 0 200 400 500 300 i *i195.211.10.0 128.213.63.2 100 0 200 400
500 i *i200.200.10.0 128.213.63.2 100 0 200 400 i *>i203.250.13.0 203.250.13.41 0 100 0 i
*>i203.250.14.0 203.250.13.41 0 100 0 i *>203.250.15.0 0.0.0.0 0 32768 i
```

En esta tabla, aparecen estas anotaciones:

Una i al comienzo indica que la entrada se detectó vía un peer iBGP.

Una i al final indica que el origen de la información de trayectoria es IGP.

Información de trayectoria: esta información es intuitiva. Por ejemplo, la red 128.213.0.0 se detecta vía la trayectoria 200 con un salto siguiente de 128.213.63.2.

Nota: Cualquier entrada generada localmente, como 203.250.15.0, tiene un salto siguiente de 0.0.0.0.

Un símbolo > indica que BGP ha elegido la mejor ruta. El BGP utiliza los pasos de decisión que se describen en el documento Algoritmo de Selección de la Mejor Trayectoria de BGP. El BGP selecciona una mejor trayectoria para alcanzar un destino, instala la trayectoria en la tabla de ruteo IP y anuncia la trayectoria a los otros peers BGP.

Nota: Observe el atributo de salto siguiente. El RTB sabe de 128.213.0.0 vía un salto siguiente de 128.213.63.2, que es el salto siguiente de eBGP que se lleva en iBGP.

Observe la tabla de ruteo IP:

```
RTB# show ip route Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP D -
EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area E1 - OSPF external type 1, E2 - OSPF
external type 2, E - EGP i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default Gateway of last resort is not set 203.250.13.0 255.255.255.255 is subnetted, 1 subnets O
203.250.13.41 [110/75] via 203.250.15.1, 02:50:45, Serial0 203.250.15.0 255.255.255.252 is
subnetted, 1 subnets C 203.250.15.0 is directly connected, Serial0 O 203.250.14.0 [110/74] via
203.250.15.1, 02:50:46, Serial0
```

Al parecer, ninguna de las entradas de BGP ha alcanzado la tabla de ruteo. Existen dos problemas aquí.

El primer problema es que el salto siguiente para estas entradas, 128.213.63.2, es inalcanzable.

No hay manera de alcanzar ese salto siguiente vía este IGP, que es OSPF. El RTB no ha detectado 128.213.63.0 vía OSPF. Usted puede ejecutar OSPF en la interfaz s0 del RTA y dejarlo pasivo; de esta manera, el RTB sabe cómo alcanzar el salto siguiente 128.213.63.2. Esta configuración de RTA aparece aquí:

```
RTA#
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 203.250.13.41 255.255.255.0

interface Ethernet0
ip address 203.250.14.1 255.255.255.0

interface Serial0
 ip address 128.213.63.1 255.255.255.252

router ospf 10
 passive-interface Serial0
 network 203.250.0.0 0.0.255.255 area 0
 network 128.213.0.0 0.0.255.255 area 0

router bgp 100
 network 203.250.0.0 mask 255.255.0.0
 neighbor 128.213.63.2 remote-as 200
 neighbor 203.250.15.2 remote-as 100
 neighbor 203.250.15.2 update-source Loopback0
```

Nota: Usted puede ejecutar el **comando bgp nexthopself** entre el RTA y el RTB para cambiar el salto siguiente.

La nueva tabla de BGP en el RTB es similar a lo siguiente:

```
RTB# show ip bgp BGP table version is 10, local router ID is 203.250.15.2 Status codes: s
suppressed, d damped, h history, * valid, > best, i - internal Origin codes: i - IGP, e - EGP, ?
- incomplete Network Next Hop Metric LocPrf Weight Path *>i128.213.0.0 128.213.63.2 0 100 0 200
i *>i192.208.10.0 128.213.63.2 100 0 200 400 500 300 i *>i195.211.10.0 128.213.63.2 100 0 200
400 500 i *>i200.200.10.0 128.213.63.2 100 0 200 400 i *>i203.250.13.0 203.250.13.41 0 100 0 i
*>i203.250.14.0 203.250.13.41 0 100 0 i *> 203.250.15.0 0.0.0.0 0 32768 i
```

Nota: Todas las entradas tienen **>**, lo que significa que el BGP puede alcanzar el salto siguiente.

Observe la tabla de ruteo:

```
RTB# show ip route Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP D -
EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area E1 - OSPF external type 1, E2 - OSPF
external type 2, E - EGP i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default Gateway of last resort is not set 203.250.13.0 255.255.255.255 is subnetted, 1 subnets O
203.250.13.41 [110/75] via 203.250.15.1, 00:04:46, Serial0 203.250.15.0 255.255.255.252 is
subnetted, 1 subnets C 203.250.15.0 is directly connected, Serial0 O 203.250.14.0 [110/74] via
203.250.15.1, 00:04:46, Serial0 128.213.0.0 255.255.255.252 is subnetted, 1 subnets O
128.213.63.0 [110/138] via 203.250.15.1, 00:04:47, Serial0
```

El segundo problema es que usted todavía no ve las entradas de BGP en la tabla de ruteo. La única diferencia es que 128.213.63.0 ahora es accesible vía OSPF. Esto es un problema de sincronización. El BGP no coloca estas entradas en la tabla de ruteo y no envía las entradas en las actualizaciones de BGP debido a una falta de sincronización con IGP.

Nota: El RTF no tiene noción de las redes 192.208.10.0 y 195.211.10.0 porque usted no ha redistribuido BGP en OSPF todavía.

En esta situación, si usted desactiva la sincronización, las entradas aparecen en la tabla de ruteo. Pero la conectividad aún está interrumpida.

Si usted desactiva la sincronización en el RTB, esto es lo que sucede:

```
RTB# show ip route Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP D -
EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area E1 - OSPF external type 1, E2 - OSPF
external type 2, E - EGP i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default Gateway of last resort is not set B 200.200.10.0 [200/0] via 128.213.63.2, 00:01:07 B
195.211.10.0 [200/0] via 128.213.63.2, 00:01:07 B 192.208.10.0 [200/0] via 128.213.63.2,
00:01:07 203.250.13.0 is variably subnetted, 2 subnets, 2 masks O 203.250.13.41 255.255.255.255
[110/75] via 203.250.15.1, 00:12:37, Serial0 B 203.250.13.0 255.255.255.0 [200/0] via
203.250.13.41, 00:01:08 203.250.15.0 255.255.255.252 is subnetted, 1 subnets C 203.250.15.0 is
directly connected, Serial0 O 203.250.14.0 [110/74] via 203.250.15.1, 00:12:37, Serial0
128.213.0.0 is variably subnetted, 2 subnets, 2 masks B 128.213.0.0 255.255.0.0 [200/0] via
128.213.63.2, 00:01:08 O 128.213.63.0 255.255.255.252 [110/138] via 203.250.15.1, 00:12:37,
Serial0
```

La tabla de ruteo parece correcta, pero no hay manera de alcanzar esas redes. El RTF en el medio no sabe cómo alcanzar las redes:

```
RTF# show ip route Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP D -
EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area E1 - OSPF external type 1, E2 - OSPF
external type 2, E - EGP i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default Gateway of last resort is not set 203.250.13.0 255.255.255.255 is subnetted, 1 subnets O
203.250.13.41 [110/11] via 203.250.14.1, 00:14:15, Ethernet0 203.250.15.0 255.255.255.252 is
subnetted, 1 subnets C 203.250.15.0 is directly connected, Serial1 C 203.250.14.0 is directly
connected, Ethernet0 128.213.0.0 255.255.255.252 is subnetted, 1 subnets O 128.213.63.0 [110/74]
via 203.250.14.1, 00:14:15, Ethernet0
```

Cuando usted desactiva la sincronización en esta situación, el problema todavía existe. Pero necesitará la sincronización más adelante para otros problemas. Redistribuya el BGP en OSPF en el RTA, con una métrica de 2000:

```
RTA#
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 203.250.13.41 255.255.255.0

interface Ethernet0
ip address 203.250.14.1 255.255.255.0

interface Serial0
```

```
 ip address 128.213.63.1 255.255.255.252

router ospf 10
 redistribute bgp 100 metric 2000 subnets
 passive-interface Serial0
 network 203.250.0.0 0.0.255.255 area 0
 network 128.213.0.0 0.0.255.255 area 0

router bgp 100
 network 203.250.0.0 mask 255.255.0.0
 neighbor 128.213.63.2 remote-as 200
 neighbor 203.250.15.2 remote-as 100
 neighbor 203.250.15.2 update-source Loopback0
```

La tabla de ruteo es similar a lo siguiente:

```
RTB# show ip route Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP D -
EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area E1 - OSPF external type 1, E2 - OSPF
external type 2, E - EGP i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default Gateway of last resort is not set O E2 200.200.10.0 [110/2000] via 203.250.15.1,
00:00:14, Serial0 O E2 195.211.10.0 [110/2000] via 203.250.15.1, 00:00:14, Serial0 O E2
192.208.10.0 [110/2000] via 203.250.15.1, 00:00:14, Serial0 203.250.13.0 is variably subnetted,
2 subnets, 2 masks O 203.250.13.41 255.255.255.255 [110/75] via 203.250.15.1, 00:00:15, Serial0
O E2 203.250.13.0 255.255.255.0 [110/2000] via 203.250.15.1, 00:00:15, Serial0 203.250.15.0
255.255.255.252 is subnetted, 2 subnets C 203.250.15.8 is directly connected, Loopback1 C
203.250.15.0 is directly connected, Serial0 O 203.250.14.0 [110/74] via 203.250.15.1, 00:00:15,
Serial0 128.213.0.0 is variably subnetted, 2 subnets, 2 masks O E2 128.213.0.0 255.255.0.0
[110/2000] via 203.250.15.1, 00:00:15,Serial0 O 128.213.63.0 255.255.255.252 [110/138] via
203.250.15.1, 00:00:16, Serial0
```

Las entradas de BGP han desaparecido porque OSPF tiene una mejor distancia que iBGP. La distancia de OSPF es 110, mientras que la distancia de iBGP es 200.

Desactive la sincronización en el RTA, de modo que el RTA pueda anunciar 203.250.15.0. Esta acción es necesaria porque el RTA no se sincroniza con OSPF debido a la diferencia en las máscaras. Mantenga la sincronización desactivada en el RTB, de modo que el RTB pueda anunciar 203.250.13.0. Esta acción es necesaria en el RTB por la misma razón.

Ahora, haga uso de la interfaz s1 del RTB para ver cómo se ven las rutas. También, habilite OSPF en el serial 1 del RTB para dejarlo pasivo. Este paso permite que el RTA sepa del salto siguiente 192.208.10.5 vía IGP. Si usted no realiza este paso, pueden ocurrir loops de ruteo porque, para alcanzar el salto siguiente 192.208.10.5, usted necesita ir en sentido contrario a través de eBGP. Estas son las nuevas configuraciones del RTA y RTB:

```
RTA#
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 203.250.13.41 255.255.255.0

interface Ethernet0
ip address 203.250.14.1 255.255.255.0
```

```
interface Serial0
 ip address 128.213.63.1 255.255.255.252

router ospf 10
 redistribute bgp 100 metric 2000 subnets
 passive-interface Serial0
 network 203.250.0.0 0.0.255.255 area 0
 network 128.213.0.0 0.0.255.255 area 0

router bgp 100
 no synchronization
 network 203.250.13.0
 network 203.250.14.0
 neighbor 128.213.63.2 remote-as 200
 neighbor 203.250.15.2 remote-as 100
 neighbor 203.250.15.2 update-source Loopback0

RTB#
hostname RTB

ip subnet-zero

interface Serial0
 ip address 203.250.15.2 255.255.255.252

interface Serial1
 ip address 192.208.10.6 255.255.255.252

router ospf 10
 redistribute bgp 100 metric 1000 subnets
 passive-interface Serial1
 network 203.250.0.0 0.0.255.255 area 0
 network 192.208.0.0 0.0.255.255 area 0

router bgp 100
 no synchronization
 network 203.250.15.0
 neighbor 192.208.10.5 remote-as 300
 neighbor 203.250.13.41 remote-as 100
```

Las tablas de BGP son similares a lo siguiente:

RTA# **show ip bgp** BGP table version is 117, local router ID is 203.250.13.41 Status codes: s suppressed, d damped, h history, * valid, > best, i -internal Origin codes: i - IGP, e - EGP, ? - incomplete Network Next Hop Metric LocPrf Weight Path *> 128.213.0.0 128.213.63.2 0 0 200 i *>i192.208.10.0 192.208.10.5 0 100 0 300 i *>i195.211.10.0 192.208.10.5 100 0 300 500 i * 128.213.63.2 0 200 400 500 i *> 200.200.10.0 128.213.63.2 0 200 400 i *> 203.250.13.0 0.0.0.0 0 32768 i *> 203.250.14.0 0.0.0.0 0 32768 i *>i203.250.15.0 203.250.15.2 0 100 0 i RTB# **show ip bgp** BGP table version is 12, local router ID is 203.250.15.10 Status codes: s suppressed, d damped, h history, * valid, > best, i -internal Origin codes: i - IGP, e - EGP, ? - incomplete Network Next Hop Metric LocPrf Weight Path *>i128.213.0.0 128.213.63.2 0 100 0 200 i * 192.208.10.5 0 300 500 400 200 i *> 192.208.10.0 192.208.10.5 0 0 300 i *> 195.211.10.0 192.208.10.5 0 300 500 i *>i200.200.10.0 128.213.63.2 100 0 200 400 i * 192.208.10.5 0 300 500 400 i *>i203.250.13.0 203.250.13.41 0 100 0 i *>i203.250.14.0 203.250.13.41 0 100 0 i *> 203.250.15.0 0.0.0.0 0 32768 i

Hay diferentes formas de diseñar su red para comunicarse con los dos ISP, el AS200 y el AS300. Una manera es tener un ISP primario y un ISP de respaldo. Usted puede detectar las rutas parciales de uno de los ISP y las rutas predeterminadas de ambos ISP. En este ejemplo, usted recibe las rutas parciales del AS200 y solo rutas locales del AS300. El RTA y el RTB generan las rutas predeterminadas en OSPF, con el RTB como la preferencia debido a la métrica más baja. De esta manera, usted puede balancear el tráfico saliente entre los dos ISP.

Puede ocurrir una posible asimetría si el tráfico que sale del RTA regresa vía el RTB. Esta situación puede ocurrir si usted utiliza el mismo conjunto de direcciones IP, la misma red principal, cuando se comunica con los dos ISP. Debido a la agregación, su AS entero puede verse como una entidad entera para el mundo exterior. Los puntos de entrada a su red pueden ocurrir vía el RTA o el RTB. Usted puede descubrir que todo el tráfico entrante a su AS llega vía uno solo punto, aunque tenga varios puntos a Internet. En el ejemplo, usted tiene dos redes principales diferentes cuando se comunica con los dos ISP.

Otra razón posible de la asimetría es la diferente longitud de trayectoria anunciada para alcanzar su AS. Quizás un proveedor de servicios está más cerca de un destino que de otro. En el ejemplo, el tráfico del AS400 que tiene a su red como destino siempre viene a través del RTA debido a la trayectoria más corta. Usted puede intentar efectuar esa decisión. Puede utilizar el **comando set as-path prepend** para anteponer números de trayectoria a sus actualizaciones y hacer que la longitud de trayectoria parezca más larga. Pero, con atributos como preferencia local, métrica o peso, el AS400 puede haber configurado el punto de salida para que sea AS200. En este caso, no hay nada que usted pueda hacer.

Esta configuración es la configuración final para todos los routers:

```
RTA#
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 203.250.13.41 255.255.255.0

interface Ethernet0
 ip address 203.250.14.1 255.255.255.0

interface Serial0
 ip address 128.213.63.1 255.255.255.252

router ospf 10
 redistribute bgp 100 metric 2000 subnets
 passive-interface Serial0
 network 203.250.0.0 0.0.255.255 area 0
 network 128.213.0.0 0.0.255.255 area 0
 default-information originate metric 2000

router bgp 100
 no synchronization
network 203.250.13.0
 network 203.250.14.0
 neighbor 128.213.63.2 remote-as 200
 neighbor 128.213.63.2 route-map setlocalpref in
 neighbor 203.250.15.2 remote-as 100
```

```
 neighbor 203.250.15.2 update-source Loopback0

ip classless
ip default-network 200.200.0.0

route-map setlocalpref permit 10
 set local-preference 200
```

En el RTA, la preferencia local para las rutas que vienen del AS200 está configurada en 200. Además, la red 200.200.0.0 es la opción para el candidato predeterminado. El **comando ip default-network** lo habilita para elegir el valor predeterminado.

También en este ejemplo, el uso del comando default-information originate con OSPF inserta la ruta predeterminada dentro del dominio de OSPF. Este ejemplo también utiliza este comando con el protocolo Intermediate System-to-Intermediate System (IS-IS) y BGP. Para el RIP, hay una redistribución automática en RIP de 0.0.0.0, sin configuración adicional. Para IGRP y EIGRP, la inserción de la información predeterminada en el dominio de IGP ocurre después de la redistribución de BGP en IGRP y EIGRP. Además, con IGRP y EIGRP, usted puede redistribuir una ruta estática a 0.0.0.0 en el dominio de IGP.

```
RTF#
hostname RTF

ip subnet-zero

interface Ethernet0
 ip address 203.250.14.2 255.255.255.0

interface Serial1
 ip address 203.250.15.1 255.255.255.252

router ospf 10
 network 203.250.0.0 0.0.255.255 area 0

ip classless

RTB#
hostname RTB

ip subnet-zero

interface Loopback1
 ip address 203.250.15.10 255.255.255.252

interface Serial0
 ip address 203.250.15.2 255.255.255.252
!
interface Serial1
 ip address 192.208.10.6 255.255.255.252

router ospf 10
 redistribute bgp 100 metric 1000 subnets
 passive-interface Serial1
 network 203.250.0.0 0.0.255.255 area 0
```

```
 network 192.208.10.6 0.0.0.0 area 0
 default-information originate metric 1000
!
router bgp 100
 no synchronization
 network 203.250.15.0
 neighbor 192.208.10.5 remote-as 300
 neighbor 192.208.10.5 route-map localonly in
 neighbor 203.250.13.41 remote-as 100
!
ip classless
ip default-network 192.208.10.0
ip as-path access-list 1 permit ^300$

route-map localonly permit 10
 match as-path 1
set local-preference 300
```

Para el RTB, la preferencia local para las actualizaciones que vienen del AS300 está configurada en 300. Este valor es más alto que el valor de preferencia local de las actualizaciones de iBGP que vienen del RTA. De esta manera, el AS100 selecciona RTB para las rutas locales del AS300. Cualquier otra ruta en el RTB, si existiera alguna otra ruta, se transmite internamente con una preferencia local de 100. Este valor es más bajo que la preferencia local de 200, que viene del RTA. Por lo tanto, el RTA es la preferencia.

Nota: Usted solo anunció las rutas locales del AS300. Cualquier información de trayectoria que no coincida con ^300$, se descartará. Si usted desea anunciar las rutas locales y las rutas vecinas, que son los clientes de ISP, utilice ^300_[0-9]*.

Este es el resultado de la expresión regular que indica las rutas locales del AS300:

```
RTB# show ip bgp regexp ^300$ BGP table version is 14, local router ID is 203.250.15.10 Status
codes: s suppressed, d damped, h history, * valid, > best, i - internal Origin codes: i - IGP, e
- EGP, ? - incomplete Network Next Hop Metric LocPrf Weight Path *> 192.208.10.0 192.208.10.5 0
300 0 300 RTC# hostname RTC ip subnet-zero interface Loopback0 ip address 128.213.63.130
255.255.255.192 interface Serial2/0 ip address 128.213.63.5 255.255.255.252 ! interface
Serial2/1 ip address 128.213.63.2 255.255.255.252 router bgp 200 network 128.213.0.0 neighbor
128.213.63.1 remote-as 100 neighbor 128.213.63.1 distribute-list 1 out neighbor 128.213.63.6
remote-as 400 ip classless access-list 1 deny 195.211.0.0 0.0.255.255 access-list 1 permit any
```

En el RTC, usted agrega 128.213.0.0/16 e indica las rutas específicas para la inserción en el AS100. Si el ISP se niega a realizar esta tarea, usted debe filtrar en el extremo entrante del AS100.

```
RTD#
hostname RTD

ip subnet-zero

interface Loopback0
 ip address 192.208.10.174 255.255.255.192
!
interface Serial0/0
 ip address 192.208.10.5 255.255.255.252
!
```

```
interface Serial0/1
 ip address 192.208.10.2 255.255.255.252

router bgp 300
 network 192.208.10.0
 neighbor 192.208.10.1 remote-as 500
 neighbor 192.208.10.6 remote-as 100

RTG#
hostname RTG

ip subnet-zero

interface Loopback0
 ip address 195.211.10.174 255.255.255.192

interface Serial0
 ip address 192.208.10.1 255.255.255.252

interface Serial1
 ip address 195.211.10.1 255.255.255.252

router bgp 500
 network 195.211.10.0
 aggregate-address 195.211.0.0 255.255.0.0 summary-only
 neighbor 192.208.10.2 remote-as 300
 neighbor 192.208.10.2 send-community
 neighbor 192.208.10.2 route-map setcommunity out
 neighbor 195.211.10.2 remote-as 400
!
ip classless
access-list 1 permit 195.211.0.0 0.0.255.255
access-list 2 permit any
route-map setcommunity permit 20
 match ip address 2
!
route-map setcommunity permit 10
 match ip address 1
 set community no-export
```

Una demostración del uso del filtrado de comunidades está en el RTG. Usted agrega una comunidad **no-export** a las actualizaciones de 195.211.0.0 hacia el RTD. De esta manera, el RTD no exporta esa ruta al RTB. Sin embargo, en este caso, el RTB no acepta estas rutas de todos modos.

```
RTE#
hostname RTE

ip subnet-zero

interface Loopback0
 ip address 200.200.10.1 255.255.255.0

interface Serial0
```

```
 ip address 195.211.10.2 255.255.255.252

interface Serial1
 ip address 128.213.63.6 255.255.255.252

router bgp 400
 network 200.200.10.0
 aggregate-address 200.200.0.0 255.255.0.0 summary-only
 neighbor 128.213.63.5 remote-as 200
 neighbor 195.211.10.1 remote-as 500

ip classless
```

El RTE agrega 200.200.0.0/16. Estas son las tablas de ruteo y de BGP finales para el RTA, el RTF y el RTB:

```
RTA# show ip bgp BGP table version is 21, local router ID is 203.250.13.41 Status codes: s
suppressed, d damped, h history, * valid, > best, i - internal Origin codes: i - IGP, e - EGP, ?
- incomplete Network Next Hop Metric LocPrf Weight Path *> 128.213.0.0 128.213.63.2 0 200 0 200
i *>i192.208.10.0 192.208.10.5 0 300 0 300 i *> 200.200.0.0/16 128.213.63.2 200 0 200 400 i *>
203.250.13.0 0.0.0.0 0 32768 i *> 203.250.14.0 0.0.0.0 0 32768 i *>i203.250.15.0 203.250.15.2 0
100 0 i RTA# show ip route Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -
BGP D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area E1 - OSPF external type 1, E2
- OSPF external type 2, E - EGP i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default Gateway of last resort is 128.213.63.2 to network 200.200.0.0 192.208.10.0 is variably
subnetted, 2 subnets, 2 masks O E2 192.208.10.0 255.255.255.0 [110/1000] via 203.250.14.2,
00:41:25, Ethernet0 O 192.208.10.4 255.255.255.252 [110/138] via 203.250.14.2, 00:41:25,
Ethernet0 C 203.250.13.0 is directly connected, Loopback0 203.250.15.0 is variably subnetted, 3
subnets, 3 masks O 203.250.15.10 255.255.255.255 [110/75] via 203.250.14.2, 00:41:25, Ethernet0
O 203.250.15.0 255.255.255.252 [110/74] via 203.250.14.2, 00:41:25, Ethernet0 B 203.250.15.0
255.255.255.0 [200/0] via 203.250.15.2, 00:41:25 C 203.250.14.0 is directly connected, Ethernet0
128.213.0.0 is variably subnetted, 2 subnets, 2 masks B 128.213.0.0 255.255.0.0 [20/0] via
128.213.63.2, 00:41:26 C 128.213.63.0 255.255.255.252 is directly connected, Serial0 O*E2
0.0.0.0/0 [110/1000] via 203.250.14.2, Ethernet0/0 B* 200.200.0.0 255.255.0.0 [20/0] via
128.213.63.2, 00:02:38 RTF# show ip route Codes: C - connected, S - static, I - IGRP, R - RIP, M
- mobile, B - BGP D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area E1 - OSPF
external type 1, E2 - OSPF external type 2, E - EGP i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS
level-2, * - candidate default Gateway of last resort is 203.250.15.2 to network 0.0.0.0
192.208.10.0 is variably subnetted, 2 subnets, 2 masks O E2 192.208.10.0 255.255.255.0
[110/1000] via 203.250.15.2, 00:48:50, Serial1 O 192.208.10.4 255.255.255.252 [110/128] via
203.250.15.2, 01:12:09, Serial1 203.250.13.0 is variably subnetted, 2 subnets, 2 masks O
203.250.13.41 255.255.255.255 [110/11] via 203.250.14.1, 01:12:09, Ethernet0 O E2 203.250.13.0
255.255.255.0 [110/2000] via 203.250.14.1, 01:12:09, Ethernet0 203.250.15.0 is variably
subnetted, 2 subnets, 2 masks O 203.250.15.10 255.255.255.255 [110/65] via 203.250.15.2,
01:12:09, Serial1 C 203.250.15.0 255.255.255.252 is directly connected, Serial1 C 203.250.14.0
is directly connected, Ethernet0 128.213.0.0 is variably subnetted, 2 subnets, 2 masks O E2
128.213.0.0 255.255.0.0 [110/2000] via 203.250.14.1, 00:45:01, Ethernet0 O 128.213.63.0
255.255.255.252 [110/74] via 203.250.14.1, 01:12:11, Ethernet0 O E2 200.200.0.0 255.255.0.0
[110/2000] via 203.250.14.1, 00:03:47, Ethernet0 O*E2 0.0.0.0 0.0.0.0 [110/1000] via
203.250.15.2, 00:03:33, Serial1
```

Nota: En la tabla de ruteo de RTF, se indica que la manera de alcanzar las redes locales al AS300, como 192.208.10.0, es a través del RTB. La manera de alcanzar otras redes conocidas, como 200.200.0.0, es a través del RTA. El gateway de último recurso está configurado en RTB. Si

algo le sucede a la conexión entre el RTB y el RTD, el valor predeterminado que el RTA anuncia se derriba con una métrica de 2000.

```
RTB# show ip bgp BGP table version is 14, local router ID is 203.250.15.10 Status codes: s
suppressed, d damped, h history, * valid, > best, i - internal Origin codes: i - IGP, e - EGP, ?
- incomplete Network Next Hop Metric LocPrf Weight Path *>i128.213.0.0 128.213.63.2 0 200 0 200
i *> 192.208.10.0 192.208.10.5 0 300 0 300 i *>i200.200.0.0/16 128.213.63.2 200 0 200 400 i
*>i203.250.13.0 203.250.13.41 0 100 0 i *>i203.250.14.0 203.250.13.41 0 100 0 i *> 203.250.15.0
0.0.0.0 0 32768 i RTB# show ip route Codes: C - connected, S - static, I - IGRP, R - RIP, M -
mobile, B - BGP D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area E1 - OSPF
external type 1, E2 - OSPF external type 2, E - EGP i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS
level-2, * - candidate default Gateway of last resort is 192.208.10.5 to network 192.208.10.0 *
192.208.10.0 is variably subnetted, 2 subnets, 2 masks B* 192.208.10.0 255.255.255.0 [20/0] via
192.208.10.5, 00:50:46 C 192.208.10.4 255.255.255.252 is directly connected, Serial1
203.250.13.0 is variably subnetted, 2 subnets, 2 masks O 203.250.13.41 255.255.255.255 [110/75]
via 203.250.15.1, 01:20:33, Serial0 O E2 203.250.13.0 255.255.255.0 [110/2000] via 203.250.15.1,
01:15:40, Serial0 203.250.15.0 255.255.255.252 is subnetted, 2 subnets C 203.250.15.8 is
directly connected, Loopback1 C 203.250.15.0 is directly connected, Serial0 O 203.250.14.0
[110/74] via 203.250.15.1, 01:20:33, Serial0 128.213.0.0 is variably subnetted, 2 subnets, 2
masks O E2 128.213.0.0 255.255.0.0 [110/2000] via 203.250.15.1, 00:46:55, Serial0 O 128.213.63.0
255.255.255.252 [110/138] via 203.250.15.1, 01:20:34, Serial0 O*E2 0.0.0.0/0 [110/2000] via
203.250.15.1, 00:08:33, Serial0 O E2 200.200.0.0 255.255.0.0 [110/2000] via 203.250.15.1,
00:05:42, Serial0
```

Información Relacionada

- BGP: Preguntas Frecuentes
- Configuraciones de Ejemplo de BGP a través de un Firewall PIX
- Cómo Utilizar HSRP para Proporcionar Redundancia en una Red de BGP con Varias Conexiones
- Configuración de Redundancia de Modo de Router Simple y BGP en un MSFC Cat6000
- Cómo Lograr un Ruteo Óptimo y Reducir el Consumo de Memoria de BGP
- Troubleshooting de BGP
- Troubleshooting de CPU Alto Causado por el Proceso de Router de BGP o Escaneo de BGP
- Distribución de la Carga con BGP en Entornos con una Sola Conexión y con Varias Conexiones: Configuraciones de Ejemplo
- Página de Soporte de BGP
- Soporte Técnico y Documentación - Cisco Systems