Estudos de caso de BGP

Índice

Introdução

Este documento contém cinco estudos de caso Border Gateway Protocol (BGP).

Pré-requisitos

Requisitos

Não existem requisitos específicos para este documento.

Componentes Utilizados

Este documento não se restringe a versões de software e hardware específicas.

Convenções

Consulte as Convenções de Dicas Técnicas da Cisco para obter mais informações sobre convenções de documentos.

Estudos de Caso do BGP 1

O BGP, que o RFC 1771 define, permite que você crie o roteamento sem loop do interdomain entre os sistemas autônomos (AS). Um AS está um conjunto de roteador sob uma única administração técnica. Roteadores no AS pode usar os protocolos Interior Gateway Protocols (IGP) múltiplos para trocar a informação de roteamento dentro do AS. Os roteadores podem usar um protocolo de gateway exterior aos pacotes de rota fora do AS.

Como o BGP funciona?

O BGP usa o TCP como o protocolo de transporte, na porta 179. Dois roteadores BGP formam uma conexão de TCP entre uma outra. Estes roteadores são roteadores de peer. Os mensagens de intercâmbio dos roteadores de peer para abrir e confirmar os parâmetros de conexão.

Informação de alcançabilidade de rede da troca dos roteadores BGP. Esta informação é principalmente uma indicação dos caminhos cheios que uma rota deve recolher a ordem para alcançar a rede de destino. Os trajetos são BGP AS números. Esta informação ajuda na construção de um gráfico dos AS que são sem loop. O gráfico igualmente mostra onde aplicar políticas de roteamento a fim de reforçar algumas limitações no comportamento de roteamento.

Quaisquer dois roteadores que formarem uma conexão de TCP a fim trocar informação de roteamento de BGP são “peers” ou “vizinhos”. Os peers BGP trocam inicialmente as tabelas de roteamento BGP completas. Após esta troca, os peers enviam atualizações de acréscimo como as alterações de tabela de roteamento. O BGP mantem um número de versão da tabela de BGP. O número de versão é o mesmo para todos os peers BGP. O número de versão muda sempre que o BGP atualiza a tabela com alterações de informação de roteamento. A emissão dos pacotes keepalive assegura que a conexão entre os peers BGP esteja viva. Os pacotes de notificação saem em resposta aos erros ou às condições especiais.

eBGP e iBGP

Se um AS tem múltiplos falantes BGP, o AS pode servir como um serviço de trânsito para outros AS. Como o diagrama nesta seção mostra, o AS200 é um AS de trânsito para o AS100 e o AS300.

A fim de enviar a informação aos AS externos, deve haver uma segurança da alcançabilidade para redes. A fim de assegurar a alcançabilidade de rede, estes processos ocorrem:

BGP interno (iBGP) que conecta os roteadores dentro de um AS

Redistribuição da Informação de BGP aos IGP que são executado no AS

Quando o BGP é executado entre os roteadores que pertencem a dois AS diferentes, este é chamado BGP exterior (eBGP). Quando o BGP é executado entre roteadores no mesmos AS, este está chamado iBGP.

Habilite o roteamento BGP

Complete estas etapas a fim de habilitar e configurar o BGP.

Suponha que você quer ter dois roteadores, RTA e RTB, conversando através do BGP. No primeiro exemplo, o RTA e o RTB estão em AS diferentes. No segundo exemplo, ambos os roteadores pertencem ao mesmos QUE.

Defina o processo de roteador e o número do AS a que os roteadores pertencem.

Emita este comando para habilitar o BGP em um roteador:

```
router bgp autonomous-system RTA# router bgp 100 RTB# router bgp 200
```

Estas indicações indicam que o RTA executa o BGP e pertence ao AS100. O RTB executa o BGP e pertence ao AS200.

Defina vizinhos de BGP.

A formação do vizinho de BGP indica os roteadores que tentam falar através do BGP. A seçãoFormar vizinhos de BGP explica este processo.

Forme vizinhos de BGP

Dois roteadores BGP tornam-se vizinhos depois que os roteadores estabelecem uma conexão de TCP entre si. A conexão de TCP é essencial para que os roteadores de dois peers comecem a troca das atualizações de roteamento.

Depois que a conexão de TCP está funcionando, os roteadores enviam mensagens abertas para trocar valores. Os valores trocados entre os roteadores inclui o número AS, a versão BGP executados pelo roteadores, a ID do roteador BGP, e o tempo de contenção do keepalive. Após a confirmação e a aceitação destes valores, ocorre o estabelecimento da conexão vizinha. Todo o

estado além do estabelecido é uma indicação que os dois roteadores não se formam vizinhos e os roteadores não podem trocar atualizações BGP.

Emita este **comando neighbor** para estabelecer uma conexão de TCP:

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

O *número* no comando no número AS do roteador a que você quer conectar com o BGP. O ***IP address*** é o endereço de próximo salto com a conexão direta para o eBGP. Para o iBGP, o ***IP address*** é todo o endereço IP no outro roteador.

Os dois endereços IP que você usa no **comando neighbor** dos roteadores de peer *devem* poder alcançar um ao outro. Uma maneira de verificar a alcançabilidade é um ping estendido entre os dois endereços IP. O ping estendido força o roteador fazendo o ping a usar como fonte o endereço IP que o **comando neighbor** especificar. O roteador deve usar este endereço ao invés do endereço IP da interface de que o pacote vai.

Se há alguma mudança da configuração de BGP, você *deve* restaurar a conexão vizinha para permitir que os parâmetros novos tomem efeito.

cancele o *endereço* BGP IP

Nota: O ***endereço*** é o endereço vizinho.

clear ip bgp *

Este comando cancela todas as conexões vizinha.

Por padrão, as sessões de BGP começam com o uso da versão 4 do BGP e negociam para baixo às versões anterior, caso necessário. Você pode impedir negociações e forçar a versão BGP que os roteadores usam para comunicar com um vizinho. Emita este comando no modo de configuração do roteador:

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

Está aqui um exemplo da configuração do **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
```

Neste exemplo, RTA e RTB executam o eBGP. RTB e RTC executam o iBGP. O número AS remoto aponta para um AS externo ou um AS interno, os quais indicam o eBGP ou o iBGP. Também, os pares do eBGP têm a conexão direta, mas os pares do iBGP não têm a conexão

direta. os roteadores iBGP não precisam ter a conexão direta. Mas, deve haver algum IGP que executa e permite que os dois vizinhos alcancem um ao outro.

Esta seção fornece um exemplo da informação que o comando show ip bgp neighbors indica.

Nota: Preste atenção especial ao estado BGP. Qualquer coisa a não ser o estado estabelecido indica que os pares não estão funcionando.

Nota: Observe também estes artigos:

A versão BGP, que é 4

O roteador remoto ID

Este número é o endereço IP mais alto no roteador ou na relação do loopback mais elevado, se existente.

A versão de tabela

A versão de tabela fornece o estado da tabela. Quando essa informação nova entra, a tabela aumenta a versão. Uma versão que continue a incrementar indica que há algum flap da rota que causa a atualização contínua das rotas.

```
# 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
```

BGP e interfaces de loopback

O uso de uma interface de loopback para definir vizinhos é comum com iBGP, mas não é comum com eBGP. Normalmente, você usa a interface de loopback para certificar-se que o endereço IP do vizinho fica acima e é independente de hardware que funciona corretamente. No caso do eBGP, os roteadores de peer têm freqüentemente uma conexão direta, e o loopback não se aplica.

Se você usa o endereço IP de uma interface de loopback no **comando neighbor**, você precisa alguma configuração extra no roteador vizinho. O roteador vizinho precisa informar o BGP do uso de uma interface de loopback ao invés de uma interface física para iniciar o BGP vizinho do TCP vizinho. A fim de indicar uma interface de loopback, emita este comando:

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

Este exemplo ilustra o uso deste 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
```

Neste exemplo, RTA e RTB executam dentro do AS100. No **comando neighbor**, o RTB usa a interface de loopback do RTA, 150.212.1.1. Neste caso, o RTA deve forçar o BGP para usar o endereço IP de loopback como a fonte na conexão vizinha TCP. A fim de forçar esta ação, o RTA adiciona o **update-source** ***interface-type interface-number*** de modo que o comando seja o **neighbor 190.225.11.1 update-source loopback 1.** Esta declaração força o BGP para usar o endereço IP da interface de loopback quando o BGP conversa com o vizinho 190.225.11.1.

Nota: O RTA usou o endereço IP da interface física do RTB, 190.225.11.1, como um vizinho. O uso deste endereço IP é porque o RTB não precisa nenhuma configuração especial. Refira a configuração de exemplo para o iBGP e eBGP com ou sem endereço de loopback para uma configuração de exemplo completo do cenário de rede.

eEBGP Multihop

Em alguns casos, um roteador Cisco pode executar o eBGP com um roteador da terceiros que não permite a conexão direta dos dois peers externos. Para conseguir a conexão, você pode usar e EBGP multihop. Os eEBGP multihop permitem uma conexão vizinha entre dois peers externos que não têm a conexão direta. O multihop é somente para o eBGP e não para o iBGP. Este exemplo ilustra eEBGP multihop:

```
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
```

O RTA indica um vizinho externo que não tenha a conexão direta. O RTA precisa de indicar seu uso do comando neighbor ebgp-multihop. Por outro lado, o RTB indica um vizinho que tenha a conexão direta, que é 129.213.1.2. Devido a esta conexão direta, o RTB não precisa o **comando neighbor ebgp-multihop**. Você também deve configurar um IGP ou um roteamento estático para permitir que os vizinhos sem conexão se alcancem.

O exemplo na seção dos eEBGP Multihop (Load Balancing) mostra como conseguir o balanço de carga (load balancing) com BGP em um caso onde você tenha o eBGP sobre linhas paralela.

eEBGP Multihop (load balancing)

```
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
```

Este exemplo ilustra o uso das interfaces de loopback, **update-source** , e **ebgp-multihop**. O exemplo é uma ação alternativa a fim conseguir um balanço de carga (load balancing) entre dois pacotes eBGP sobre linhas de série paralelas. Nas situações normais, o BGP escolhe uma das linhas em que enviar pacotes, e no balanço de carga (load balancing) isto não acontece. Com a introdução de interfaces de loopback, o salto seguinte para o eBGP é a interface de loopback. Você usa rotas estáticas, ou um IGP, para introduzir dois caminhos de custo iguais para alcançar o destino. O RTA tem duas escolhas para alcançar o salto seguinte 160.10.1.1: um trajeto através de 1.1.1.2 e o outro trajeto através de 2.2.2.2. O RTB tem as mesmas escolhas.

Mapas de rotas

Há um uso pesado dos mapas de rotas com BGP. No contexto BGP, o mapa de rotas é um método para controlar e alterar a informação de roteamento. O controle e a alteração da informação de roteamento ocorrem com a definição das condições para a redistribuição de rota de um protocolo de roteamento a outro. Ou o controle da informação de roteamento pode ocorrer na injeção dentro e fora do BGP. O formato do mapa de rotas segue:

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

O mapa de caracteres é simplesmente um nome que você dê ao mapa de rotas. Você pode definir múltiplas instâncias do mesmo mapa de rotas, ou o mesmo caractere de nome. O número de seqüência é simplesmente uma indicação da posição que um mapa de rotas novo deve ter na lista de mapas de rotas que você tem configurado já com o mesmo nome.

Neste exemplo, há duas amostras do mapa de rotas definido, com o nome MYMAP. O primeiro exemplo tem um número de seqüência de 10, e o segundo tem um número de seqüência de 20.

route-map mymap permit 10 (o primeiro conjunto de condição vai aqui.)

route-map MYMAP permit 20 (o segundo conjunto de condição vai aqui.)

Quando você aplica o mapa de rotas MYMAP às rotas de entrada e de saída, o primeiro conjunto de condições é aplicado através da amostra 10. Se o primeiro conjunto de condição não é encontrado, você continua para uma amostra mais alta do mapa de rotas.

comandos match and set Configuration

Cada mapa de rotas consiste em uma lista de comandos **match** and **set** configuration. O match especifica **critérios de combinação**, e set especifica uma ação **definir** se os critérios que o

comando match são encontrados.

Por exemplo, você pode definir um mapa de rotas que verifique atualizações de saída. Se há um combinação para o endereço IP 1.1.1.1, a métrica para essa atualização está ajustada ao 5. Estes comandos ilustram o exemplo:

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

Agora, se os critérios de verificação de repetição de dados são encontrados e você tem uma licença, há uma redistribuição ou um controle das rotas, porque a ação do grupo especifica. Você sai da lista.

Se os critérios de verificação de repetição de dados são encontrados e você tem uma negação, não há nenhum redistribuição ou controle da rota. Você sai da lista.

Se os critérios de verificação de repetição de dados não são encontrados e você tem um permitir ou negar, o exemplo seguinte do mapa de rotas está verificado. Por exemplo, o exemplo 20 é verificado. Esta verificação do seguinte-exemplo continua até que você saia ou termine todos os exemplos do mapa de rotas. Se você termina a lista sem uma combinação, a rota não está aceitada nem está enviada.

No software release de Cisco IOS® mais cedo do que o Cisco IOS Software Release 11.2, quando você usa mapas de rotas para filtrar atualizações BGP um pouco do que redistribui entre protocolos, você *não pode* filtrar a entrada quando você usa um comando match no endereço IP. Um filtro na saída é aceitável. O Cisco IOS Software Release 11.2 e Mais Recente não têm esta limitação.

Os comandos relacionados para a combinação são:

- match as-path
- combinar comunidade
- combinar clns
- combinar interface
- combinar endereço ip
- combinar ip next-hop
- combinar ip route-source
- combinar metrica
- combinar tipo de rota

combinar tag

Os comandos relacionados para o **grupo** são:

definir as-path

definir clns

definir etiqueta automática

definir comunidade

set interface

definir interface padrão

definir ip default próximo salto

definir nível

defomor a preferência local

definir métrica

definir o tipo métrico

definir o salto seguinte

definir a origem

definir a etiqueta

definir o peso

Olhe alguns exemplos do mapa de rotas:

Exemplo 1

Supor que a corrida BGP protocolo de informação de roteamento (RIF) da corrida RTA e RTB, e RTA e RTC. O RTA obtém atualizações através do BGP e redistribui as atualizações PARA RASGAR-SE. Supor que o RTA quer redistribuir neste caso às rotas RTB sobre 170.10.0.0 com

umas rotas métricas de 2 e todas outras com um métrico de 5., você pode usar esta configuração:

```
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
```

Neste exemplo, se uma rota combina o endereço IP 170.10.0.0, a rota tem um métrico de 2. Então, você sao da lista do mapa de rotas. Se não há nenhuma combinação, você continua abaixo da lista do mapa de rotas, que indica o ajuste de tudo outro ao 5 métrico.

Nota: Faça sempre a pergunta “o que acontece às rotas que não combinam algumas das instruções compatíveis?” Estas rotas deixam cair, por padrão.

Exemplo 2

Supor que, no exemplo 1, você não quer o AS100 para aceitar atualizações sobre 170.10.0.0. Você não pode aplicar mapas de rotas no de entrada quando você combina com um endereço IP como a base. Conseqüentemente, você deve usar um mapa de rota externa no 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
```

Agora que você sente mais confortável com como começar o BGP e como definir um vizinho, olhe como começar a troca da informação de rede.

Há múltiplas formas de enviar a informação de rede com uso do BGP. Estas seções atravessam os métodos um por um:

comando network

O formato do **comando network** é:

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

O **comando network** controla as redes que se originam desta caixa. Este conceito é diferente da configuração familiar com protocolo Interior Gateway Routing (IGRP) e RIP. Com este comando, você não tenta executar o BGP em uma determinada interface. Ao invés, você tenta indicar o BGP de qual redes BGP deve originar desta caixa. O comando usa uma parcela da máscara porque o BGP versão 4 (BGP4) pode cuidar subnetting e supernetting. Um máximo de 200 entradas do **comando network** é aceitável.

O **comando network** funciona se o roteador conhece a rede que você tenta anunciar, se conectado, estática, ou aprendido dinâmicamente.

Um exemplo do **comando network** é:

```
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 exemplo indica que o roteador A gera uma entrada de rede para 192.213.0.0/16. O /16 indica que você usa uma superrede do endereço classe C e você anuncia os primeiros dois octetos, ou primeiros 16 bits.

Nota: Você precisa da rota estática para conseguir que o roteador gere 192.213.0.0 porque a rota estática combina uma entrada de compatibilidade na tabela de roteamento.

Redistribuição

O **comando network** é uma maneira de anunciar suas redes através do BGP. Uma outra maneira é redistribuir seu IGP no BGP. Seu IGP pode ser IGRP, protocolo Open Shortest Path First (OSPF), RIP, Enhanced Interior Gateway Routing Protocol (EIGRP), ou um outro protocolo. Esta redistribuição pode parecer assustadora porque agora você despeja todas suas rotas internas no BGP; algumas destas rotas podem ter sido aprendidas através do BGP e você não precisa enviá-las para fora outra vez. Aplique a filtração cuidadosa para certificar-se de que você envia às rotas somente Internet que você quer anunciar e não a todas as rotas que você tem. Aqui está um exemplo:

O RTA anuncia que 129.213.1.0 e o RTC anunciam 175.220.0.0. Olhe a configuração de RTC:

Se você emite o **comando network**, você tem:

```
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.
```

Se ao invés você usa redistribuição, você tem:

```
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 redistribuição causa as origens de 129.213.1.0 pelo seu AS. Você não é a fonte de 129.213.1.0; O AS100 é a fonte. Assim você tem que usar filtros para impedir a fonte fora dessa rede pelo seu AS. A configuração correta é:

```
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
```

Você usa o **comando access-list** para controlar as redes que originam o AS200.

A redistribuição do OSPF no BGP é levemente diferente da redistribuição para outros IGP. A introdução simples de **redistribui ospf 1** sob o **roteador BGP** não funciona. As palavras-chaves específicas tais como **interno**, **externo**, e **nssa externo** são necessárias para redistribuir rotas respectivas. Refira a compreendendo a redistribuição das rotas OSPF no BGP para mais detalhes.

Rotas estáticas e redistribuição

Você pode sempre usar rotas estáticas para originar uma rede ou uma sub-rede. A única diferença é que o BGP considera estas rotas para ter uma origem que esteja incompleta, ou desconhecida. Você pode realizar o mesmo resultado nesse exemplo na seção de redistribuição realizada com esta:

```
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
....
```

A interface **null0** significa desconsidere o pacote. Assim se você obtém o pacote e há uma combinação mais específica do que 175.220.0.0, que existe, o roteador envia o pacote à combinação específica. Se não, o roteador desconsidera o pacote. Este método é uma boa maneira anunciar uma supernet.

Este original discutiu como você pode usar métodos diferentes para originar rotas fora do seu AS. Lembre-se que estas rotas são geradas em adição a outras rotas BGP que o BGP aprendido através dos vizinhos, internos ou externos. O BGP repassa informações que o BGP aprende de um peer a outros peers. A diferença é que as rotas que geram do **comando network**, redistribução, ou a estática indicam o seu AS como a origem destas redes.

A redistribução é sempre o método de injeção do BGP no IGP.

Aqui está um exemplo:

```
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: Você não precisa da rede 150.10.0.0 ou da rede 160.10.0.0 no RTC a menos que você queira que o RTC gere estas redes assim como repassar estas redes enquanto elas vêm dentro do AS100 e do AS200. Além disso, a diferença é que o **comando network** adiciona uma propaganda extra para estas mesmas redes, que indique que o AS300 também é uma origem para estas rotas.

Nota: Recorde que o BGP não aceita as atualizações que originaram de suas próprias AS. Esta recusa assegura uma topologia sem loop do interdomain.

Por exemplo, suponha que o AS200, do exemplo nesta seção, tenha uma conexão BGP direta no AS100. O RTA gera uma rota 150.10.0.0 e envia a rota ao AS300. Então, o RTC passa esta rota ao AS200 e mantem a origem como o AS100. O RTB passa 150.10.0.0 ao AS100 com a origem ainda AS100. O RTA observa que a atualização originou do suas próprias AS e ignoram a atualização.

iBGP

Você usa o iBGP se um AS quer para atuar como um sistema transitório a outros AS. É verdadeiro que você pode fazer a mesma coisa aprendendo através do eBGP, redistribuindo no IGP, e então redistribuindo outra vez em outro AS? Sim, mas iBGP oferece mais maneiras flexíveis e eficientes de trocar a informação dentro do AS. Por exemplo, o iBGP fornece maneiras de controlar a melhor saída do AS com uso da preferência local. A seção atributo de preferência local fornece mais informação sobre a preferência 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: Recorde que quando um auto-falante de BGP receber uma atualização de outros auto-falantes de BGP no seus próprios AS (iBGP), o auto-falante de BGP que recebe a atualização não redistribui essa informação a outros auto-falantes de BGP no seus próprios AS. O auto-falante de BGP que recebe a atualização redistribui a informação a outros auto-falantes de BGP fora do seu AS. Portanto, sustente uma malha cheia entre os alto-falantes iBGP dentro do AS.

No diagrama nesta seção, RTA e RTB executam iBGP. O RTA e o RTD também executam o iBGP. As atualizações BGP que vêm do RTB ao RTA transmitem ao RTE, que é fora do AS. As atualizações não transmitem ao RTD, que é dentro do AS. Portanto, faça um peering iBGP entre o RTB e o RTD a fim de não quebrar o fluxo das atualizações.

O algoritmo de decisão BGP

Depois que o BGP recebe atualizações sobre destinos diferentes dos sistemas diferente autônomo, o protocolo deve escolher trajetos para alcançar um destino específico. BGP escolhe somente um caminho único para alcançar um destino específico.

O BGP baseia a decisão em diferentes **atributos** , tais como o salto seguinte, as influências administrativas, a preferência local, a origem da rota, o comprimento de trajeto, o código de origem, o métrico, e outro atributos.

O BGP propaga sempre o melhor caminho aos vizinhos. Consulte o algoritmo de seleção de melhor caminho BGP para obter mais informações.

A seção Estudos de Caso BGP 2 explica estes atributos e seu uso.

Estudos de Caso do BGP 2

Atributo AS_PATH

Sempre que uma atualização da rota passa um AS, o número do AS prepended a essa atualização. O atributo AS_PATH é realmente a lista de números AS que uma rota atravessou a fim alcançar um destino. Um AS_SET é um conjunto matemático pedido {} de todos os AS que foram atravessados. A seção do CIDR exemplo 2 (conjunto as) deste documento fornece um exemplo do AS_SET.

No exemplo nesta seção, o RTB anuncia a rede 190.10.0.0 no AS200. Quando essa rota atravessa o AS300, o RTC adiciona seus próprios número AS à rede. Assim quando 190.10.0.0 alcança o RTA, a rede tem dois AS os números anexos: primeiro 200, então 300. Para o RTA, o trajeto para alcançar 190.10.0.0 é (300, 200).

O mesmo processo aplica-se a 170.10.0.0 e a 180.10.0.0. O RTB tem que tomar o trajeto (300, 100); O RTB atravessa o AS300 e AS100 a fim de alcançar 170.10.0.0. O RTC tem que atravessar o trajeto (200) a fim alcançar 190.10.0.0 e o trajeto (100) a fim alcançar 170.10.0.0.

Atributo de origem

A origem é um atributo imperativo que defina a origem da informação de caminho. O atributo de origem pode assumir três valores:

IGP — A informação de alcançabilidade da camada de rede (NLRI) é interior ao até à data das origens. Isto acontece normalmente quando você emite o **comando bgp network**. Um i na tabela de BGP indica o IGP.

EGP - O NLRI é instruído através do Protocolo de Gateway Exterior (EGP). Um e na tabela de BGP indica o EGP.

INCOMPLETO - O NLRI é desconhecido ou instruído através de outros meios. INCOMPLETO ocorre geralmente quando você redistribui rotas de outros protocolos de roteamento no BGP e a origem da rota está incompleta. Um ? na tabela de BGP indica INCOMPLETO.

```
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
```
O RTA alcança 170.10.0.0 através de 300 i. O "300 i" significa que o próximo trajeto AS é 300 e a origem da rota é IGP. O RTA igualmente alcança 190.10.50.0 através do i. Este "i" significa que a entrada está no mesmos AS e a origem é o IGP. O RTE alcança 150.10.0.0 através de 100 i. O "100 i" significa que o AS seguinte é 100 e a origem é IGP. O RTE também alcança 190.10.0.0 através de 100?. Os "100?" significa que o próximo AS é 100 e que a origem está incompleta e vem de uma rota estática.

Atributo de próximo salto BGP

O atributo de próximo salto BGP é o endereço IP do salto seguinte a usar-se a fim de alcançar um determinado destino.

Para o eBGP, o salto seguinte é sempre o endereço IP do vizinho que o **comando neighbor** especifica. No exemplo nesta seção, o RTC anuncia 170.10.0.0 ao RTA com um salto seguinte de 170.10.20.2. O RTA anuncia 150.10.0.0 ao RTC com um salto seguinte de 170.10.20.1. Para o iBGP, o protocolo indica que o salto seguinte que o eBGP anuncia deve ser levado no iBGP. Devido a esta regra, o RTA anuncia 170.10.0.0 a seu peer RTB do iBGP com um salto seguinte de 170.10.20.2. Assim, de acordo com o RTB, o salto seguinte para alcançar 170.10.0.0 é 170.10.20.2 e *não* 150.10.30.1.

Certifique-se de que o RTB pode alcançar 170.10.20.2 através do IGP. Se não, o RTB derruba pacotes com destino 170.10.0.0 porque o endereço de próximo salto é inacessível. Por exemplo, se o RTB executa o iGRP, você pode igualmente executar o iGRP na rede RTA 170.10.0.0. Você quer fazer a voz passiva do iGRP no link ao RTC de modo que o BGP seja somente trocado.

```
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: O RTC anuncia 170.10.0.0 ao RTA com um salto seguinte igual a 170.10.20.2.

Nota: O RTA anuncia 170.10.0.0 ao RTB com um salto seguinte igual a 170.10.20.2. O salto seguinte do eBGP é levado em iPBP.

Tome o cuidado especial quando você trata das redes de multia-cesso e de multi-acesso sem banda larga (NBMA). As seções Salto Seguinte BGP (redes de multi-acesso) e Salto Seguinte BGP (NBMA) fornecem mais detalhes.

Salto seguinte BGP (redes multi-acesso)

Este exemplo mostra como o salto seguinte se comporta em uma rede de multi-acesso tal como Ethernet.

Suponha que o RTC e o RTD no AS300 executam o OSPF. O RTC executa o BGP com RTA. O RTC pode alcançar a rede 180.20.0.0 através de 170.10.20.3. Quando o RTC envia uma atualização BGP ao RTA referente ao 180.20.0.0, usa o RTC como o salto seguinte 170.10.20.3. O RTC não usa seu próprio endereço IP, 170.10.20.2. O RTC usa este endereço porque a rede entre o RTA, o RTC, e o RTD é uma rede de multi-acesso. O RTA usa o RTD como um salto seguinte para alcançar 180.20.0.0 é mais apreciável do que o salto extra através do RTC.

Nota: O RTC anuncia 180.20.0.0 ao RTA com um salto seguinte 170.10.20.3.

Se o meio comum ao RTA, ao RTC, e ao RTD não é multi-accesso, mas NBMA, ocorrem complicações adicionais.

Salto seguinte BGP (NBMA)

O meio comum aparece como uma nuvem no diagrama. Se o meio comum é um relé de tramas ou qualquer nuvem NBMA, o comportamento exato é como se você tenha uma conexão de Ethernet. O RTC anuncia 180.20.0.0 ao RTA com um salto seguinte de 170.10.20.3.

O problema é que o RTA não tem Circuitos Virtuais Diretos Permanentes (PVC) ao RTD e não pode alcançar o salto seguinte. Neste caso, a distribuição falha.

O **comando next-hop-self** remedeia esta situação.

comando next-hop-self

Para situações com o salto seguinte, como no exemplo do salto seguinte BGP (NBMA), você pode usar o **comando next-hop-self**. A sintaxe é:

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

O **comando next-hop-self** permite que você force o BGP para usar um endereço IP específico como o salto seguinte.

Para o exemplo do salto seguinte BGP (NBMA), esta configuração resolve o problema:

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

O RTC anuncia 180.20.0.0 com um salto seguinte igual a 170.10.20.2.

Backdoor de BGP

Neste diagrama, RTA e RTC executam o eBGP. RTB e RTC executam o eBGP. O RTA e o RTB executam algum tipo do IGP, seja RIP, IGRP, ou um outro protocolo. Por definição, as atualizações do eBGP têm uma distância de 20, que seja menos do que as distâncias IGP. As distâncias padrão são:

- 120 para o RIP
- 100 para o IGRP

90 para o EIGRP

110 para o OSPF

O RTA recebe atualizações sobre 160.10.0.0 através de dois protocolos de roteamento:

eBGP com uma distância de 20

IGP com uma distância que seja maior que 20

Por padrão, o BGP tem estas distâncias:

Distância externa - 20

Distância interna - 200

Distância local - 200

Mas você pode usar o **comando distance** para mudar as distâncias padrão:

```
distance bgp external-distance internal-distance local-distance
```

O RTA escolhe o eBGP através do RTC devido à distância mais curta.

Se você quer que o RTA aprenda sobre 160.10.0.0 através de RTB (IGP), você tem duas opções:

Mude a distância externa do eBGP ou da distância IGP.

Nota: Esta mudança não é recomendada.

Use o BGP backdoor.

O BGP backdoor faz à rota IGP a rota preferida.

Emita o comando network address backdoor.

A rede configurada é a rede que você quer alcançar através do IGP. Para o BGP, esta rede obtém o mesmo tratamento que localmente uma rede atribuída, a não ser que as atualizações BGP não anunciem esta rede.

```
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
```
A rede 160.10.0.0 é tratada como uma entrada local, mas não anunciada como uma entrada da rede normal.

O RTA aprende 160.10.0.0 do RTB através do EIGRP com distância 90. O RTA igualmente aprende o endereço do RTC através do eBGP com distância 20. Normalmente o eBGP é a preferência, mas devido ao **comando network backdoor**, o EIGRP é a preferência.

Sincronização

Antes da discussão de sincronização, olhe este cenário. O RTC no AS300 envia atualizações sobre 170.10.0.0. RTA e RTB executam o iBGP, assim que RTB obtém a atualização e pode alcançar 170.10.0.0 através do salto seguinte 2.2.2.1. Recorde que o salto seguinte está levado através do iBGP. A fim de alcançar o salto seguinte, o RTB deve enviar o tráfego ao RTE.

Suponha que o RTA não tem a rede redistribuída 170.10.0.0 no IGP. Neste momento, o RTE não tem nenhuma ideia que 170.10.0.0 existe.

Se o RTB começa anunciar ao AS400 que o RTB pode alcançar 170.10.0.0, tráfego que vem do RTD ao RTB com destino 170.10.0.0 com fluxo de entrada e queda no RTE.

Estados de sincronização que, se seu AS passa o tráfego de outro AS para um terceiro AS, o BGP deve anunciar uma rota antes que todos os roteadores no seu AS aprendam sobre a rota através do IGP. O BGP espera até que o IGP propague a rota dentro do AS. Então, o BGP anuncia a rota aos peers externos.

No exemplo nesta seção, o RTB espera para ouvir sobre 170.10.0.0 através do IGP. Então, o RTB começa a enviar a atualização ao RTD. Você pode fazer o RTB pensar que o IGP propagou a informação se você adiciona uma rota estática no RTB esses pontos a 170.10.0.0. Certifique-se de que outros roteadores podem alcançar 170.10.0.0.

Desabilite a sincronização

Em alguns casos, você não precisa sincronização. Se você não passa o tráfego de um AS diferente por seu AS, você pode desabilitar a sincronização. Você pode igualmente desabilitar a sincronização se todos os roteadores no seu AS executam o BGP. Desabilitar essa característica pode permitir você leve menos rotas em seu IGP e permita que o BGP convirja mais rapidamente.

A desabilitação de sincronização não é automática. Se todos seus roteadores no AS executem o BGP e você não executar o IGP, não há como o roteador saber. Seu roteador espera indefinidamente por uma atualização IGP sobre uma determinada rota antes que o roteador envie a rota aos peer externos. Você tem que desabilitar a sincronização manualmente neste caso de modo que o roteamento possa funcionar corretamente:

```
router bgp 100
no synchronization
```
Nota: Certifique-se de que você emita o **comando clear ip bgp address** para zerar a sessão.

```
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 ponderação

O atributo de ponderação é um atributo das Cisco-definições. Este atributo usa o peso para selecionar um melhor caminho. O peso é atribuído localmente ao roteador. O valor faz somente sentido ao roteador específico. O valor não é propagado nem é levado por algumas das atualizações da rota. Um peso pode ser um número de 0 a 65.535. Os trajetos que o roteador origina têm um peso de 32.768 por padrão, e outros trajetos têm um peso de 0.

As rotas com um valor de um peso mais alto têm a preferência quando existem rotas múltiplas com mesmo destino. Olhe o exemplo nesta seção. O RTA aprendeu sobre a rede 175.10.0.0 do AS4. O RTA propaga a atualização ao RTC. O RTB igualmente aprendeu sobre a rede 175.10.0.0 do AS4. O RTB propaga a atualização ao RTC. O RTC agora tem duas maneiras de alcançar 175.10.0.0 e tem que decidir qual a maneira de ir. Se você ajusta o peso das atualizações no RTC que vêm do RTA de modo que o peso seja maior do que o peso das atualizações que vêm do RTB, você força o RTC a usar o RTA como um salto seguinte para alcançar 175.10.0.0. Diversos métodos conseguem este peso ajustado:

Use o **comando neighbor**.

vizinho {*IP address* | *grupo de peer*} peso *peer*

Use as listas de acessos AS_PATH.

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

Use mapas de rotas.

```
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.
```

O RTA, que tem um valor de um peso mais alto, tem a preferência como o salto seguinte.

Você pode conseguir o mesmo resultado com IP AS_PATH e listas de filtro.

```
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$ ...
```

Você igualmente pode conseguir o mesmo resultado com o uso dos mapas de rotas.

```
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: Você pode alterar o peso para preferir o trajeto do MPLS VPN BGP com trajeto IGP como um backup.

Nota: Para mais informação, refira este documento da comunidade do apoio de Cisco que descreve como configurar o roteador para ter um caminho preferido em preliminar e em condições de falha e para o redistribuir na recuperação do caminho principal: Preferindo o trajeto do MPLS VPN BGP com backup IGP

Atributo de preferência local

A preferência local é uma indicação ao AS sobre qual trajeto tem a preferência para sair do AS a fim alcançar uma determinada rede. Um trajeto com uma preferência local mais alta tem mais preferência. O valor padrão para a preferência local é 100.

Ao contrário do atributo de ponderação, que somente é relevante ao roteador local, a preferência local é um atributo que os roteadores trocam no mesmos AS.

Você definiu o local de preferência com a introdução do **comando bgp default local-preference value**. Você pode igualmente ajustar a preferência local com mapas de rotas, porque o exemplo nesta seção demonstra:

Nota: É necessário executar um soft reset (isto é, cancele o processo BGP no roteador) para que as mudanças sejam consideradas. A fim cancelar o processo BGP, use o comando **clear ip bgp [soft][in/out] comando onde soft**indica um soft reset sem quebrar a sessão e o **[in/out]** especifica a configuração de entrada ou de saída. Se a **entrada/saída** não for especificada ambas a entrada e a saída serão restauradas.

O **comando bgp default local-preference** ajusta a preferência local nas atualizações fora do roteador que vão aos pares no mesmos AS. No diagrama nesta seção, o AS256 recebe atualizações sobre 170.10.0.0 de dois lados diferentes da organização. A preferência local ajuda-o a determinar que maneira de retirar o AS256 a fim alcançar essa rede. Suponha que o RTD é a preferência do ponto de saída. Esta configuração ajusta a preferência local para as atualizações que vêm do AS300 a 200 e para as atualizações que vêm do AS100 a 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
```

Nesta configuração, o RTC ajusta a preferência local de todas as atualizações a 150. O mesmo RTD ajusta a preferência local de todas as atualizações a 200. Há uma troca da preferência local dentro do AS256. Conseqüentemente, o RTC e o RTD realizam que a rede 170.10.0.0 tem uma preferência local mais alta quando as atualizações vêm do AS300 ao invés do AS100. Todo o tráfego no AS256 que tem que rede enquanto um destino transmite com RTD como um ponto de saída.

O uso dos mapas de rotas fornece mais flexibilidade. No exemplo nesta seção, todas as atualizações que o RTD recebe estão etiquetadas com a preferência local 200 quando as atualizações alcançam o RTD. As atualizações que vêm de AS34 igualmente são etiquetadas com a preferência local de 200. Esta etiqueta pode ser desnecessária. Por este motivo, você pode usar mapas de rotas para especificar as atualizações específicas que precisam de ser etiquetadas com uma preferência local específica. Aqui está um exemplo:

```
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
```

Com esta configuração, toda a atualização que vier do AS300 tem uma preferência local de 200. Todas as outras atualizações, tais como as atualizações que vêm de AS34, têm um valor de 150.

Atributo de métrica

O atributo de métrica igualmente tem o nome MULTI_EXIT_DISCRIMINATOR, MED (BGP4), ou INTER_AS (BGP3). O atributo é uma sugestão aos vizinhos externos sobre a preferência do trajeto no AS. O atributo fornece uma maneira dinâmica para influenciar outro AS de forma a alcançar uma determinada rota quando há múltiplos pontos de entrada naquele AS. Um valor de métrica mais baixo é preferido mais.

Ao contrário da preferência local, a métrica é trocada entre ASs. Um métrico é carregado no AS mas não sai do AS. Quando uma atualização entra no AS com um determinado métrico, esse

métrico está usado para fazer decisões dentro do AS. Quando a mesma atualização passar por um terceiro AS, essa métrica retorna 0. O diagrama nesta seção mostra o grupo métrico. O valor padrão métrico é 0.

A menos que um roteador receba outras direções, o roteador compara a métrica de caminhos dos vizinhos no mesmos AS. Para que o roteador compare o métrico dos vizinhos que vêm de AS diferentes, você precisa emitir um comando de configuração especial always-compare-med BGP no roteador.

Nota: Há dois comandos da configuração de BGP que podem influenciar na seleção de trajetos baseados no Multi-Exit Discriminator (MED). Os comandos são o comando bgp deterministic-med e o comando bgp always-compare-med. Uma introdução do **comando bgp deterministic-med** assegura a comparação da variável MED na escolha da rota quando diferentes peers anunciam no mesmos AS. Uma introdução do **comando bgp always-compare-med** assegura a comparação do MED para trajetos dos vizinhos em AS diferentes. O **comando bgp always-compare-med** é útil quando os provedores de serviço múltiplos ou as empresas concordam com uma política uniforme de ajuste do MED. Refira a como o comando bgp deterministic-med difere do comando bgp always-compare-med para compreender como estes comandos influenciam a seleção de trajeto BGP.

No diagrama nesta seção, o AS100 obtém a informação sobre a rede 180.10.0.0 através de três roteadores diferentes: RTC, RTD, e RTB. O RTC e o RTD estão no AS300, e o RTB está no AS400.

Neste exemplo, a comparação Como-PATH no RTA pelo comando bgp bestpath as-path ignore é ignorada. É configurada para forçar o BGP para cair sobre ao atributo seguinte para a comparação da rota (neste caso métrica ou MED). Se o comando é omitido, o BGP instalará a rota 180.10.0.0 do roteador RTC como aquele tem o Como-PATH o mais curto.

Suponha que você ajustou o métrico que vem do RTC a 120, o métrico que vem do RTD a 200, e o métrico que vem do RTB aos 50. Por padrão, um roteador compara o medidor que vem dos vizinhos no mesmos AS. Conseqüentemente, o RTA pode somente comparar o métrico que vem do RTC ao métrico que vem do RTD. O RTA escolhe o RTC como o melhor salto seguinte porque 120 é menos que 200. Quando o RTA obtém uma atualização do RTB com métrica 50, o RTA não pode comparar o métrico a 120 porque o RTC e o RTB estão em AS diferentes. O RTA deve escolher baseado em alguns outros atributos.

A fim forçar o RTA para comparar o métrico, você deve emitir o **comando bgp always-compare-med** no RTA. Estas configurações ilustram este processo:

```
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
```

Com estas configurações, o RTA escolhe o RTC como o salto seguinte, com consideração do terno de que todos atributos restantes são os mesmos. A fim incluir o RTB na comparação métrica, você deve configurar o RTA desta maneira:

```
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
```

Neste caso, o RTA escolhe o RTB como o melhor salto seguinte a fim de alcançar a rede 180.10.0.0.

Você pode também ajusta o métrico durante a redistribuição das rotas no BGP se você emite o **comando default-metric number**.

Suponha que, no exemplo nesta seção, o RTB injeta uma rede através da estática no AS100. Está aqui a configuração:

```
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 comunidade

O atributo de comunidade é um atributo transitivo opcional na escala de 0 a 4.294.967.200. O atributo de comunidade é uma maneira de agrupar destinos em uma determinada comunidade e de aplicar decisões de roteamento de acordo com aquelas comunidades. As decisões de roteamento são aceitar, preferir, e redistribuir, entre outros.

Você pode usar mapas de rotas para ajustar os atributos de comunidade. O comando configurar mapa de rotas tem esta sintaxe:

set community *community-number* **[additive] [***well-known-community***]**
Alguns predefinidos, as comunidades bem conhecidas para o uso neste comando são:

no-export - Não anuncie aos peers eBGP. Mantenha esta rota dentro de um AS.

no-advertise - Não anuncie esta rota a nenhum peer, interno ou externo.

Internet - Anuncie esta rota à comunidade da internet. Todo o roteador pertence a esta comunidade.

local-as - Use nos cenários de confederação para impedir a transmissão dos pacotes fora do local AS.

Estão aqui dois exemplos dos mapas de rotas que ajustam à comunidade:

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

  ou

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

Se você não ajusta a palavra-chave **aditiva**, 200 substituem toda a velha comunidade já existente. Se você usa a palavra-chave de aditivo, uma adição de 200 à comunidade ocorre. Mesmo se você ajusta o atributo de comunidade, este atributo não transmite aos vizinhos por padrão. A fim enviar o atributo a um vizinho, você deve usar este comando:

neighbor {*ip-address* **|** *peer-group-name***} send-community**
Aqui está um exemplo:

```
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
```

No Cisco IOS Software Release 12.0 e Mais Recente, você pode configurar as comunidades em três formatos diferentes: decimal, hexadecimal, e AA: NN. Por padrão, o Cisco IOS Software usa o formato decimal mais velho. A fim de configurar e indicar no AA: O NN, emite o **comando ip bgp-community new-format global configuration**. A primeiro parte de AA: O NN representa o número AS, e a segunda parte representa um número 2-byte.

Aqui está um exemplo:

Sem o comando ip bgp-community new-format na configuração global, uma introdução do comando show ip bgp 6 0 0 0 indica o valor do atributo de comunidade no formato decimal. Neste

exemplo, o valor do atributo de comunidade 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
```

Agora, emita o **comando ip bgp-community new-format** global neste roteador.

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

Com o **comando ip bgp-community new-format global configuration**, os indicadores do valor de comunidade no AA: Formato do NN. O valor aparece como 100:20 na saída do **comando show ip bgp 6 0 0 0 n**este exemplo:

```
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
```

Estudos de Caso do BGP 3

Filtração BGP

Um número de métodos diferentes do filtro permitem que você controle a emissão e receba-a das atualizações BGP. Você pode filtrar atualizações BGP com informação de rota como base, ou com informação de caminho ou comunidades como base. Todos os métodos conseguem os mesmos resultados. A escolha de um método sobre um outro método depende da configuração de rede específica.

Filtragem de rota

A fim de restringir a informação de roteamento que o roteador aprende ou anuncia, você pode filtrar o BGP com o uso das atualizações de roteamento para ou de um vizinho específico. Você define uma lista de acessos e aplica a lista de acessos às atualizações de ou para um vizinho. Emita este comando no modo de configuração do roteador:

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

Neste exemplo, o RTB origina a rede 160.10.0.0 e envia a atualização ao RTC. Se o RTC quer parar a propagação das atualizações ao AS100, você deve definir uma lista de acessos para filtrar aquelas atualizações e para aplicar a lista de acessos durante uma comunicação com o 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.
```

O uso das listas de acessos é um pouco complicado quando você trata com super-redes que pode causar alguns conflitos.

Suponha que, no exemplo nesta seção, o RTB tem sub-redes diferentes de 160.10.x.x. Seu objetivo é filtrar atualizações e anunciar somente 160.0.0.0/8.

Nota: A notação de /8 significa que você usa 8 bit da máscara de sub-rede, que parte da extrema esquerda do endereço IP. Este endereço é equivalente a 160.0.0.0 255.0.0.0.

O comando access-list 1 permite 160.0.0.0 0.255.255.255 permite 160.0.0.0/8, 160.0.0.0/9, e assim por diante. A fim de restringir a atualização a somente 160.0.0.0/8, você deve usar uma lista de acesso estendida deste 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 somente.

Refira como obstruir umas ou várias redes de um BGP peer para exemplos de configurações em como filtrar redes dos BGP peers. O método usa o **comando distribute-list** com uma lista de controles de acesso padrão e prolongadas (ACL), assim como a filtração da lista de prefixo.

Filtração do trajeto

Um outro tipo de filtração é filtração do trajeto.

Você pode especificar uma lista de acessos de entrada e atualizações de saída com uso do BGP AS informação de trajetos. No diagrama nesta seção, você pode obstruir atualizações sobre 160.10.0.0 de modo que não vá ao AS100. Para obstruir as atualizações, defina uma lista de acessos no RTC que previna a transmissão ao AS100 de todas as atualizações que originarem do AS200. Execute estes 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 exemplo para o RTC envia atualizações sobre 160.10.0.0 ao 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 .*
```

O **comando access-list 1** neste exemplo força a recusa de todas as atualizações com informação de caminho que começa com 200 e termina com 200. O ^200$ no comando é uma “expressão regular”, em que o ^ significa que “começa com” e $ significa “termina com”. Desde que o RTB envia atualizações sobre 160.10.0.0 com informação de trajeto que começa com 200 e termina com 200, a combinação das atualizações da lista de acessos. A lista de acessos rejeita estas atualizações.

O . * é uma outra expressão regular em que. significa “todo o caractere” e * significa “a repetição desse caractere”. Assim. * representa toda a informação de caminho, a qual é necessária para permitir a transmissão de todas as atualizações restantes.

O que acontece se, em vez de usar ^200$, você usar ^200? Com um AS400, como no diagrama nesta seção, as atualizações que o AS400 origina têm a informação de caminho do formulário (200, 400). Nesta informação de caminho, 200 são primeiros e 400 são últimos. Estas atualizações combinam a lista de acessos ^200 porque a informação de caminho começa com 200. A lista de acessos impede a transmissão destas atualizações ao RTA, que não é a exigência.

A fim de verificar se você executou a expressão regular correta, emita o comando show ip bgp regexp regular-expression. Este comando mostra todos os trajetos que combinaram a configuração da expressão regular.

AS Regular Expression

Esta seção explica a criação de uma expressão regular.

Uma expressão regular é um padrão para combinar contra uma série de entrada. Quando você constrói uma expressão regular, você especifica uma série que a entrada deva combinar. No caso do BGP, você especifica uma corda que consista na informação de caminho que uma entrada deve combinar.

No exemplo na seção Filtro de Caminho , você especificou a série ^200$. Você quis a informação de caminho que vem dentro das atualizações para combinar a série a fim de tomar uma decisão.

Uma expressão regular compreende:

Faixa

Uma escala é uma seqüência dos caracteres dentro dos suportes quadrados esquerdos e adequados. Um exemplo é [abcd].

Átomo

Um átomo é um único caractere. Aqui estão alguns exemplos:

.

O . combina todo caractere único.

^

O ^ combina o começo da série de entrada.

$

O $ combina a extremidade da série de entrada.

\

O \ combina o caractere.

-

O _combina uma vírgula (,), o colchete esquerdo ({), o colchete direito (}), o começo da série de entrada, a extremidade da série de entrada, ou um espaço.

Parte

Uma parte é um destes símbolos, que segue um átomo:

`*`

O * combina 0 ou mais seqüências do átomo.

`+`

O + combina 1 ou mais seqüências do átomo.

`?`

O ? combina o átomo ou a série nula.

Ramo

Um ramo é 0 ou peças mais concatenadas.

Estão aqui alguns exemplos das expressões regulares:

`a*`

Esta expressão indica toda a ocorrência da letra “a”, que não inclui nenhum.

`a+`

Esta expressão indica que pelo menos uma ocorrência da letra “a” deve esta presente.

`ab?a`

Esta expressão combina o “aa” ou o “aba”.

`_100_`

Esta expressão significa através do AS100.

`_100$`

Esta expressão indica uma origem do AS100.

`^100 .*`

Esta expressão indica a transmissão do AS100.

`^$`

Esta expressão indica origens deste AS.

Refira a utilização de expressões regulares no BGP para exemplos de configuração do filtragem de expressão regular.

Filtração da comunidade do BGP

Este documento cobriu a filtragem de rota e a filtração do AS-path. Um outro método é filtração da comunidade. A seção atributo de comunidade discute a comunidade, e esta seção fornece alguns exemplos de como usar a comunidade.

Neste exemplo, você quer que o RTB ajuste o atributo de comunidade às rotas de BGP que o RTB anuncia tais que o RTC não propaga estas rotas aos peers externos. Use o **atributo de comunidade não exportação**.

```
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 exemplo usa o **comando route-map setcommunity** a fim ajustar a comunidade a **não exportação**.

Nota: O **comando neighbor send-community** é necessário a fim enviar este atributo ao RTC.

Quando o RTC obtém as atualizações com o atributo NO_EXPORT, o RTC não propaga as atualizações ao peer RTA externo.

Neste exemplo, o RTB ajustou o atributo de comunidade ao **aditivo 100 200**. Esta ação adiciona o valor 100 200 a todo o valor de comunidade existente antes da transmissão ao 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
```

Uma lista de comunidade é um grupo das comunidades que você usa em uma cláusula de **combinação** do mapa de rotas. A lista de comunidade permite que você filtre ou ajuste atributos com diferentes listas dos números da comunidade como base.

ip community-list *community-list-number* **{permit | deny}** *community-number*

Por exemplo, você pode definir este mapa de rotas, **combinar-com-comunidade**:

```
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.
```

Você pode usar a lista de comunidade a fim filtrar como base ou ajustar determinados parâmetros, como o peso e métrico, em determinadas atualizações com o valor de comunidade. No segundo exemplo nesta seção, o RTB enviou atualizações ao RTC com uma comunidade de 100 200. Se o RTC quer ajustar como base o peso com aqueles valores, você pode fazer isto:

```
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
```

Neste exemplo, toda a rota que tiver 100 no atributo de comunidade combina a lista 1. O peso desta rota é ajustado a 20. Alguma rota que tiver somente 200 como a comunidade combina a lista 2 e tiver um peso de 20. As palavras-chave **exatas do** estado que a comunidade consiste em somente 200 e nada mais. A última lista de comunidade garante que outras atualizações não deixam cair. Recorde que qualquer coisa que não combina cai, por padrão. A palavra-chave internet indica todas as rotas porque todas as rotas são membros da comunidade da internet.

Refira a utilização BGP Valores de Comunidade para Controlar Política de Roteamento em uma Revide de Provedor Upstream para mais informação.

Vizinhos de BGP e mapas de rotas

Você pode usar o **comando neighbor** conjuntamente com mapas de rotas aos parâmetros do filtro ou do grupo em entrante e em atualizações de saída.

Os mapas de rotas associados com a **declaração vizinha** não têm nenhum efeito em atualizações recebidas quando você combina baseado no endereço IP:

neighbor *ip-address* **route-map** *route-map-name*

Suponha que, no diagrama nesta seção, você quer que o RTC aprenda do AS200 sobre as redes que são locais ao AS200 e nada mais. Também, você quer ajustar o peso nas rotas aceitadas a 20. Use uma combinação de listas de acessos do **vizinho** e **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$
```

Todas as atualizações que originarem do AS200 têm a informação de caminho que começa com 200 e termina com 200. Estas atualizações são permitidas. Qualquer outra atualização cai.

Suponha que você quer:

Uma aceitação das atualizações que originam do AS200 e têm um peso de 20

A gota das atualizações que originam do AS400

Um peso de 10 para outras atualizações

```
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 indicação ajusta um peso de 20 para as atualizações que são locais ao AS200. A declaração igualmente ajusta um peso de 10 para as atualizações que estão além do AS400, e deixa cair as atualizações que vêm do AS400.

Uso do comando set as-path prepend

Em algumas situações, você deve manipular a informação de caminho a fim de manipular o processo de decisão BGP. O comando que você usa com um mapa de rotas é:

set as-path prepend *as-path# as-path#*

Suponha que, no diagrama na seção vizinhos de BGP e nos mapas de rotas da , o RTC anuncia sua própria rede 170.10.0.0 a dois AS, AS100S e AS200S diferentes. Quando a informação é propagada ao AS600, o Roteadores no AS600 tem a informação de alcançabilidade de rede sobre 170.10.0.0 através de duas rotas diferentes. A primeira rota é através do AS100 com trajeto (100, 300), e segundo é através do AS400 com trajeto (400, 200, 300). Se todos atributos restantes são os mesmos, o AS600 escolhe o caminho mais curto e escolhe a rota através do AS100.

O AS300 obtém todo o tráfego através do AS100. Se você quer influenciar esta decisão da extremidade do AS300, você pode fazer o trajeto com o AS100 parecer ser mais longo do que o trajeto que atravessa o AS400. Você pode fazer este se você prepend números AS à informação do trajeto existente que está anunciada ao AS100. Uma prática comum é repetir seu próprio número AS desta maneira:

```
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
```

Devido a esta configuração, o AS600 recebe atualizações sobre 170.10.0.0 através do AS100 com informação de caminho de: (100, 300, 300, 300). Esta informação de caminho é mais longa do que (400, 200, 300) esse AS600 recebido do AS400.

Grupos de paridade BGP

Um grupo de paridade BGP é um grupo de vizinhos de BGP com as mesmas políticas de atualização. Os mapas de rotas, distribuem lista, e as listas de filtro ajustam tipicamente políticas da atualização. Você não define as mesmas políticas para cada vizinho separado; em lugar de, você define um nome de grupo de paridade e atribui estas políticas ao grupo de paridade.

Os membros do grupo de paridade herdam todas as opções de configuração do grupo de paridade. Você pode igualmente configurar membros para cancelar estas opções se as opções não afetam atualizações de saída. Você pode somente cancelar as opções que são ajustadas na entrada.

A fim definir um grupo de paridade, emita este comando:

neighbor *peer-group-name* **peer-group**

Este exemplo aplica aos grupos de paridade BGP vizinhos internos e 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 configuração define um grupo de paridade com o nome **internalmap**. A configuração define algumas políticas para o grupo, tal como um mapa de rotas **SETMETRIC** para ajustar o métrico a 5 e a duas listas de filtro diferentes, 1 e 2. A configuração aplica o grupo de paridade a todos os vizinhos internos, RTE, RTF, e RTG. Também, a configuração define uma lista de filtro separada 3 para o vizinho RTE. Esta lista de filtro cancela a lista de filtro 2 interna do grupo de paridade.

Nota: Você pode somente cancelar as opções que afetam atualizações de entrada.

Agora, olhe em como você pode usar a grupo de paridade com vizinhos externos. Com o mesmo diagrama nesta seção, você configura o RTC com um **externalmap do** grupo de paridade e aplica o grupo de paridade aos vizinhos 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: Nestas configurações, você define as indicações do **remoto-como** fora do grupo de paridade porque você deve definir um AS externo diferente. Também, você cancela as atualizações de entrada de vizinho 1.1.1.2 com a atribuição da lista de filtro 3.

Para obter mais informações sobre dos grupo de paridade, refira grupos de BGP peer.

Nota: No Cisco IOS Software Release 12.0(24)S, Cisco introduziu a característica dos grupos de paridade da atualização dinâmica BGP. A característica está disponível no Cisco IOS Software Release mais recente também. A característica introduz um algoritmo novo que calcula dinamicamente e aperfeiçoa grupos da atualização de vizinhos que compartilham das mesmas políticas de saída. Estes vizinhos podem compartilhar as mesmas mensagens de atualização. Nas versões anteriores do Cisco IOS Software, o grupo de mensagens da atualização BGP era com base em configurações do grupo de paridade. Este método para agrupar atualizações limitou políticas de saída e configurações de sessão específicas. A característica do grupo de paridade da atualização dinâmica BGP separa a réplica do grupo da atualização da configuração do grupo de paridade. Esta separação melhora o tempo de convergência e a flexibilidade da configuração vizinha. Refira o grupo de paridade da atualização dinâmica BGP para mais detalhes.

Estudos de Caso do BGP 4

CIDR e endereços agregados

Uma das principais melhoras do BGP4 sobre BGP3 é o roteamento entre domínios sem classe (CIDR). O CIDR ou supernetting são uma maneira nova de olhar endereços IP. Com CIDR, não há nenhuma noção das classes, tais como a classe A, B, ou o C. por exemplo, rede 192.213.0.0 era uma vez uma rede de classe C ilegal. Agora, a rede é um super-rede legal, 192.213.0.0/16. O "16" representa o número de bit na máscara de sub-rede, quando você conta da extrema

esquerda do endereço IP. Esta representação é similar a 192.213.0.0 255.255.0.0.

Você usa agregados a fim minimizar o tamanho das tabelas de roteamento. A agregação é o processo que combina as características de diversas rotas diferentes de tal maneira que a propaganda de uma rota única é possível. Neste exemplo, o RTB gera a rede 160.10.0.0. Você configura o RTC para propagar uma super-rede dessa rota 160.0.0.0 ao 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
```

O RTC propaga o endereço agregado 160.0.0.0 ao RTA.

Comandos aggregate

Há um amplo intervalo dos comandos aggregate. Você deve compreender como cada um trabalha a fim ter o comportamento da agregação que você deseja.

O primeiro comando é esse do exemplo na seção CIDR e em endereços agregados:

aggregate-address *address-mask*

Este comando anuncia rota do prefixo e todas as rotas mais específicas. O comando aggregate-address 160.0.0.0 propaga uma rede adicional 160.0.0.0 mas não impede a propagação de 160.10.0.0 ao RTA. O resultado é a propagação de redes 160.0.0.0 e de 160.10.0.0 ao RTA, que é a propaganda do prefixo e da rota mais específica.

Nota: Você não pode agregar um endereço se você não tem uma rota mais específica desse endereço na tabela de roteamento de BGP.

Por exemplo, o RTB não pode gerar um agregado para 160.0.0.0 se o RTB não tem uma entrada dos mais específico de 160.0.0.0 na tabela de BGP. Uma injeção da rota dos mais específico na tabela de BGP é possível. A injeção da rota pode ocorrer através de:

- Atualizações recebidas de outros AS
- Redistribuição de um IGP ou de uma estática no BGP
- O **comando network**, por exemplo, **rede 160.10.0.0**

Se você quer que o RTC propague a rede 160.0.0.0 somente e **não** a rota mais específica, emita este comando:

aggregate-address *address mask* **summary-only**
Este comando anuncia somente o prefixo. O comando suprime todas as rotas mais específicas.

O comando aggregate 160.0.0.0 255.0.0.0 summary-only propaga a rede 160.0.0.0 e suprime a rota 160.10.0.0 mais específica.

Nota: Se você agrega uma rede que injete em seu BGP através da **instrução de rede**, a entrada de rede injeta sempre em atualizações BGP. Esta injeção ocorre mesmo que você use o **comando aggregate summary-only**. O exemplo na seção CIDR Exemplo 1 discute esta situação.

aggregate-address *address-mask* **as-set**
Este comando anuncia o prefixo e as rotas mais específicas. Mas o comando inclui a informação do **recurso na** informação de caminho das atualizações de roteamento.

aggregate 129.0.0.0 255.0.0.0 as-set
A seção CIDR Exemplo 2 (as-set) discute este comando.

Se você quer suprimir rotas mais específicas quando você faz a agregação, defina um mapa de rotas e aplique o mapa de rotas aos agregados. A ação permite que você seja seletivo sobre quais rotas mais específicas para suprimir.

aggregate-address *address-mask* **suppress-map** *map-name*
Este comando anuncia o prefixo e as rotas mais específicas. Mas o comando suprime a propaganda com uma base do mapa de rotas. Suponha que, com o diagrama na seção CIDR e endereços agregados, você quer agregar 160.0.0.0, suprime a rota 160.20.0.0 mais específica, e permitir a propagação de 160.10.0.0. Use este mapa de rotas:

```
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 definição do **mapa de omissões**, há uma supressão das atualizações de todos os pacotes que a lista de acessos permitir.

Então, aplique o mapa de rotas à indicação **agregada**.

```
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
```

Está aqui uma outra variação:

aggregate-address *address-mask* **attribute-map** *map-name*
Este comando permite que você ajuste os atributos, tais como métrico, na altura da emissão dos agregados. A fim de ajustar a origem dos agregados ao IGP, aplique este mapa de rotas ao **comando aggregate attribute-map**:

```
route-map SETMETRIC
set origin igp

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

Para mais informação, refira a compreendendo a agregação de rota em BGP.

CIDR Exemplo 1

Pedido: Permita que o RTB anuncie o prefixo 160.0.0.0 e suprima todas as rotas mais específicas. O problema com este pedido é que a rede 160.10.0.0 é local ao AS200, assim que significa que o AS200 é o autor de 160.10.0.0. Você não pode mandar o RTB gerar um prefixo para 160.0.0.0 sem a geração de uma entrada para 160.10.0.0, mesmo se você usa o **comando aggregate summary-only**. O RTB gera ambas as redes porque o RTB é o autor de 160.10.0.0. Há duas soluções a este problema.

A primeira solução é usar uma rota estática e redistribui-la no BGP. O resultado é que o RTB anuncia o agregado com uma origem de incompleta (?).

```
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
```

Na segunda solução, além da rota estática, você adiciona uma entrada para o **comando network**. Esta entrada tem o mesmo efeito, salvo que a entrada ajusta a origem da atualização ao 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
```

CIDR Exemplo 2 (as-set)

Você usa a indicação **as-set** na agregação para reduzir o tamanho da informação de caminho. Com o **as-set**, o número AS é listado somente uma vez, não importa quantas vezes o número AS apareceu nos caminhos múltiplos que foram agregados. Você usa o **comando aggregate as-set** nas situações em que a agregação da informação causa a perda de informação no que diz respeito ao atributo de trajeto. Neste exemplo, o RTC obtém atualizações sobre 160.20.0.0 do RTA e atualizações sobre 160.10.0.0 do RTB. Suponha que o RTC quer a rede agregada 160.0.0.0/8 e envie a rede ao RTD. O RTD não conhece a origem dessa rota. Se você adiciona a indicação de **as-set agregada**, você força o RTC para gerar a informação de caminho sob a forma de um grupo {}. Esse grupo inclui toda a informação de caminho, independentemente de que trajeto veio primeiramente.

```
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:

O RTC não tem uma indicação de **as-set**. O RTC envia uma atualização 160.0.0.0/8 ao RTD com informação de caminho (300), como se a rota originou do 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}.
```
Os dois assuntos seguintes, confederação BGP e refletores de rota, são para os provedores de serviço da Internet (ISP) que querem um controle adicional da explosão do ibgp peering dentro de seus AS.

Confederação BGP

A implementação de confederação de BGP reduz a malha do iBGP dentro do AS. O truque é dividir um AS em múltiplos ASs e atribuir ao grupo inteiro a uma única confederação. Cada AS sozinho pussui em iBGP engrenado inteiramente e tem conexões a outros AS dentro da confederação. Mesmo que estes AS tenham eBGP peers aos AS dentro da confederação, os AS trocam o roteamento como se usassem o iBGP. Desta maneira, a confederação preserva o salto seguinte, métrico, e a informação de preferência de local. Ao mundo exterior, a confederação parece ser uma única AS.

A fim de configurar uma confederação BGP, emita este comando:

bgp confederation identifier *autonomous-system*

O identificador de confederação é o número AS do grupo de confederação.

A introdução deste comando executa peering entre o AS múltiplo dentro da confederação:

bgp confederation peers autonomous-system [*autonomous-system***]**

Está aqui um exemplo de confederação:

Suponha que você tenha um AS500 que consiste em nove auto-falantes de BGP. Outros auto-

falantes não-BGP também existem, mas você tem somente interesse nos auto-falantes de BGP que têm conexões eBGP a outros AS. Se você quer fazer uma malha completa do iBGP dentro do AS500, você precisa nove conexões de peer para cada roteador. Você precisa oito peers do iBGP e um peer do eBGP aos AS externos.

Se você usa a confederação, você pode dividir o AS500 no AS múltiplo: AS50, AS60, e AS70. Você dá ao identificador de confederação um AS de 500. O mundo exterior vê somente um AS, AS500. Para cada um do AS50, do AS60, e do AS70, você define uma malha cheia de peers do iBGP, e você define a lista de peers da confederação com o **comando bgp confederation peers**.

Está aqui uma configuração de exemplo dos roteadores RTC, RTD, e RTA:

Nota: O RTA não tem nenhum conhecimento do AS50, do AS60, ou do AS70. O RTA tem somente o conhecimento do 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)
```

Refletores de rota

Uma outra solução para a explosão do ibgp peering dentro do AS são os refletores de rota (RR). Enquanto a seção do iBGP demonstra, um auto-falante de BGP não anuncia uma rota que o auto-falante de BGP aprendido através de um outro alto-falante iBGP a um terceiro alto-falante iBGP. Você pode relaxar um pouco esta limitação e fornecer o controle adicional, que permite que um roteador anuncie, ou reflita, rotas ensinadas pelo iBGP a outros alto-falantes iBGP. Esta rota de reflexão reduz o número de peers do iBGP dentro do AS.

Em casos normais, mantenha uma malha completa do iBGP entre o RTA, o RTB, e o RTC dentro do AS100. Se você utiliza o conceito RR, o RTC pode ser elegido como um RR. Desta maneira, o RTC tem um ibgp peering parcial com RTA e RTB. Peering entre o RTA e o RTB não é necessário porque o RTC é um RR para as atualizações que vêm do RTA e do RTB.

neighbor route-reflector-client
O roteador com este comando é o RR, e os vizinhos em que os pontos do comando são os

clientes desse RR. No exemplo, a configuração de RTC tem o **comando neighbor route-reflector-client** que aponta os endereços IP RTA e RTB. A combinação do RR e dos clientes é um "conjunto". Neste exemplo, em formulário RTA, RTB, e RTC um conjunto com um único RR dentro do AS100.

Outros peers do iBGP do RR que não são clientes são "nonclients".

Um AS pode ter mais de um RR. Nesta situação, um RR trata outros RR apenas como todo o outro alto-falante iBGP. Outros RR podem pertencer ao mesmo conjunto (grupo de cliente) ou a outros conjuntos. Em uma configuração simples, você pode dividir o AS em conjuntos múltiplos. Você configura cada RR com outros RR como peer que não é cliente inteiramente em uma topologia em malha. Os clientes não devem peer com os alto-falantes iBGP fora do conjunto do cliente.

Considere este diagrama. Formulário RTA, RTB, e RTC um único conjunto. O RTC é o RR. Para o RTC, o RTA e o RTB são clientes e qualquer outra coisa é um nonclient. Recorde que o **comando neighbor route-reflector-client** aponta para clientes de um RR. O mesmo RTD é o RR para os clientes RTE e RTF. O RTG é um RR em um terceiro conjunto.

Nota: O RTD, RTC, e o RTG é engrenado inteiramente, mas roteadores dentro de um conjunto não são. Quando um RR receber uma rota, as rotas RR como mostras desta lista. Contudo, esta atividade depende do tipo do peer:

Rotas de um peer que não é cliente - Reflete a todos os clientes dentro do conjunto.

Rotas de um peer do cliente - Reflete a todos os peers não cliente e igualmente peer cliente.

Rotas de um peer do eBGP - Envia a atualização a todo o cliente e peer que não é cliente.

Está aqui a configuração de BGP relativa dos roteadores RTC, RTD, e 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
```

Porque há uma reflexão das rotas ensinadas pelo iBGP, pode haver um loop de informação de roteamento. O esquema RR tem alguns métodos para evitar este laço:

originator-id - Este é uns atributos de BGP opcionais, nontransitive com comprimento 4 bytes. Um RR cria este atributo. O atributo leva o Router ID (RID) do autor da rota no local AS. Se, devido à configuração deficiente, a informação de roteamento vem para trás ao autor, a informação está ignorada.

cluster-list - A seção múltiplos RR dentro de um conjunto cobre a lista do conjunto.

RR múltiplos dentro de um conjunto

Geralmente, um conjunto de clientes tem um único RR. Neste caso, o Router ID do RR identifica o conjunto. A fim de aumentar a redundância e evitar pontos de falha únicos, um conjunto pode ter mais de um RR. Você precisa de configurar todos os RR no mesmo conjunto com 4-byte um conjunto ID de modo que um RR possa reconhecer atualizações dos RR no mesmo conjunto.

Uma lista do conjunto é uma seqüência do conjunto ID que a rota passe. Quando um RR reflete uma rota dos clientes RR aos nonclients fora do conjunto, o RR adiciona o cluster local ID à lista do conjunto. Se esta atualização tem uma lista vazia do conjunto, o RR cria um. Com este atributo, um RR pode identificar se a informação de roteamento tem o loop ao mesmo conjunto devido à configuração deficiente. Se o cluster local ID é encontrado na lista do conjunto, a propaganda é ignorada.

No diagrama nesta seção, o RTD, o RTE, o RTF, e RTH pertencem a um conjunto. o RTD e RTH são RR para o mesmo conjunto.

Nota: Há uma redundância porque RTH engrenou o peering inteiramente com todos os RR. Se o RTD vai para baixo, RTH toma o lugar do RTD.

Está aqui a configuração de RTH, de RTD, de RTF, e de 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: Você não precisa do comando bgp cluster-id para o RTC porque somente um RR existe nesse conjunto.

Nota importante: Esta configuração não usa grupo de paridade. Não use grupos de paridade se os clientes dentro de um conjunto não têm peers diretos entre um outros do iBGP e os clientes trocam atualizações com o RR. Se você configura grupos de paridade, uma retirada potencial à fonte de uma rota no RR transmite a todos os clientes dentro do conjunto. Esta transmissão pode causar problemas.

A reflexão do cliente-à-cliente BGP do subcomando de roteador é permitida por padrão no RR. Se você desliga a reflexão do cliente-à-cliente BGP no RR e você faz o BGP peering redundante entre os clientes, você pode com segurança usar peer-group. Refira limitações dos peer-group para mais informação.

RR e auto-falantes de BGP convencionais

Um AS pode ter os auto-falantes de BGP que não compreendem o conceito dos RR. Este

documento chama estes roteadores convencionais auto-falantes de BGP. O esquema RR permite que tais auto-falantes de BGP convencionais coexistam. Estes roteadores podem ser membros de um grupo de cliente ou um grupo não cliente. A existência destes roteadores permite fácil e gradual migração do modelo atual do iBGP ao modelo RR. Você pode começar a criar conjuntos se você configura um roteador único como outros RRs e RR clientes normal iBGP peers. Então, você pode criar mais conjuntos gradualmente.

Neste diagrama, o RTD, o RTE, e o RTF têm o conceito da reflexão de rota. O RTC, o RTA, e o RTB são roteadores “convencionais”. Você não pode configurar estes roteadores como RR. Você pode fazer a malha normal do iBGP entre estes roteadores e RTD. Mais tarde, quando você está pronto para fazer o upgrade, você pode fazer a RTC um RR com clientes RTA e RTB. Os clientes não têm que compreender o esquema da reflexão de rota; somente os RR exigem upgrade.

Está aqui a configuração do RTD e do 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
```

Quando você estiver pronto para promover o RTC e fazer o RTC um RR, para remover a malha cheia do iBGP e para mandar o RTA e o RTB se transformar clientes do RTC.

Evite o laço da informação de roteamento

Até agora, este documento mencionou dois atributos que você pode usar para impedir dar laços potencial da informação: **autor-ID** e **cluster-list**.

Outro meio para controlar os laços é por mais limitações sobre a cláusula do **set** de mapas de rota externa. A cláusula **set** para mapas de rota externa não afeta as rotas que refletem os peers do iBGP.

Você pode igualmente pôr mais limitações sobre o nexthop-**auto**, que é uma opção de configuração por vizinho. Quando você usa o **nexthop-self** no RR, a cláusula afeta somente o salto seguinte de rotas aprendidas do eBGP porque o salto seguinte de rotas refletidas não deve ser mudado.

Route Flap Dampering

O Cisco IOS Software Release 11.0 introduziu um silenciador de rota. O silenciador de rota é um mecanismo para minimizar a instabilidade de causas de variabilidade. O silenciador de rota igualmente reduz a oscilação sobre a rede. Você define critérios para identificar rotas de comportamento deficiente. Uma rota que bata obtém uma pena de 1000 para cada flap. Assim que a pena cumulativa alcançar "suprimir limite" predefinido, ocorre a supressão do anúncio da rota. A pena deteriora exponencial baseado em uma estadia preconfigured "meia-vida". Uma vez que as diminuições da pena abaixo "de um limite de reuso" predefinido, remova a supressão do anúncio da rota ocorrem.

O retardar da rota não se aplica às rotas que são externos ao AS e aprendido através do iBGP. Desta maneira, o retardar da rota evita uma penalidade mais elevada para os pares do iBGP para as rotas externos ao AS.

A pena deteriora em uma granularidade de 5 segundos. Remova a supressão das rotas está em uma granularidade de 10 segundos. O roteador mantem a informação de umedecimento até que a pena se transforme menos do que a metade "do limite de reuso". Nesse ponto, o roteador remove a informação.

Inicialmente, umedecer-se por padrão. Se há uma necessidade, esta característica pode ser dada à habilitação da opção no futuro. Estes comandos controlam o silenciamento da rota:

umedecimento BGP - Gira sobre o umedecimento.

não silenciamento BGP - Desliga o silenciamento.

silenciamento bgp *dempo de meia-vida* - Muda o tempo de meia-vida.

O comando A que ajusta todos os parâmetros ao mesmo tempo é:

silenciamento bgp *reuso de meia-vida suprime tempo-máximo-supressão*

Esta lista detalha a sintaxe:

meia-vida-tempo - A escala é 1 - 45 minutos, e a opção padrão é 15 minutos.

valor-reuso - A escala é 1 - 20.000, e o padrão é 750.

suprimir-valor - A escala é 1 - 20.000, e o padrão é 2000.

MAX-suprimir-tempo - Esta é a duração máxima para a supressão de uma rota. A escala é 1 - 255 minutos, e o padrão é 4 vezes o tempo da meia-vida.

```
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
```

A configuração do RTB é para o silenciar a rota com parâmetros padrão. Se você supor que o link do eBGP ao RTD é estável, a tabela de BGP RTB olha como esta:

```
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
```

A fim de simular um flap da rota, emita o comando clear ip bgp 192.208.10.6 no RTD. A tabela de BGP RTB olha como esta:

```
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
```

A entrada de BGP para 192.208.10.0 está em um estado da história. Esta colocação significa que você não tem um melhor caminho à rota, mas informação sobre a oscilação da rota ainda 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
```

A rota recebeu uma pena para bater, mas a pena está ainda abaixo do “limite de supressão”. o padrão é 2000. A supressão da rota não ocorreu ainda. Se a rota bate algumas mais vezes, você vê:

```
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
```

A rota foi umedecida, ou suprimida. A rota é reusada quando a pena alcança “o valor de reuso”. Neste caso, o valor de reuso é o padrão, 750. A informação de silenciamento é removida quando a pena se torna-se menor do que a metade do limite de reuso. Neste caso, a remoção ocorre quando a pena se transforma 375 (750/2=375). Estes comandos mostram e informações estatísticas claras de flap:

show ip bgp flap-statistics - Estatísticas do flap dos indicadores para todos os trajetos.

show ib pgp flap-statistics regexp ***expressão regular*** - Estatísticas do flap dos indicadores para todos os trajetos que combinam a expressão regular.

show IP BGP flap-statistics filter-list list - Mostra as estatísticas de flap para todos os trajetos que passam o filtro.

show ip bgp flap-statistics ***A.B.C.D m.m.m.m — Mostra*** estatísticas de flap para uma entrada única.

show ip bgp flap-statistics ***A.B.C.D m.m.m.m longer-prefix*** **- Mostra** estatísticas de flap paa entradas mais específicas.

show ip bgp neighbor [dampened-routes] | [[flap-statistics] - Estatísticas do flap dos indicadores para todos os trajetos de um vizinho.

clear ip bgp flap-statistics - Limpar as estatísticas do flap para todas as rotas.

clear ip bgp flap-statistics regexp ***regular-expression*** - Limpa as estatísticas para todos os trajetos que combinam a expressão regular.

clear ip bgp flap-statistics filter-list list - Limpa as estatísticas de flap para todos os trajetos que passam o filtro.

clear ip bgp flap-statistics ***A.B.C.D m.m.m.m - Limpa*** as estatísticas de flap para entradas únicas.

clear ip bgp ***A.B.C.D flap-statistics*** **- Limpa** as estatísticas de flap de um vizinho.

Como o BGP seleciona um trajeto

Agora que você esta familiarizado com os atributos de BGP e a terminologia, refira o algoritmo de seleção de caminho do melhor BGP.

Estudos de Caso do BGP 5

Exemplo de design prático

Esta seção contem um exemplo de design que mostre a configuração e as tabelas de roteamento enquanto as tabelas aparecem realmente em roteadores Cisco.

Esta seção mostra como construir ponto por ponto esta configuração e o que pode ir mal ao longo do caminho. Sempre que você tem um AS que conecta a dois ISP através do eBGP, execute sempre o iBGP dentro do seu AS a fim de ter o melhor controle de suas rotas. Neste exemplo, o iBGP executa o AS100 interno entre o RTA e o RTB, e o OSPF é executado como um IGP. Suponha que você conecta a dois ISP, AS200S e AS300S. Este é o primeiro lote das configurações para todos os roteadores:

Nota: Estas configurações não são as configurações finais.

```
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
```

Sempre use o **comando network** ou redistribua entradas estáticas no BGP para anunciar redes. Este método é melhor do que uma redistribuição do IGP no BGP. Este exemplo usa o **comando network** para injetar redes no BGP.

Aqui, você começa com a relação do S1 na parada de RTB, como se o link entre o RTB e o RTD não existe. Esta é a tabela de BGP 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
```

Nesta tabela, estas notações aparecem:

Um i no início - Indica que a entrada era instruída através de um peer do iBGP.

Um i na extremidade - Indica que a origem da informação de caminho é IGP.

Informação de caminho - Esta informação é intuitiva. Por exemplo, a rede 128.213.0.0 é instruída através do trajeto 200 com um salto seguinte de 128.213.63.2.

Nota: Toda a entrada localmente gerada, tal como 203.250.15.0, tem um salto seguinte 0.0.0.0.

Um símbolo **>** - Indica que o BGP escolheu a melhor rota. O BGP usa as etapas da decisão essas os esboços do algoritmo de seleção de caminho do melhors BGP do documento. O BGP escolhe um melhor caminho para alcançar um destino, instala o trajeto na tabela de IP Routing, e anuncia o trajeto a outros bgp peers.

Nota: Observe o atributo de próximo salto. O RTB sabe sobre 128.213.0.0 através de um salto seguinte de 128.213.63.2, que seja o salto seguinte do eBGP levado no iBGP.

Olhe a tabela de IP Routing:

```
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
```

Aparentemente, nenhuma das entradas de BGP alcançou a tabela de roteamento. Dois problemas existem aqui.

O primeiro problema é que o salto seguinte para estas entradas, 128.213.63.2, é inacessível. Não há nenhuma maneira de alcançar esse salto seguinte através deste IGP, que é OSPF. O RTB não aprendeu sobre 128.213.63.0 através do OSPF. Você pode executar o OSPF na relação do S0 RTA e fazê-lo passivo; desta maneira, o RTB sabe alcançar o salto seguinte 128.213.63.2. Esta configuração RTA aparece aqui:

```
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: Você pode emitir o **comando bgp nexthopself** entre o RTA e o RTB a fim mudar o salto seguinte.

A tabela de BGP nova no RTB olha como esta:

```
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 as entradas têm **>**, assim que significa que o BGP pode alcançar o salto seguinte.

Olhe a tabela de roteamento:

```
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
```

O segundo problema é que você ainda não vê as entradas de BGP na tabela de roteamento. A única diferença é que 128.213.63.0 é agora alcançável através do OSPF. Este problema é uma questão de sincronização. O BGP não põe estas entradas na tabela de roteamento e não envia as entradas nas atualizações BGP devido a uma falta da sincronização com o IGP.

Nota: O RTF não tem nenhuma noção de redes 192.208.10.0 e 195.211.10.0 porque você não redistribuiu o BGP no OSPF ainda.

Neste cenário, se você desliga a sincronização, as entradas aparecem na tabela de roteamento. Mas a conectividade é ainda quebrada.

Se você desliga a sincronização no RTB, isto é o que acontece:

```
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
```

A tabela de roteamento olha muito bem, mas não há nenhuma maneira de alcançar aquelas redes. O RTF no meio não sabe alcançar as 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
```

Quando você desliga a sincronização nesta situação, o problema ainda existe. Mas você precisa a sincronização mais tarde para outras edições. Redistribua o BGP no OSPF no RTA, com um métrico 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
```

A tabela de roteamento olha como esta:

```
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
```

As entradas de BGP desapareceram porque o OSPF tem uma distância melhor do que o iBGP. A distância OSPF é 110, quando a distância do iBGP for 200.

Desligue a sincronização no RTA de modo que o RTA possa anunciar 203.250.15.0. Esta ação é necessária porque o RTA não sincroniza com o OSPF devido à diferença nas máscaras. Mantenha a sincronização fora no RTB de modo que o RTB possa anunciar 203.250.13.0. Esta ação é necessária no RTB pela mesma razão.

Agora, traga acima a relação do S1 RTB para ver como as rotas olham. Também, permita o OSPF na série 1 do RTB de fazê-la passiva. Esta etapa permite que o RTA saiba sobre o salto seguinte 192.208.10.5 através do IGP. Se você não toma esta etapa, os loop de roteamento ocorrem porque, a fim alcançar o salto seguinte 192.208.10.5, você precisa ir a outra maneira através do eBGP. Estas são as configurações novas do RTA e do 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
```

As tabelas de BGP olham como esta:

```
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
```

Há umas múltiplas formas de projetar sua rede ao dois ISP diferentes, ao AS200 e ao AS300. Uma maneira é ter um ISP principal e um apoio ISP. Você pode aprender rotas parciais de uma dos ISP e das rotas padrão a ambos os ISP. Neste exemplo, você recebe rotas parciais do AS200 e somente rotas local do AS300. o RTA e o RTB geram rotas padrão no OSPF, com o RTB como a preferência devido ao métrico mais baixo. Desta maneira, você pode equilibrar o tráfego de saída entre os dois ISP.

A assimetria potencial pode ocorrer se o tráfego que deixa o RTA volta através do RTB. Esta situação pode ocorrer se você usa o mesmo pool dos endereços IP, a mesma rede principal, quando você fala aos dois ISP. Devido à agregação, seu AS inteiro pode olhar como uma

entidade inteira ao mundo exterior. Os pontos de entrada a sua rede podem ocorrer através do RTA ou do RTB. Você pode descobrir que todo o tráfego de entrada ao seu AS chega através de um único ponto, mesmo que você tenha múltiplos pontos na Internet. No exemplo, você tem duas redes principais diferentes quando você fala aos dois ISP.

Um outro motivo potencial para a assimetria está a um comprimento de trajeto anunciado diferente para alcançar o seu AS. Talvez um provedor de serviços é mais perto de um determinado destino do que outro. No exemplo, tráfego do AS400 que tem sua rede enquanto o destino entra sempre através do RTA devido ao trajeto mais curto. Você pode tentar afetar essa decisão. Você pode usar o **comando set as-path prepend** a fim de prepend os números do trajeto a suas atualizações e fazer o olhar o comprimento de trajeto mais longo. Mas, com atributos tais como a preferência local, métrico, ou o peso, o AS400 pode ter ajustado o ponto de saída para ser AS200. Neste caso, não há nada que você pode fazer.

Esta configuração é a configuração final para todos os roteadores:

```
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
```

No RTA, a preferência local para as rotas que vêm do AS200 é ajustada a 200. Também, a rede 200.200.0.0 é a escolha para o candidato padrão. O **comando ip default-network** permite-o

escolher a opção.

Igualmente neste exemplo, o uso do comando default-information originate com OSPF injeta a rota padrão dentro do domínio de OSPF. Este exemplo igualmente usa este comando com protocolo do Intermediate System-to-Intermediate System (protocolo IS-IS) e BGP. Para o RIP, há uma redistribuição automática no RIP de 0.0.0.0, sem configuração adicional. Para o IGRP e o EIGRP, a injeção da informação da opção no domínio IGP ocorre após a redistribuição do BGP no IGRP e no EIGRP. Também, com IGRP e EIGRP, você pode redistribuir uma rota estática a 0.0.0.0 no domínio 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 o RTB, a preferência local para as atualizações que vêm do AS300 é ajustada a 300. Este valor é mais alto do que o valor da preferência local das atualizações do iBGP que vêm do RTA. Desta maneira, o AS100 escolhe o RTB para as rotas local do AS300. Todas as outras rotas no RTB, se outras rotas existem, transmitem internamente com uma preferência local de 100. Este valor é mais baixo do que a preferência local de 200, que vem do RTA. Assim o RTA é a preferência.

Nota: Você anunciou somente as rotas local do AS300. Alguma informação de caminho que não combinar quedas ^300$. Se você quer anunciar as rotas locais e as rotas vizinhas, que são os clientes do ISP, use ^300_[0-9]*.

Está aqui a saída da expressão regular que indica as rotas local do 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
```

No RTC, você agrega 128.213.0.0/16 e indica as rotas específicas para a injeção no AS100. Se o ISP recusa fazer esta tarefa, você deve filtrar no fim entrante do 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
```

Uma demonstração do uso da filtração da comunidade está no RTG. Você adiciona uma comunidade **no-export** às atualizações 195.211.0.0 para o RTD. Desta maneira, o RTD não exporta essa rota para o RTB. Contudo, neste caso, o RTB não aceita estas rotas de qualquer maneira.

```
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
```

o RTE agrega 200.200.0.0/16. Estão aqui o BGP e as tabelas de roteamento para o RTA, o RTF, e o RTB finais:

```
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: A tabela de roteamento RTF indica que a maneira de alcançar as redes locais ao AS300, tal como 192.208.10.0, é com o RTB. A maneira de alcançar outras redes conhecidas, tais como 200.200.0.0, é com o RTA. O Gateway of Last Resort é ajustado ao RTB. Se algo acontece à conexão entre o RTB e o RTD, a padrão que o RTA anuncia retrocede dentro com um métrico 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
```

Informações Relacionadas

- BGP: Perguntas mais freqüentes
- Configurações de amostra do BGP através de um PIX Firewall
- Como usar o HSRP para fornecer redundância em uma rede BGP multihomed
- Configurando a redundância de modo de roteador único e BGP em um MSFC do Cat6000
- Alcance um Roteamento Ideal e Reduza o Consumo de Memória BGP
- Troubleshooting de BGP
- Troubleshooting de CPU Elevada Gerados pelo Processo de BGP Scanner ou BGP Router
- Compartilhamento de carga com o BGP no ambientes únicos e multihomed: Configurações de exemplo
- Página de suporte de BGP
- Suporte Técnico e Documentação - Cisco Systems