Examinar estudos de caso do Border Gateway Protocol

Contents

Introdução

Este documento descreve cinco estudos de caso do 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.

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

Conventions

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.

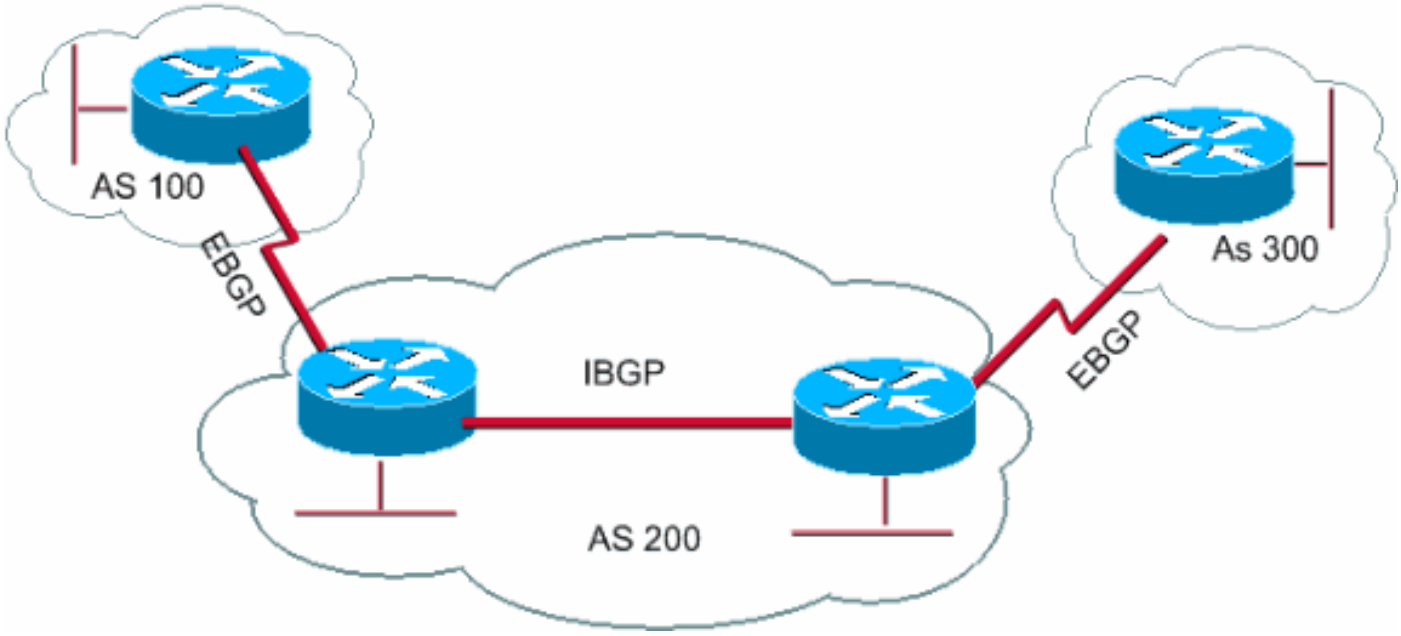
eBGP e iBGP

Se um AS tem múltiplos falantes BGP, o AS pode servir como um serviço de trânsito para outros AS. Como o próximo 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.



O BGP é executado entre roteadores no mesmo AS

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.

1. 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:

   <#root>

   ```
   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.

2. Defina vizinhos de BGP.

   A formação do vizinho de BGP indica os roteadores que tentam falar através do BGP. A próxima seção explica esse 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 neighbor comando para estabelecer uma conexão TCP:

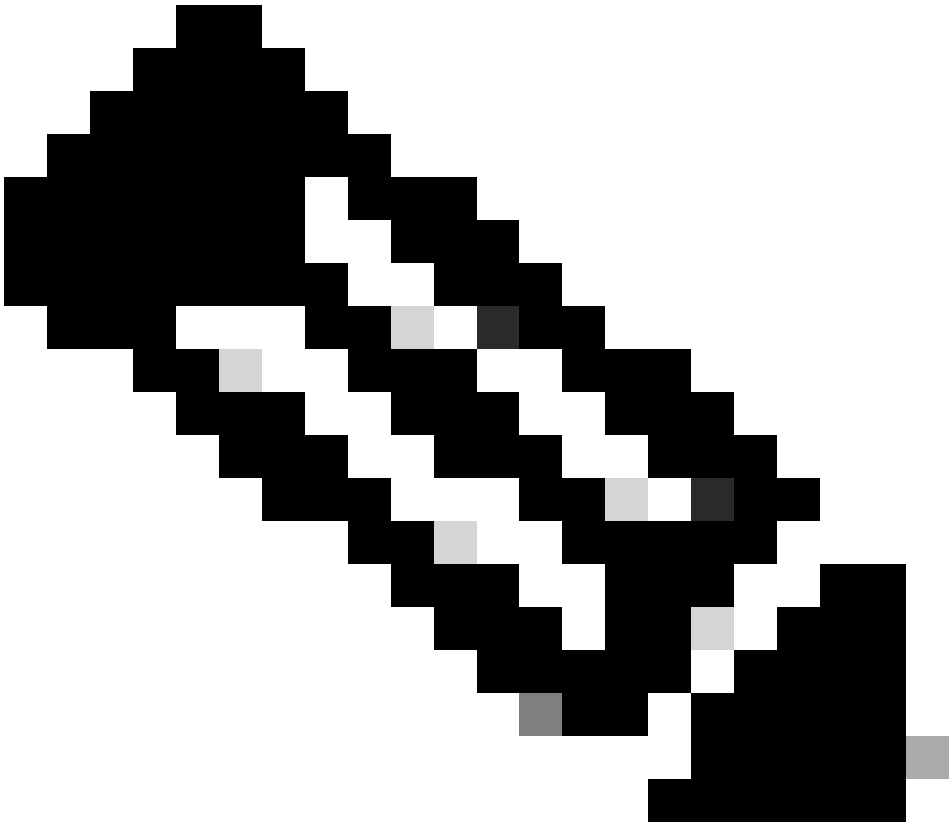
```
<#root>

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 pares devem ser capazes de 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 que está fazendo ping a usar como origem o endereço IP que o neighbor comando especifica. 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. .

- clear ip bgp address



Observação: o endereço é o endereço vizinho

-

clear ip bgp *

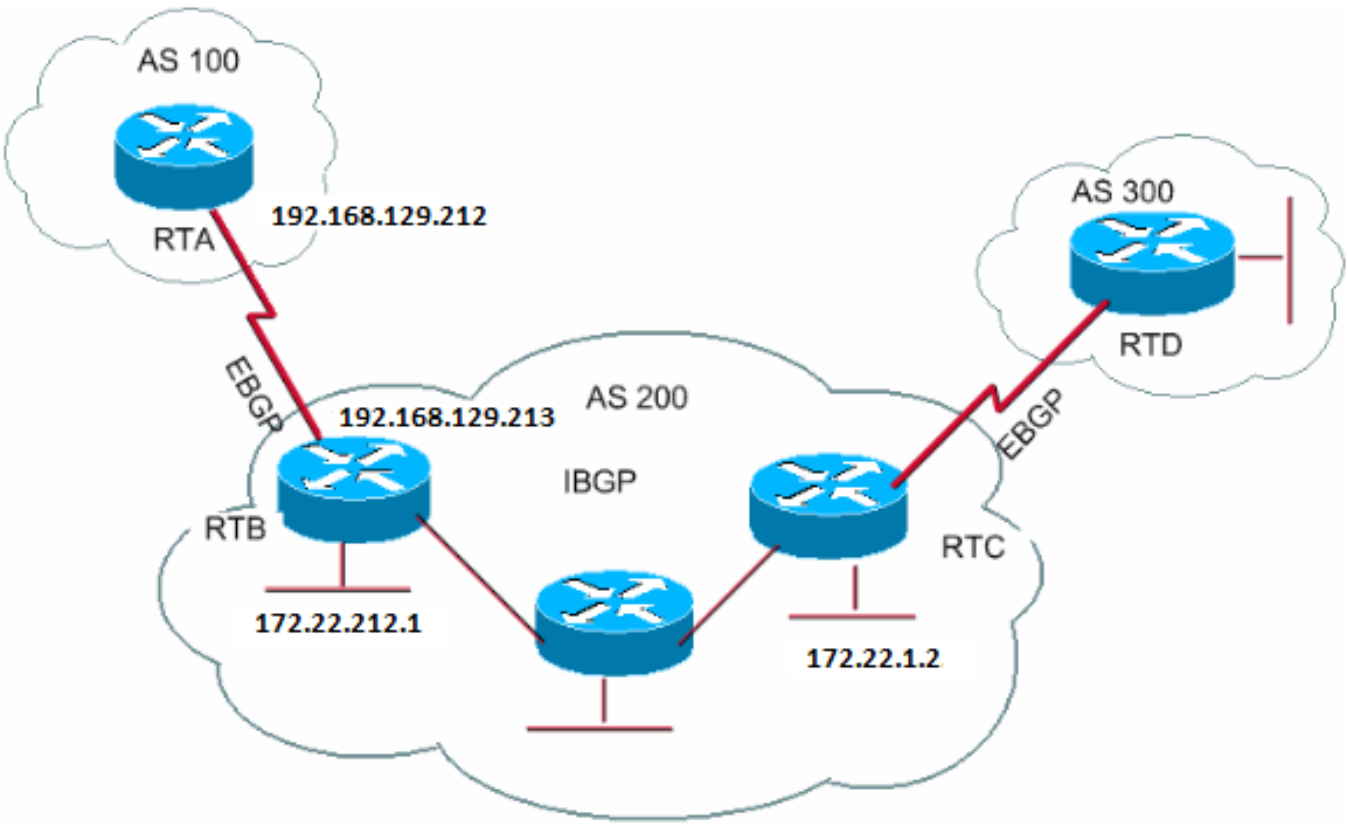
Este comando cancela todas as conexões vizinha.

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:

```
<#root>
```

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

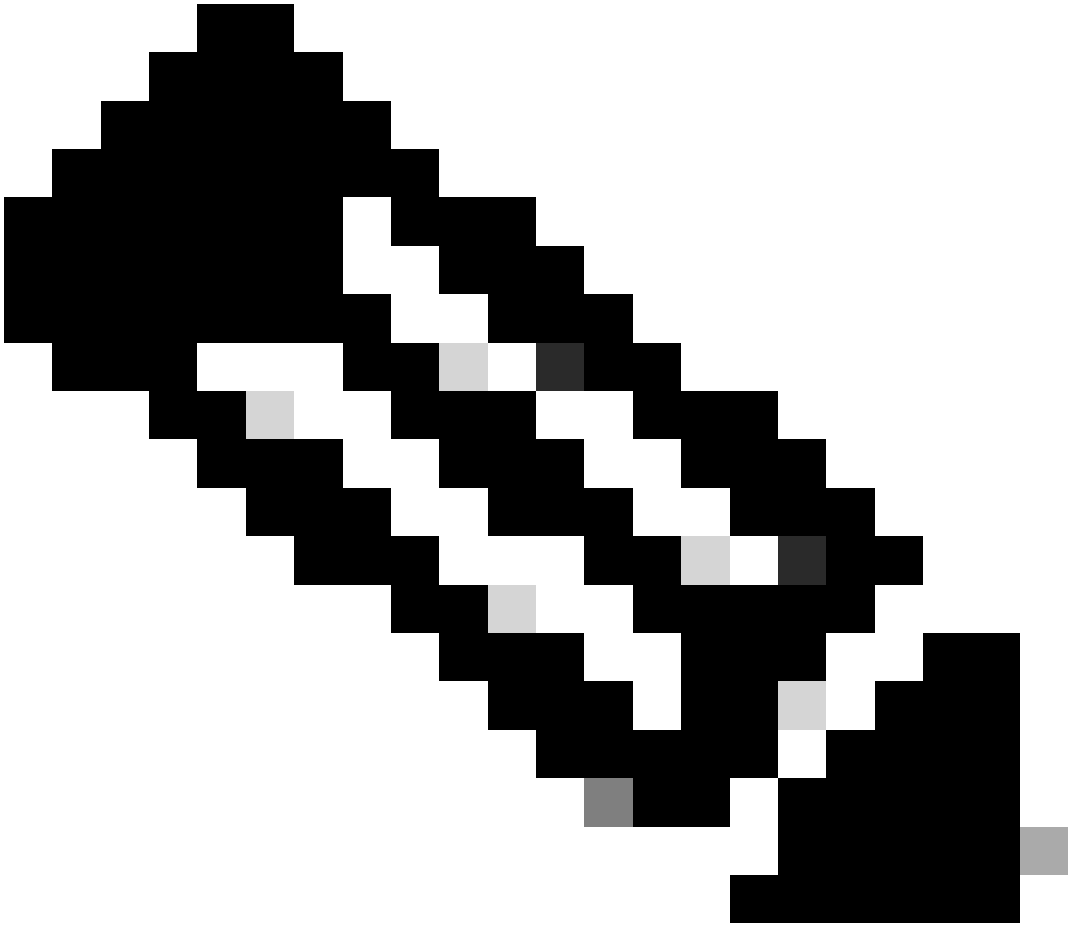
Aqui está um exemplo da configuração do neighbor comando:



```
RTA#
router bgp 100
 neighbor 192.168.129.213 remote-as 200

RTB#
router bgp 200
 neighbor 192.168.129.212 remote-as 100
 neighbor 172.22.1.2 remote-as 200

RTC#
router bgp 200
 neighbor 172.22.212.1 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. Além disso, os peers eBGP têm conexão direta, mas os peers iBGP não têm conexão direta. Os roteadores iBGP não precisam ter conexão direta. Mas deve haver algum IGP que seja executado e permita que os dois vizinhos se alcancem.

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



Observação: preste atenção especial ao estado do BGP. Qualquer estado diferente de Established indica que os peers não estão ativos. Além disso, observe estes próximos itens:

- 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.

<#root>

```
Router#
```

show ip bgp neighbors

```
  BGP neighbor is 192.168.129.213, remote AS 200, external link
  BGP version 4, remote router ID 172.22.12.1
```

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

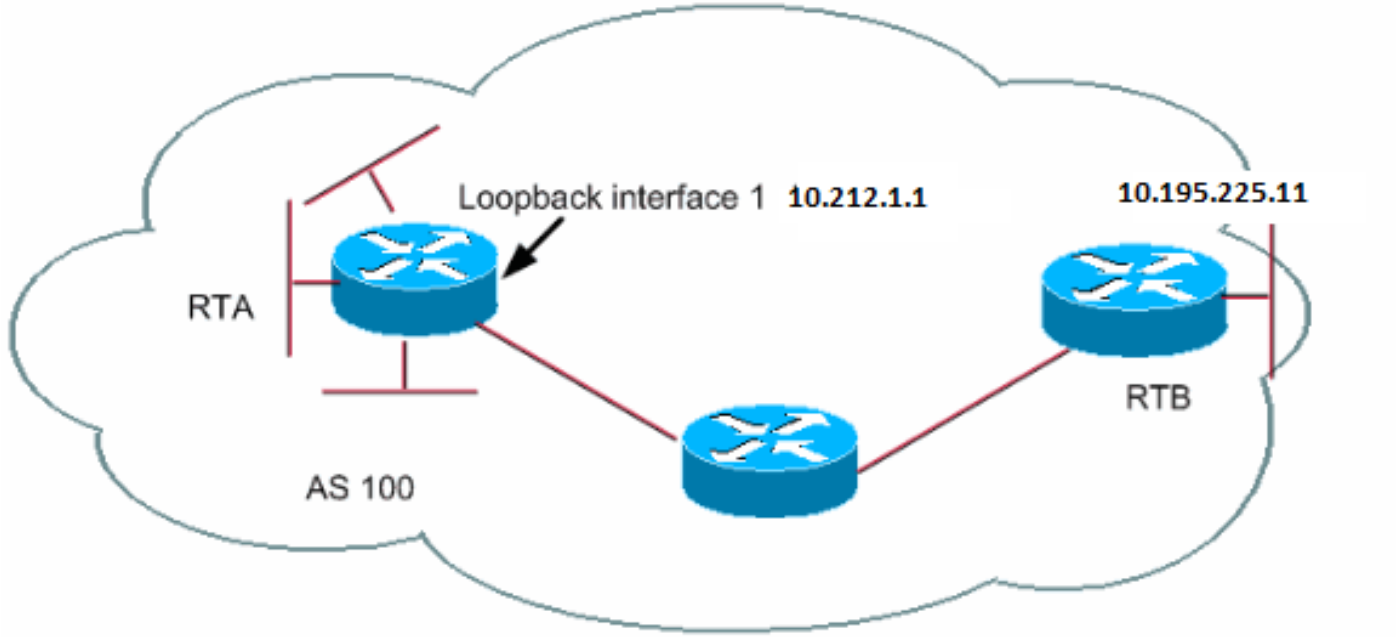
O uso de uma interface de loopback para definir vizinhos é comum com o iBGP, mas não com o eBGP. Normalmente, você usa a interface de loopback para certificar-se que o endereço IP do vizinho fica acima e é independente de hardware que funciona corretamente. No caso do eBGP, os roteadores de peer têm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

Se você usar o endereço IP de uma interface de loopback no neighbor comando, precisará de 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:

<#root>

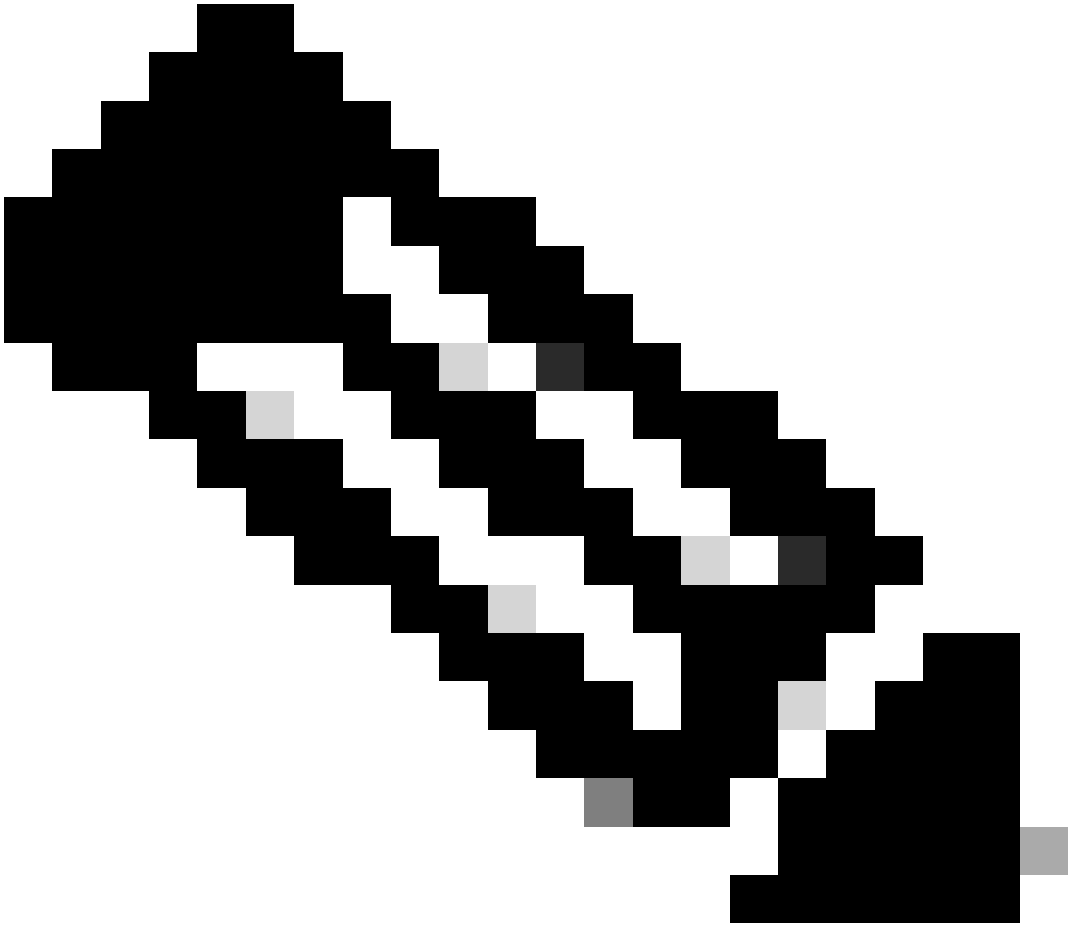
neighbor <ip-address> update-source <interface>

Este exemplo ilustra o uso deste comando:




```
RTA#
router bgp 100
 neighbor 10.195.225.11 remote-as 100
 neighbor 10.195.225.11 update-source loopback 1

RTB#
router bgp 100
 neighbor 10.212.1.1 remote-as 100
```

Neste exemplo, RTA e RTB executam dentro do AS100. No neighbor comando, o RTB usa a interface de loopback do RTA, 10.212.1.1. Neste caso, o RTA deve forçar o BGP para usar o endereço IP de loopback como a fonte na conexão vizinha TCP. Para forçar essa ação, o RTA adiciona **update-source interface-type interface-number** de modo que o comando seja neighbor 10.195.225.11 update-source loopback 1. Essa instrução força o BGP a usar o endereço IP da interface de loopback quando o BGP se comunica com o vizinho 10.195.225.11.

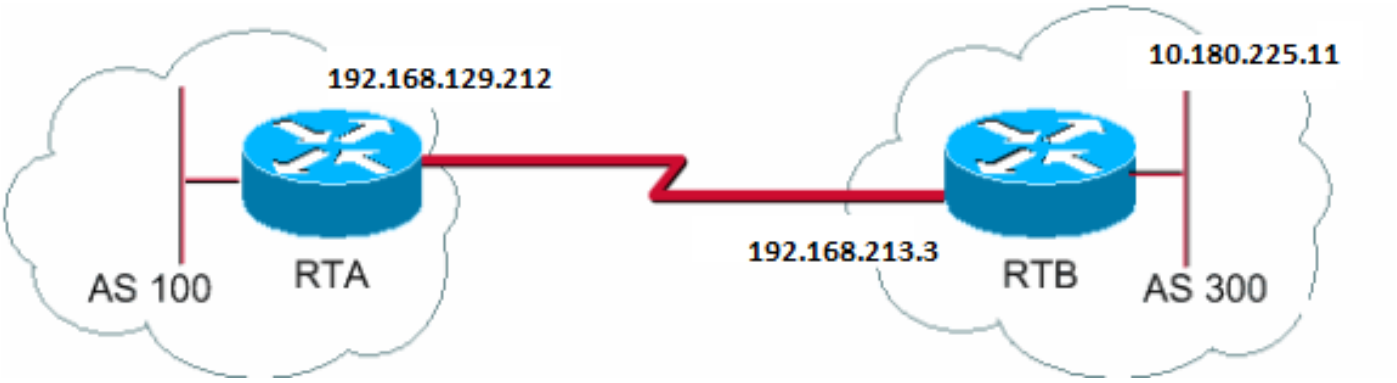


Observação: o RTA usou o endereço IP da interface física do RTB, 10.195.225.11, como um vizinho. O uso deste endereço IP é porque o RTB não precisa nenhuma configuração especial. Refira a configuração de exemplo para o iBGP e eBGP com ou sem endereço de loopback para uma configuração de exemplo completo 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:

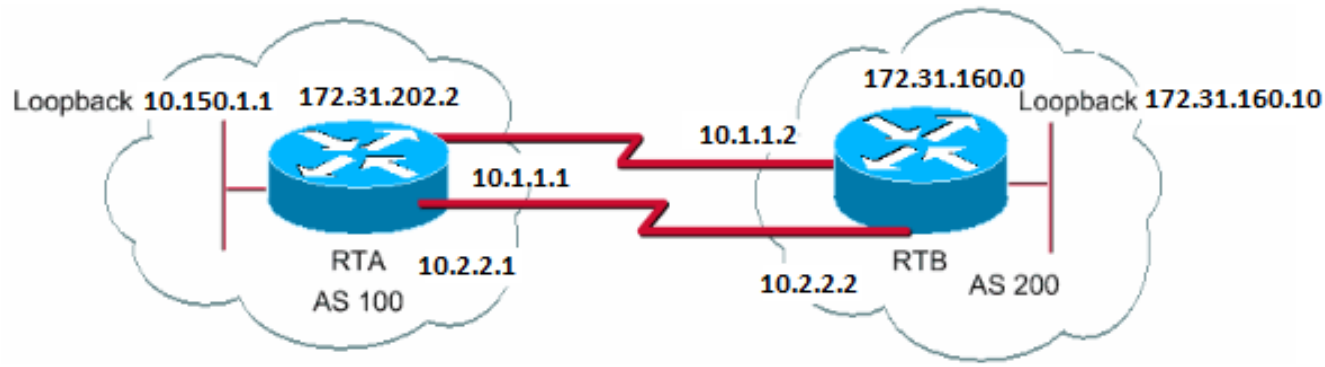



```
RTA#
router bgp 100
 neighbor 10.180.225.11 remote-as 300
 neighbor 10.180.225.11 ebgp-multihop

RTB#
router bgp 300
 neighbor 192.168.129.212 remote-as 100
```

O RTA indica um vizinho externo que não tenha a conexão direta. O RTA precisa indicar seu uso do comando neighbor ebgp-multihop. Por outro lado, o RTB indica um vizinho que tem conexão direta, que é 192.168.129.212. Devido a esta conexão direta, o RTB não precisa do neighbor ebgp-multihop comando. Você também deve configurar um IGP ou roteamento estático para permitir que os vizinhos sem conexão alcancem um ao outro.

O exemplo na seção BGP multihop (Load Balancing) mostra como alcançar o balanceamento de carga com BGP em um caso onde você tem eBGP sobre linhas paralelas.

eEBGP Multihop (Balanceamento de carga)




```
RTA#
int loopback 0
 ip address 10.150.1.1 255.255.255.0

router bgp 100
```

```
 neighbor 172.31.160.10 remote-as 200
 neighbor 172.31.160.10 ebgp-multihop
 neighbor 172.31.160.10 update-source loopback 0
 network 172.31.202.2

ip route 172.31.160.0 255.255.0.0 10.1.1.2
ip route 172.31.160.0 255.255.0.0 10.2.2.2

RTB#
int loopback 0
 ip address 172.31.160.10 255.255.255.0

router bgp 200
 neighbor 10.150.1.1 remote-as 100
 neighbor 10.150.1.1 update-source loopback 0
 neighbor 10.150.1.1 ebgp-multihop
 network 172.31.160.0

ip route 172.31.202.2 255.255.0.0 10.1.1.1
ip route 172.31.202.2 255.255.0.0 10.2.2.1
```

Este exemplo ilustra o uso de 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 opções para alcançar o próximo salto 172.31.160.10: um caminho através de 10.1.1.2 e o outro caminho através de 10.2.2.2. 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. Assim é o formato do mapa de rotas:

<#root>

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 e de set configuração. A correspondência especifica um match critério e o conjunto especifica uma set ação se os critérios que o match comando impõe forem atendidos.

Por exemplo, você pode definir um mapa de rotas que verifique atualizações de saída. Se há um combinação para o endereço IP 10.1.1.1, a métrica para essa atualização está ajustada ao 5. Estes comandos ilustram o exemplo:

<#root>

match ip address 10.1.1.1

set metric 5

Agora, se os critérios de correspondência forem atendidos e você tiver um permit, haverá uma redistribuição ou controle das rotas, conforme especificado pela ação de definição. Você sai da lista.

Se os critérios de correspondência forem atendidos e você tiver um deny, não haverá redistribuição ou controle da rota. Você sai da lista.

Se os critérios de correspondência não forem atendidos e você tiver um permit ou deny , a próxima instância do mapa de rotas será verificada. 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 terminar a lista sem uma correspondência, a rota será not accepted nor forwarded.

Nas versões do Cisco IOS® Software anteriores ao Cisco IOS Software Release 11.2, quando você usa mapas de rotas para filtrar atualizações BGP em vez de redistribuir entre protocolos, você não pode filtrar na entrada quando 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 match são:

- matchas-path
- match community
- matchclns
- match interface
- matchip address
- matchip nexthop
- matchip route-source

- matchmetric

- match route-type

- match tag

Os comandos relacionados para set são:

- set as-path

- set clns

- set automatic-tag

- set community

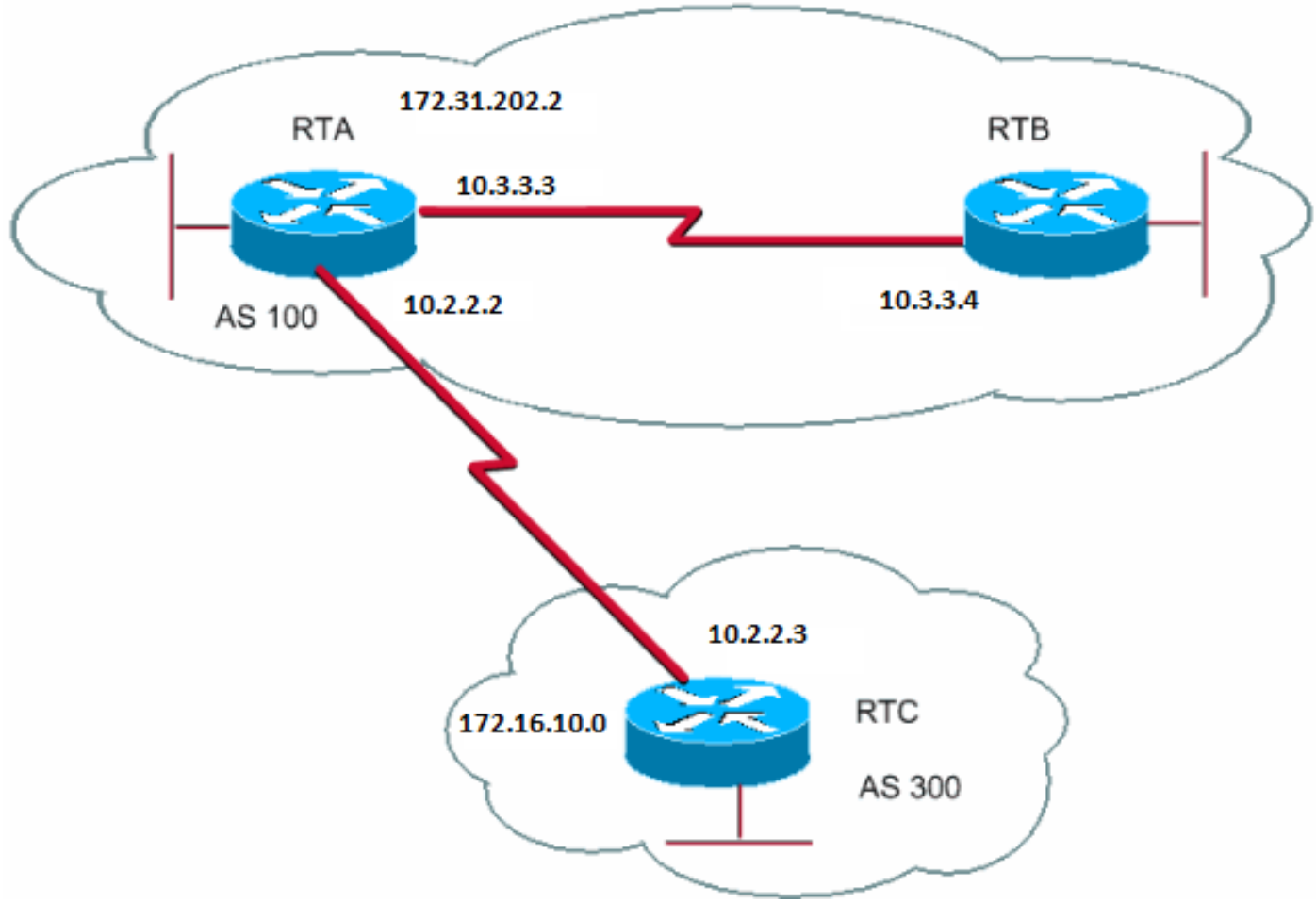
- set interface

- set default interface

- set ip default nexthop

- set level
- set local-preference
- set metric
- set metric-type
- set nexthop
- set origin
- set tag
- set weight

Olhe alguns exemplos do mapa de rotas:



Exemplos de mapa de rotas

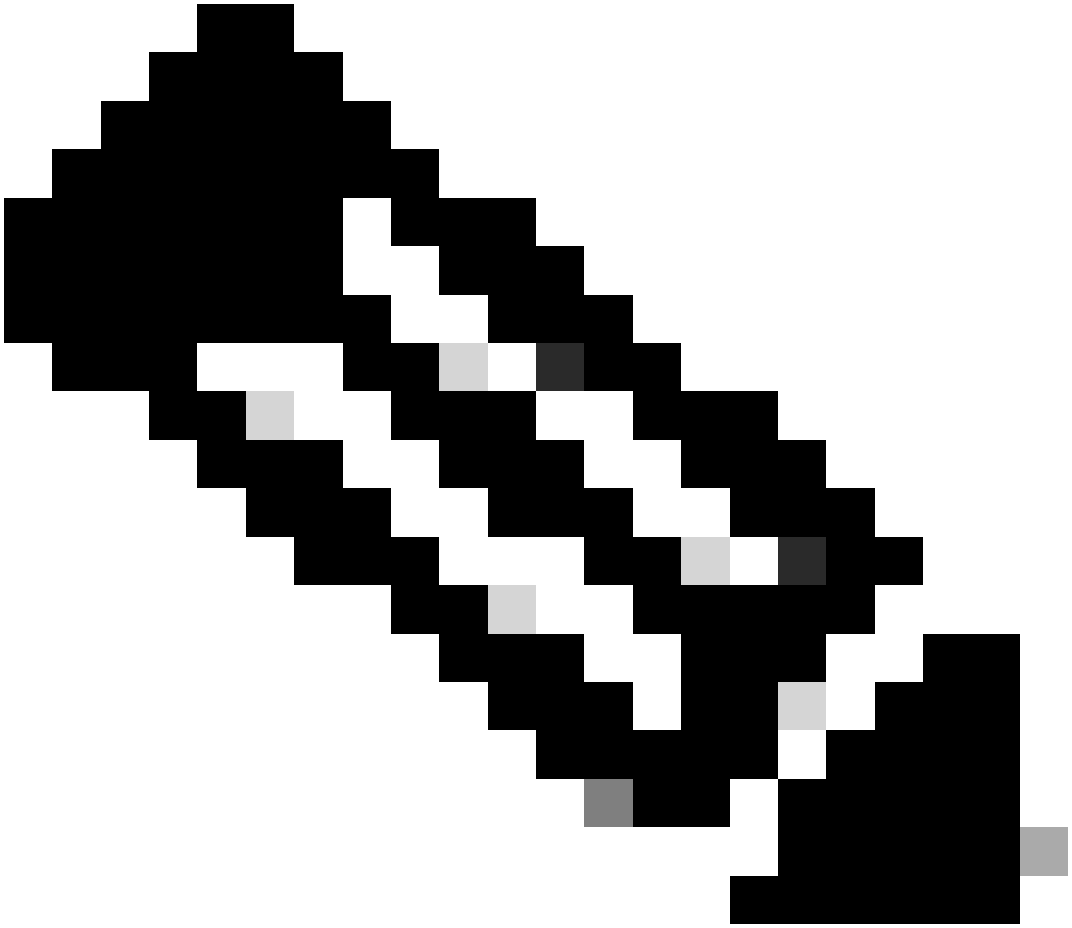
Exemplo 1

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. Suponha que o RTA deseja redistribuir para rotas RTB sobre 172.16.10.0 com uma métrica de 2 e todas as outras rotas com uma métrica de 5. Nesse caso, você pode usar esta configuração:

```
RTA#
router rip
 network 10.3.0.0
 network 10.2.0.0
 network 172.31.202.2
 passive-interface Serial0
 redistribute bgp 100 route-map SETMETRIC

router bgp 100
 neighbor 10.2.2.3 remote-as 300
 network 172.31.202.2

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

route-map SETMETRIC permit 20
 set metric 5
```

```
access-list 1 permit 172.16.10.0 0.0.255.255
```

Neste exemplo, se uma rota combina o endereço IP 172.16.10.0, a rota tem um métrico de 2. Então, você sao da lista do mapa de rotas. Se não houver correspondência, você prosseguirá pela lista de mapas de rotas, que indica que todo o resto está definido como métrica 5.



Observação: sempre faça a pergunta "O que acontece com as rotas que não correspondem a nenhuma das instruções de correspondência?" Estas rotas deixam cair, por padrão.

Exemplo 2

Suponha que, no Exemplo 1, você não deseja que o AS100 aceite atualizações sobre 172.16.10.0. 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 172.16.10.0
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.2.2.2 route-map STOPUPDATES out

route-map STOPUPDATES permit 10
 match ip address 1

access-list 1 deny 172.16.10.0 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
- Redistribuição
- Rotas estáticas e redistribuição

comando network

O formato do network comando é:

<#root>

network <network-number> mask <network-mask>

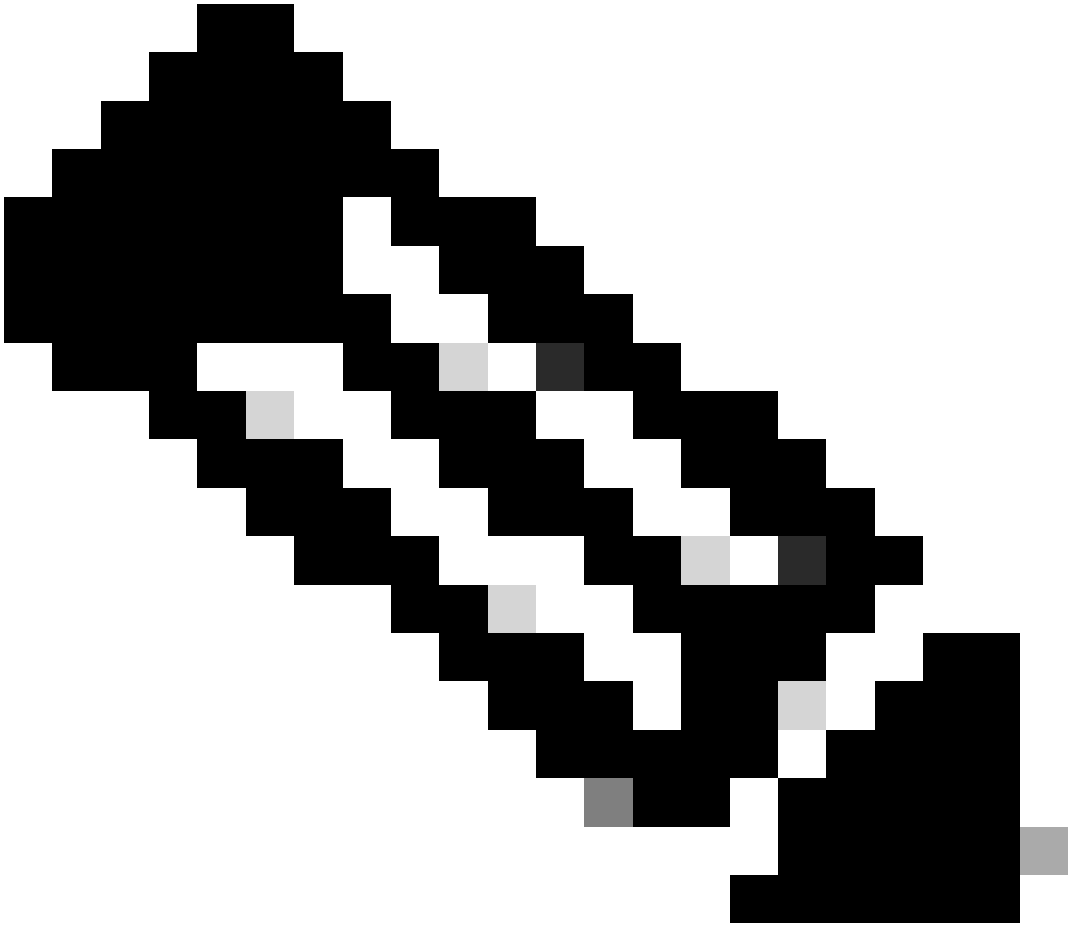
O network comando controla as redes que se originam dessa caixa. Este conceito é diferente da configuração familiar com protocolo Interior Gateway Routing (IGRP) e RIP. Com este comando, você não tenta executar o BGP em uma determinada interface. Em vez disso, você tenta indicar ao BGP quais redes o 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 network comando são aceitáveis.

O network comando funciona se o roteador conhece a rede que você tenta anunciar, seja conectada, estática ou aprendida dinamicamente.

Um exemplo do comando network é:

```
RTA#
router bgp 1
 network 192.168.213.0 mask 255.255.0.0

ip route 192.168.213.0 255.255.0.0 null 0
```

Este exemplo indica que o roteador A gera uma entrada de rede para 192.168.213.0/16. O /16 indica que você usa uma superrede do endereço classe C e você anuncia os primeiros dois octetos, ou primeiros 16 bits.

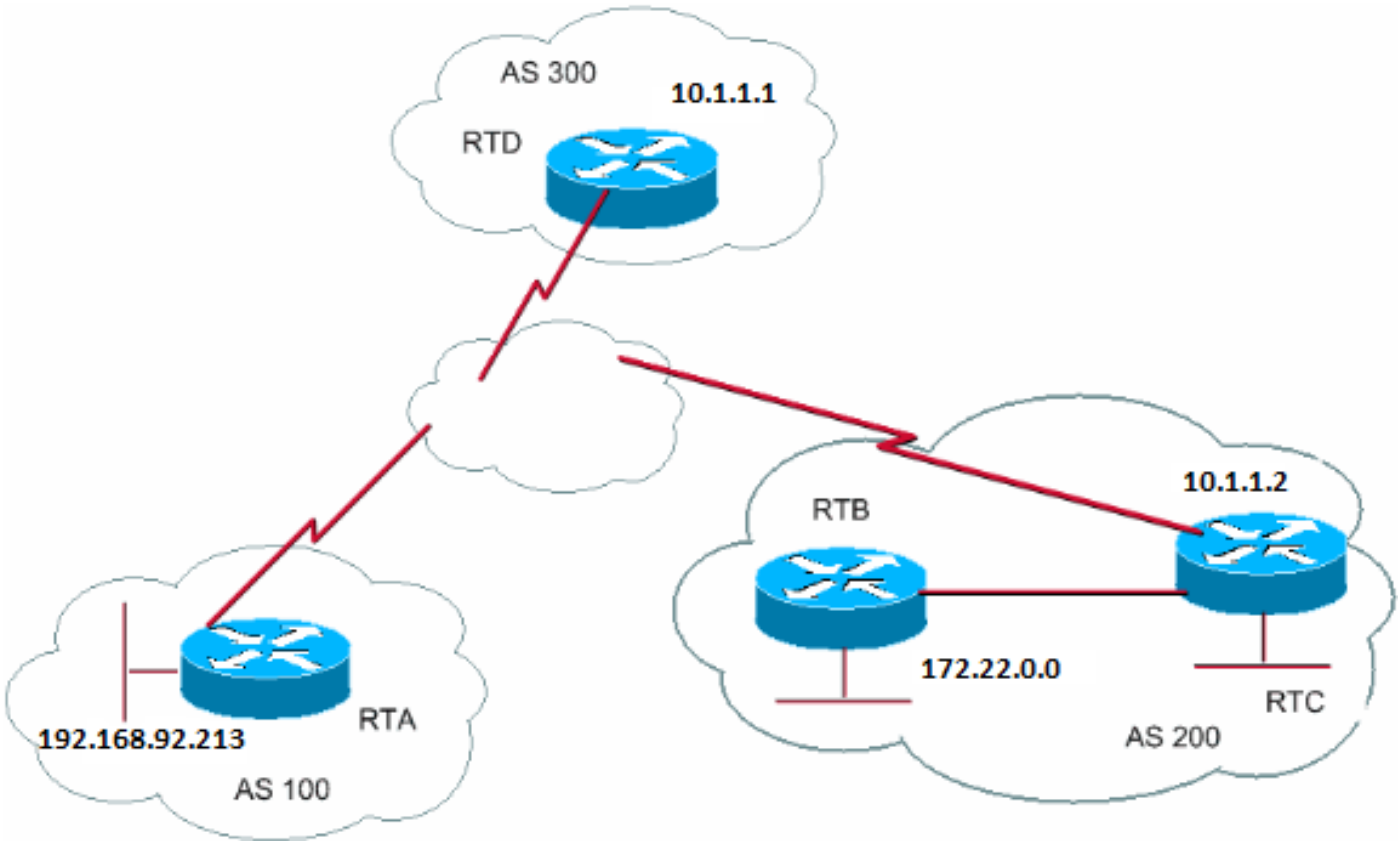


Observação: você precisa da rota estática para fazer com que o roteador gere 192.168.213.0, pois a rota estática coloca uma entrada correspondente na tabela de roteamento.

Redistribuição

O network comando é uma forma de anunciar suas redes via 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. Essa redistribuição pode parecer assustadora porque agora você despeja todas as suas rotas internas no BGP; algumas dessas rotas podem ter sido aprendidas via BGP e você não precisa enviá-las novamente. Ao filtrar, tenha cuidado para garantir que você envie para as rotas somente de Internet que deseja anunciar e não para todas as rotas que possui. Exemplo:

O RTA anuncia que 192.168.92.213 e o RTC anunciam 172.22.0.0. Olhe a configuração de RTC:



Se você emitir o networkcomando, terá:

```
RTC#
router eigrp 10
 network 172.22.0.0
 redistribute bgp 200
 default-metric 1000 100 250 100 1500

router bgp 200
 neighbor 10.1.1.1 remote-as 300
 network 172.22.0.0 mask 255.255.0.0

!--- This limits the networks that your AS originates to 172.22.0.0.
```

Se ao invés você usa redistribução, você tem:

```
RTC#
router eigrp 10
network 172.22.0.0
redistribute bgp 200
default-metric 1000 100 250 100 1500

router bgp 200
neighbor 10.1.1.1 remote-as 300
redistribute eigrp 10
```

```
!--- EIGRP injects 192.168.92.213 again into BGP.
```

Esta redistribuição causa as origens de 192.168.92.213 pelo seu AS. Você não é a origem de 192.168.92.213; AS100 é a origem. Portanto, você deve usar filtros para evitar que a origem saia dessa rede pelo seu AS. A configuração correta é:

```
RTC#
router eigrp 10
 network 172.22.0.0
 redistribute bgp 200
 default-metric 1000 100 250 100 1500

router bgp 200
 neighbor 10.1.1.1 remote-as 300
 neighbor 10.1.1.1 distribute-list 1 out
 redistribute eigrp 10

access-list 1 permit 172.22.0.0 0.0.255.255
```

Você usa o access-listcomando para controlar as redes que se originam do AS200.

A redistribuição do OSPF no BGP é levemente diferente da redistribuição para outros IGP. A simples questão de redistribute ospf 1under router bgpnão funciona. Palavras-chave específicas, como internal, external, e **nssa-external** são necessárias para redistribuir as respectivas rotas. Consulte Entender a redistribuição de rotas OSPF no BGP para obter 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 obter o mesmo resultado que o exemplo na seção de redistribuçãorealizado com:

```
RTC#
router eigrp 10
 network 172.22.0.0
 redistribute bgp 200
 default-metric 1000 100 250 100 1500

router bgp 200
 neighbor 10.1.1.1 remote-as 300
 redistribute static

ip route 172.22.0.0 255.255.255.0 null0
```

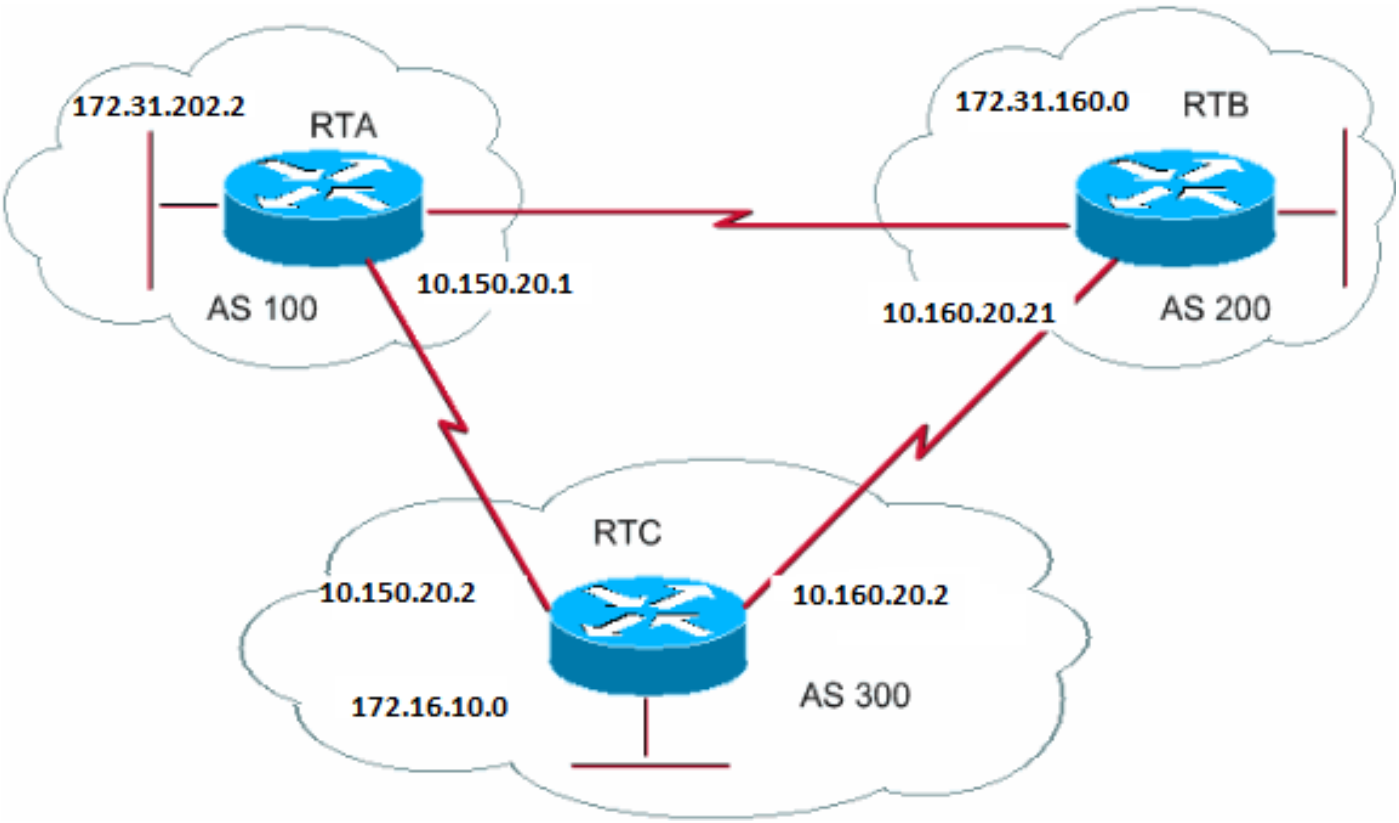
A null0 interface significa desconsiderar o pacote. Portanto, se você receber o pacote e houver uma correspondência mais específica do que

172.22.0.0, que existe, o roteador enviará o pacote para a correspondência específica. Se não, o roteador desconsidera o pacote. Este método é uma boa maneira anunciar uma supernet.

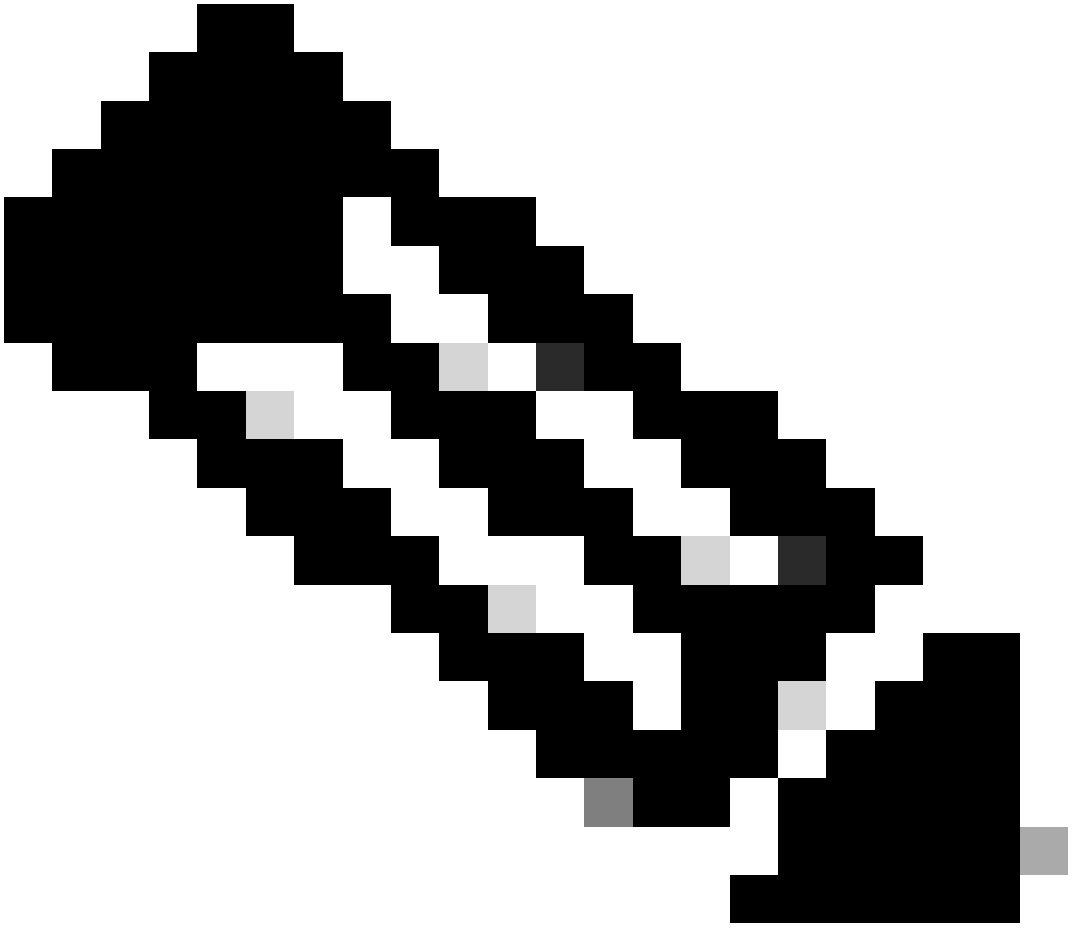
Este original discutiu como você pode usar métodos diferentes para originar rotas fora do seu AS. Lembre-se que estas rotas são geradas em adição a outras rotas BGP que o BGP aprendido através dos vizinhos, internos ou externos. O BGP repassa informações que o BGP aprende de um peer a outros peers. A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

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

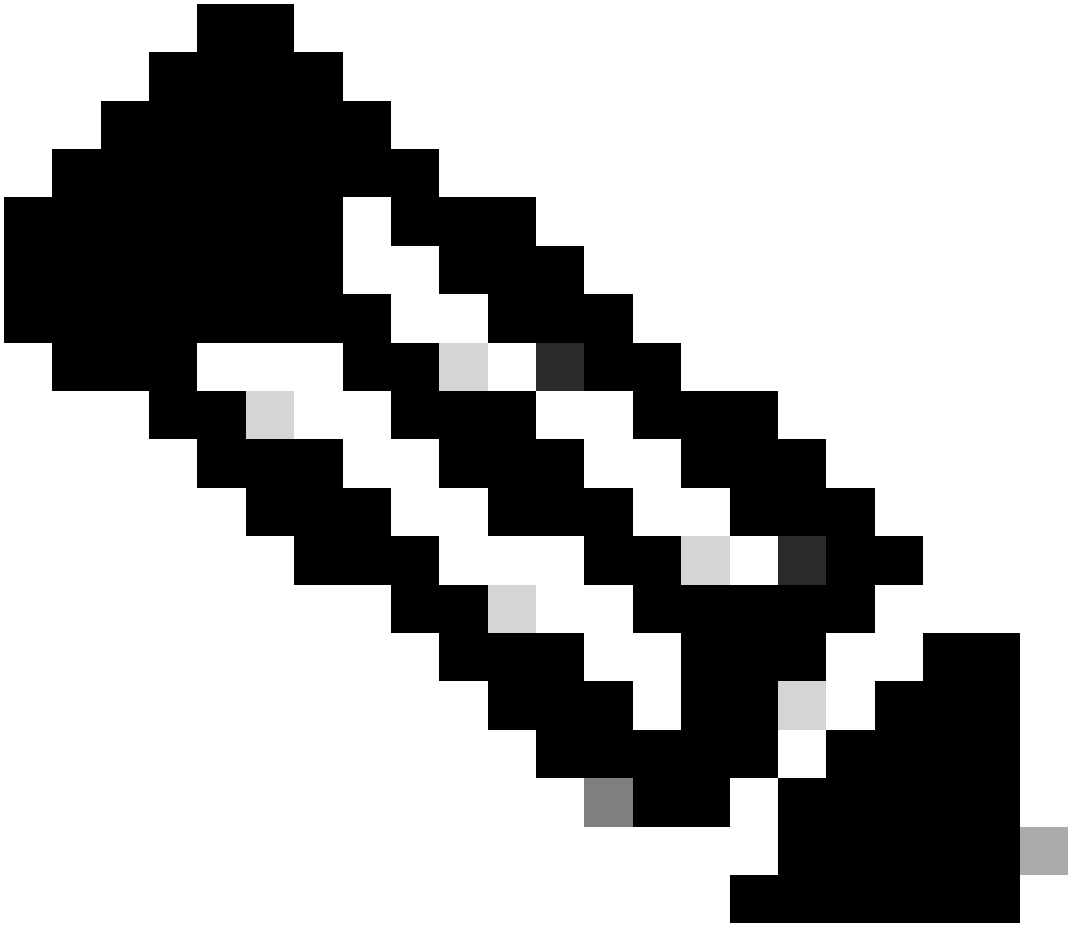
Aqui está um exemplo:




```
RTA#
router bgp 100
 neighbor 10.150.20.2 remote-as 300
 network 172.31.202.2

RTB#
router bgp 200
 neighbor 10.160.20.2 remote-as 300
 network 172.31.160.0

RTC#
router bgp 300
 neighbor 10.150.20.1 remote-as 100
 neighbor 10.160.20.21 remote-as 200
 network 170.10.00
```



Observação: Você não precisa da rede 172.31.202.2 ou da rede 172.31.160.0 no RTC, a menos que você queira que o RTC gere essas redes, bem como passe sobre essas redes quando elas vêm 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.



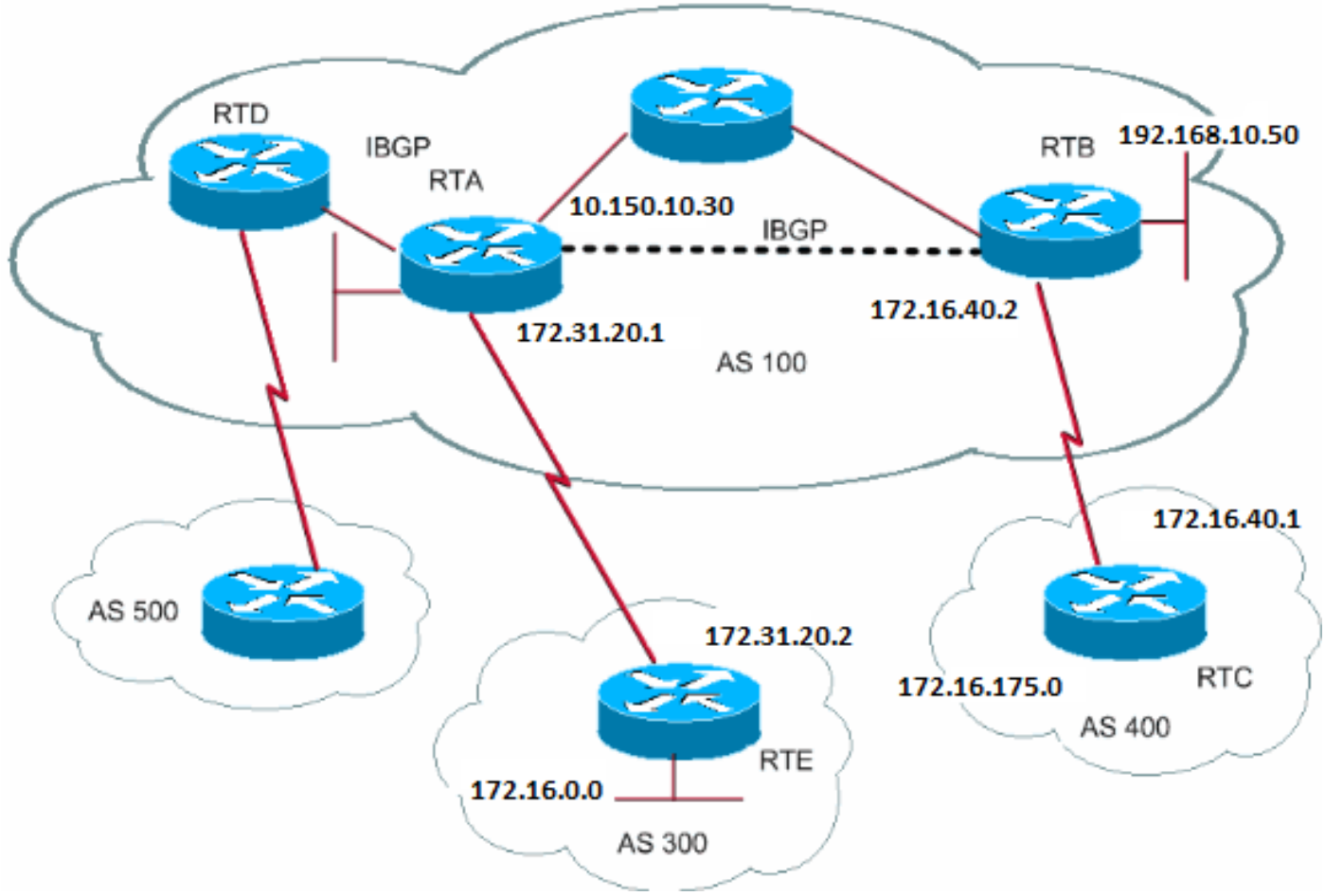
Observação: lembre-se de que o BGP não aceita atualizações originadas de seu próprio AS. Esta recusa assegura uma topologia sem loop do interdomain.

Por exemplo, suponha que o AS200, do exemplo nesta seção, tenha uma conexão BGP direta no AS100. O RTA gera uma rota 172.31.202.2 e envia a rota ao AS300. Então, o RTC passa esta rota ao AS200 e mantem a origem como o AS100. O RTB passa 172.31.202.2 ao AS100 com a origem ainda AS100. O RTA observa que a atualização originou do suas próprias AS e ignoram a atualização.

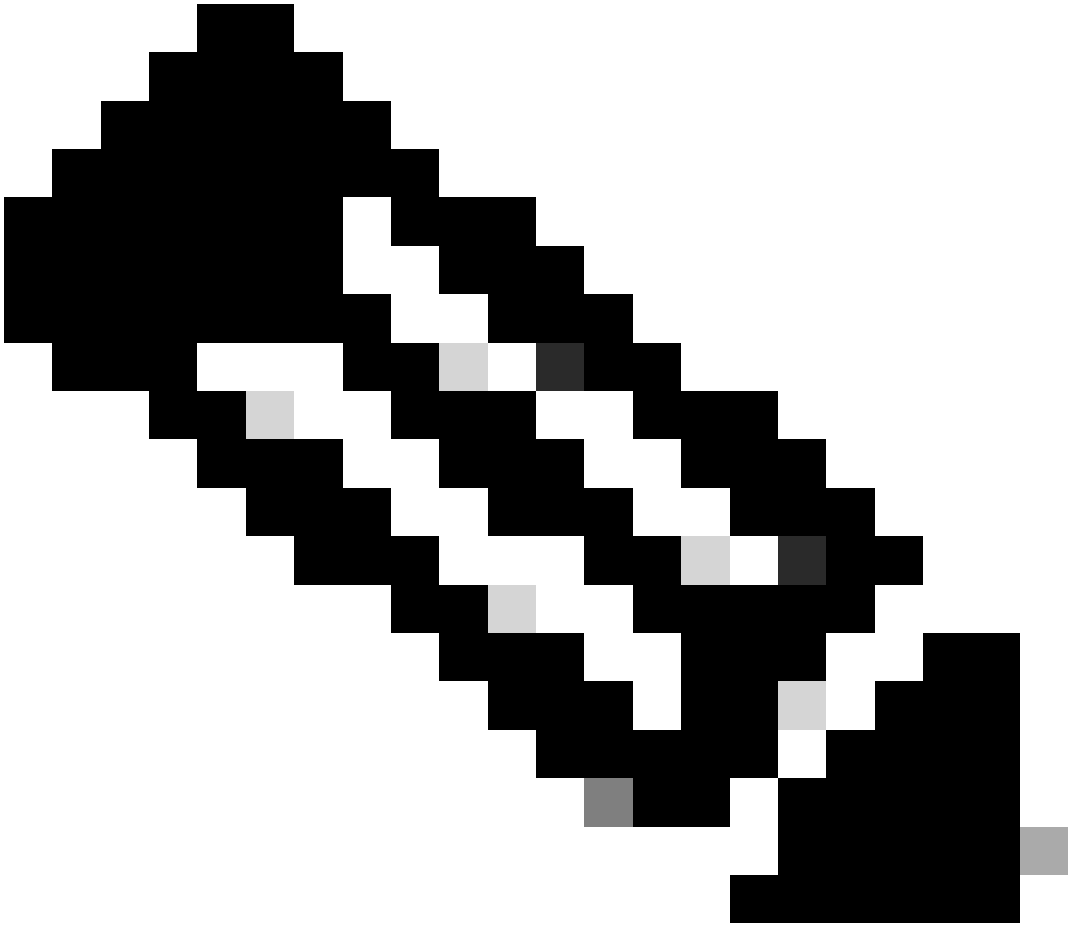
iBGP

Você usa o iBGP se um AS deseja atuar como um sistema de trânsito para outro AS. Você pode fazer a mesma coisa se aprender via eBGP, redistribuir no IGP e depois redistribuir novamente em outro AS. Mas o iBGP oferece mais flexibilidade e maneiras mais eficientes de trocar informações dentro de um AS. Por exemplo, o iBGP fornece maneiras de controlar a melhor saída do AS com uso da preferência local. O

atributo sectionLocal Preference fornece mais informações sobre preferências locais.




```
RTA#
router bgp 100
 neighbor 192.168.10.50 remote-as 100
 neighbor 172.31.20.2 remote-as 300
 network 172.31.202.2

RTB#
router bgp 100
 neighbor 10.150.10.30 remote-as 100
 neighbor 172.16.40.1 remote-as 400
 network 192.168.10.150

RTC#
router bgp 400
 neighbor 172.16.40.2 remote-as 100
 network 172.16.0.0
```



Observação: lembre-se de que quando um alto-falante BGP recebe uma atualização de outros alto-falantes BGP em seu próprio AS (iBGP), o alto-falante BGP que recebe a atualização não redistribui essa informação para outros alto-falantes BGP em seu próprio 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.

O RTA e o RTB executam o 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.

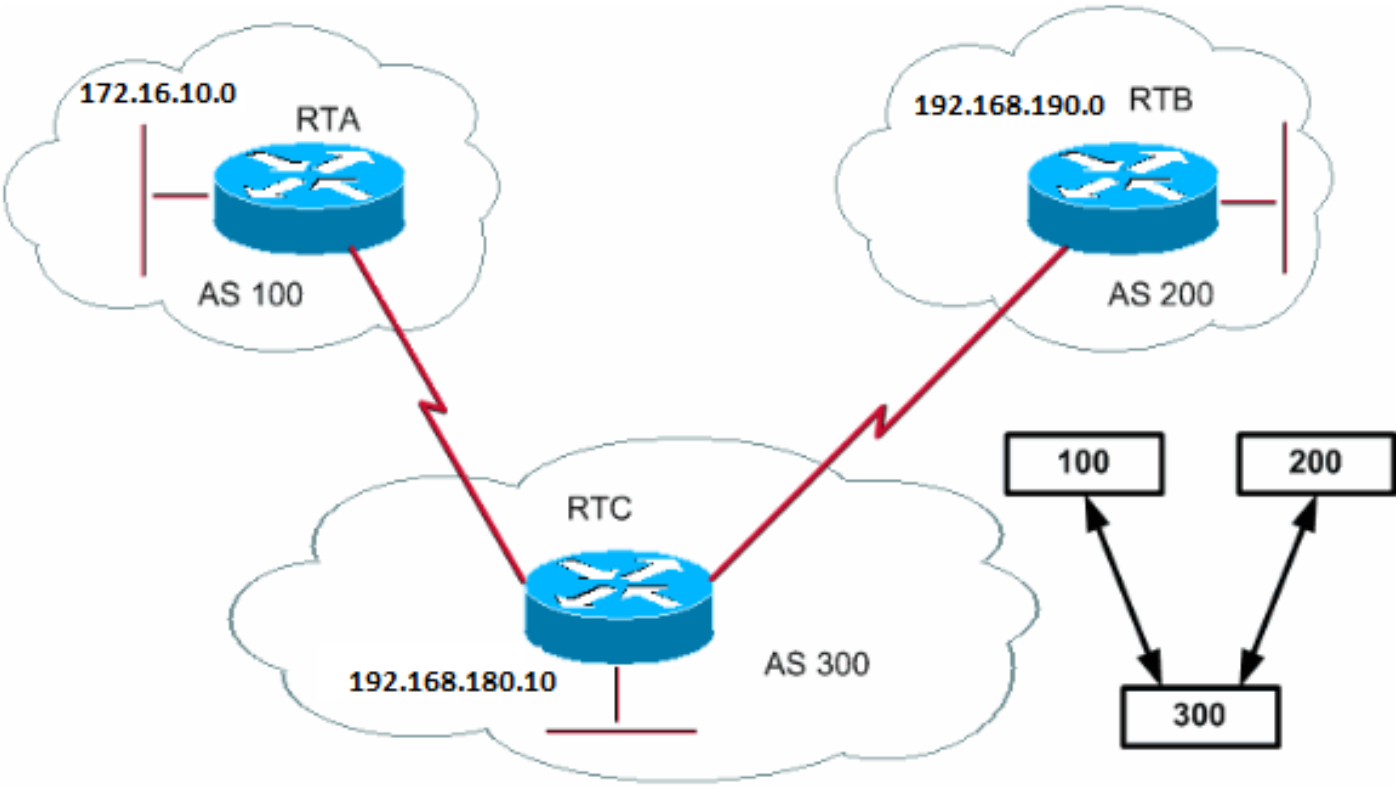
O BGP baseia a decisão em attributes diferentes, como próximo salto, pesos administrativos, preferência local, origem da rota, comprimento do caminho, código de origem, métrica e outros atributos.

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

A próxima seçãoexplica esses 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 Exemplo de CIDR 2 (as-set) deste documento fornece um exemplo de AS_SET.

No exemplo nesta seção, o RTB anuncia a rede 192.168.190.0 no AS200. Quando essa rota atravessa o AS300, o RTC adiciona seus próprios número AS à rede. Quando 192.168.190.0 alcança o RTA, a rede tem dois números AS anexados: primeiro 200, depois 300. Para o RTA, o trajeto para alcançar 192.168.190.0 é (300, 200).

O mesmo processo aplica-se a 172.16.10.0 e a 192.168.180.10. O RTB tem que tomar o caminho (300, 100); o RTB atravessa o AS300 e depois o AS100 para alcançar 172.16.10.0. O RTC tem que atravessar o trajeto (200) a fim alcançar 192.168.190.0 e o trajeto (100) a fim alcançar 172.16.10.0.

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:

-

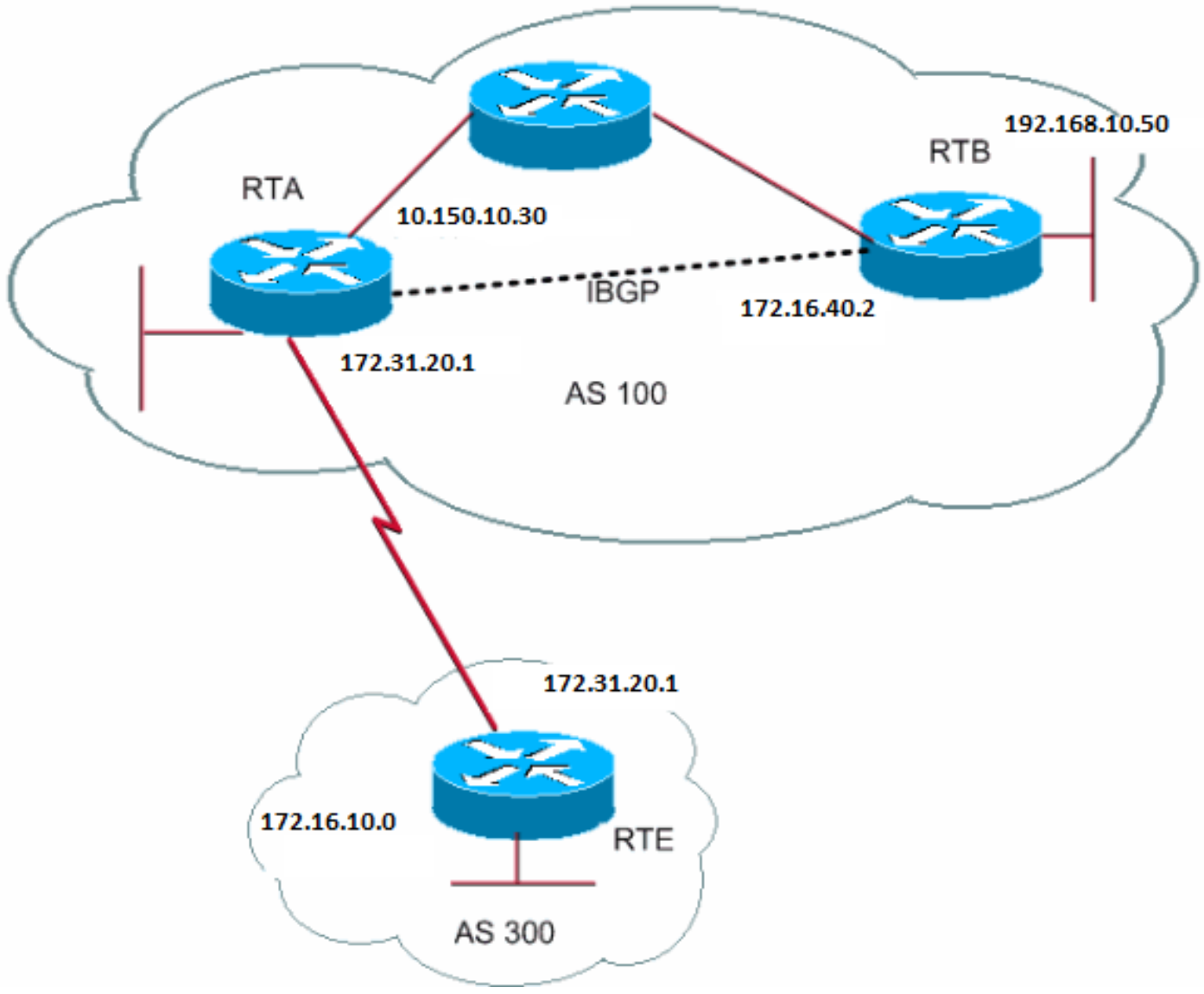
IGP - A informação de alcançabilidade da camada de rede (NLRI) é interior à origem do AS. Isso normalmente acontece quando você emite o **bgp network**comando . Nin na tabela BGP indica IGP.

-

EGP - O NLRI é descoberto por meio do exterior gateway protocol (EGP). Anena tabela de BGP indica EGP.

-

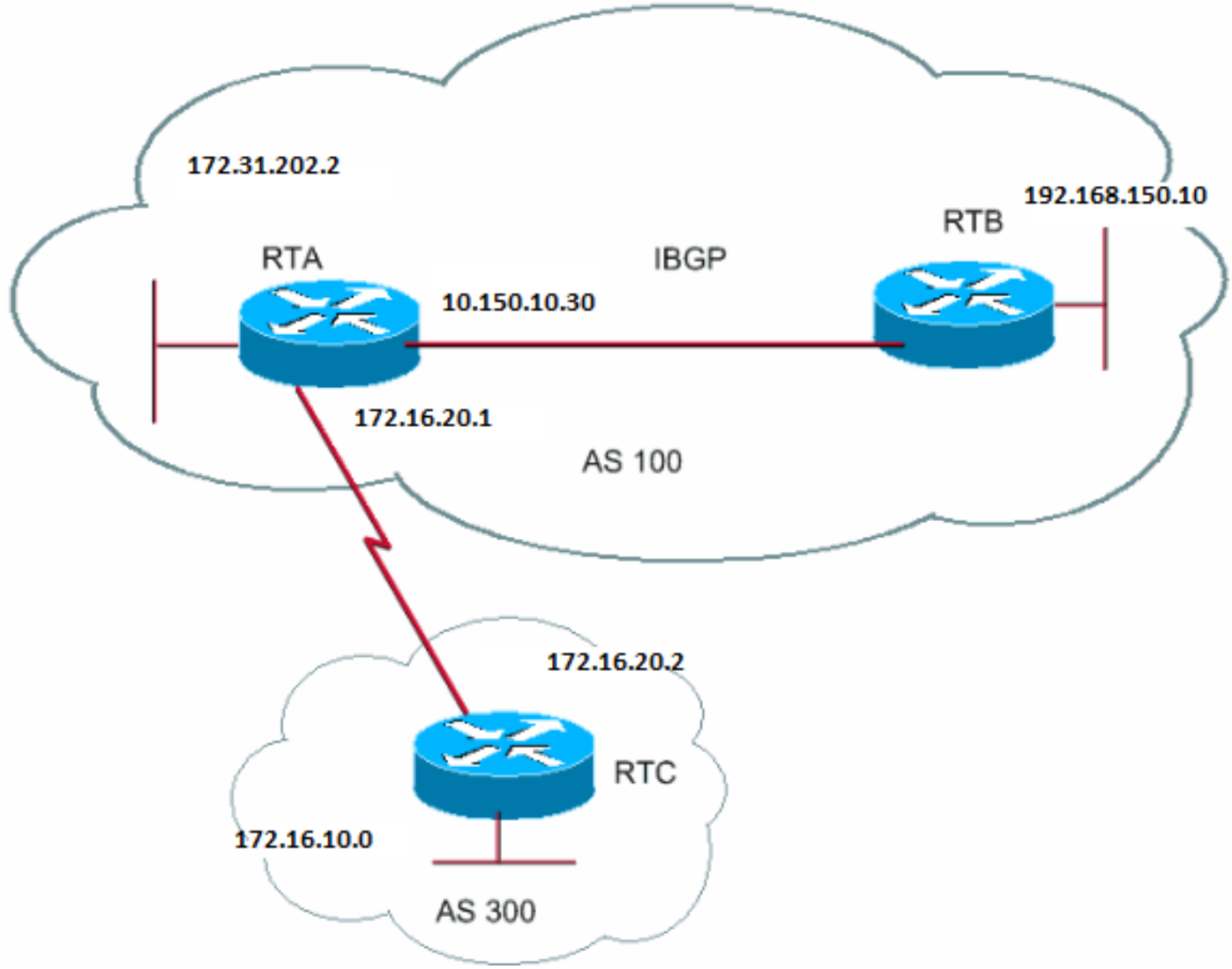
INCOMPLETO - O NLRI é desconhecido ou instruído através de outros meios. INCOMPLETO ocorre geralmente quando você redistribui rotas de outros protocolos de roteamento no BGP e a origem da rota está incompleta. Um ?na tabela BGP indica INCOMPLETO.



```
RTA#
router bgp 100
 neighbor 192.168.10.50 remote-as 100
 neighbor 172.31.20.2 remote-as 300
 network 172.31.202.2
 redistribute static

ip route 192.168.190.0 255.255.0.0 null0

RTB#
router bgp 100
 neighbor 10.150.10.30 remote-as 100
 network 192.168.10.150

RTE#
router bgp 300
 neighbor 172.31.20.1 remote-as 100
 network 172.16.10.0
```

O RTA alcança 172.16.10.0 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 192.168.10.150 através do i. Este "i" significa que a entrada está no mesmos AS e a origem é o IGP. O RTE alcança 172.31.202.2 através de 100 i. O "100 i" significa que o AS seguinte é 100 e a origem é IGP. O RTE também alcança 192.168.190.0 através de 100?. O "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



Atributo de próximo salto BGP

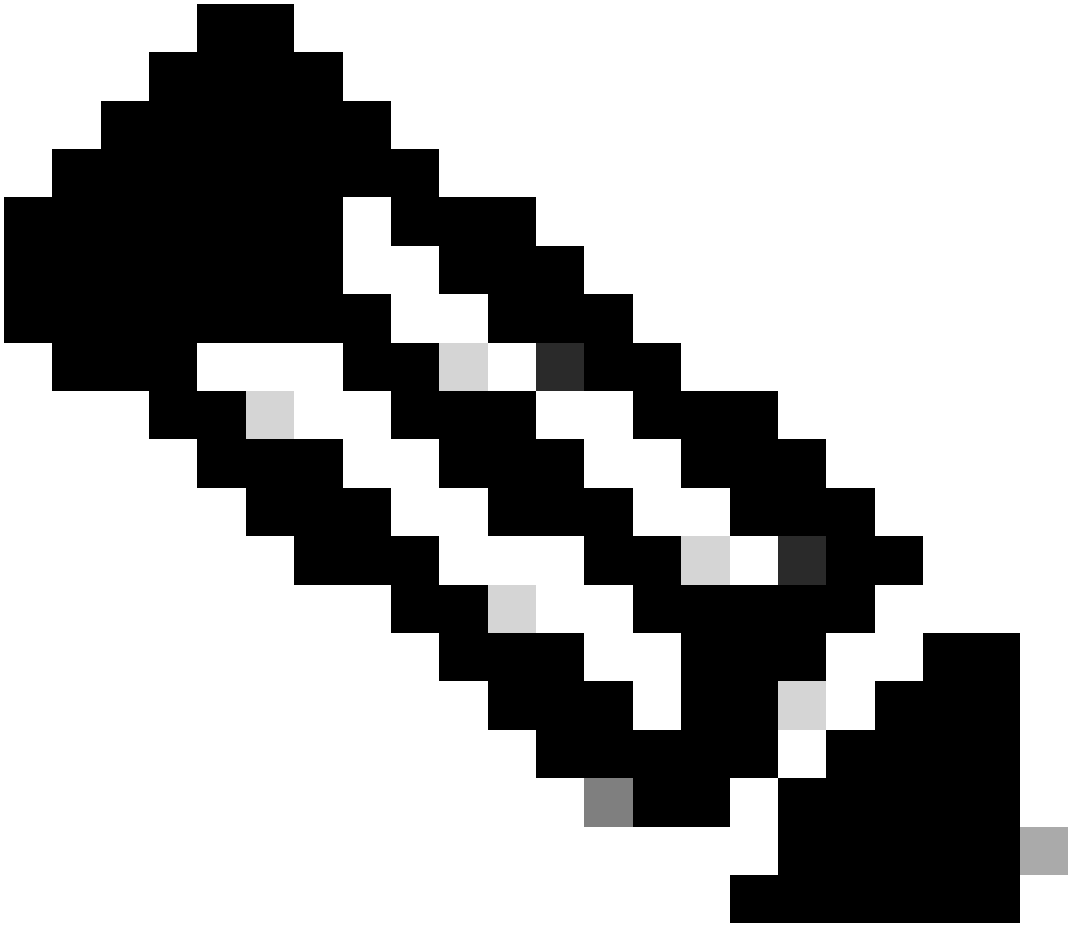
O atributo de próximo salto BGP é o endereço IP do salto seguinte a usar-se a fim de alcançar um determinado destino.

Para o eBGP, o próximo salto é sempre o endereço IP do vizinho que o neighbor comando especifica. No exemplo nesta seção, o RTC anuncia 172.16.10.0 ao RTA com um salto seguinte de 172.31.20.2. O RTA anuncia 172.31.202.2 ao RTC com um salto seguinte de 172.31.20.1. Para o iBGP, o protocolo afirma que o próximo salto que o eBGP anuncia deve ser transportado para o iBGP. Devido a esta regra, o RTA anuncia 172.16.10.0 a seu peer RTB do iBGP com um salto seguinte de 172.31.20.2. Com base no RTB, o próximo salto para acessar 172.16.10.0 é 172.31.20.2 *e não*10.150.10.30.

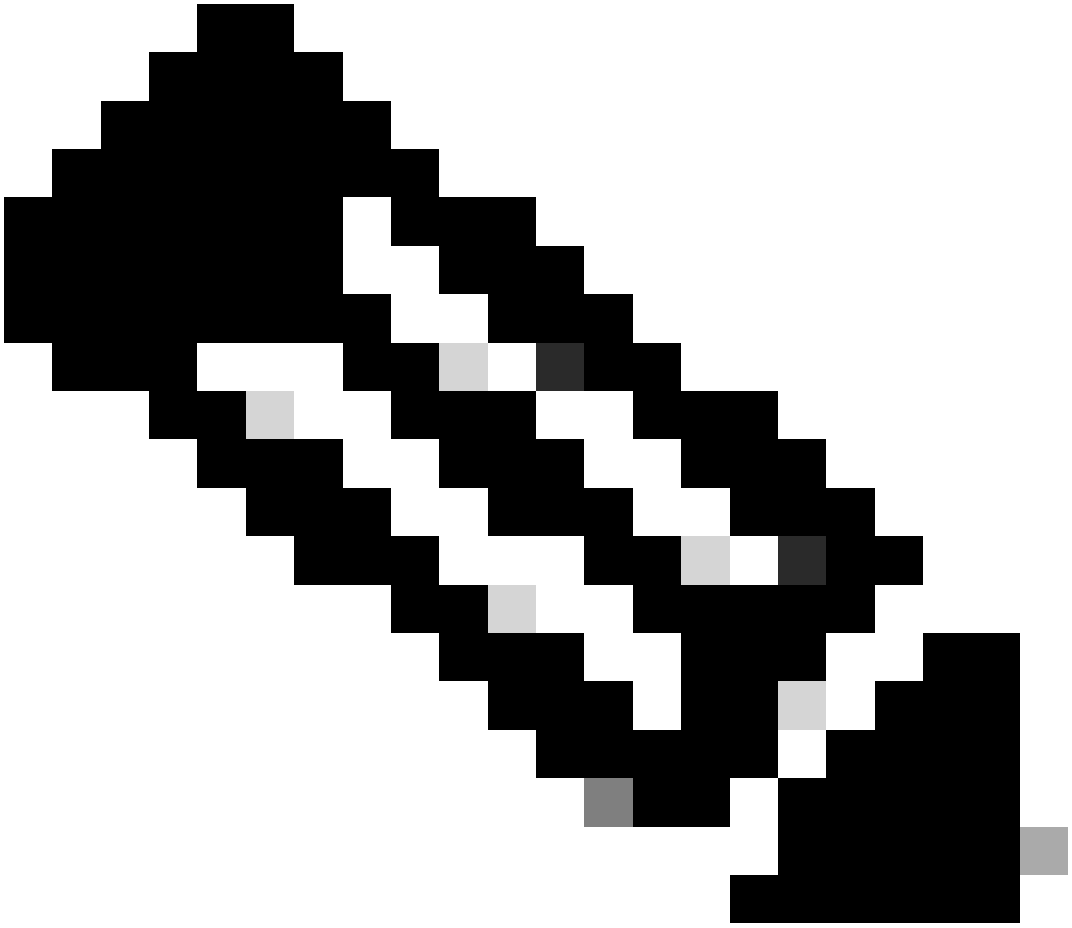
Certifique-se de que o RTB pode alcançar 172.31.20.2 através do IGP. Se não, o RTB derruba pacotes com destino 172.16.10.0 porque o endereço de próximo salto é inacessível. Por exemplo, se o RTB executa o iGRP, você pode igualmente executar o iGRP na rede RTA 172.16.10.0. Você quer fazer a voz passiva do iGRP no link ao RTC de modo que o BGP seja somente trocado.

```
RTA#
router bgp 100
 neighbor 172.31.20.2 remote-as 300
 neighbor 192.168.150.10 remote-as 100
 network 172.31.202.2

RTB#
```

```
router bgp 100
 neighbor 10.150.10.30 remote-as 100

RTC#
router bgp 300
 neighbor 172.31.20.1 remote-as 100
 network 172.16.10.0
```



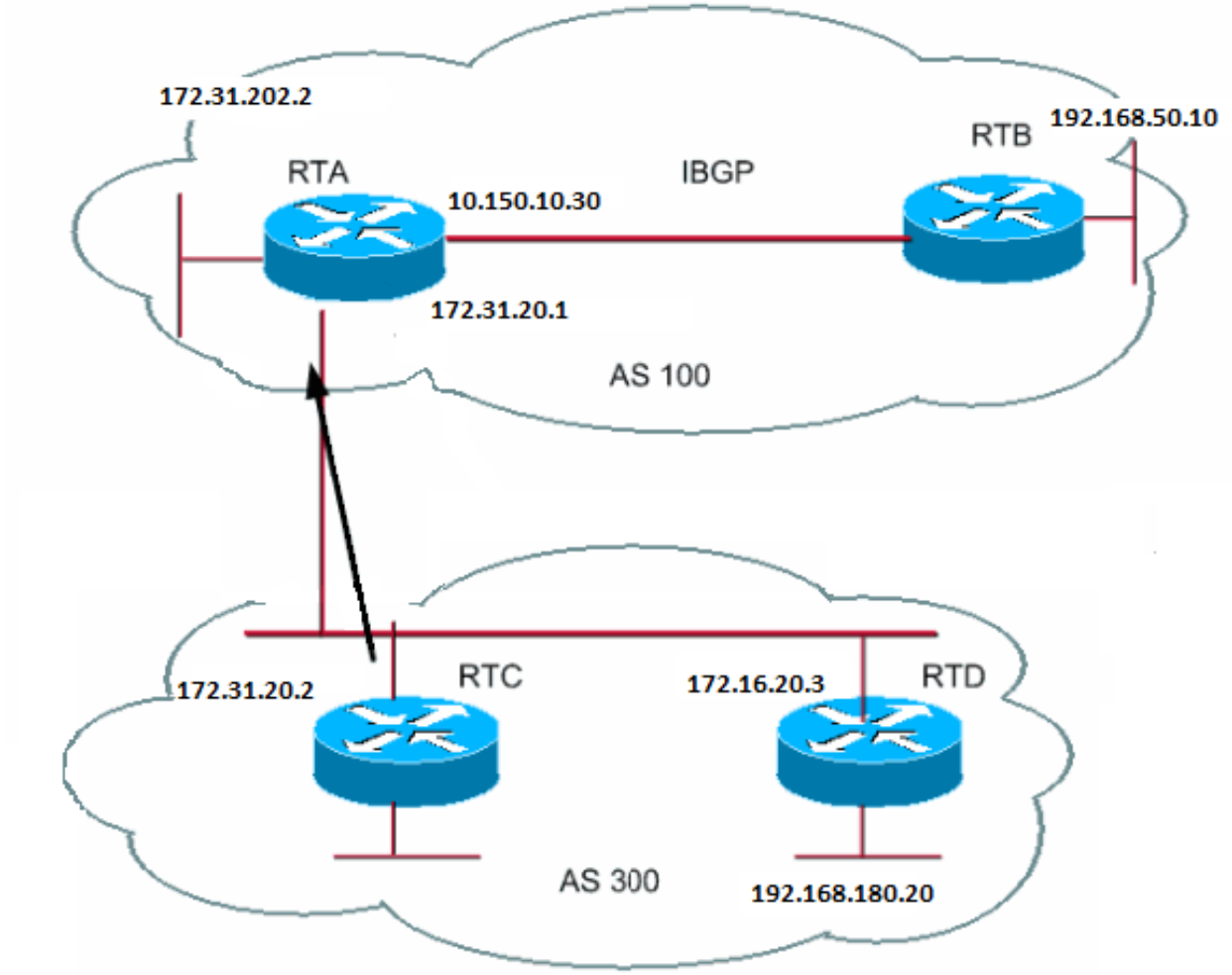
Observação: o RTC anuncia 172.16.10.0 ao RTA com um próximo salto igual a 172.31.20.2.



Observação: o RTA anuncia 172.16.10.0 ao RTB com um salto seguinte igual a 172.31.20.2. O salto seguinte do eBGP é levado em iPBP.

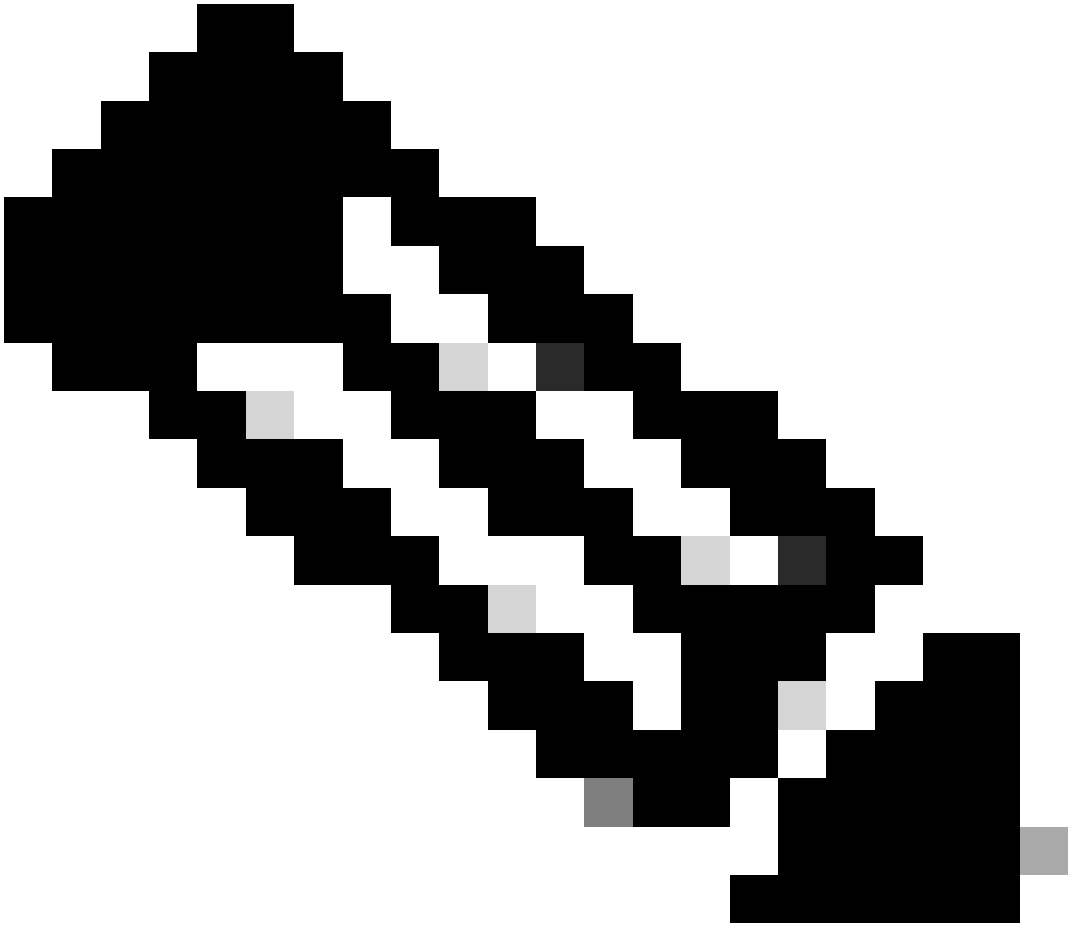
Tome especial cuidado ao lidar com redes multiacesso e multiacesso sem broadcast (NBMA). As seções Próximo salto BGP (redes multiacesso) e Próximo salto BGP (NBMA) fornecem mais detalhes.

Salto seguinte BGP (redes multi-acesso)



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

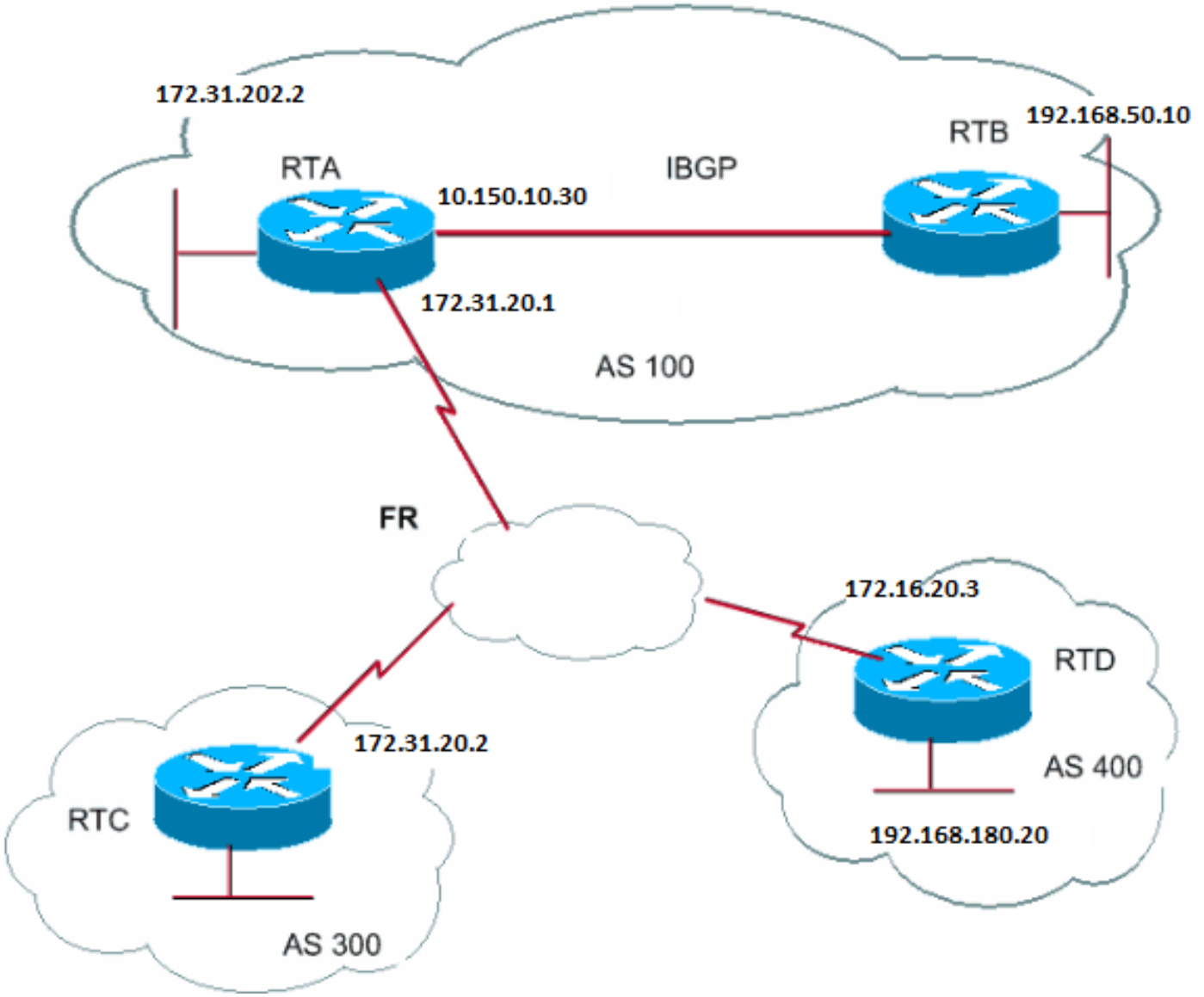
Suponha que o RTC e o RTD no AS300 executam o OSPF. O RTC executa o BGP com RTA. O RTC pode alcançar a rede 192.168.180.20 através de 172.16.20.3. Quando o RTC envia uma atualização BGP ao RTA referente ao 192.168.180.20, usa o RTC como o salto seguinte 172.16.20.3. O RTC não usa seu próprio endereço IP, 172.31.20.2. 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 192.168.180.20 é mais apreciável do que o salto extra através do RTC.



Observação: o RTC anuncia 192.168.180.20 ao RTA com um salto seguinte 172.16.20.3.

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 192.168.180.20 ao RTA com um salto seguinte de 172.16.20.3.

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 next-hop-selfcomando corrige essa situação.

comando next-hop-self

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

<#root>

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

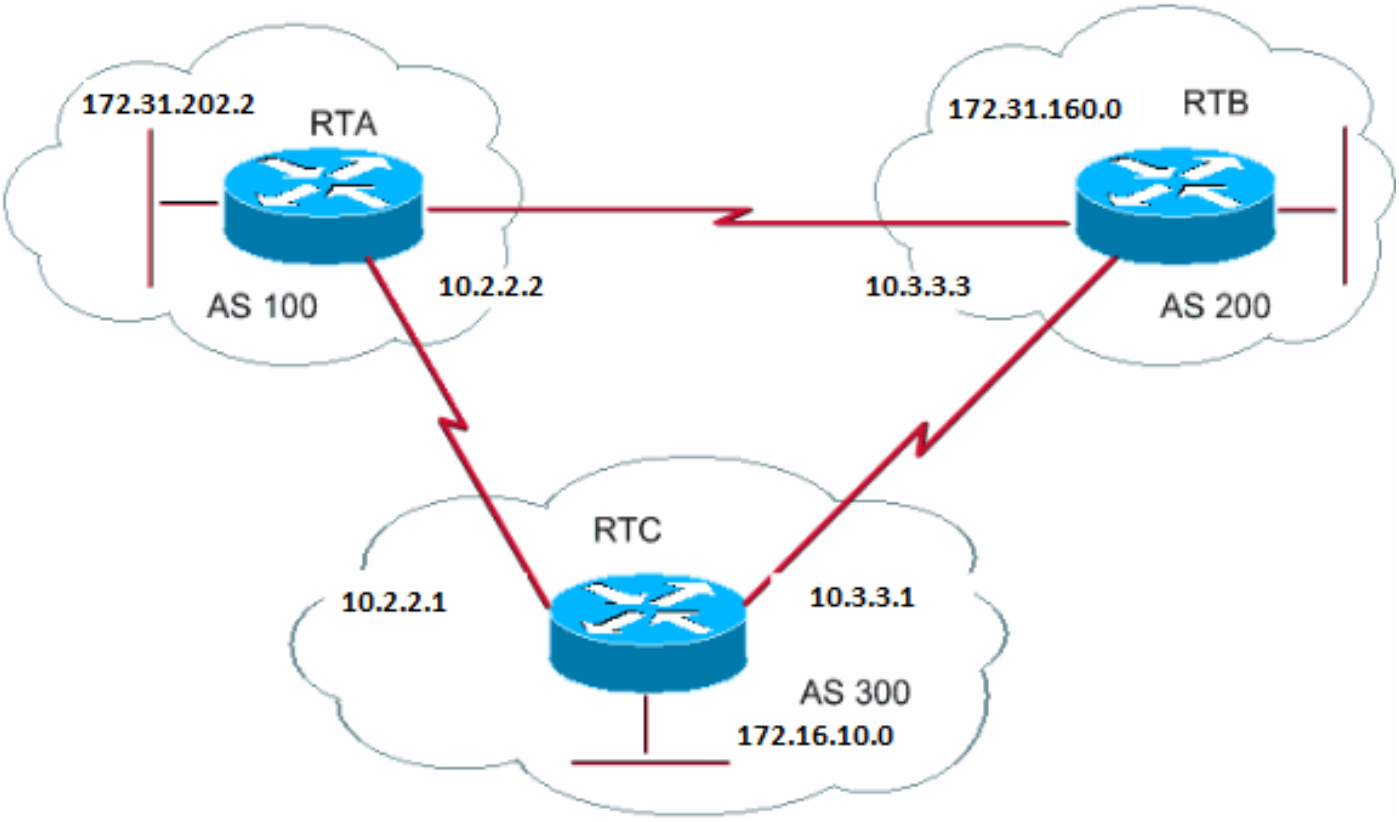
O next-hop-selfcomando permite forçar o BGP a usar um endereço IP específico como o próximo salto.

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

```
RTC#
router bgp 300
 neighbor 172.31.20.1 remote-as 100
 neighbor 172.31.20.1 next-hop-self
```

O RTC anuncia 192.168.180.20 com um salto seguinte igual a 172.31.20.2.

Backdoor de BGP



No diagrama anterior, 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 172.31.160.0 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 distance comando para alterar as distâncias padrão:

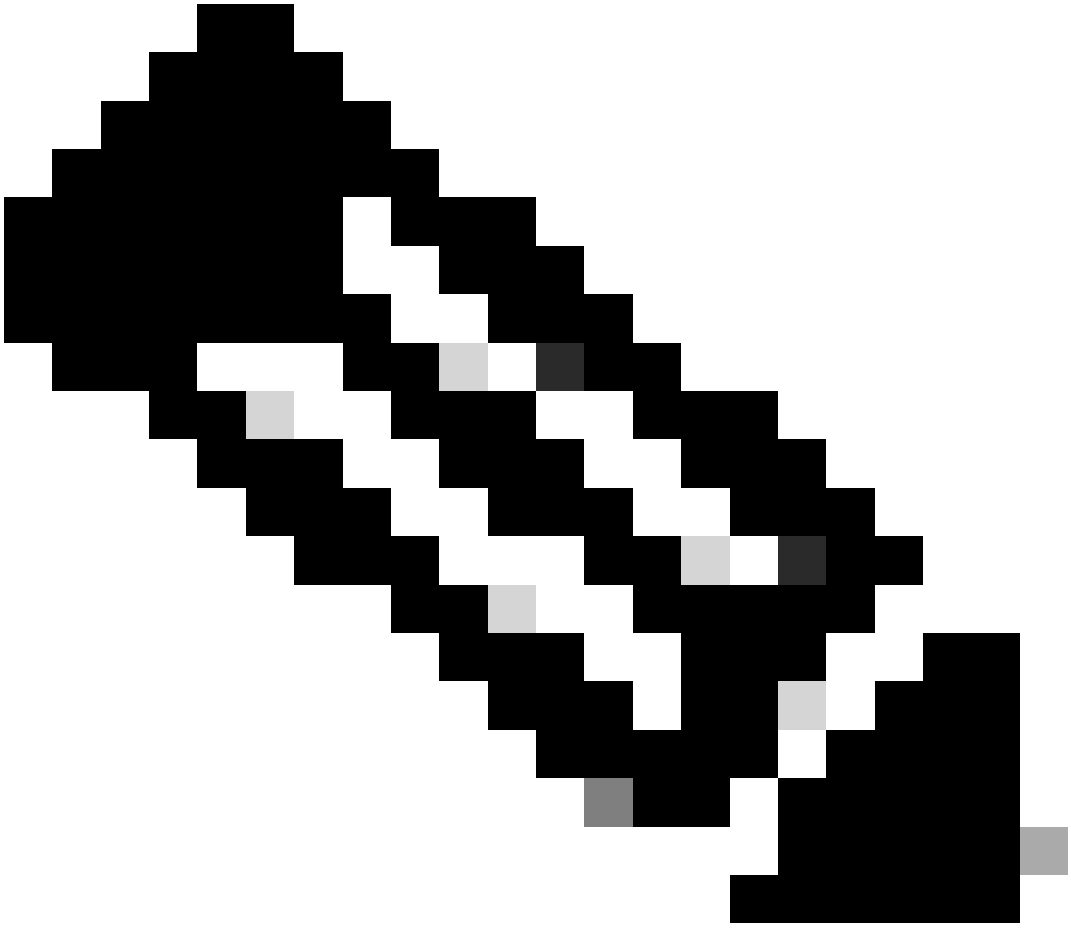
<#root>

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 172.31.160.0 através de RTB (IGP), você tem duas opções:

- Mude a distância externa do eBGP ou da distância IGP.



Observação: esta alteração não é recomendada.

-

Use o BGP backdoor.

O BGP backdoor faz à rota IGP a rota preferida.

Emita o comando **networkaddressbackdoor**.

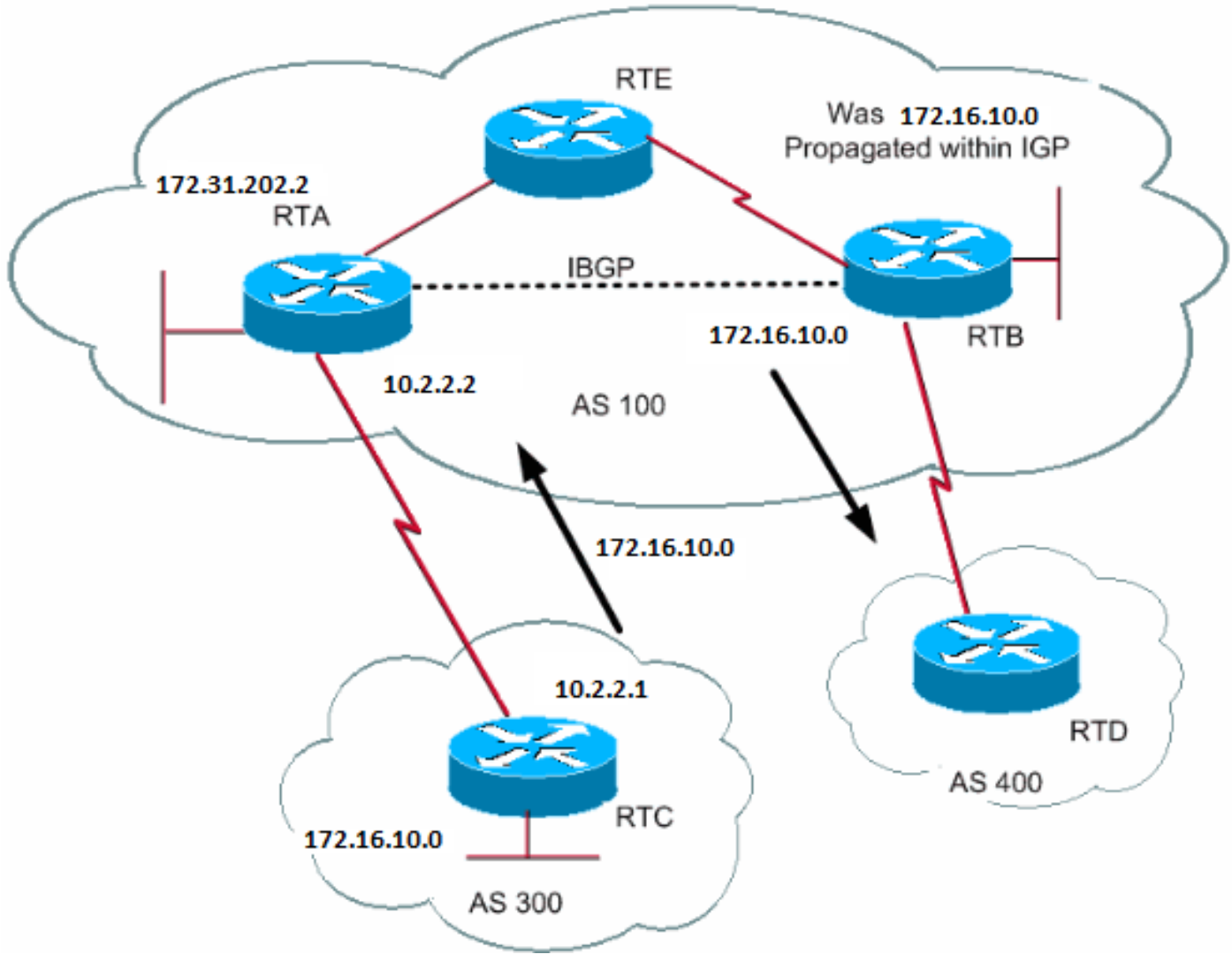
A rede configurada é a rede que você quer alcançar através do IGP. Para o BGP, esta rede obtém o mesmo tratamento que localmente uma rede atribuída, a não ser que as atualizações BGP não anunciem esta rede.

```
RTA#
router eigrp 10
 network 172.31.202.2

router bgp 100
 neighbor 10.2.2.1 remote-as 300
 network 172.31.160.0 backdoor
```

A rede 172.31.160.0 é tratada como uma entrada local, mas não é anunciada como uma entrada de rede normal.

O RTA aprende 172.31.160.0 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 172.16.10.0. RTA e RTB executam o iBGP,

assim que RTB obtém a atualização e pode alcançar 172.16.10.0 através do salto seguinte 10.2.2.1. 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 172.16.10.0 no IGP. Neste momento, o RTE não tem nenhuma ideia que 172.16.10.0 existe.

Se o RTB começa a anunciar ao AS400 que o RTB pode alcançar 172.16.10.0, o tráfego que vem do RTD para o RTB com o destino 172.16.10.0 flui e cai no RTE.

A sincronização afirma que, se o seu AS passar o tráfego de outro AS para um terceiro AS, o BGP não deve anunciar uma rota antes que todos os roteadores no seu AS tenham aprendido 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.

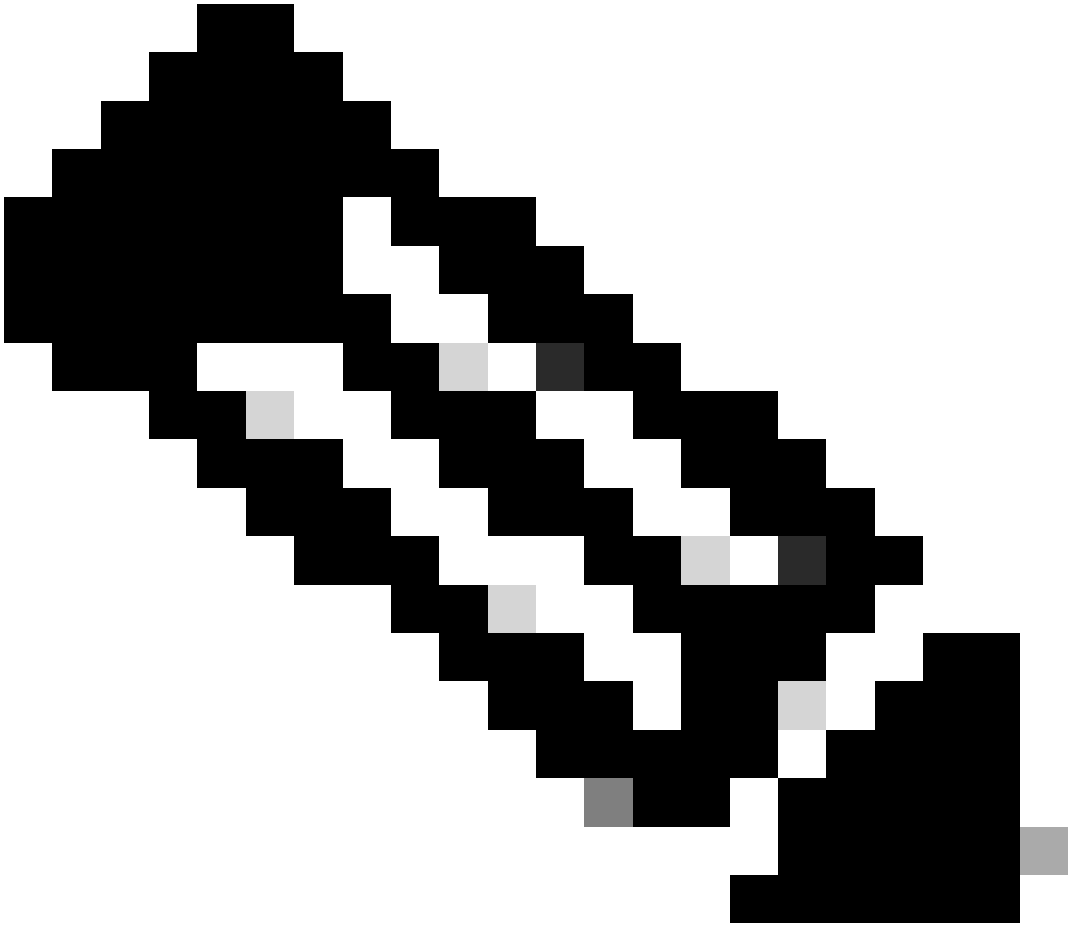
No exemplo nesta seção, o RTB espera para ouvir sobre 172.16.10.0 através do IGP. Então, o RTB começa a enviar a atualização ao RTD. Você pode fazer o RTB pensar que o IGP propagou a informação se você adiciona uma rota estática no RTB esses pontos a 172.16.10.0. Certifique-se de que outros roteadores podem alcançar 172.16.10.0.

Desabilite a sincronização

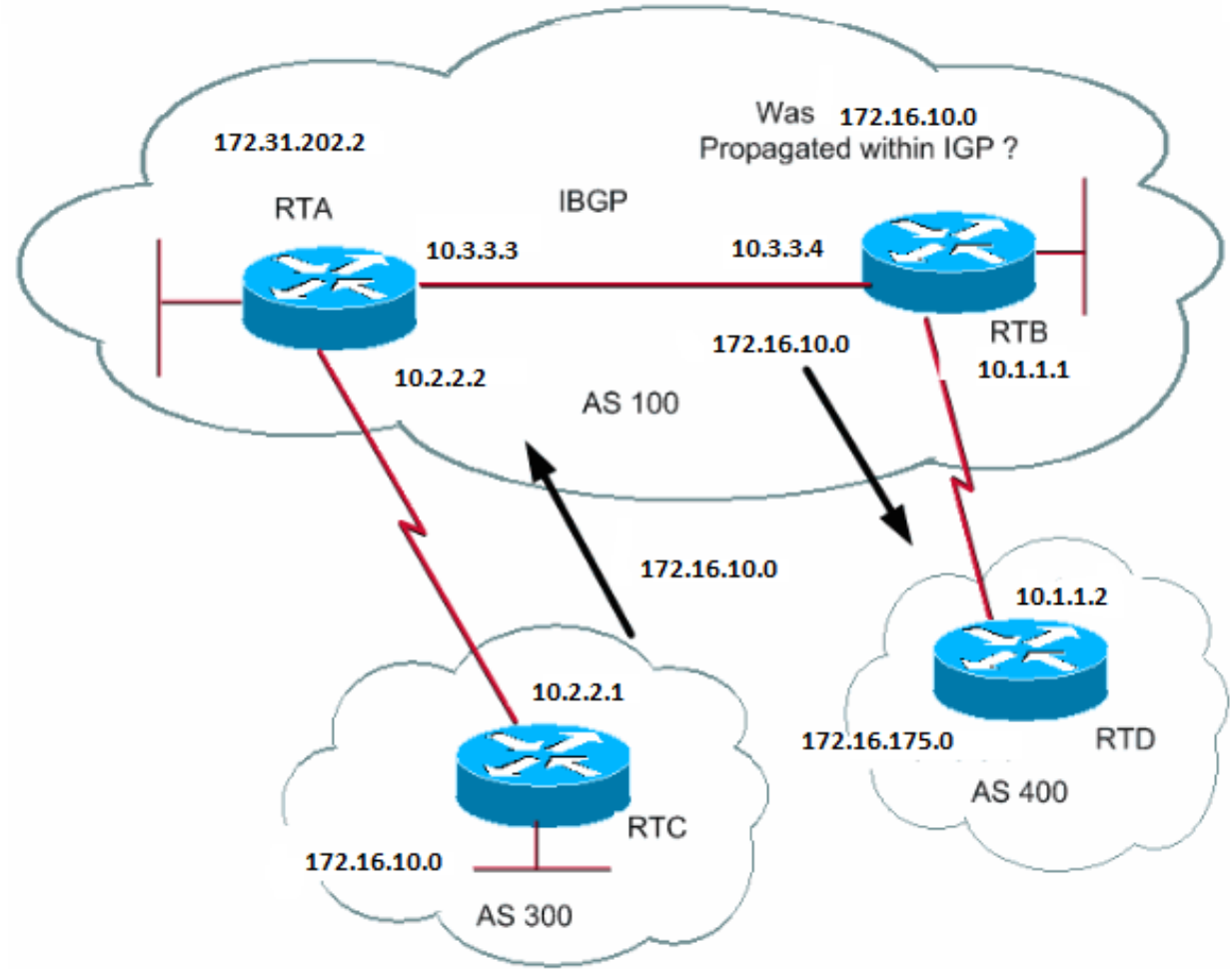
Em alguns casos, você não precisa sincronização. Se você não passa o tráfego de um AS diferente por seu AS, você pode desabilitar a sincronização. Você pode igualmente desabilitar a sincronização se todos os roteadores no seu AS executam o BGP. Desabilitar essa característica pode permitir você leve 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
```

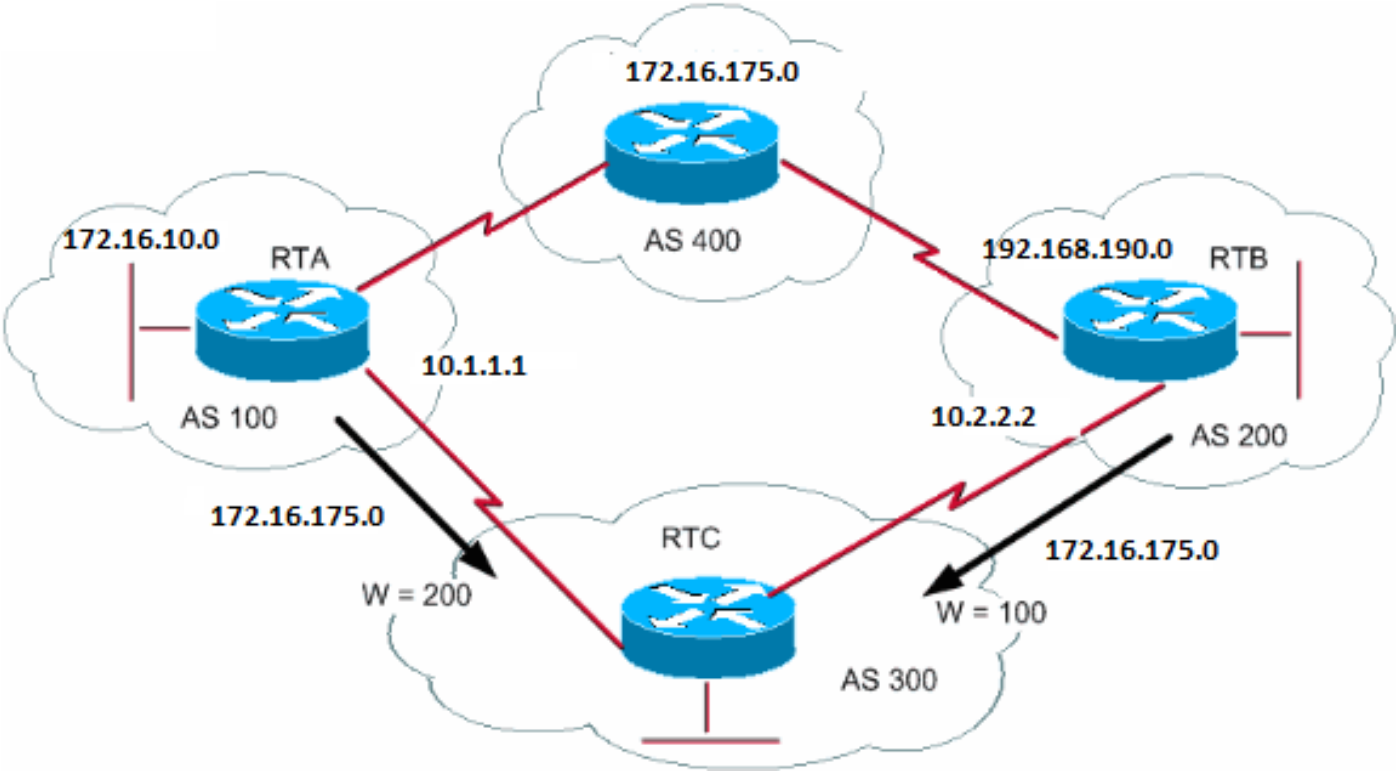


Observação: certifique-se de emitir o comando clear ip bgp address para reiniciar a sessão.




```
RTB#
router bgp 100
 network 172.31.202.2
 neighbor 10.1.1.2 remote-as 400
 neighbor 10.3.3.3 remote-as 100
 no synchronization

!--- RTB puts 172.16.10.0 in its IP routing table and advertises the network
!--- to RTD, even if RTB does not have an IGP path to 172.16.10.0.

RTD#
router bgp 400
 neighbor 10.1.1.1 remote-as 100
 network 172.16.0.0

RTA#
router bgp 100
 network 172.31.202.2
 neighbor 10.3.3.4 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 172.16.0.0 do AS4. O RTA propaga a atualização ao RTC. O RTB igualmente aprendeu sobre a rede 172.16.0.0 do AS4. O RTB propaga a atualização ao RTC. O RTC agora tem duas maneiras de alcançar 172.16.0.0 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 172.16.0.0. Diversos métodos conseguem este peso ajustado:

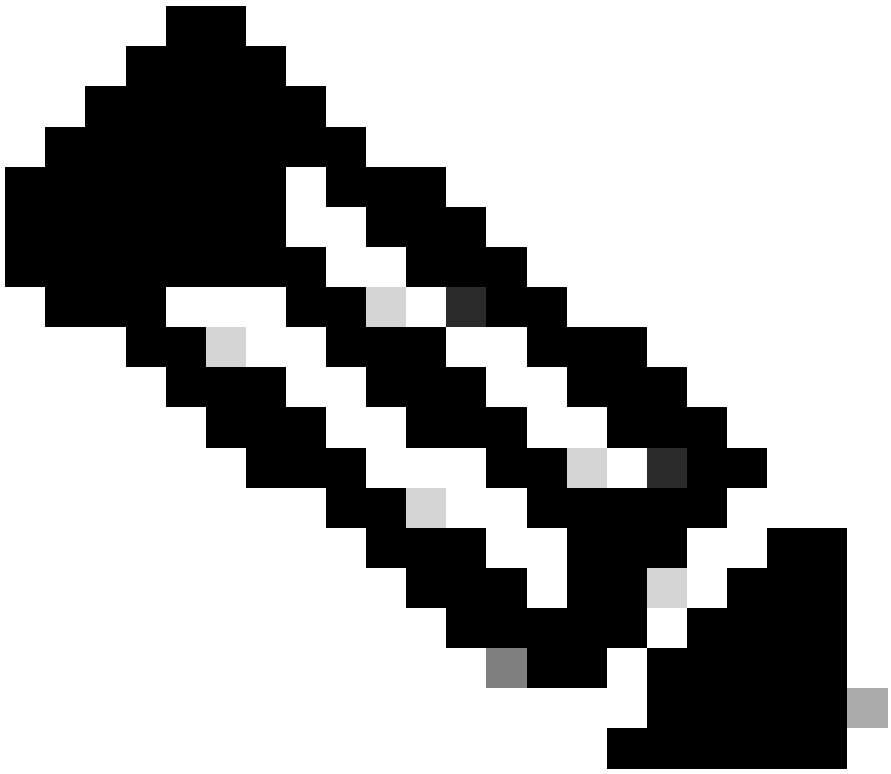
- Use o comando neighbor.

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

- Use as listas de acessos AS_PATH.

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

neighbor <ip-address>filter-list <access-list-number>weight <weight>



Observação: em alguns cenários, pode haver poucos comandos que não estão disponíveis em algumas versões de software.

- Use mapas de rotas.

```
RTC#
router bgp 300
 neighbor 10.1.1.1 remote-as 100
 neighbor 10.1.1.1 weight 200

!--- The route to 172.16.0.0 from RTA has a 200 weight.

 neighbor 10.2.2.2 remote-as 200
 neighbor 10.2.2.2 weight 100

!--- The route to 172.16.0.0 from RTB has a 100 weight.
```

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

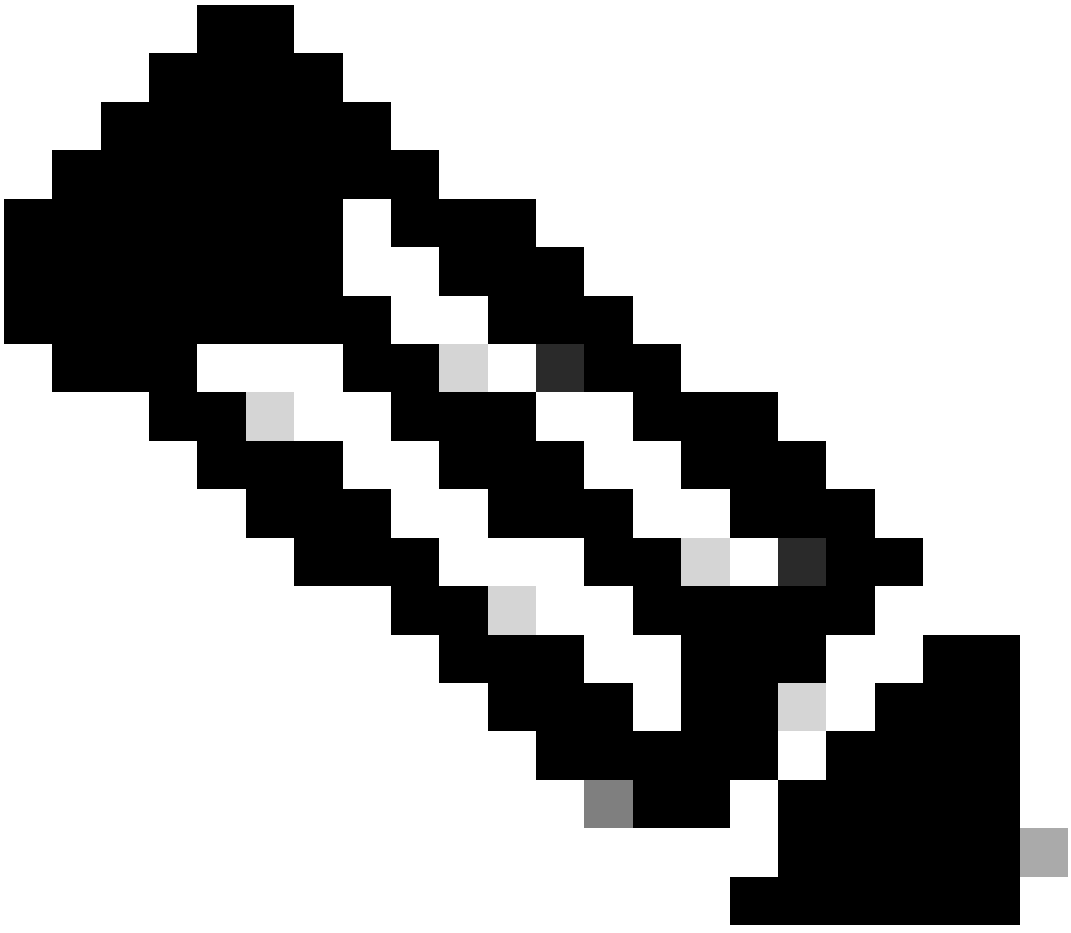
Você pode conseguir o mesmo resultado com IP AS_PATH e listas de filtro.

```
RTC#
router bgp 300
 neighbor 10.1.1.1 remote-as 100
 neighbor 10.1.1.1 filter-list 5 weight 200
 neighbor 10.2.2.2 remote-as 200
 neighbor 10.2.2.2 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.

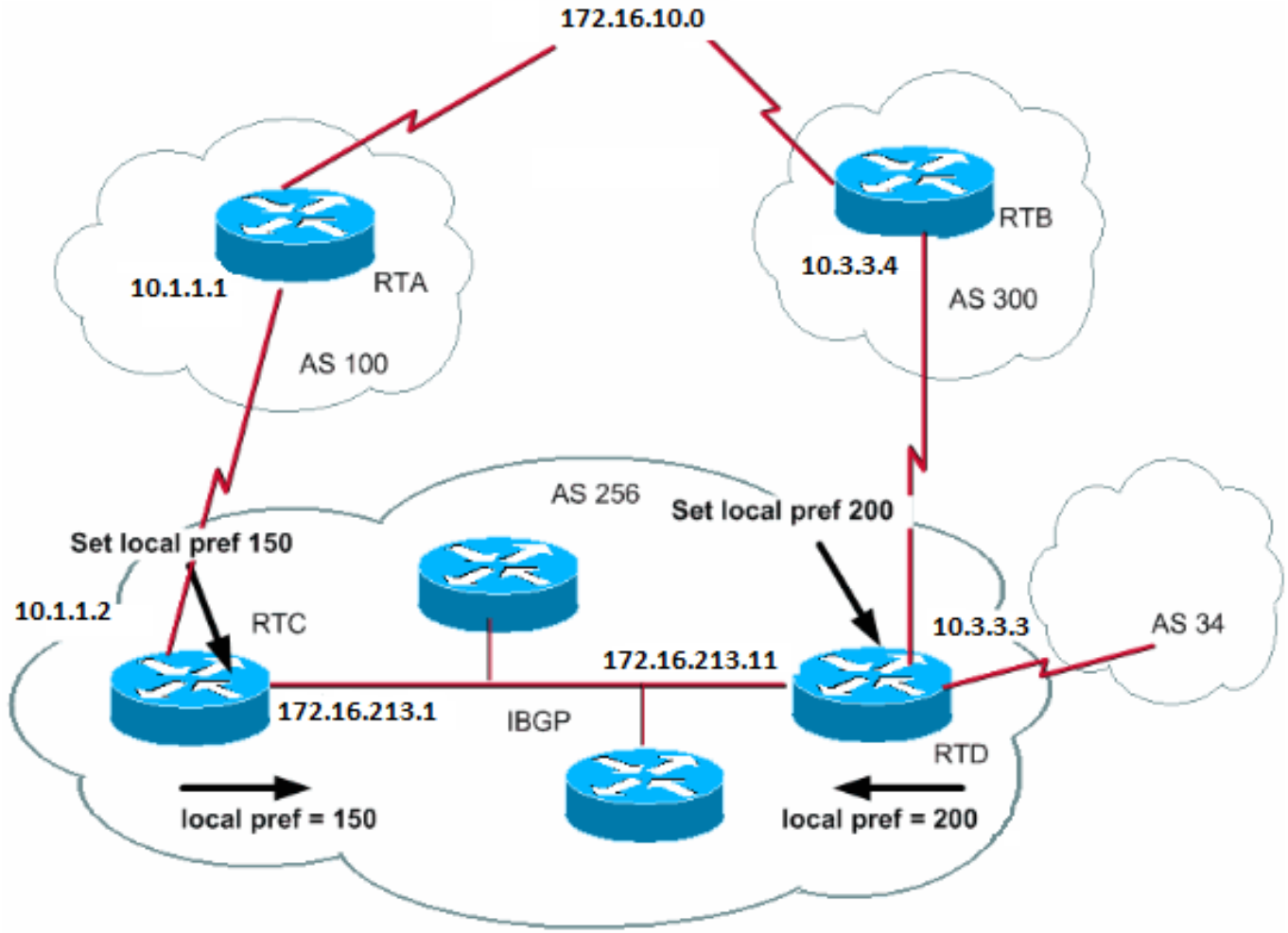
```
RTC#
router bgp 300
 neighbor 10.1.1.1 remote-as 100
 neighbor 10.1.1.1 route-map setweightin in
 neighbor 10.2.2.2 remote-as 200
 neighbor 10.2.2.2 route-map setweightin in
...
ip as-path access-list 5 permit ^100$
```

```
...

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.
```



Observação: você pode modificar o peso para preferir o caminho BGP de VPN MPLS com o caminho IGP como um backup.

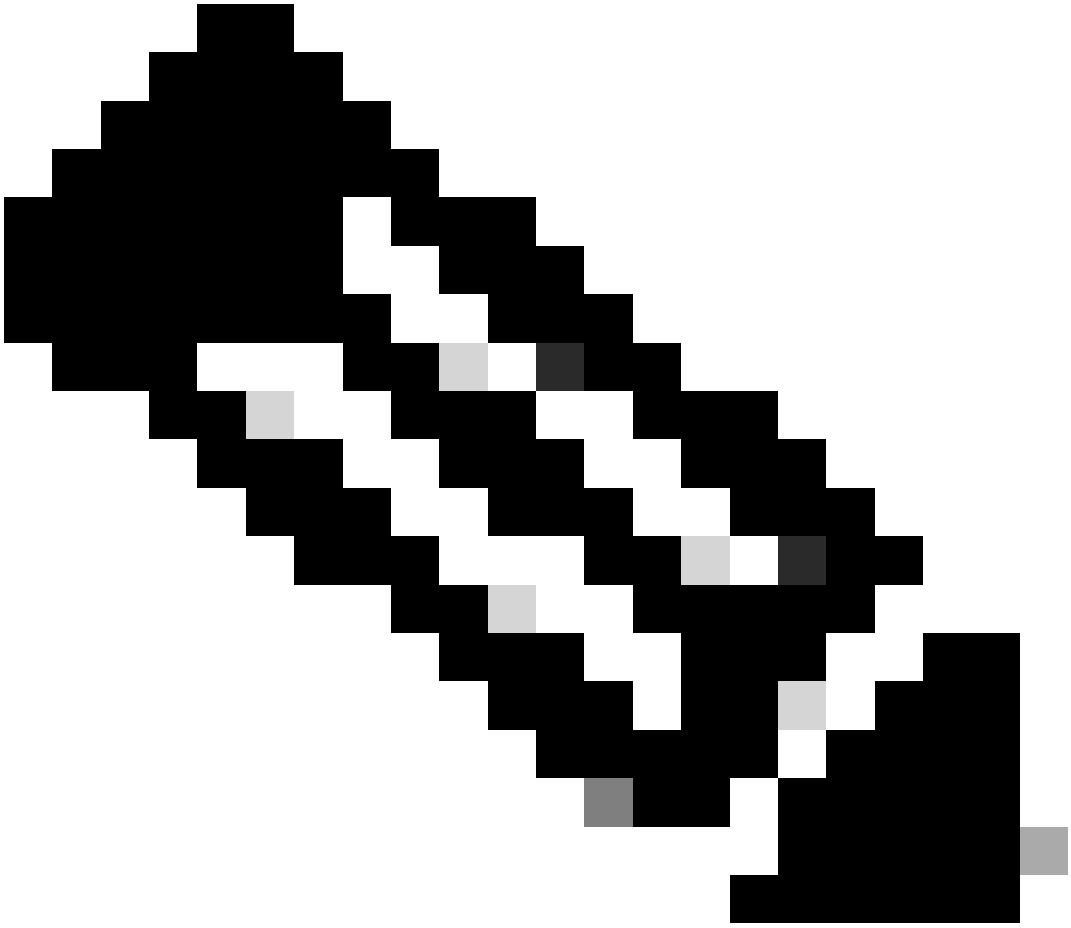
Atributo de preferência local



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:



Observação: é necessário executar uma reinicialização suave (isto é, limpar o processo bgp no roteador) para que as alterações sejam levadas em consideração. Para limpar o processo bgp, use o comandoclear ip bgp [soft][in/out] onde soft indica uma reinicialização suave e não desmonta a sessão e [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 172.16.10.0 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 10.1.1.1 remote-as 100
  neighbor 10.213.11.2 remote-as 256
 bgp default local-preference 150

RTD#
router bgp 256
 neighbor 10.3.3.4 remote-as 300
 neighbor 10.213.11.1 remote-as 256
 bgp default local-preference 200
```

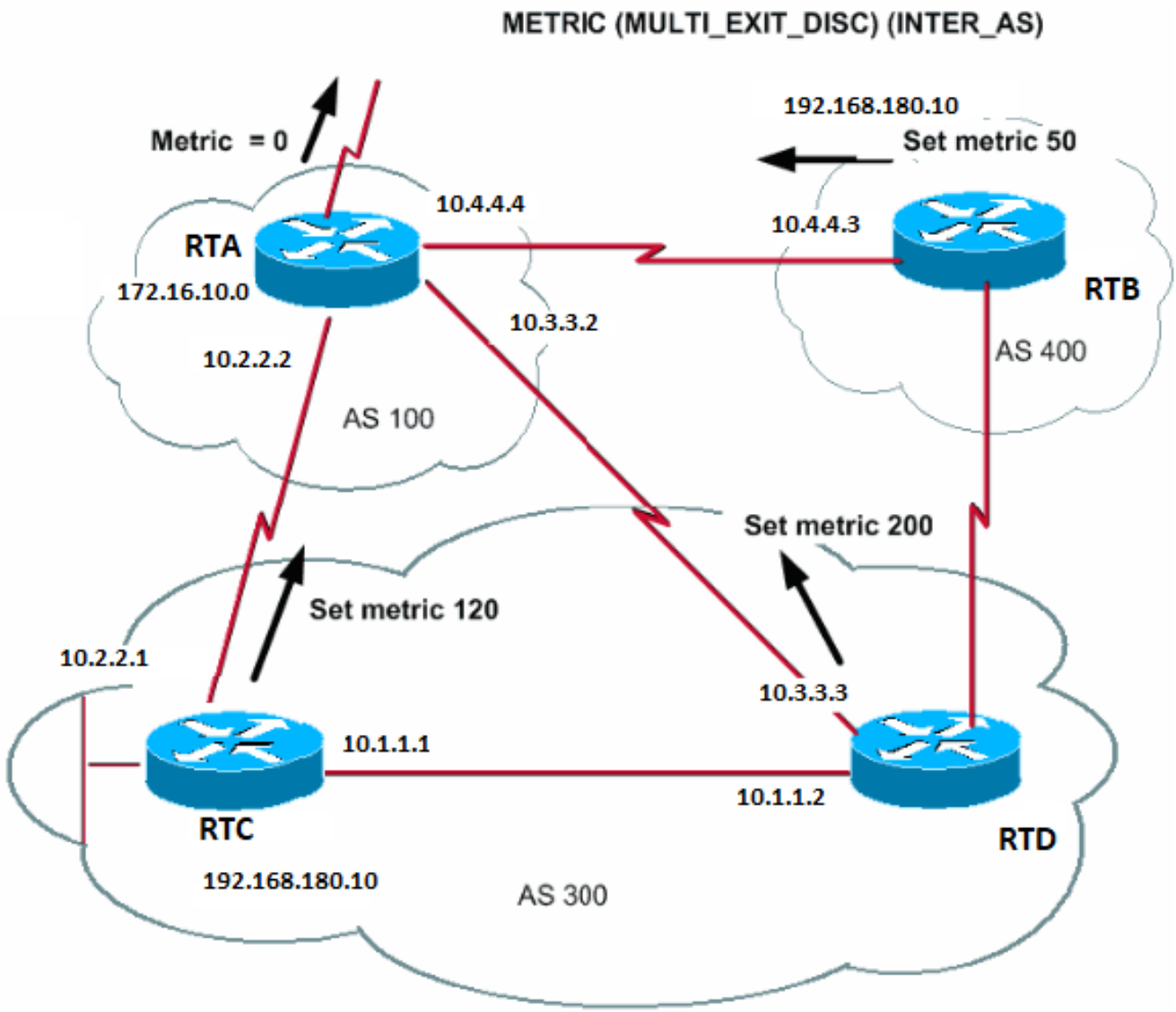
Nesta configuração, o RTC ajusta a preferência local de todas as atualizações a 150. O mesmo RTD ajusta a preferência local de todas as atualizações a 200. Há uma troca da preferência local dentro do AS256. Conseqüentemente, o RTC e o RTD realizam que a rede 172.16.10.0 tem uma preferência local mais 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 10.3.3.4 remote-as 300
 neighbor 10.3.3.4 route-map setlocalin in
 neighbor 10.213.11.1 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.

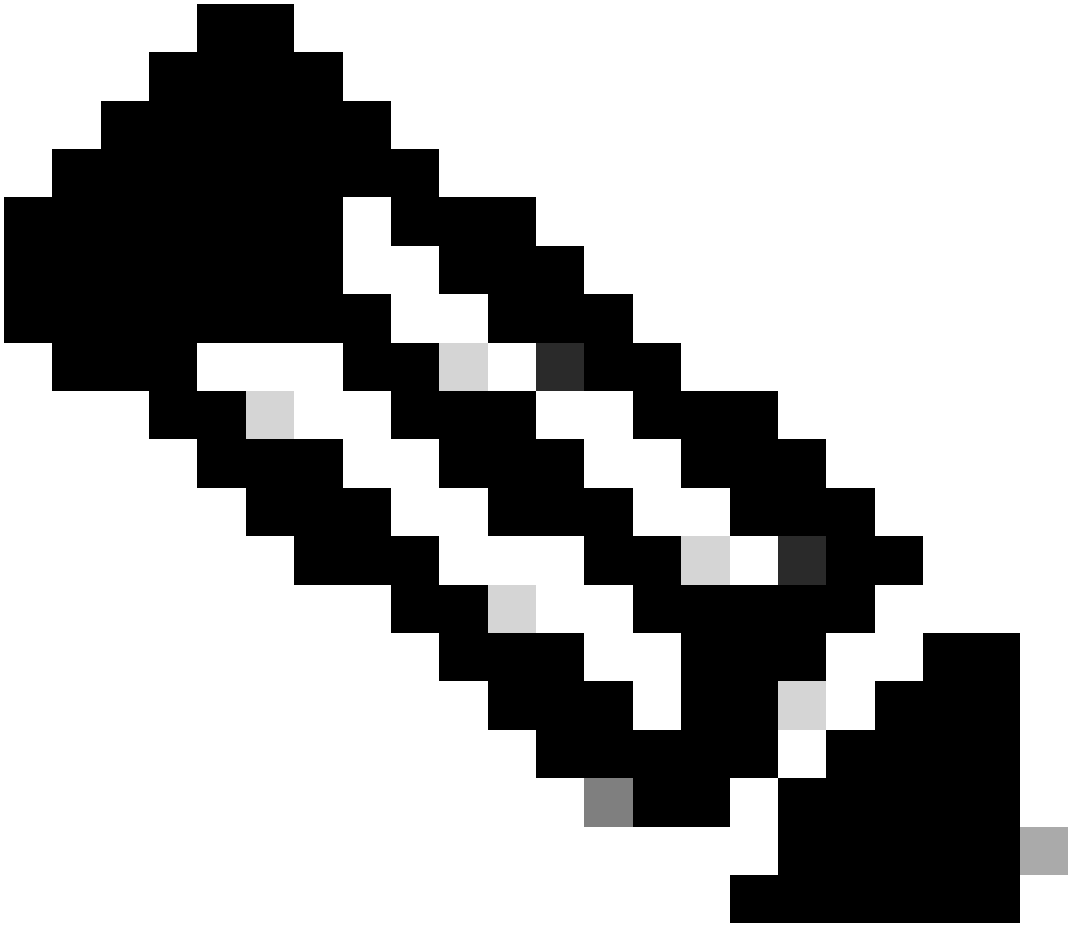
Atributo de métrica



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.



Observação: há dois comandos de configuração BGP que podem influenciar a seleção de caminho baseada no discriminador de várias saídas (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 informações sobre a rede 192.168.180.10 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 de AS-Path no RTA pelo comando **bgp bestpath as-path ignore** é ignorada. Ele é configurado para forçar o BGP a cair no próximo atributo para comparação de rota (neste caso métrica ou MED). Se o comando for omitido, o BGP poderá instalar a rota

192.168.180.10 do roteador RTC, pois ela tem o caminho AS 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 10.2.2.1 remote-as 300
 neighbor 10.3.3.3 remote-as 300
 neighbor 10.4.4.3 remote-as 400
 bgp bestpath as-path ignore

RTC#
router bgp 300
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.2.2.2 route-map setmetricout out
 neighbor 10.1.1.2 remote-as 300

route-map setmetricout permit 10
   set metric 120

RTD#
router bgp 300
 neighbor 10.3.3.2 remote-as 100
 neighbor 10.3.3.2 route-map setmetricout out
 neighbor 10.1.1.1 remote-as 300

route-map setmetricout permit 10
 set metric 200

RTB#
router bgp 400
 neighbor 10.4.4.4 remote-as 100
 neighbor 10.4.4.4 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 10.3.3.3 remote-as 300
```

```
 neighbor 10.4.4.3 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 192.168.180.10.

Você também pode definir métricas durante a redistribuição de rotas no BGP se você emitir o comando **default-metricnumber**.

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 192.168.180.10 255.255.0.0 null 0

!--- This causes RTB to send out 192.168.180.10 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 aplicar decisões de roteamento que correspondam a essas 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:

<#root>

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
```

or

```
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 10.3.3.3 remote-as 300
 neighbor 10.3.3.3 send-community
 neighbor 10.3.3.3 route-map setcommunity out
```

No Cisco IOS Software Release 12.0 e posterior, você pode configurar comunidades em três formatos diferentes: decimal, hexadecimal e AA:NN. Por padrão, o Cisco IOS Software usa o formato decimal mais velho. Para configurar e exibir em AA:NN, execute o comando **ip bgp-community new**-global configuration format. A primeira parte de AA:NN representa o número AS e a segunda parte representa um número de 2 bytes.

Aqui está um exemplo:

Sem o comando **ip bgp-community new-format** na configuração global, uma emissão do comando **show ip bgp 10.6.0.0** exibe o valor do atributo de comunidade no formato decimal. Neste exemplo, o valor do atributo de comunidade aparece como6553620.

<#root>

```
Router#
```

show ip bgp 10.6.0.0

```
BGP routing table entry for 10.6.0.0/8, 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 (10.255.255.1)
      Origin IGP, metric 0, localpref 100, valid, external, best
```

Community: 6553620

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

```
<#root>

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 de configuração global ip bgp-community new**-format, o valor da comunidade é exibido no formato AA:NN. O valor aparece como`100:20` na saída do **comando show ip bgp 10.6.0.0** neste exemplo:

<#root>

```
Router#
```

show ip bgp 10.6.0.0

```
BGP routing table entry for 10.6.0.0/8, 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 (10.255.255.1)
      Origin IGP, metric 0, localpref 100, valid, external, best
```

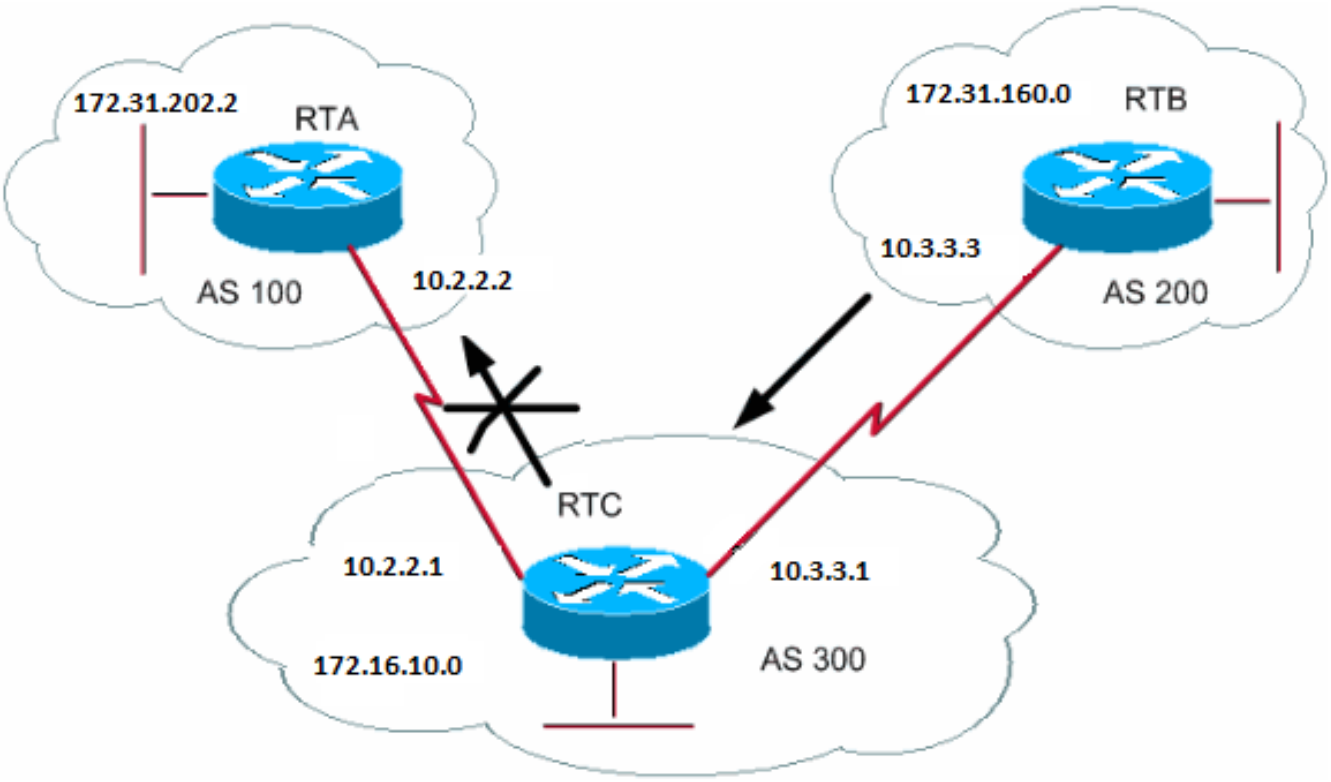
Community: 100:20

Estudos de Caso do BGP 3

Filtro 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.

Filtro 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:

```
<#root>
```

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

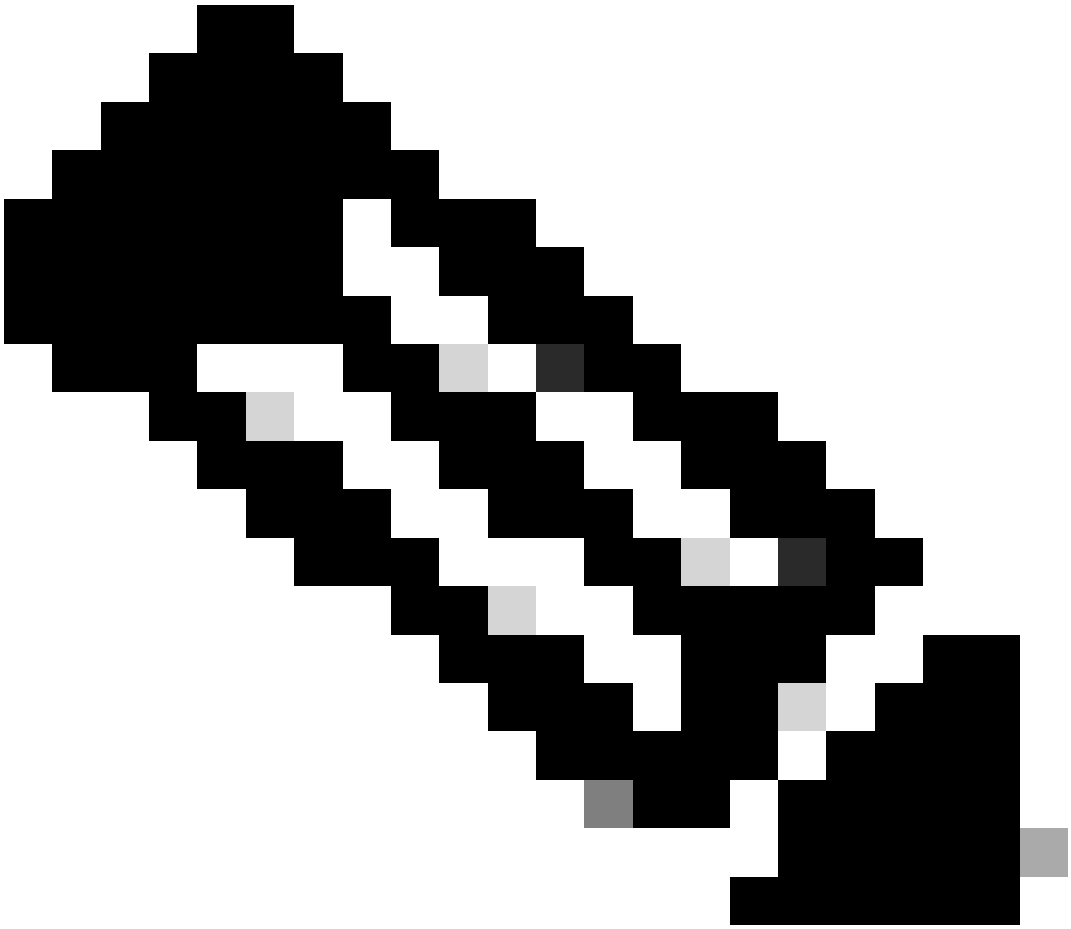
Neste exemplo, o RTB origina a rede 172.31.160.0 e envia a atualização ao RTC. Se o RTC quer parar a propagação das atualizações ao AS100, você deve definir uma lista de acessos para filtrar aquelas atualizações e para aplicar a lista de acessos durante uma comunicação com o RTA:

```
RTC#
router bgp 300
```

```
 network 172.16.10.0
 neighbor 10.3.3.3 remote-as 200
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.2.2.2 distribute-list 1 out

access-list 1 deny 172.31.160.0 0.0.255.255

access-list 1 permit 0.0.0.0 255.255.255.255

!--- Filter out all routing updates about 160.10.x.x.
```

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 192.168.160.0/8.



Observação: a notação /8 significa que você usa 8 bits de máscara de sub-rede, que começam da extrema esquerda do endereço IP. Este endereço é equivalente a 192.168.160.0 255.0.0.0.

O comando access-list 1 permit 192.168.160.0 0.255.255.25 5permite 192.168.160.0/8, 192.168.160.0/9 e assim por diante. A fim de restringir a atualização a somente 192.168.160.0/8, você deve usar uma lista de acesso estendida deste formato:

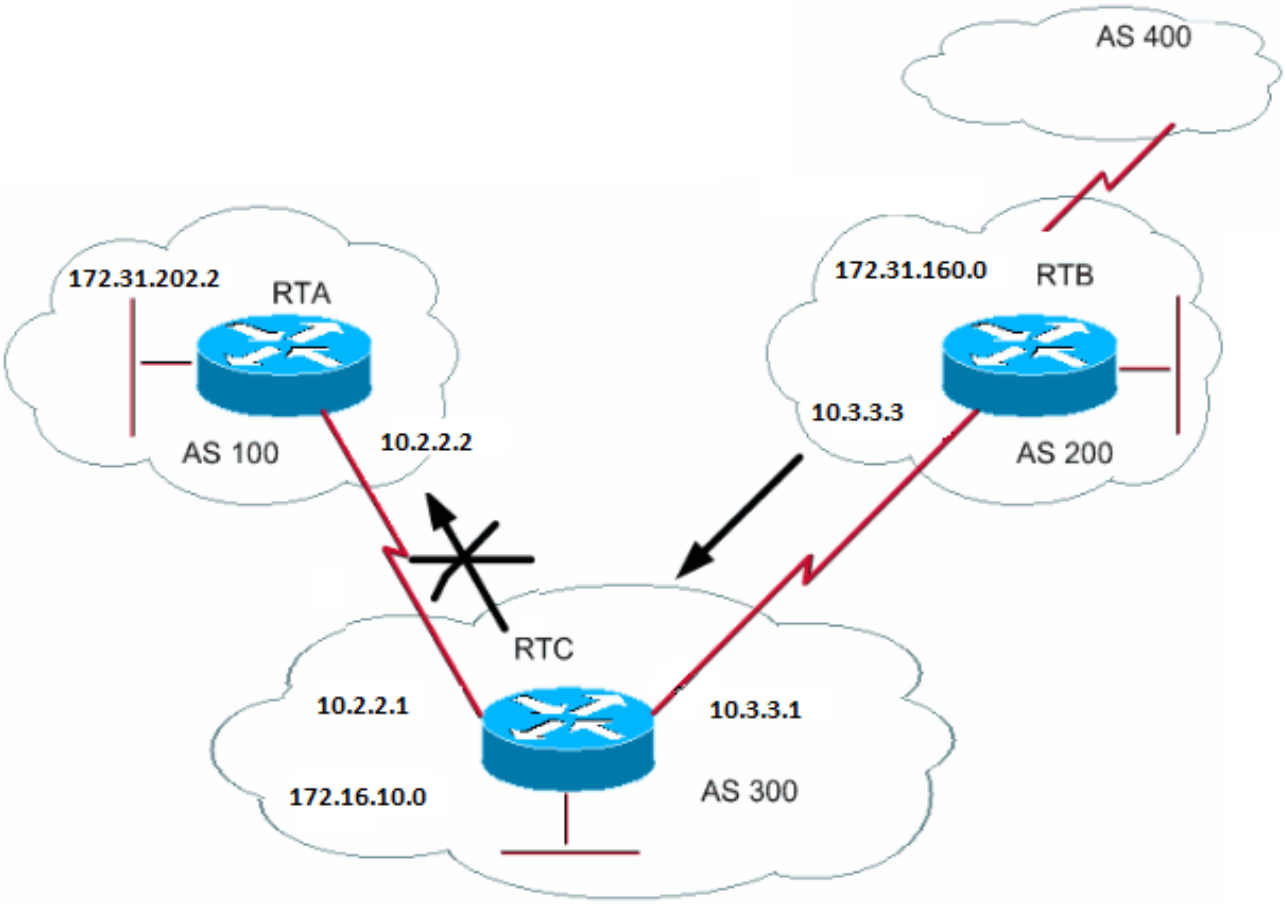
```
<#root>

access-list 101 permit ip 192.168.160.0 0.255.255.255 255.0.0.0 0.0.0.0.
```

Esta lista permite 192.168.160.0/8 somente.

Consulte Block One or More Networks From a BGP Peer para obter configurações de exemplo sobre como filtrar redes de peers BGP. O método usa o comando **distribute-list** com listas de controle de acesso (ACLs) padrão e estendidas, bem como a capacidade de filtrar a lista de prefixos.

Filtro de caminho

Você também pode filtrar caminhos.



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 172.31.160.0 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:

<#root>

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

<#root>

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

Este exemplo para o RTC envia atualizações sobre 172.31.160.0 ao RTA:

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

!--- The 1 is the access list number below.

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

O access-list 1comando neste exemplo força a negação de quaisquer atualizações com informações de caminho que começam com 200 e terminam com 200. O ^200$ no comando é uma “expressão regular”, em que o ^ significa que “começa com” e $ significa “termina com”. Como o RTB envia atualizações sobre 172.31.160.0 com informações de caminho que começam com 200 e terminam com 200, as atualizações correspondem à lista de acesso. A lista de acessos rejeita estas atualizações.

O `.*`é outra expressão regular em que o `.`significa "qualquer caractere" e o`*`significa "a repetição desse caractere". So `.*`representa qualquer informação de caminho, que é necessária para permitir a transmissão de todas as outras atualizações.

O que acontece se, em vez do uso de`^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. Essas atualizações correspondem à lista de acesso`^200` porque as informações de caminho começam com 200. A lista de acessos impede a transmissão destas atualizações ao RTA, que não é a exigência.

Para verificar se você implementou a expressão regular correta, execute o comando **show ip bgp regexpregular-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 da seção **Path Filter** , você especificou a string^200**$.** Você queria que as informações de caminho que vêm dentro das atualizações coincidissem com a sequência de caracteres para decidir.

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 **.** corresponde a qualquer caractere único.

  ^

  - O ^ combina o começo da série de entrada.

  $

- O $ combina a extremidade da série de entrada.

\

- O\corresponde ao caractere.

_

- O_corresponde a uma vírgula (`,`), chave esquerda (`{`), chave direita (`}`), o início da cadeia de caracteres de entrada, o fim da cadeia de caracteres de entrada ou um espaço.

•

Peça

Uma peça é um destes símbolos, que vem depois de um átomo:

*

- O * combina 0 ou mais seqüências do átomo.

+

- O + combina 1 ou mais seqüências do átomo.

?

- O?corresponde ao átomo ou à string nula.

- **Filial**

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.

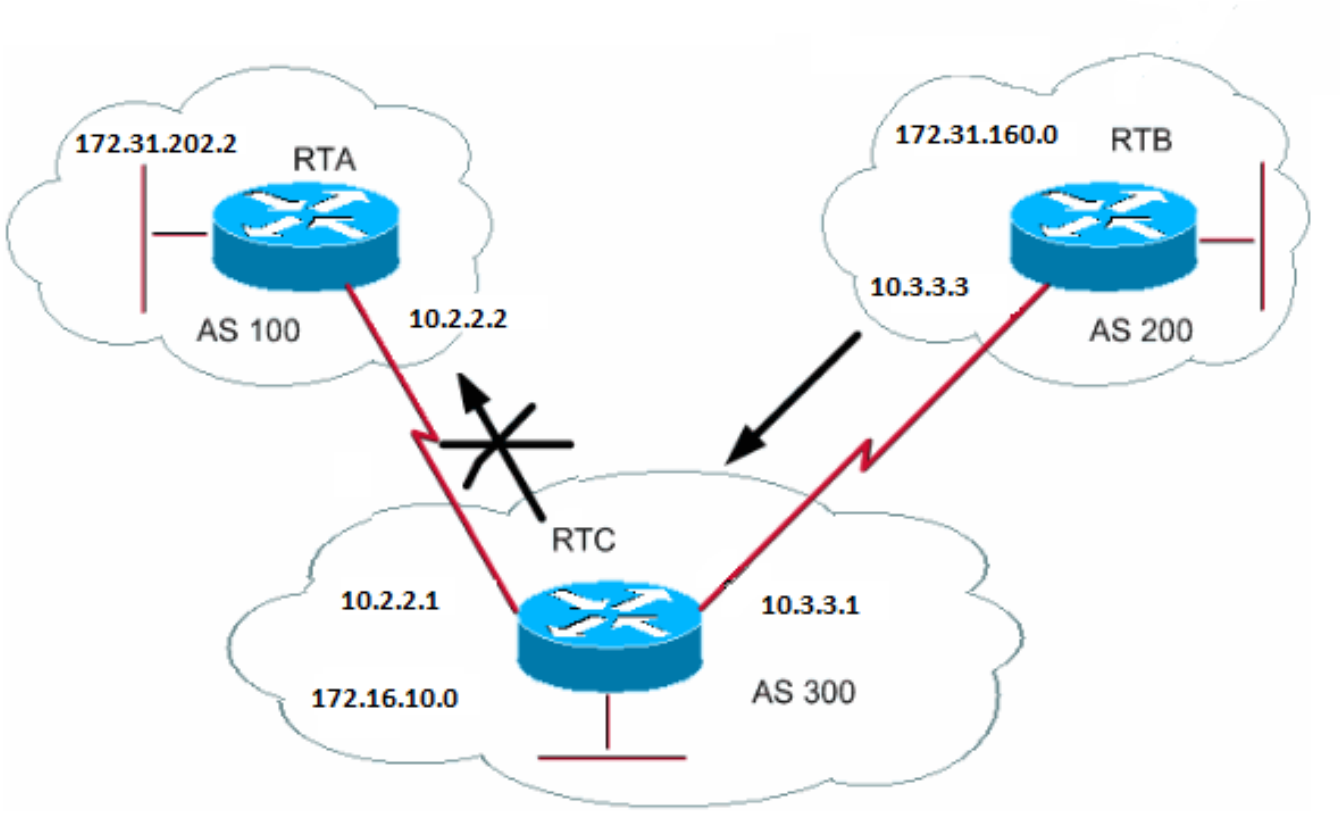
```
^$
```

- Esta expressão indica origens deste AS.

Consulte Usar expressões regulares em BGP para obter configurações de exemplo de filtragem de expressões regulares.

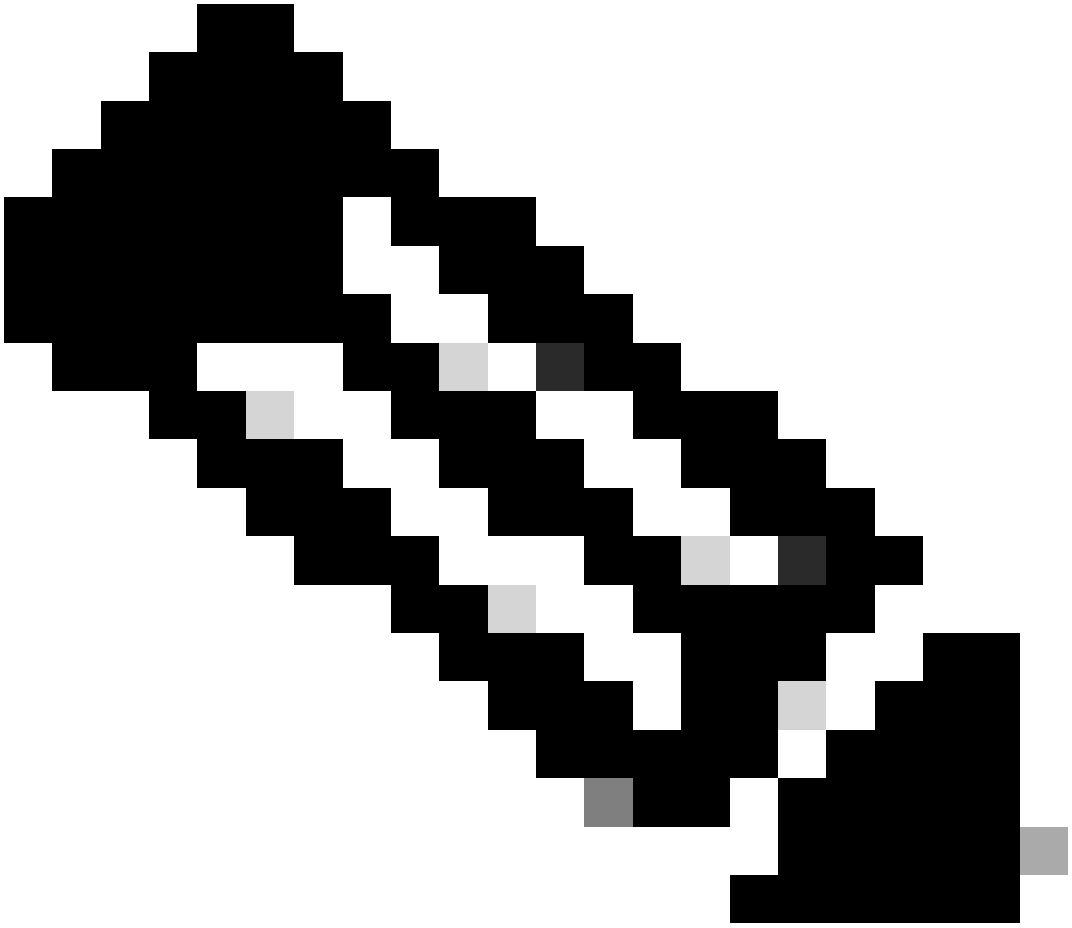
Filtro de comunidade BGP

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.

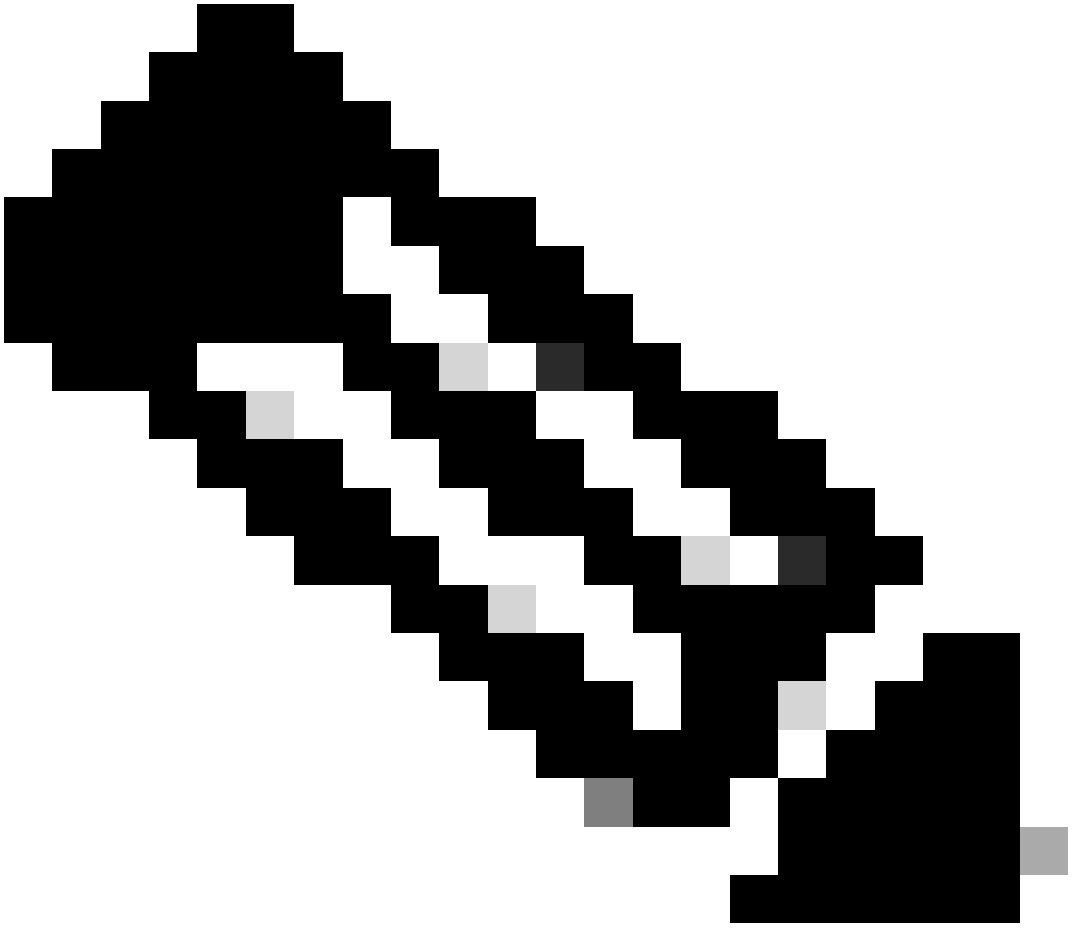


Neste exemplo, você quer que o RTB ajuste o atributo de comunidade às rotas de BGP que o RTB anuncia tais que o RTC não propaga estas rotas aos peers externos. Use o atributo no-exportcommunity.

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

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

access-list 1 permit 0.0.0.0 255.255.255.255
```



Observação: este exemplo usa o comandoroute-map setcommunity para definir a comunidade como no-export.



Observação: o **neighbor send-community** comando é necessário para 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 definiu o atributo de comunidade para**100 200 additive** . Esta ação adiciona o valor 100 200 a qualquer valor comunitário atual antes da transmissão ao RTC.

```
RTB#
router bgp 200
 network 172.31.160.0
 neighbor 10.3.3.1 remote-as 300
```

```
 neighbor 10.3.3.1 send-community
 neighbor 10.3.3.1 route-map setcommunity out

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

access-list 2 permit 0.0.0.0 255.255.255.255
```

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.

<#root>

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.
```

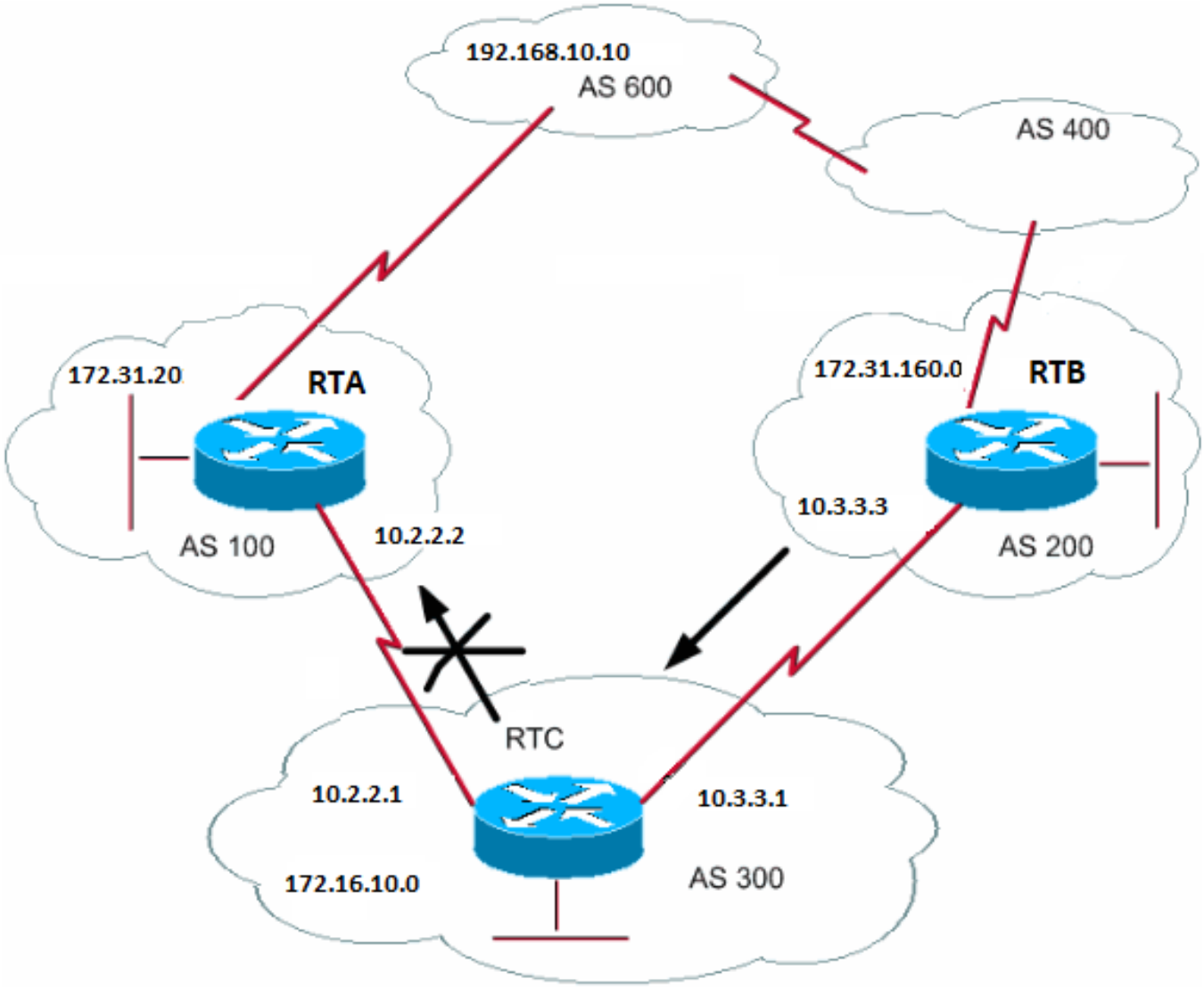
Você pode usar a lista de comunidade a fim filtrar como base ou ajustar determinados parâmetros, como o peso e métrico, em determinadas atualizações com o valor de comunidade. No segundo exemplo nesta seção, o RTB enviou atualizações ao RTC com uma comunidade de 100 200. Se o RTC quer ajustar como base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

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

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

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

route-map check-community permit 30
 match community 3

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

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.

Consulte Configurar e controlar uma rede de provedor upstream com valores de comunidade BGP para obter mais informações.

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:

<#root>

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 172.16.10.0
 neighbor 10.3.3.3 remote-as 200
 neighbor 10.3.3.3 route-map stamp in

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

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

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 172.16.10.0
 neighbor 10.3.3.3 remote-as 200
 neighbor 10.3.3.3 route-map stamp in

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

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

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

Esta indicação ajusta um peso de 20 para as atualizações que são locais ao AS200. A instrução também define um peso de 10 para atualizações que estão atrás do AS400 e descarta 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:

<#root>

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

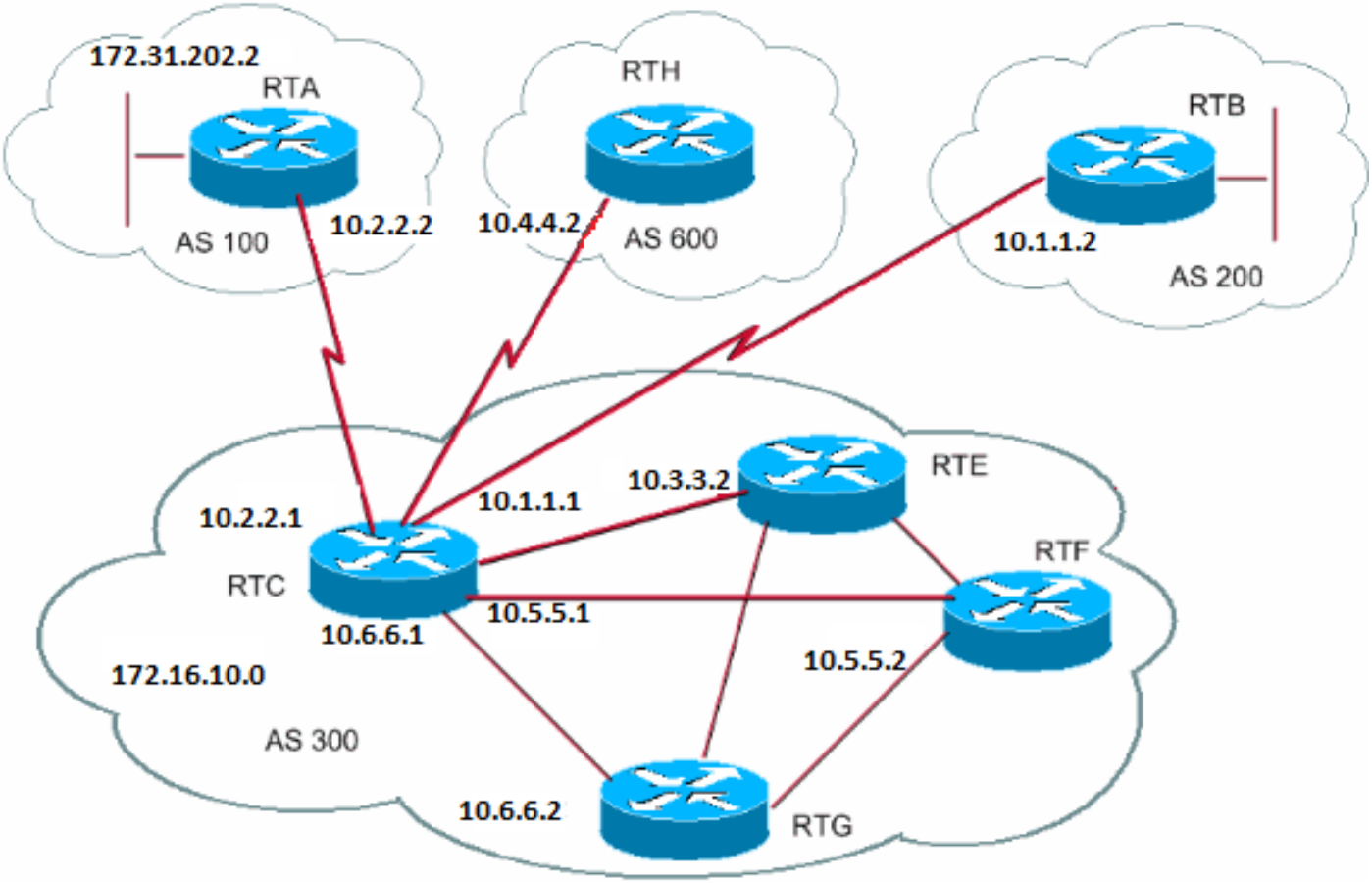
Suponha que, no diagrama na seção Vizinhos BGP e Mapas de Rota, o RTC anuncia sua própria rede 172.16.10.0 para dois ASs diferentes, AS100 e AS200. Quando a informação é propagada ao AS600, os roteadores no AS600 têm a informação de alcançabilidade de rede sobre 172.16.10.0 através de duas rotas diferentes. 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 isso se você anexar números AS à informação de caminho atual que é anunciada ao AS100. Uma prática comum é repetir seu próprio número AS desta maneira:

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

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

Devido a esta configuração, o AS600 recebe atualizações sobre 172.16.10.0 através do AS100 com informações 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 vez disso, você define um nome de grupo de peer e atribui essas políticas ao grupo de peer.

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:

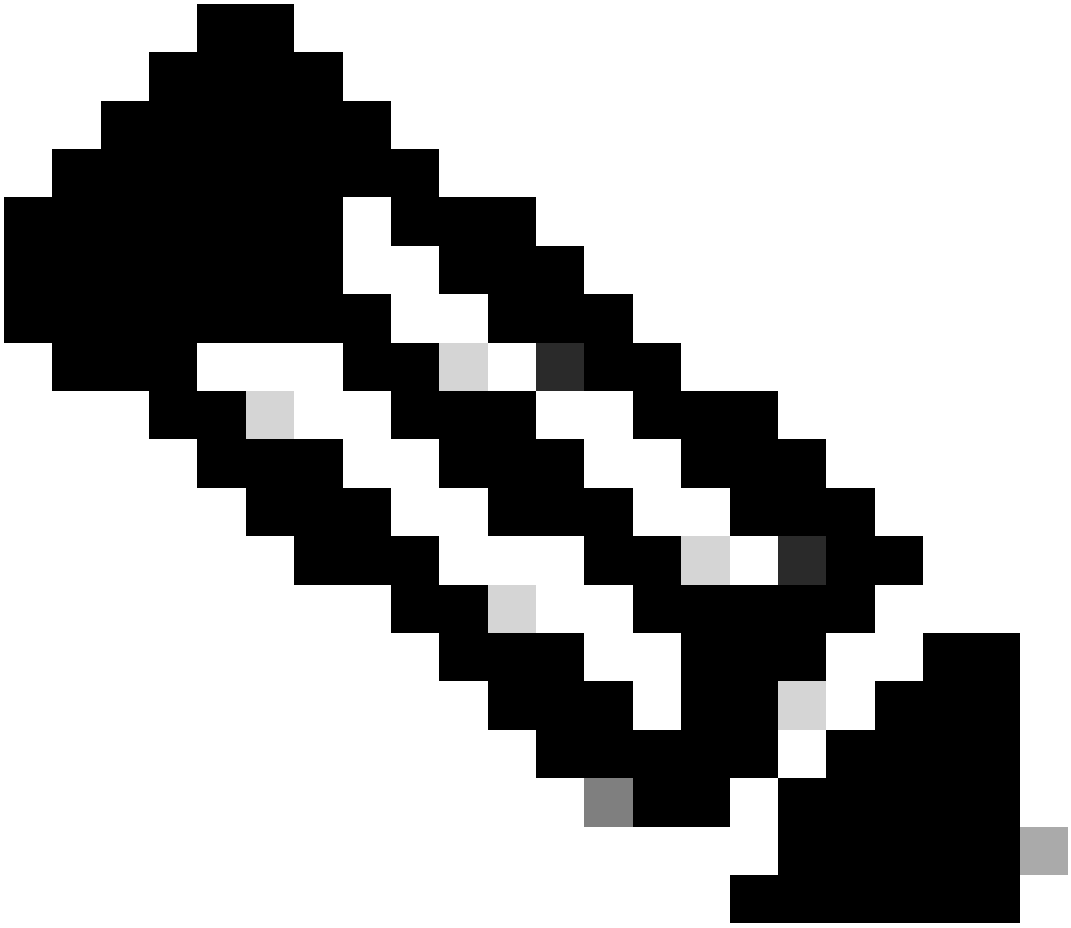
```
<#root>
```

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 10.5.5.2 peer-group internalmap
 neighbor 10.6.6.2 peer-group internalmap
 neighbor 10.3.3.2 peer-group internalmap
 neighbor 10.3.3.2 filter-list 3 in
```

Esta 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.

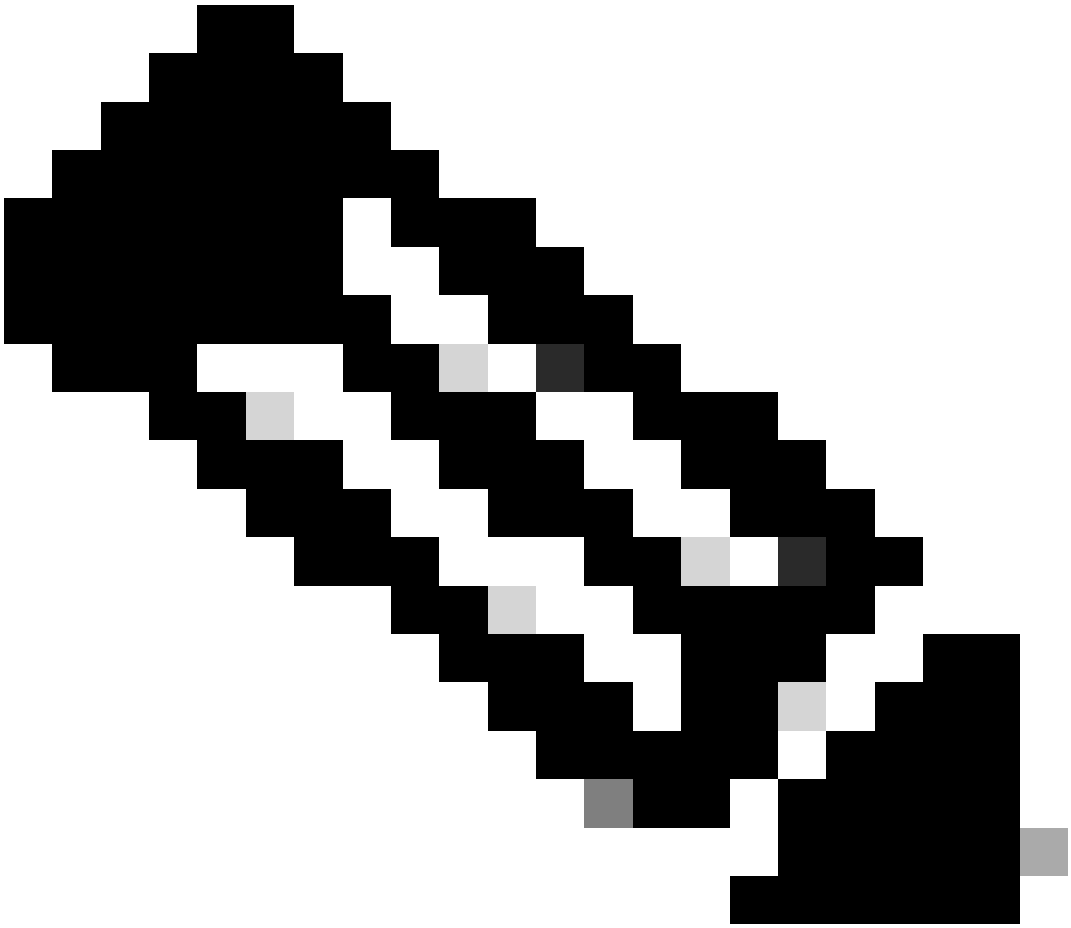


Observação: você só pode substituir opções que afetam atualizações de entrada.

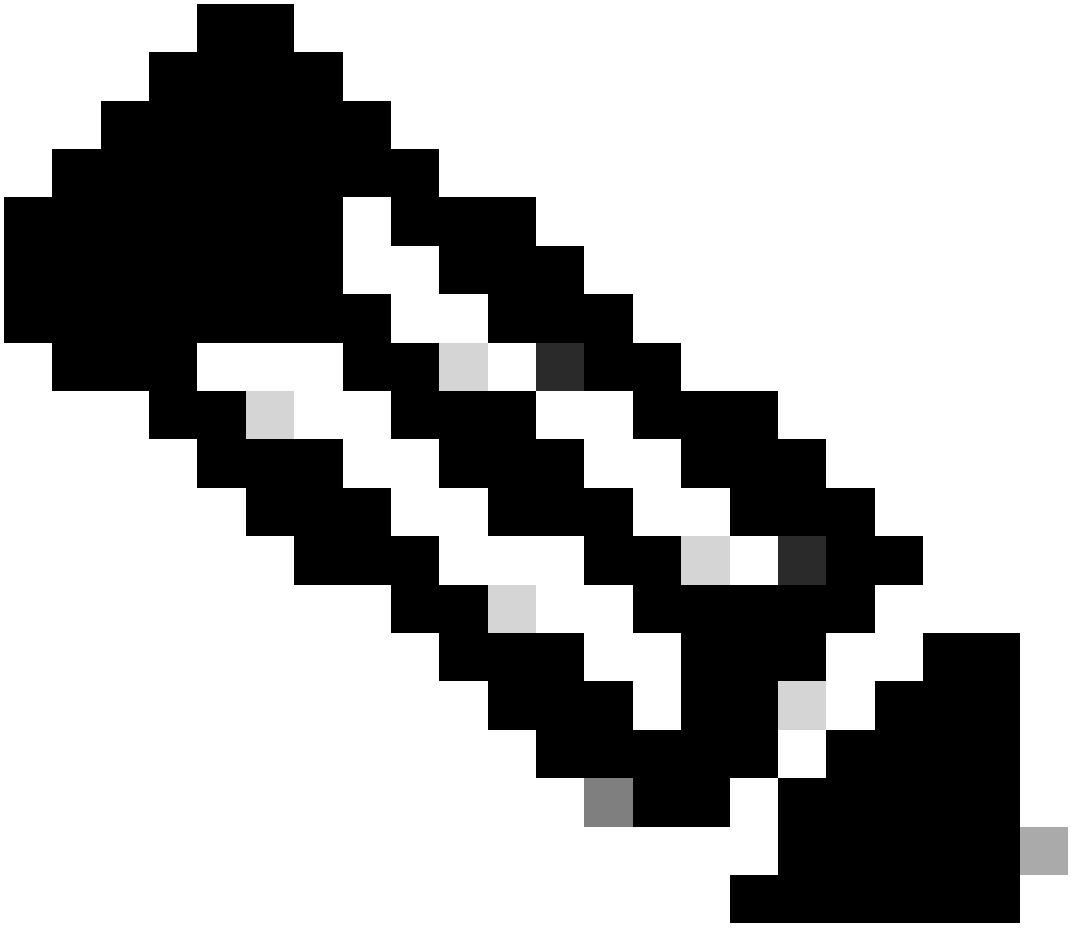
Agora, olhe em como você pode usar a grupo de paridade com vizinhos externos. Com o mesmo diagrama nesta seção, você configura o RTC com um externalmap do grupo de paridade e aplica o grupo de paridade aos vizinhos externos.

```
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 10.2.2.2 remote-as 100
```

```
neighbor 10.2.2.2 peer-group externalmap
neighbor 10.4.4.2 remote-as 600
neighbor 10.4.4.2 peer-group externalmap
neighbor 10.1.1.2 remote-as 200
neighbor 10.1.1.2 peer-group externalmap
neighbor 10.1.1.2 filter-list 3 in
```



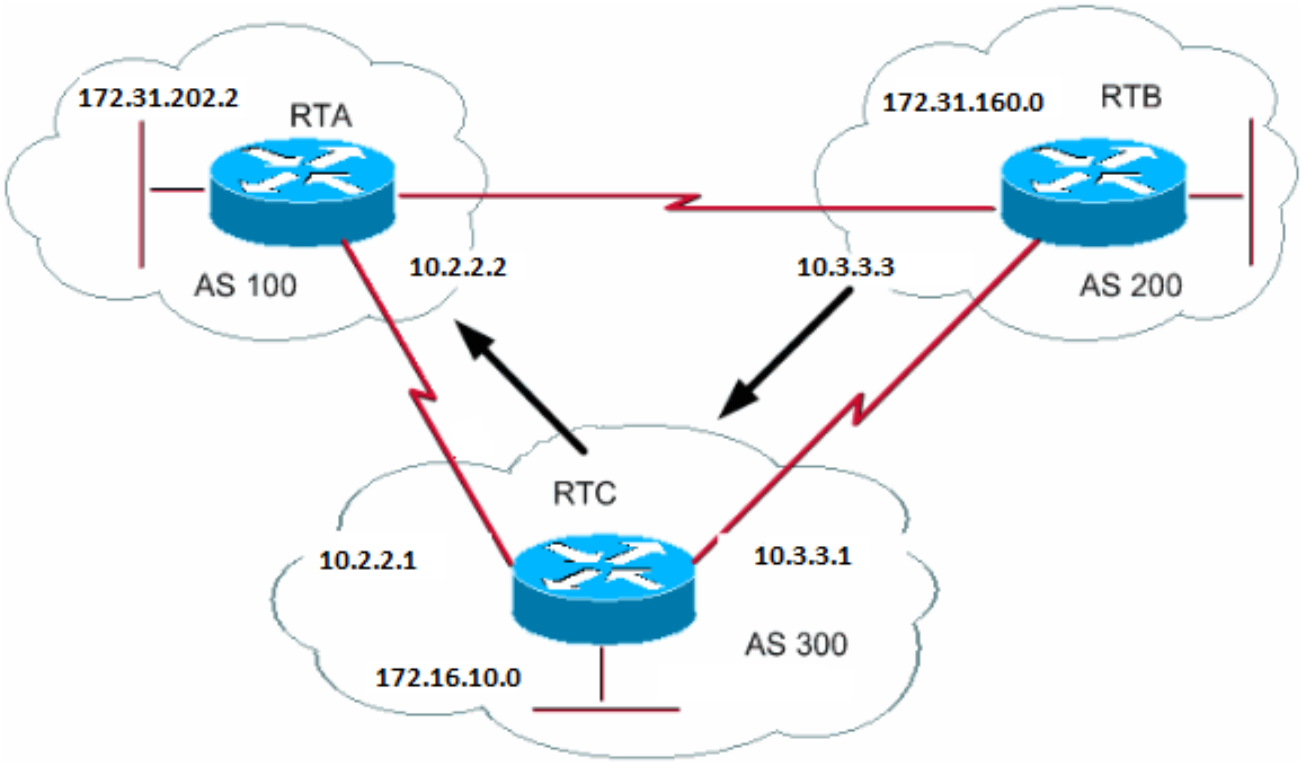
Observação: Nessas configurações, você define as instruções remote-as fora do grupo de peer porque deve definir ASs externos diferentes. Também, você cancela as atualizações de entrada de vizinho 10.1.1.2 com a atribuição da lista de filtro 3. Para obter mais informações sobre dos grupo de paridade, refira grupos de BGP peer.



Observação: no Cisco IOS Software Release 12.0(24)S, a Cisco apresentou o recurso BGP Dynamic Update Peer Groups. 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 melhorias feitas no BGP4 em comparação com o BGP3 foi a inclusão do Classless Interdomain Routing (CIDR). O CIDR ou supernetting são uma maneira nova de olhar endereços IP. Com o CIDR, não há noção de classes, como a classe A, B ou C. Por exemplo, a rede 192.168.213.0 já foi uma rede de classe C ilegal. Agora, a rede é um super-rede legal, 192.168.213.0/16. O 16 representa o número de bits na máscara de sub-rede, quando você conta da extrema esquerda do endereço IP. Esta representação é similar a 192.168.213.0 255.255.0.0.

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 172.31.160.0. Você configura o RTC para propagar uma super-rede dessa rota 192.168.160.0 ao RTA:

```
RTB#
router bgp 200
 neighbor 10.3.3.1 remote-as 300
 network 172.31.160.0

#RTC
router bgp 300
 neighbor 10.3.3.3 remote-as 200
 neighbor 10.2.2.2 remote-as 100
 network 172.16.10.0
 aggregate-address 192.168.160.0 255.0.0.0
```

O RTC propaga o endereço agregado 192.168.160.0 ao RTA.

Comandos aggregate

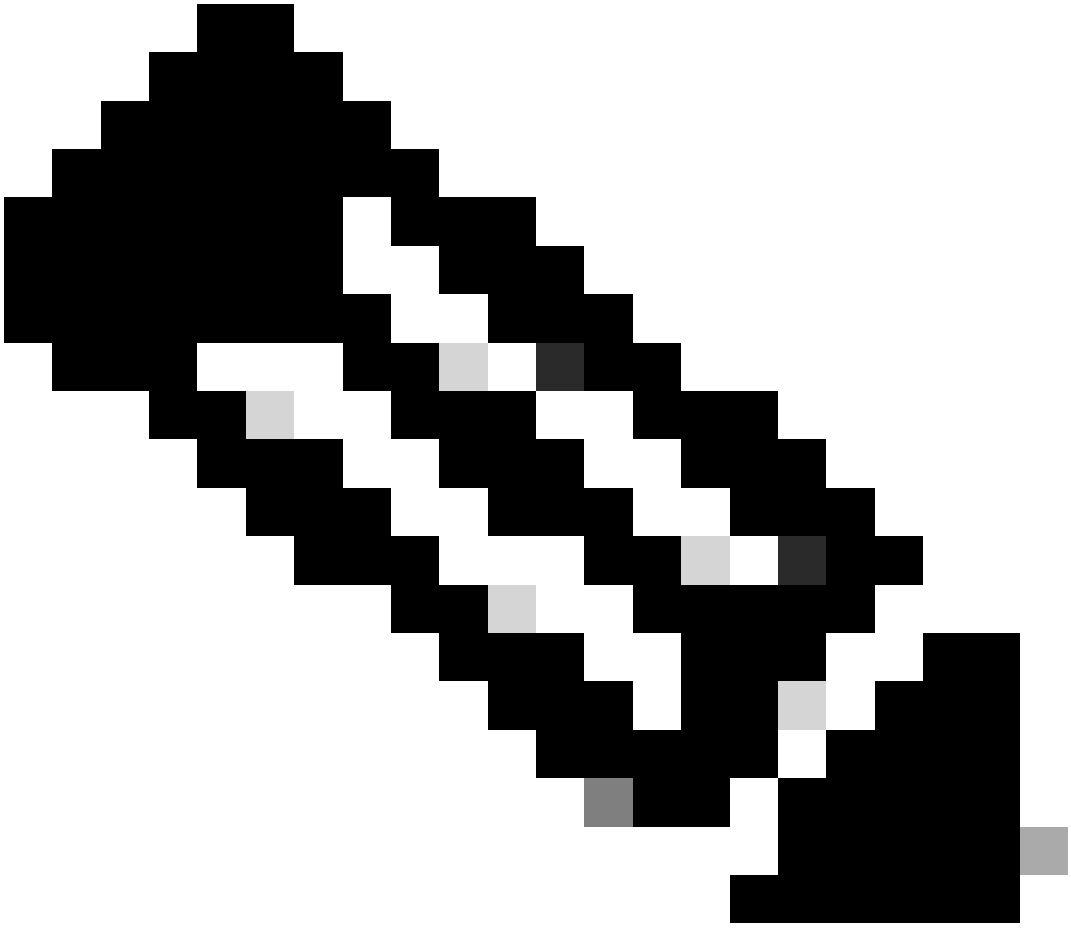
Há um amplo intervalo dos comandos aggregate. Você deve compreender como cada um trabalha a fim ter o comportamento da agregação que você deseja.

O primeiro comando é aquele do exemplo na seção CIDR e nos endereços agregados:

<#root>

aggregate-address address-mask

Este comando anuncia rota do prefixo e todas as rotas mais específicas. O comando **aggregate-address 192.168.160.0** propaga uma rede adicional 192.168.160.0, mas não impede a propagação de 172.31.160.0 para o RTA. O resultado é a propagação de redes 192.168.160.0 e de 172.31.160.0 ao RTA, que é a propaganda do prefixo e da rota mais específica.



Observação: você não pode agregar um endereço se não tiver uma rota mais específica desse endereço na tabela de roteamento BGP.

Por exemplo, o RTB não pode gerar um agregado para 192.168.160.0 se o RTB não tem uma entrada mais específica de 192.168.160.0 na tabela 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 172.31.160.0

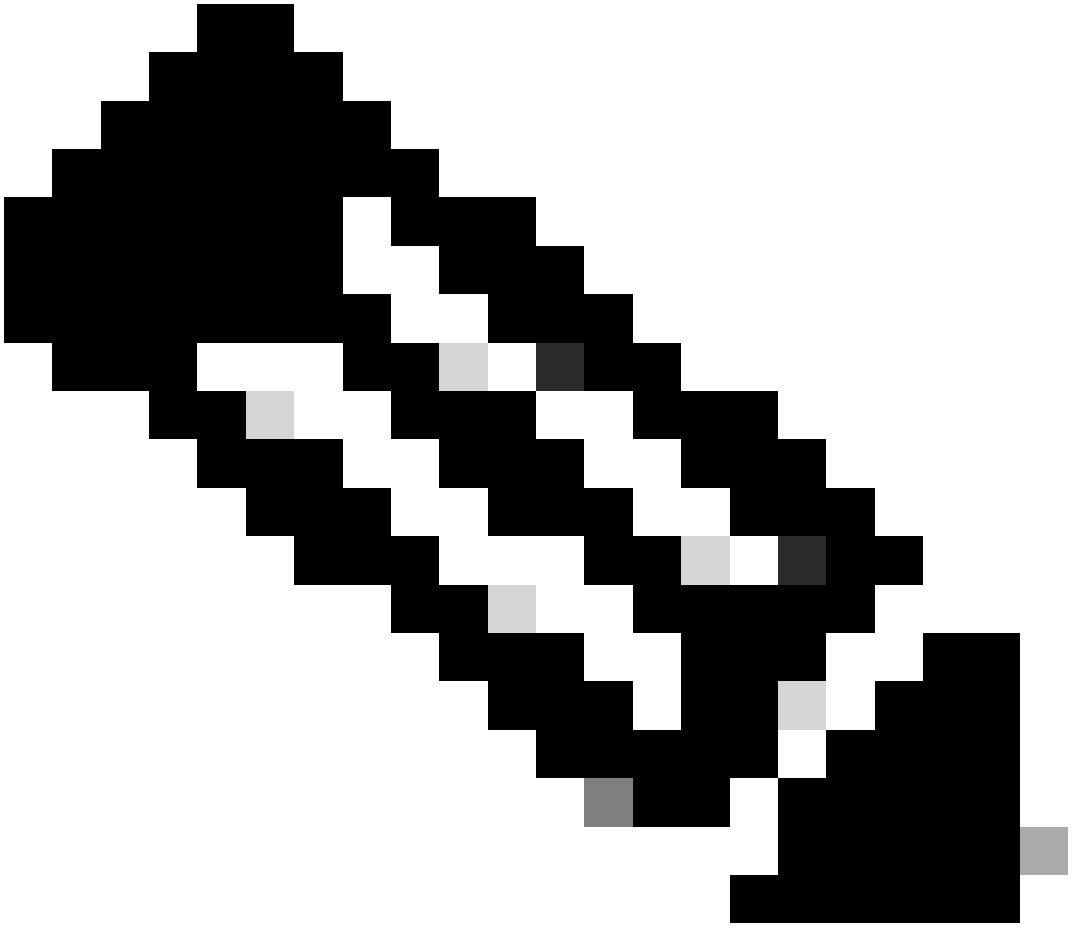
Se você quiser que o RTC propague somente a rede 192.168.160.0 e nãoa rota mais específica, emita este comando:

<#root>

aggregate-address <address> <mask> summary-only

Este comando anuncia somente o prefixo. O comando suprime todas as rotas mais específicas.

O comando **aggregate 192.168.160.0 255.0.0.0 summary-only** propaga a rede 192.168.160.0 e suprime a rota mais específica 172.31.160.0.



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

<#root>

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.

<#root>

aggregate 192.168.0.0 255.0.0.0 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.

<#root>

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 192.168.160.0, suprime a rota 192.168.160.20 mais específica, e permitir a propagação de 172.31.160.0. Use este mapa de rotas:

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

access-list 1 permit 192.168.160.20 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 10.3.3.3 remote-as 200
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.2.2.2 remote-as 100
 network 172.16.10.0
 aggregate-address 192.168.160.0 255.0.0.0 suppress-map CHECK
```

Está aqui uma outra variação:

<#root>

aggregate-address <address> <mask> attribute-map <map-name>

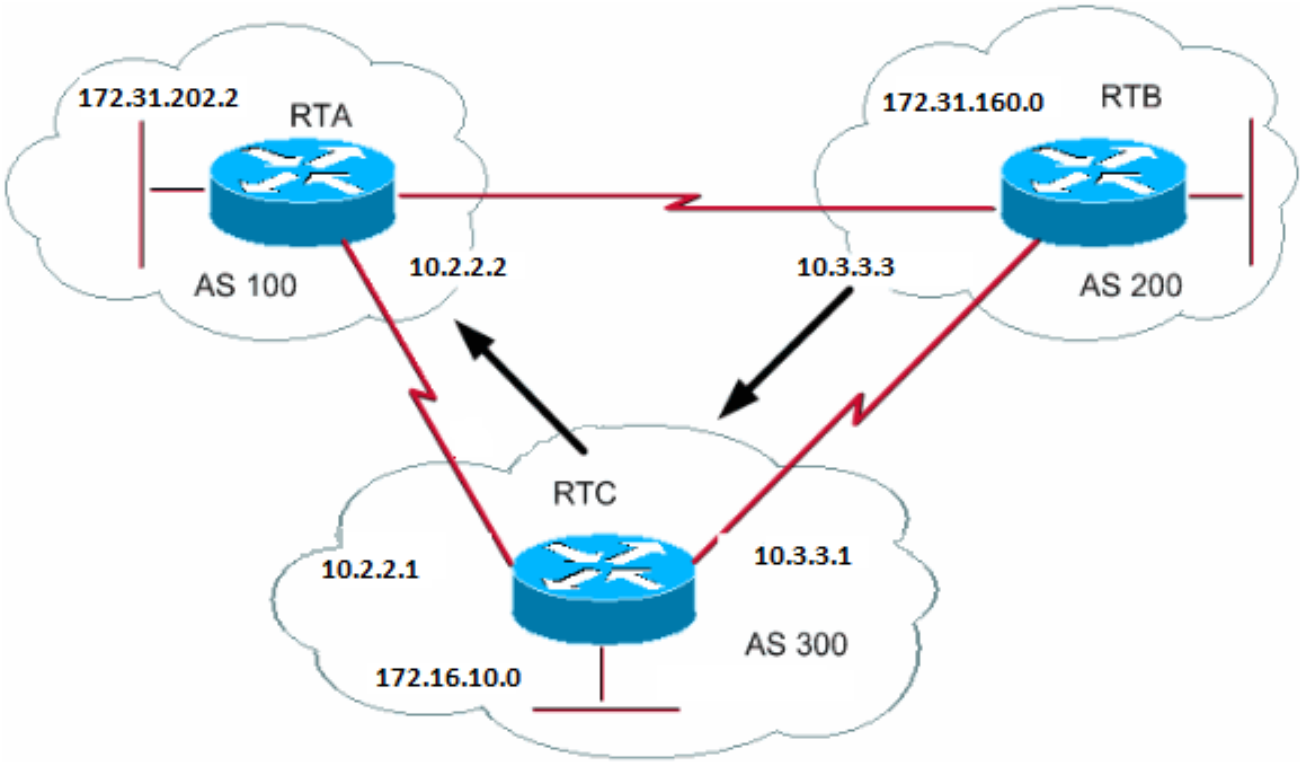
Este comando permite que você ajuste os atributos, tais como métrico, na altura da emissão dos agregados. A fim de ajustar a origem dos agregados ao IGP, aplique este mapa de rotas ao comando aggregate attribute-map:

```
route-map SETMETRIC
 set origin igp
```

```
aggregate-address 192.168.160.0 255.0.0.0 attribute-map SETORIGIN
```

Para obter mais informações, consulte Entender a agregação de rotas no BGP.

CIDR Exemplo 1



Solicitação: permita que o RTB anuncie o prefixo 192.168.160.0 e suprima todas as rotas mais específicas. O problema com essa solicitação é que a rede 172.31.160.0 é local para o AS200, o que significa que o AS200 é o originador de 172.31.160.0. Você não pode mandar o RTB gerar um prefixo para 192.168.160.0 sem a geração de uma entrada para 172.31.160.0, mesmo se você usa o comando aggregate summary-only. O RTB gera ambas as redes porque o RTB é o autor de 172.31.160.0. 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 (?).

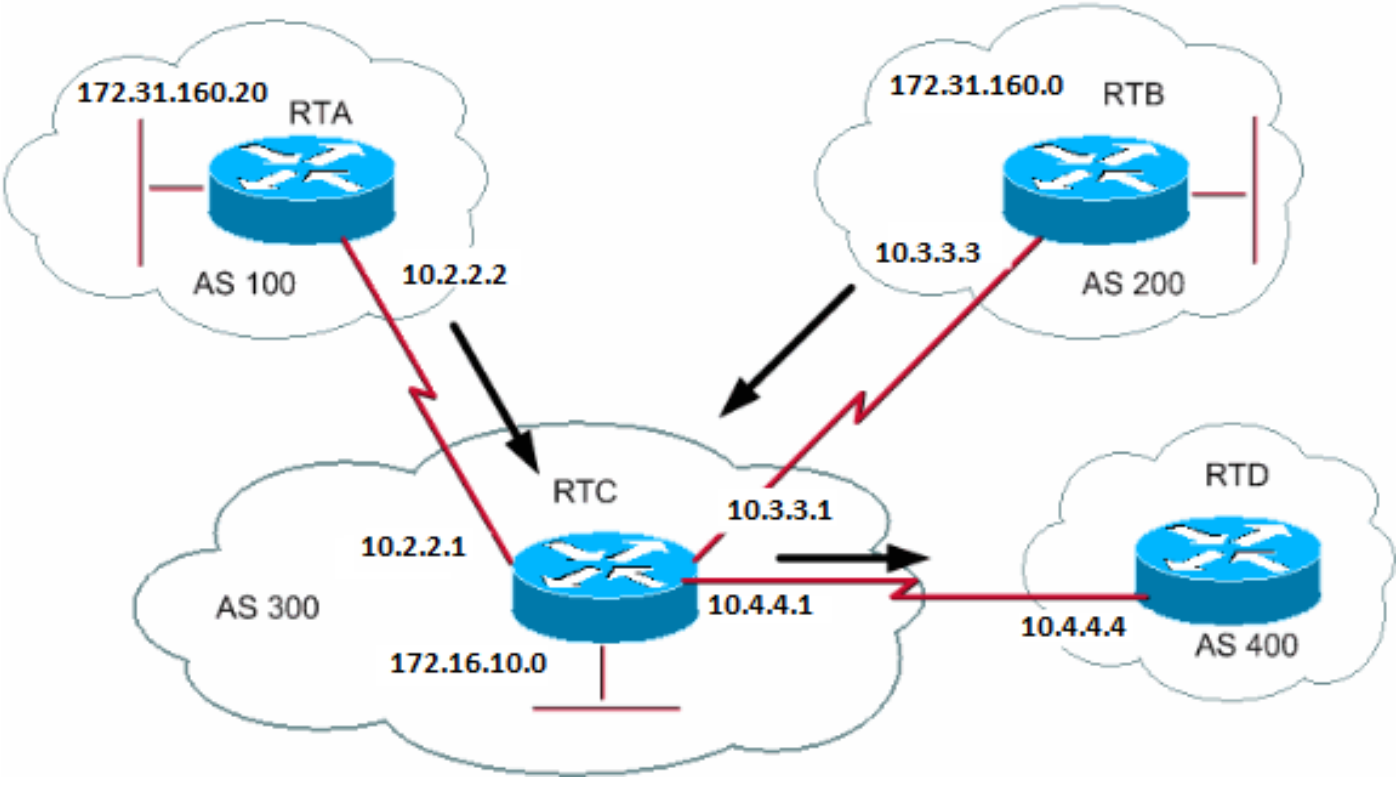
```
RTB#
router bgp 200
 neighbor 10.3.3.1 remote-as 300
 redistribute static

!--- This generates an update for 192.168.160.0 !--- with the origin path as "incomplete".

ip route 192.168.160.0 255.0.0.0 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 192.168.160.0 mask 255.0.0.0

!--- This entry marks the update with origin IGP.

 neighbor 10.3.3.1 remote-as 300
 redistribute static

ip route 192.168.160.0 255.0.0.0 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 192.168.160.20 do RTA e atualizações sobre 172.31.160.0 do RTB. Suponha que o RTC quer a rede agregada 192.168.160.0/8 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 172.31.160.0
 neighbor 10.3.3.1 remote-as 300

RTA#
router bgp 100
 network 192.168.160.20
 neighbor 10.2.2.1 remote-as 300
```

Caso 1:

O RTC não tem uma indicação de as-set. O RTC envia uma atualização 192.168.160.0/8 ao RTD com informação de caminho (300), como se a rota originou do AS300.

```
RTC#
router bgp 300
 neighbor 10.3.3.3 remote-as 200
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.4.4.4 remote-as 400
 aggregate 192.168.160.0 255.0.0.0 summary-only

!--- This command causes RTC to send RTD updates about 192.168.160.0/8
!--- with no indication that 192.168.160.0 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 10.3.3.3 remote-as 200
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.4.4.4 remote-as 400
 aggregate 192.168.160.0 255.0.0.0 summary-only
 aggregate 192.168.160.0 255.0.0.0 as-set

!--- This command causes RTC to send RTD updates about 192.168.160.0/8
!--- with an indication that 192.168.160.0 belongs to a set {100 200}.
```

Os próximos dois assuntos, Confederação de BGP e Refletores de Rota, são para provedores de serviços de Internet (ISPs) que querem maior controle da explosão de peering de iBGP dentro de seus ASs.

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:

```
<#root>
```

bgp confederation identifier <autonomous-system>

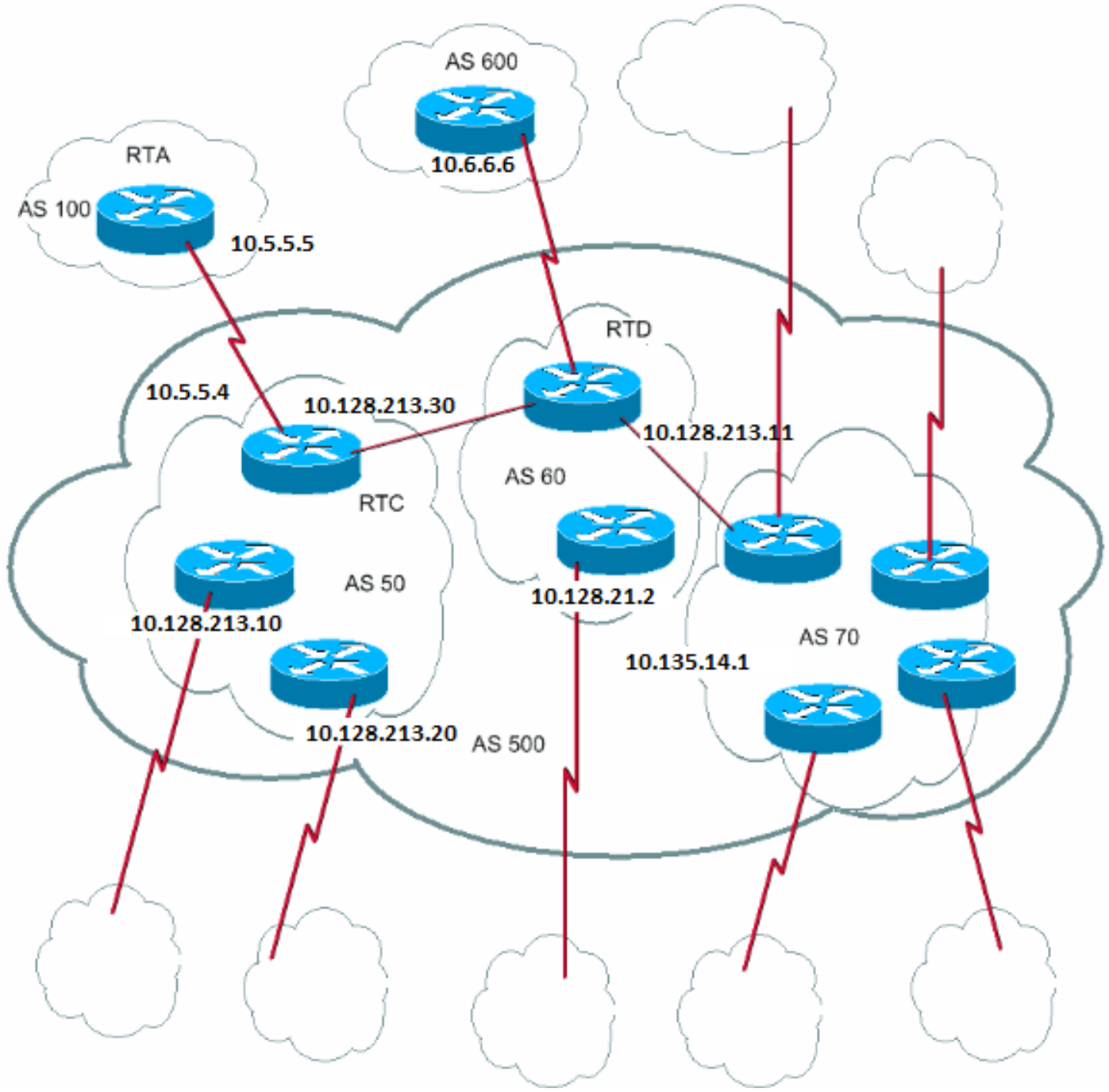
O identificador de confederação é o número AS do grupo de confederação.

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

```
<#root>
```

bgp confederation peers <autonomous-system> <autonomous-system>

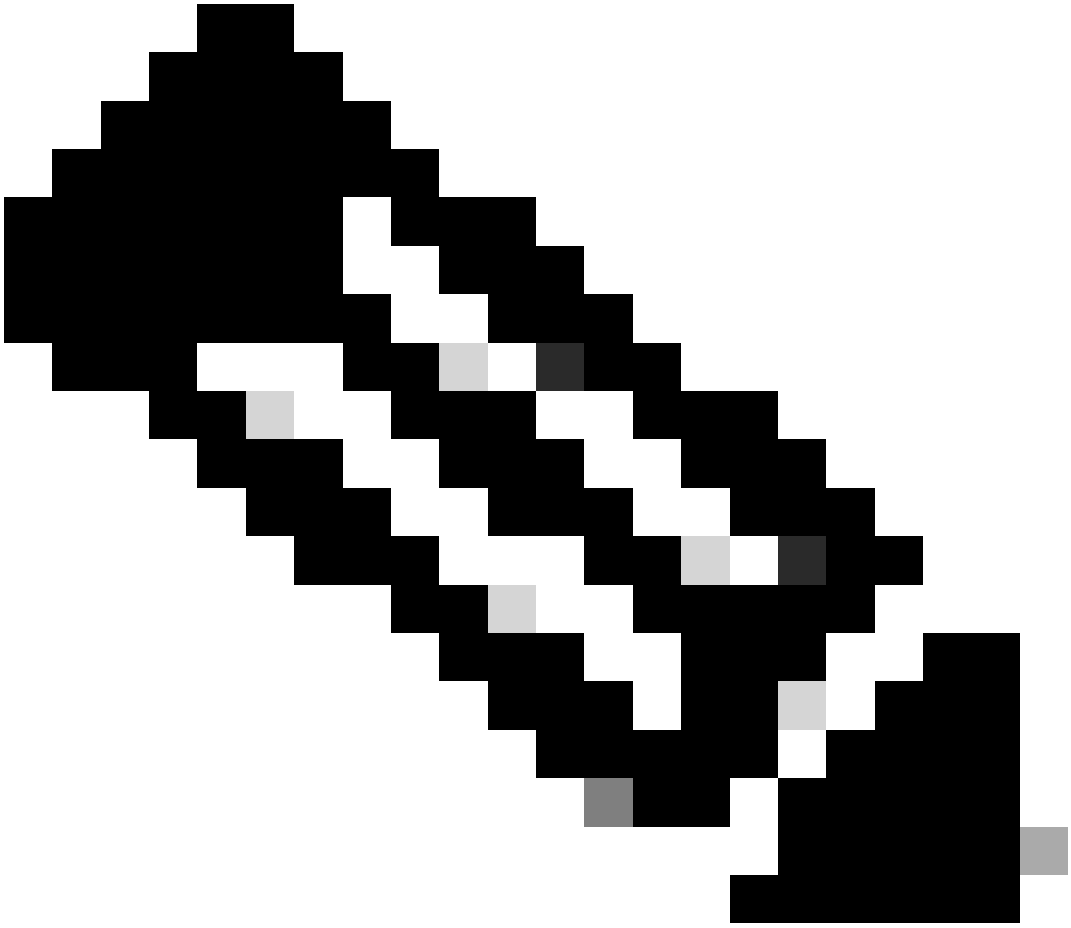
Está aqui um exemplo de confederação:



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 confederação, você pode dividir o AS500 em vários ASs: 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:



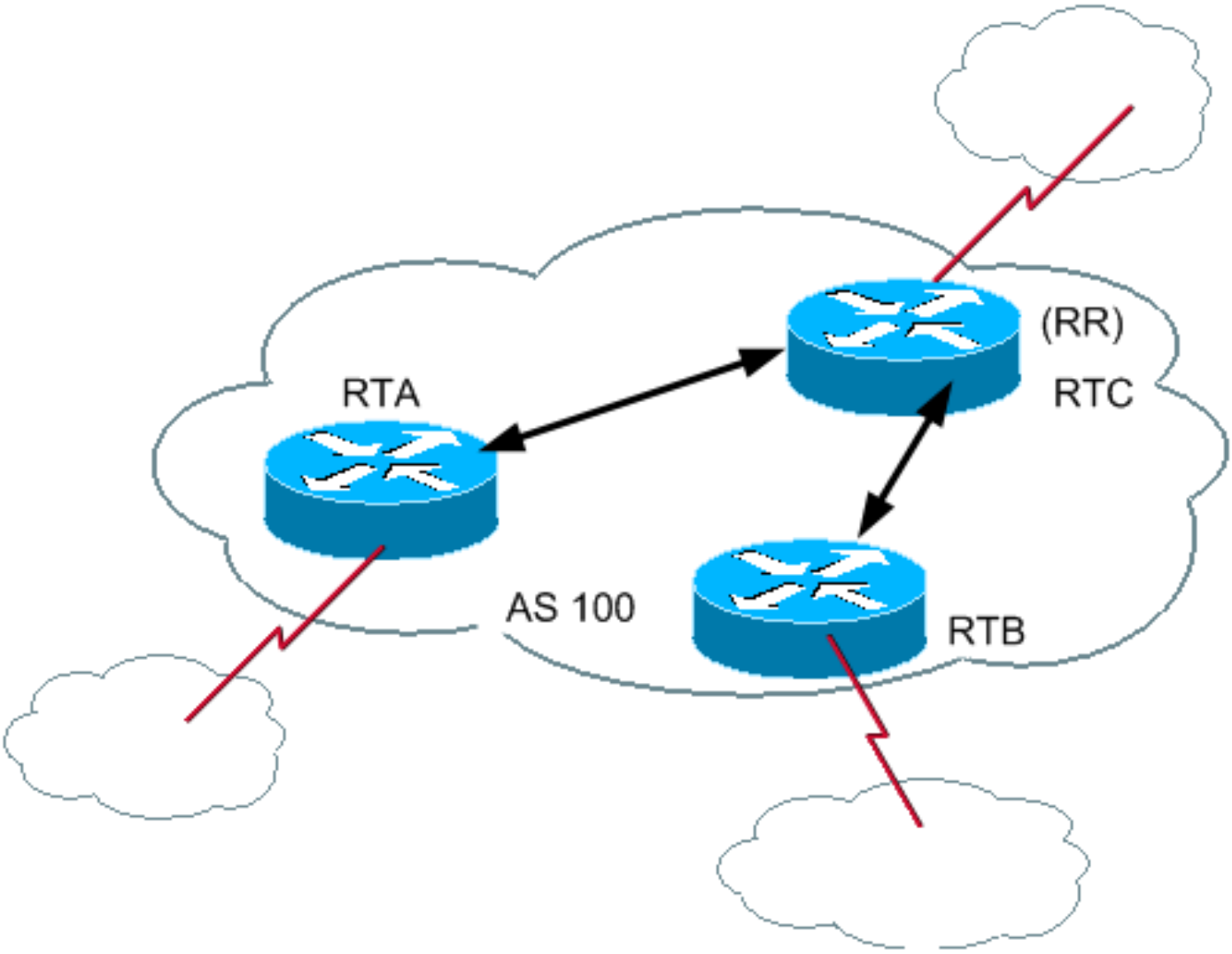
Observação: o RTA não tem conhecimento do AS50, AS60 ou AS70. O RTA tem somente o conhecimento do AS500.

```
RTC#
router bgp 50
 bgp confederation identifier 500
 bgp confederation peers 60 70
 neighbor 10.128.213.10 remote-as 50 (IBGP connection within AS50)
 neighbor 10.128.213.20 remote-as 50 (IBGP connection within AS50)
 neighbor 10.128.213.11 remote-as 60 (BGP connection with confederation peer 60)
 neighbor 10.128.213.14 remote-as 70 (BGP connection with confederation peer 70)
 neighbor 10.5.5.5 remote-as 100 (EBGP connection to external AS100)

RTD#
router bgp 60
```

```
 bgp confederation identifier 500
 bgp confederation peers 50 70
 neighbor 10.128.210.2 remote-as 60 (IBGP connection within AS60)
 neighbor 10.128.213.30 remote-as 50(BGP connection with confederation peer 50)
 neighbor 10.128.213.14 remote-as 70 (BGP connection with confederation peer 70)
 neighbor 10.6.6.16 remote-as 600 (EBGP connection to external AS600)

RTA#
router bgp 100
 neighbor 10.5.5.4 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). Como a seção iBGP demonstra, um alto-falante BGP não anuncia uma rota que o alto-falante BGP aprendeu através de outro alto-falante iBGP para 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.



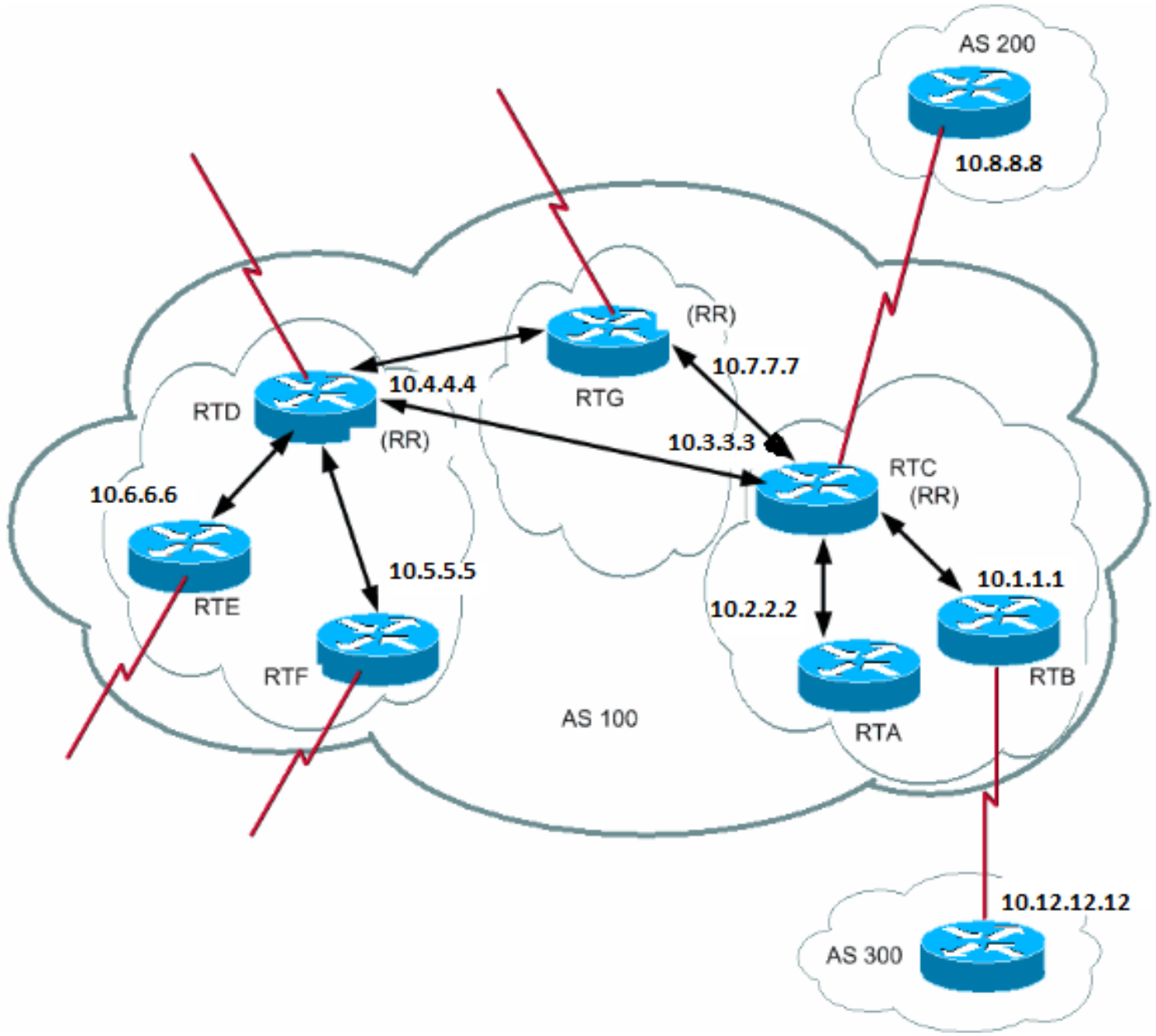
Em casos normais, mantenha uma malha completa do iBGP entre o RTA, o RTB, e o RTC dentro do AS100. Se você utiliza o conceito RR, o RTC pode ser elegido como um RR. Desta maneira, o RTC tem um ibgp peering parcial com RTA e RTB. Peering entre o RTA e o RTB não é necessário porque o RTC é um RR para as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

<#root>

neighbor <ip address> route-reflector-client

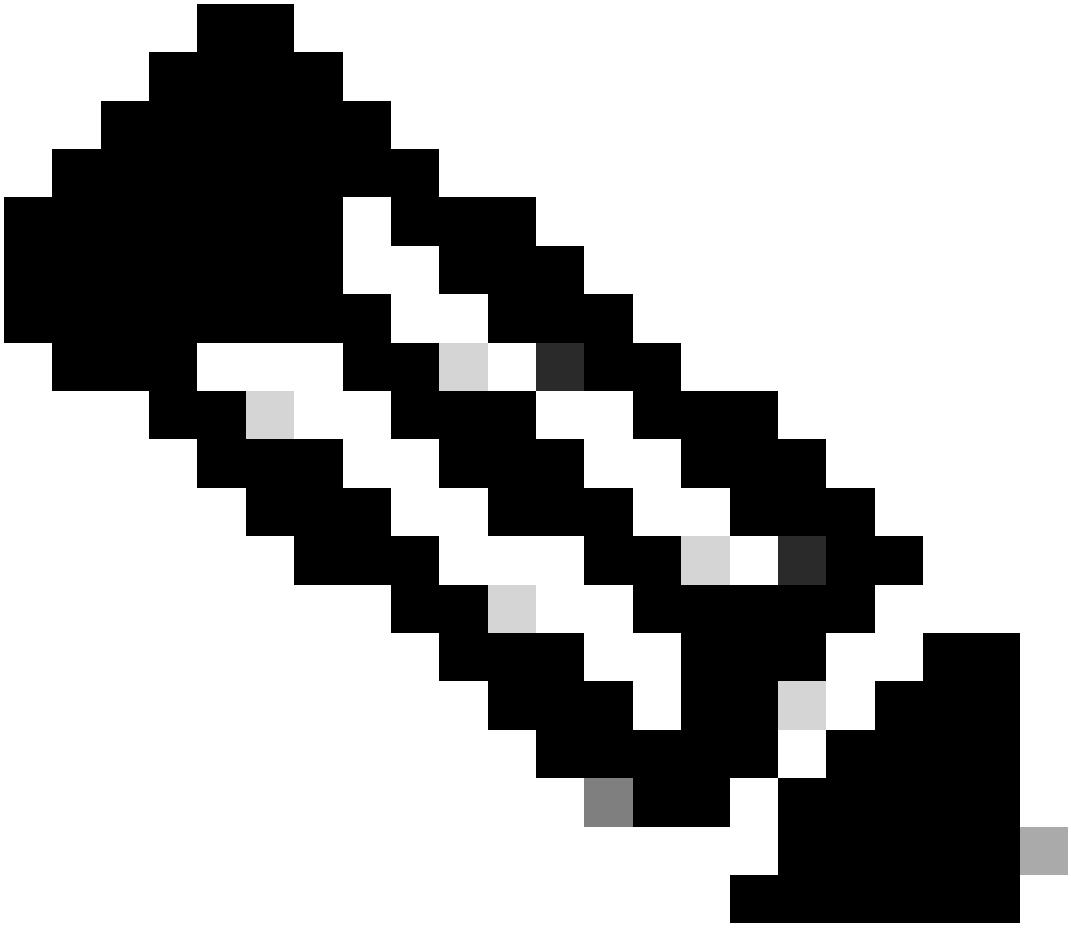
O roteador com este comando é o RR, e os vizinhos em que os pontos do comando são os clientes desse RR. No exemplo, a configuração de RTC tem o comando neighbor route-reflector-client que aponta os endereços IP RTA e RTB. A combinação do RR e dos clientes é um “conjunto”. Neste exemplo, em formulário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

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



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 fazer peer com alto-falantes iBGP fora do cluster de clientes.

No diagrama anterior, o RTA, o RTB e o RTC formam um único cluster. 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.



Observação: o RTD, o RTC e o RTG estão totalmente integrados, mas os roteadores dentro de um cluster não estão.

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:

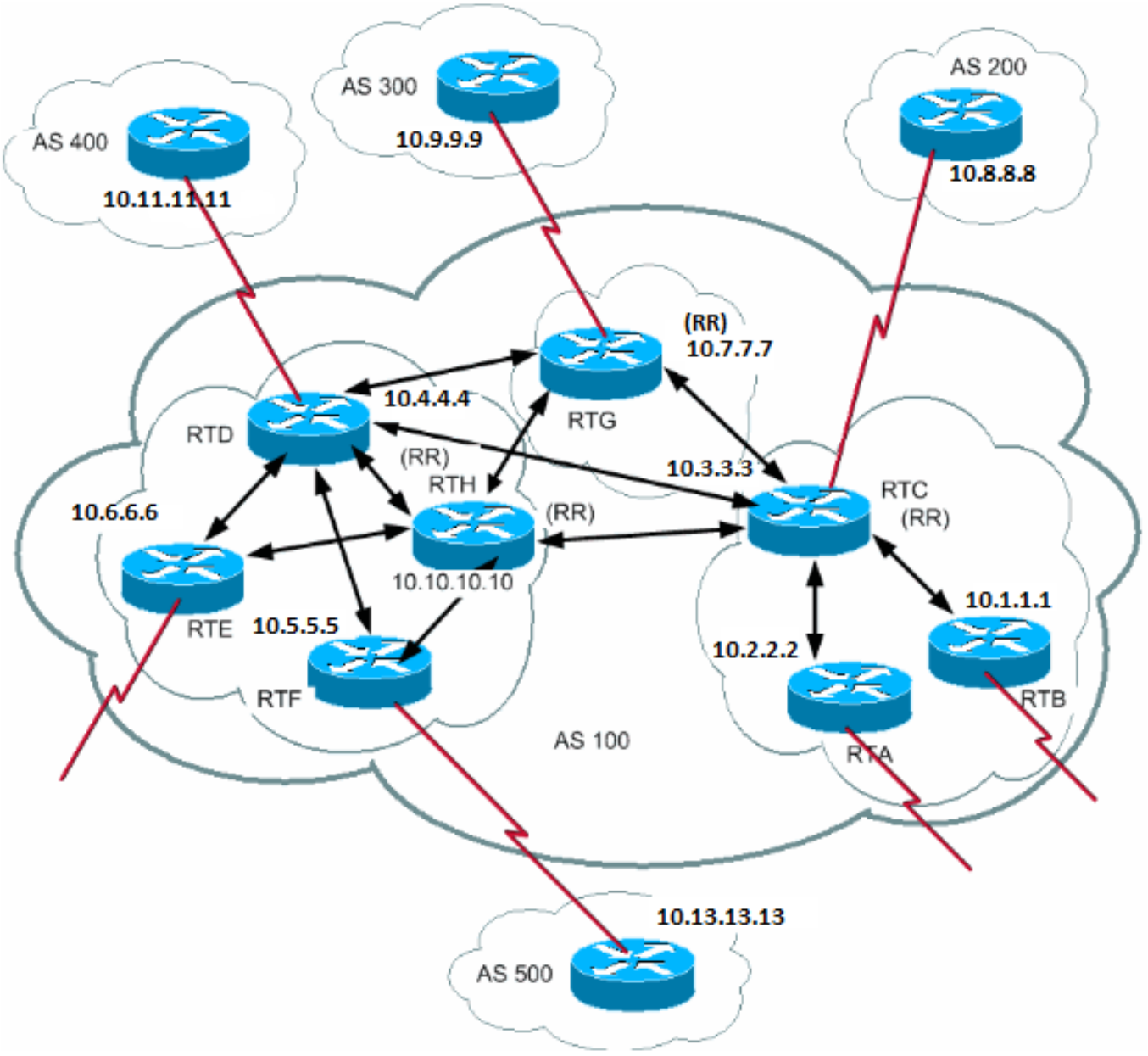
```
RTC#
router bgp 100
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.2.2.2 route-reflector-client
 neighbor 10.1.1.1 remote-as 100
 neighbor 10.1.1.1 route-reflector-client
 neighbor 10.7.7.7 remote-as 100
 neighbor 10.4.4.4 remote-as 100
 neighbor 10.8.8.8 remote-as 200

RTB#
router bgp 100
 neighbor 10.3.3.3 remote-as 100
 neighbor 10.12.12.12 remote-as 300

RTD#
router bgp 100
 neighbor 10.6.6.16 remote-as 100
 neighbor 10.6.6.16 route-reflector-client
 neighbor 10.5.5.5 remote-as 100
 neighbor 10.5.5.5 route-reflector-client
 neighbor 10.7.7.7 remote-as 100
 neighbor 10.3.3.3 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 Vários RRs dentro de um Cluster cobre a lista de clusters.

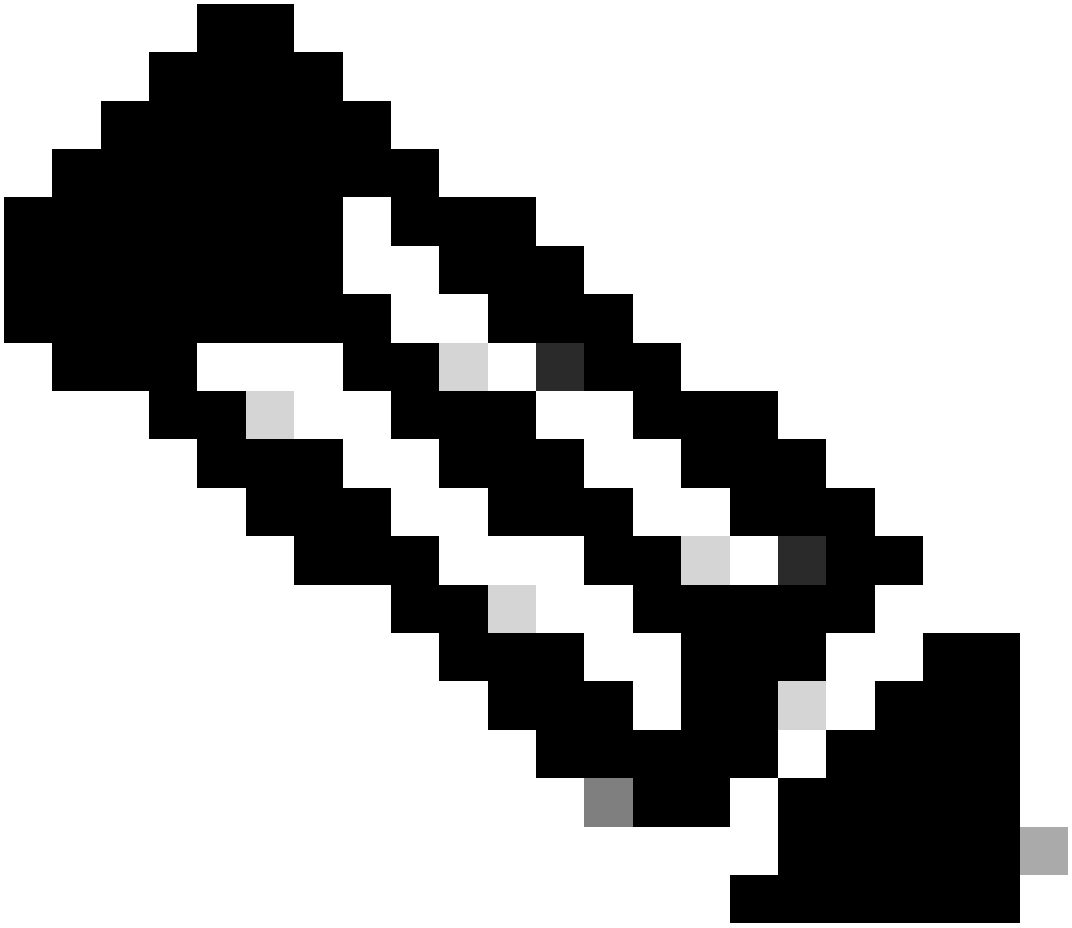
RR múltiplos dentro de um conjunto



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.

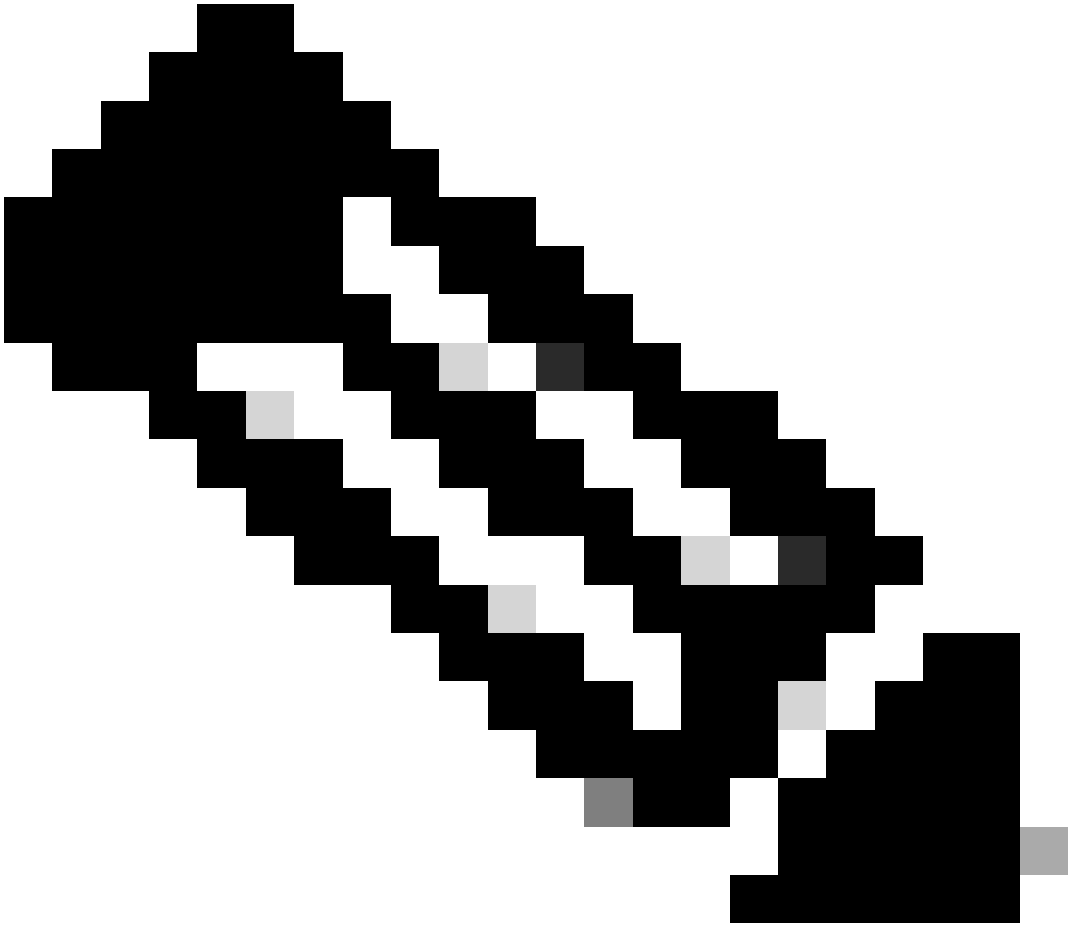


Observação: há redundância porque o RTH tem um peering totalmente combinado com todos os RRs. Se o RTD vai para baixo, RTH toma o lugar do RTD.

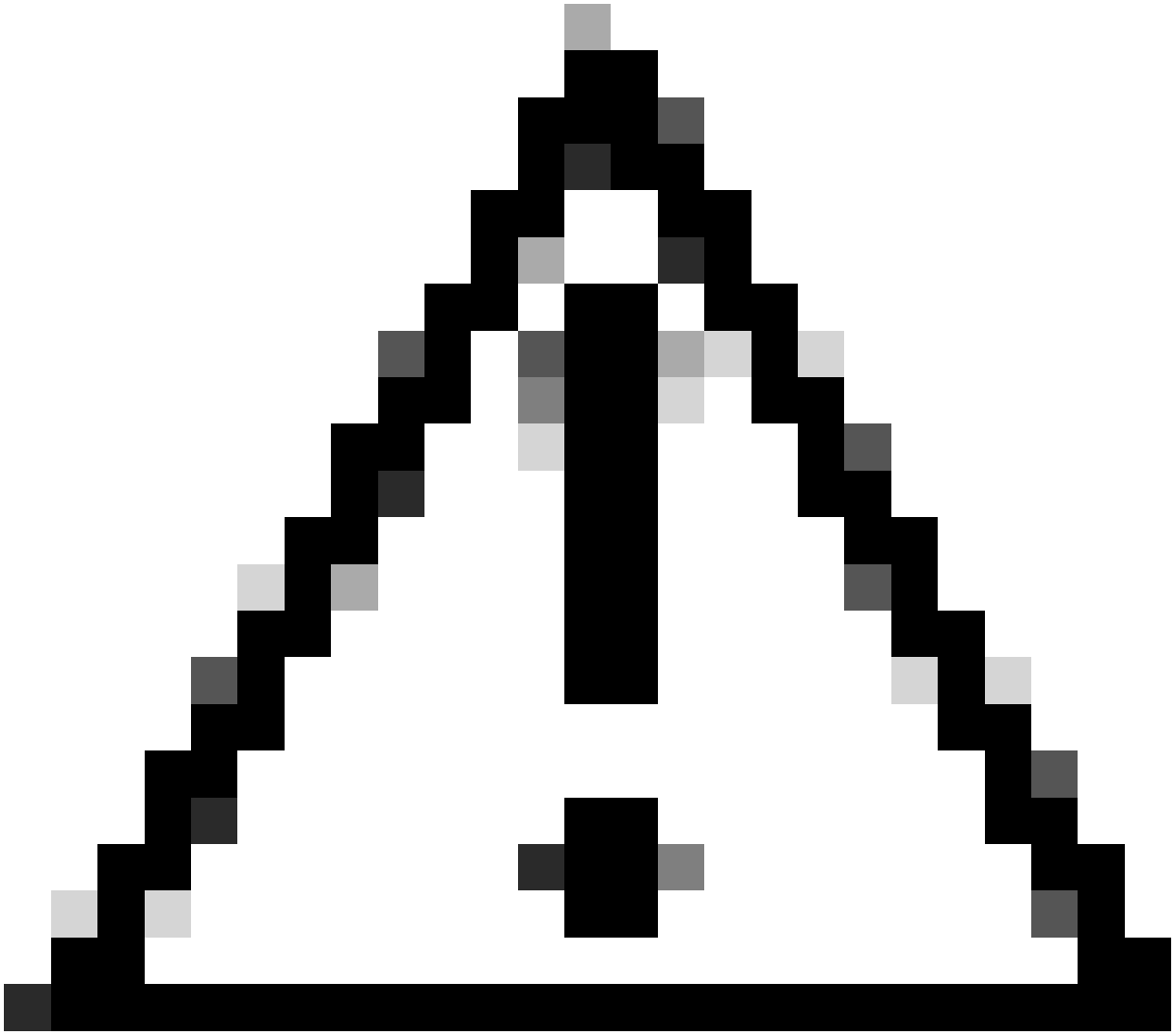
Está aqui a configuração de RTH, de RTD, de RTF, e de RTC:

```
RTH#
router bgp 100
 neighbor 10.4.4.4 remote-as 100
 neighbor 10.5.5.5 remote-as 100
 neighbor 10.5.5.5 route-reflector-client
 neighbor 10.6.6.16 remote-as 100
 neighbor 10.6.6.16 route-reflector-client
 neighbor 10.7.7.7 remote-as 100
```

```
 neighbor 10.3.3.3 remote-as 100
 neighbor 10.9.9.9 remote-as 300
 bgp cluster-id 10

RTD#
router bgp 100
 neighbor 10.10.10.10 remote-as 100
 neighbor 10.5.5.5 remote-as 100
 neighbor 10.5.5.5 route-reflector-client
 neighbor 10.6.6.16 remote-as 100
 neighbor 10.6.6.16 route-reflector-client
 neighbor 10.7.7.7 remote-as 100
 neighbor 10.3.3.3 remote-as 100
 neighbor 10.11.11.11 remote-as 400
 bgp cluster-id 10

RTF#
router bgp 100
 neighbor 10.10.10.10 remote-as 100
 neighbor 10.4.4.4 remote-as 100
 neighbor 10.13.13.13 remote-as 500

RTC#
router bgp 100
 neighbor 10.1.1.1 remote-as 100
 neighbor 10.1.1.1 route-reflector-client
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.2.2.2 route-reflector-client
 neighbor 10.4.4.4 remote-as 100
 neighbor 10.7.7.7 remote-as 100
 neighbor 10.10.10.10 remote-as 100
 neighbor 10.8.8.8 remote-as 200
```



Observação: você não precisa do comando bgp cluster-id para o RTC porque existe apenas um RR nesse cluster.



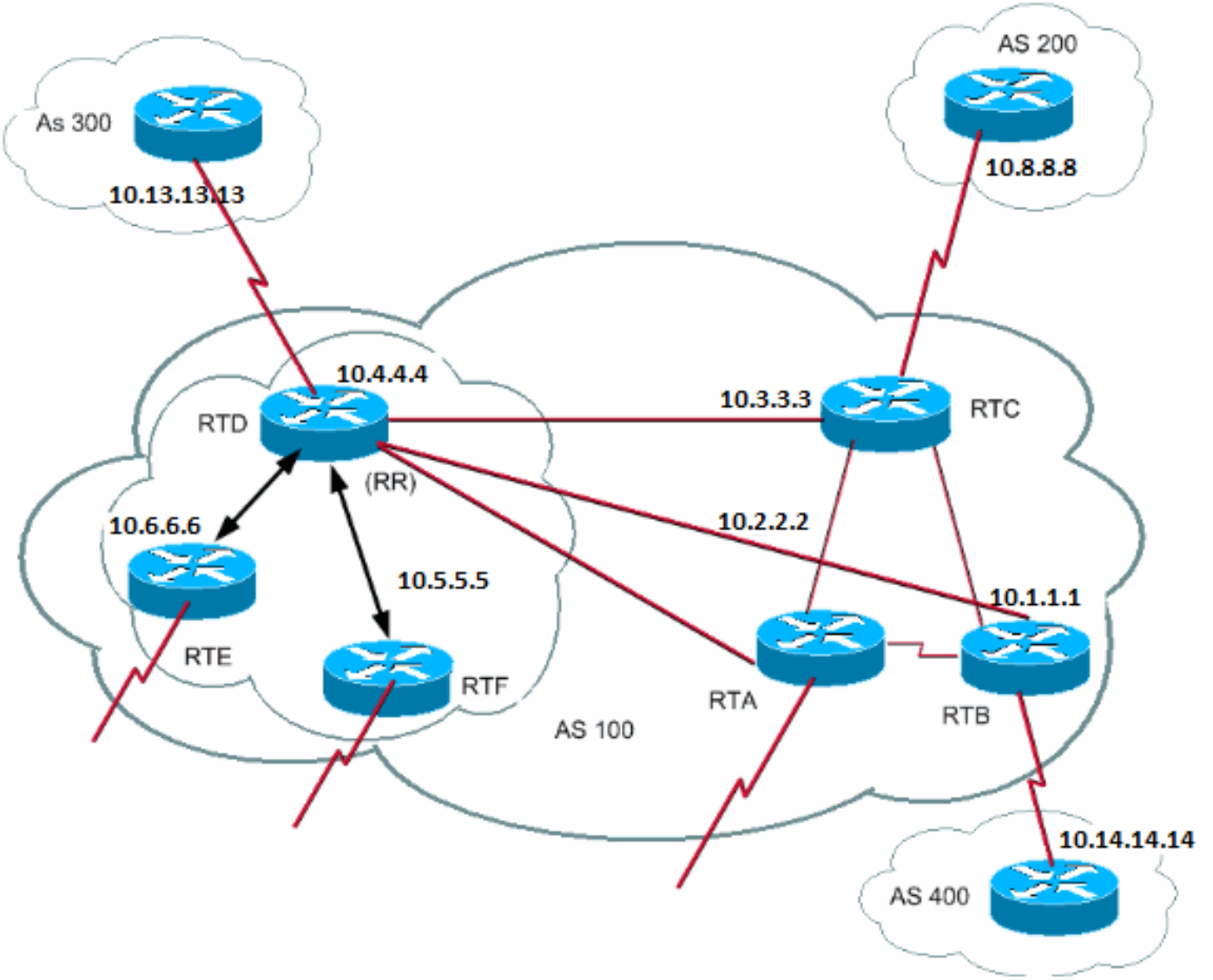
Cuidado: esta configuração não usa grupos de pares. Não use grupos de paridade se os clientes dentro de um conjunto não têm peers diretos entre um outros do iBGP e os clientes trocam atualizações com o RR. Se você configura grupos de paridade, uma retirada potencial à fonte de uma rota no RR transmite 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. Consulte Limitações de Grupos de Pares para obter mais informações.

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. RTC, RTA e 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 precisam entender o esquema de reflexão de rota; somente os RRs exigem a atualização.

Está aqui a configuração do RTD e do RTC:

```
RTD#
router bgp 100
 neighbor 10.6.6.16 remote-as 100
 neighbor 10.6.6.16 route-reflector-client
 neighbor 10.5.5.5 remote-as 100
 neighbor 10.5.5.5 route-reflector-client
 neighbor 10.3.3.3 remote-as 100
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.1.1.1 remote-as 100
 neighbor 10.13.13.13 remote-as 300
```

```
RTC#
router bgp 100
 neighbor 10.4.4.4 remote-as 100
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.1.1.1 remote-as 100
 neighbor 10.14.14.14 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 evitar o loop de informações em potencial: **originator-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 também pode colocar mais restrições em **next-hop-self**, que é uma opção de configuração por vizinho. Quando você usa **next-hop**-selfo n RRs, a cláusula afeta somente o próximo salto de rotas aprendidas do eBGP porque o próximo salto de rotas refletidas não deve ser alterado.

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 atinge um limite de supressão predefinido, ocorre a supressão do anúncio de rota. A penalidade decai exponencialmente com base em um tempo de meia-vida pré-configurado. Quando a pena diminui sob um limite de reutilização predefinido, o anúncio de rota não é mais suprimido.

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. As rotas não são suprimidas em uma granularidade de 10 segundos. O roteador mantém as informações de redução até que a pena se torne menos da metade do limite de reutilização. Nesse ponto, o roteador remove a informação.

Inicialmente, umedecer-se por padrão. Se houver necessidade, esse recurso poderá receber a habilitação padrão no futuro. Estes comandos controlam o silenciamento da rota:

- **umedecimento BGP - Gira sobre o umedecimento.**

- **não silenciamento BGP - Desliga o silenciamento.**

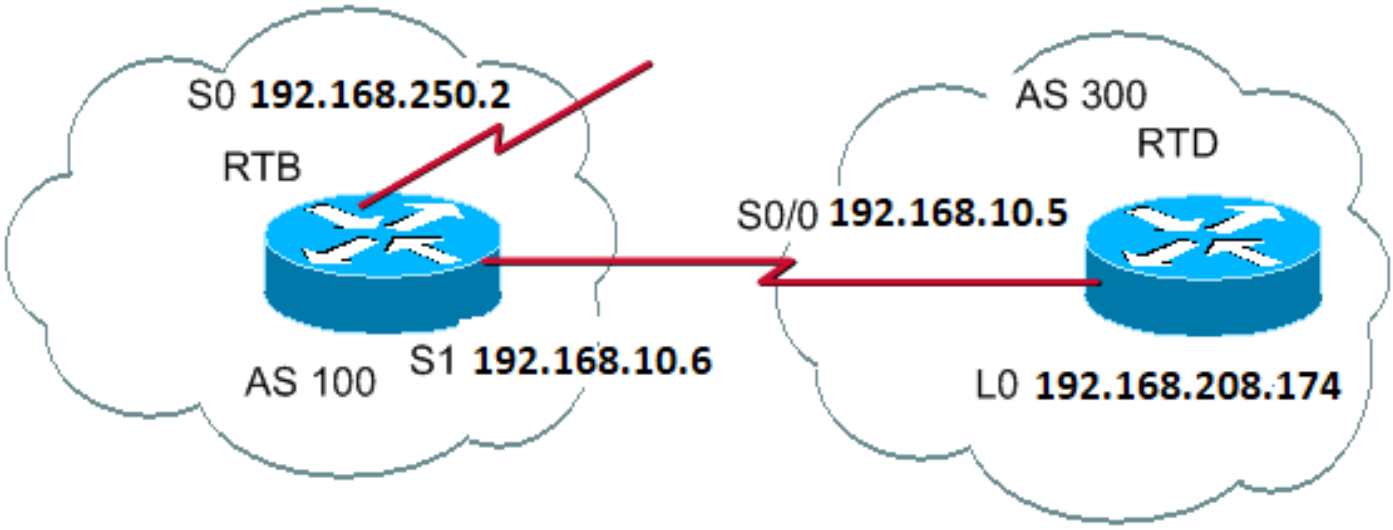
- **bgp dampeninghalf-life-time**— Altera o tempo de meia-vida.

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

- **bgp dampeninghalf-life-timereusesuppressmaximum-suppress-time**

Esta lista detalha a sintaxe:

- **half-life-time**— O intervalo é de 1-45 minutos, e o padrão atual é de 15 minutos.
- **reuse-value**— O intervalo é 1-20.000 e o padrão é 750.
- **suppress-value**— O intervalo é 1-20.000 e o padrão é 2000.
- **max-suppress-time**— Esta é a duração máxima para a supressão de uma rota. O intervalo é de 1 a 255 minutos e o padrão é 4 vezes o tempo de meia-vida.




```
RTB#
hostname RTB

interface Serial0
 ip address 192.168.250.2 255.255.255.252

interface Serial1
 ip address 192.168.10.6 255.255.255.252

router bgp 100
 bgp dampening
 network 192.168.250.15
 neighbor 192.168.10.5 remote-as 300
```

```
RTD#
hostname RTD

interface Loopback0
 ip address 192.168.208.174 255.255.255.192

interface Serial0/0
 ip address 192.168.10.5 255.255.255.252

router bgp 300
 network 192.168.10.0
 neighbor 192.168.10.6 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:

<#root>

```
RTB#
```

show ip bgp

```
BGP table version is 24, local router ID is 192.168.250.2 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
*> 192.168.10.0     192.168.10.5           0             0 300 i
*> 192.168.250.15     0.0.0.0                0         32768 i
```

A fim de simular um flap da rota, emita o comando clear ip bgp 192.168.10.6 no RTD. A tabela de BGP RTB olha como esta:

```
<#root>

RTB#
```

show ip bgp

```
BGP table version is 24, local router ID is 192.168.250.2 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 192.168.10.0     192.168.10.5           0             0 300 i
*> 192.168.250.15     0.0.0.0                0         32768 i
```

A entrada de BGP para 192.168.10.0 está em um estado de histórico. Esta colocação significa que você não tem um melhor caminho à rota, mas informação sobre a oscilação da rota ainda existe.

```
<#root>
```

```
RTB#
```

show ip bgp 192.168.10.0

```
BGP routing table entry for 192.168.10.0 255.255.255.0, version 25
Paths: (1 available, no best path)
300 (history entry)
    192.168.10.5 from 192.168.10.5 (192.168.208.174)
Origin IGP, metric 0, external
Dampinfo: penalty 910, flapped 1 times in 0:02:03
```

A rota recebeu uma penalidade por oscilação, mas a penalidade ainda está 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:

```
    <#root>

RTB#
```

show ip bgp

```
BGP table version is 32, local router ID is 192.168.250.2 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 192.168.10.0     192.168.10.5           0             0  300 i
*> 192.168.250.15     0.0.0.0               0       32768  i

RTB#
```

show ip bgp 192.168.10.0

```
BGP routing table entry for 192.168.10.0 255.255.255.0, version 32
```

```
Paths: (1 available, no best path)
300, (suppressed due to dampening)
192.168.10.5 from 192.168.10.5 (192.168.208.174)
      Origin IGP, metric 0, valid, external
Dampinfo: penalty 2615, flapped 3 times in 0:05:18 , reuse in 0:27:00
```

A rota foi amortecida 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 ip bgp flap-statistics regexpregular-expression** — Exibe estatísticas de flap para todos os caminhos que correspondem à expressão regular.
- **show ip bgp flap-statistics filter-listlist**— Exibe estatísticas de flap para todos os caminhos que passam pelo filtro.
- **show ip bgp flap-statisticsA.B.C.D m.m.m.m**— Exibe estatísticas de flap para uma única entrada.
- **show ip bgp flap-statisticsA.B.C.D m.m.mlonger-prefix — Exibe estatísticas de flap para entradas mais específicas.**
- **show ip bgp neighbor [dampened-routes] | [flap-statistics]** — Exibe estatísticas de flap para todos os caminhos de um vizinho.
- **clear ip bgp flap-statistics - Limpar as estatísticas do flap para todas as rotas.**
- **clear ip bgp flap-statistics regexpregular-expression**— Limpa as estatísticas de flap para todos os caminhos que correspondem à

expressão regular.

- **clear ip bgp flap-statistics filter-listlist — Limpa as estatísticas de flap para todos os caminhos que passam pelo filtro.**

- **clear ip bgp flap-statisticsA.B.C.D m.m.m.m**— Limpa as estatísticas de flap para uma única entrada.

- **clear ip bgpA.B.C.Dflap-statistics** — Limpa as estatísticas de sincronismo para todos os caminhos de um vizinho.

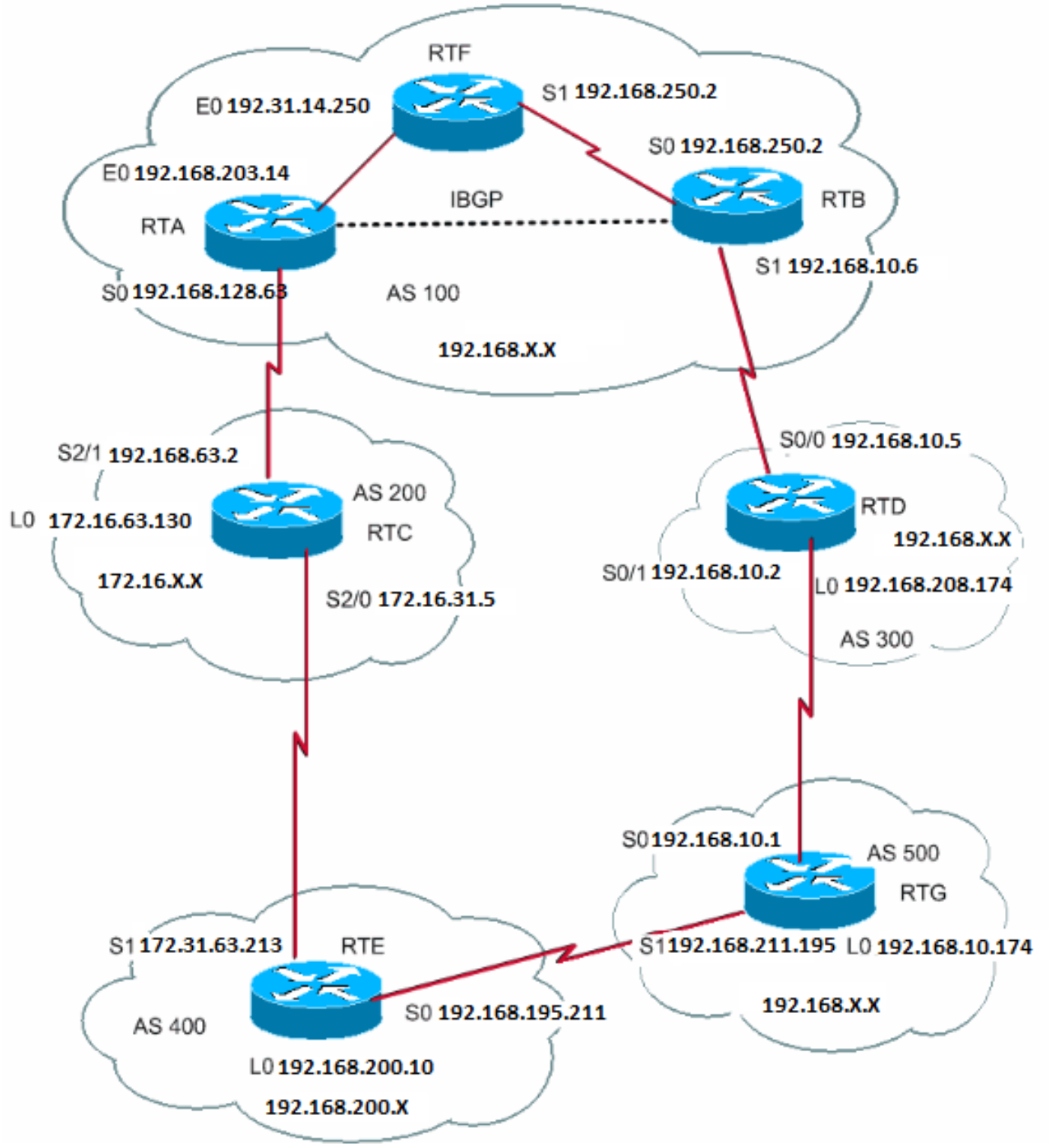
Como o BGP seleciona um trajeto

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

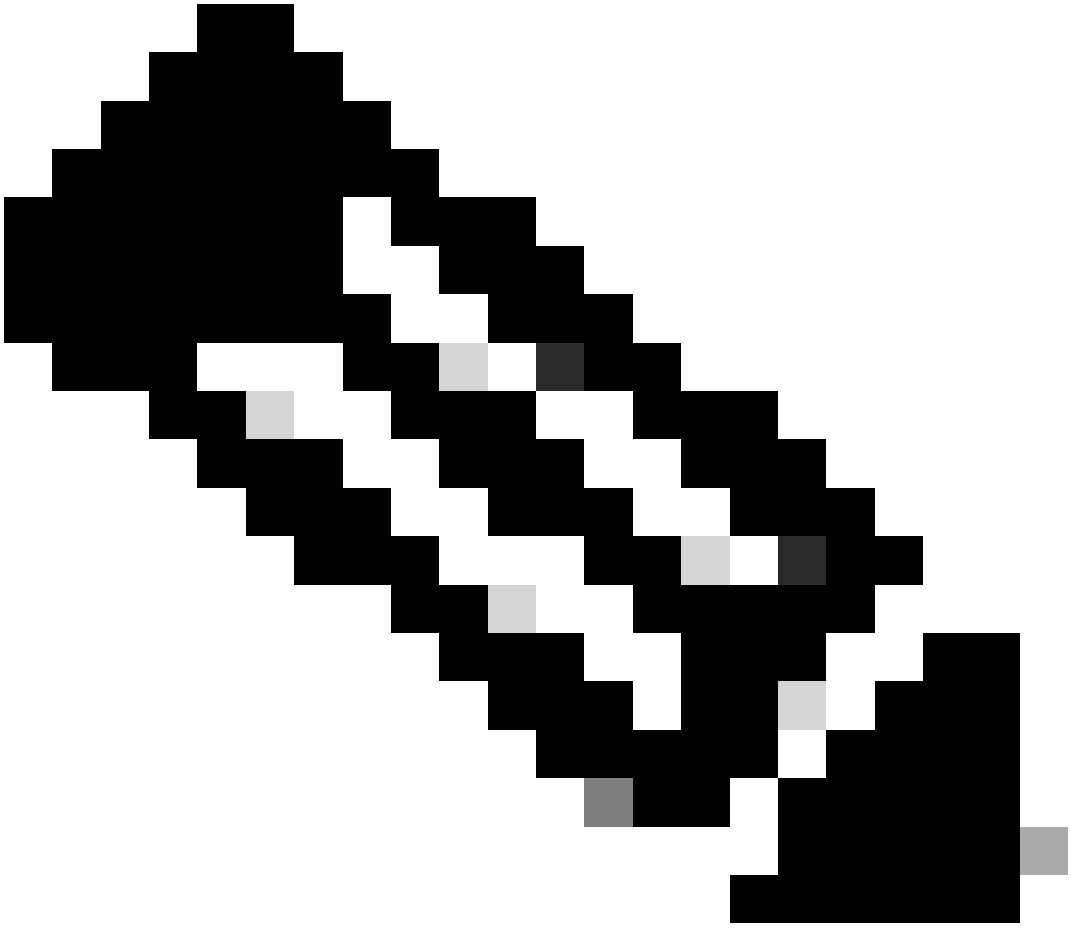
Estudos de Caso do BGP 5

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:



Observação: essas configurações não são as configurações finais.

```
RTA#
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 192.168.203.250 255.255.255.0

interface Ethernet0
ip address 192.168.203.14 255.255.255.0

interface Serial0
```

```
 ip address 192.168.128.63 255.255.255.252

router ospf 10
 network 192.168.203.25 0.0.255.255 area 0

router bgp 100
 network 192.168.203.13
 network 192.168.250.14
 neighbor 172.31.63.250 remote-as 200
 neighbor 192.168.250.2 remote-as 100
 neighbor 192.168.250.2 update-source Loopback0

RTF#
hostname RTF

ip subnet-zero

interface Ethernet0
 ip address 172.31.14.250 255.255.255.0

interface Serial1
 ip address 172.16.15.250 255.255.255.252

router ospf 10
 network 192.168.203.25 0.0.255.255 area 0

RTB#
hostname RTB

ip subnet-zero

interface Serial0
 ip address 192.168.250.2 255.255.255.252

interface Serial1
 ip address 192.168.10.6 255.255.255.252

router ospf 10
 network 192.168.203.25 0.0.255.255 area 0

router bgp 100
network 192.168.250.15
 neighbor 192.168.10.5 remote-as 300
 neighbor 192.168.203.250 remote-as 100

RTC#
hostname RTC

ip subnet-zero

interface Loopback0
 ip address 192.168.128.6330 255.255.255.192

interface Serial2/0
 ip address 172.16.31.5 255.255.255.252
!
interface Serial2/1
 ip address 172.31.63.250 255.255.255.252

router bgp 200
 network 172.31.10.0
 neighbor 192.168.128.63 remote-as 100
```

```
 neighbor 172.31.63.213 remote-as 400

RTD#
hostname RTD

ip subnet-zero

interface Loopback0
ip address 192.168.208.174 255.255.255.192

interface Serial0/0
 ip address 192.168.10.5 255.255.255.252
!
interface Serial0/1
 ip address 192.168.10.2 255.255.255.252

router bgp 300
 network 192.168.10.0
 neighbor 192.168.10.1 remote-as 500
 neighbor 192.168.10.6 remote-as 100

RTE#
hostname RTE

ip subnet-zero

interface Loopback0
ip address 192.168.200.10 255.255.255.0

interface Serial0
 ip address 192.168.195.211 255.255.255.252

interface Serial1
 ip address 172.31.63.213 255.255.255.252
 clockrate 1000000

router bgp 400
 network 192.168.10.10
 neighbor 172.16.31.5 remote-as 200
 neighbor 192.168.211.195 remote-as 500

RTG#
hostname RTG

ip subnet-zero

interface Loopback0
 ip address 192.168.211.19574 255.255.255.192

interface Serial0
 ip address 192.168.10.1 255.255.255.252

interface Serial1
 ip address 192.168.211.195 255.255.255.252

router bgp 500
 network 192.168.211.10
 neighbor 192.168.10.2 remote-as 300
 neighbor 192.168.195.211 remote-as 400
```

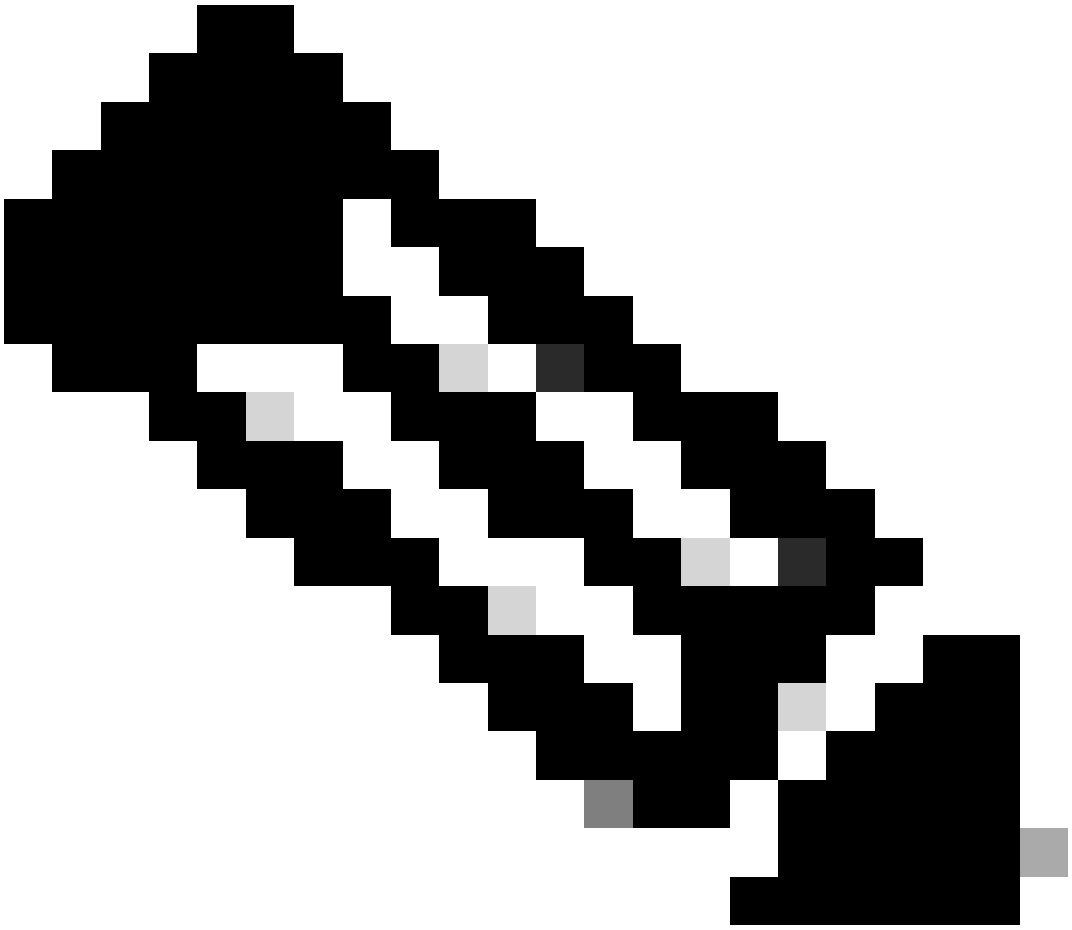
Sempre use o network comando ou redistribua entradas estáticas no BGP para anunciar redes. Este método é melhor do que uma redistribuição do IGP no BGP. Este exemplo usa o network comando para injetar redes no BGP.

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:

```
<#root>

RTB#

show ip bgp BGP

table version is 4, local router ID is 192.168.250.2 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
*i172.31.10.0      172.31.63.250           0    100      0 200 i
*i192.168.10.0     172.31.63.250                100      0 200 400 500
300 i
*i192.168.211.10     172.31.63.250                100      0 200 400 500 i
*i192.168.10.10     172.31.63.250                100      0 200 400 i
*>i192.168.203.13    192.168.203.250          0    100      0 i
*>i192.168.250.14    192.168.203.250          0    100      0 i
*>192.168.250.15     0.0.0.0                  0         32768 i
```

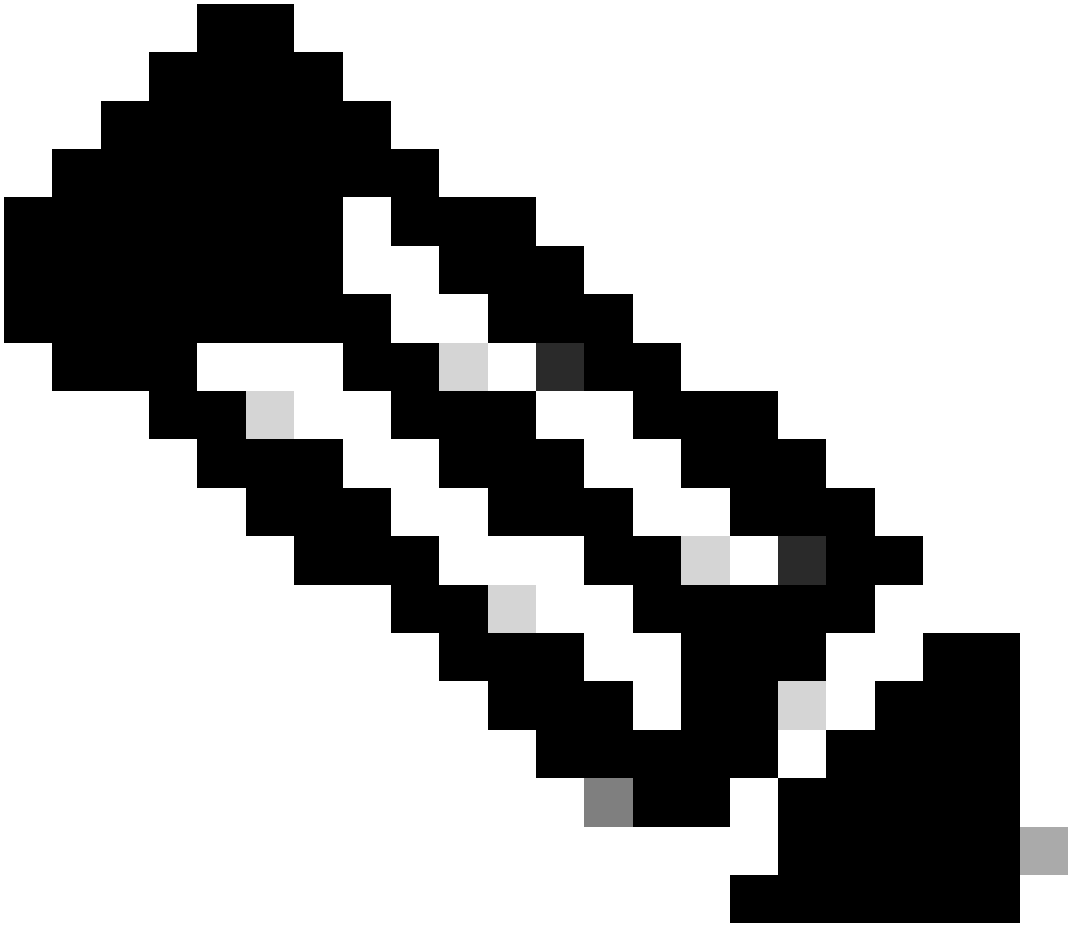
Nesta tabela, estas notações aparecem:

- Aniat the starting—Indica que a entrada foi aprendida através de um peer iBGP.
- Aniat the end—Indica que a origem das informações de caminho é IGP.
- Informações de caminho — Essas informações são intuitivas. Por exemplo, a rede 172.31.10.0 é instruída através do trajeto 200 com um salto seguinte de 172.31.63.250.



Observação: qualquer entrada gerada localmente, como 192.168.250.15, tem um próximo salto 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.



Observação: observe o atributo Próximo Salto. O RTB sabe sobre 172.31.10.0 através de um salto seguinte de 172.31.63.250, que seja o salto seguinte do eBGP levado no iBGP.

Olhe a tabela de IP Routing:

<#root>

RTB#

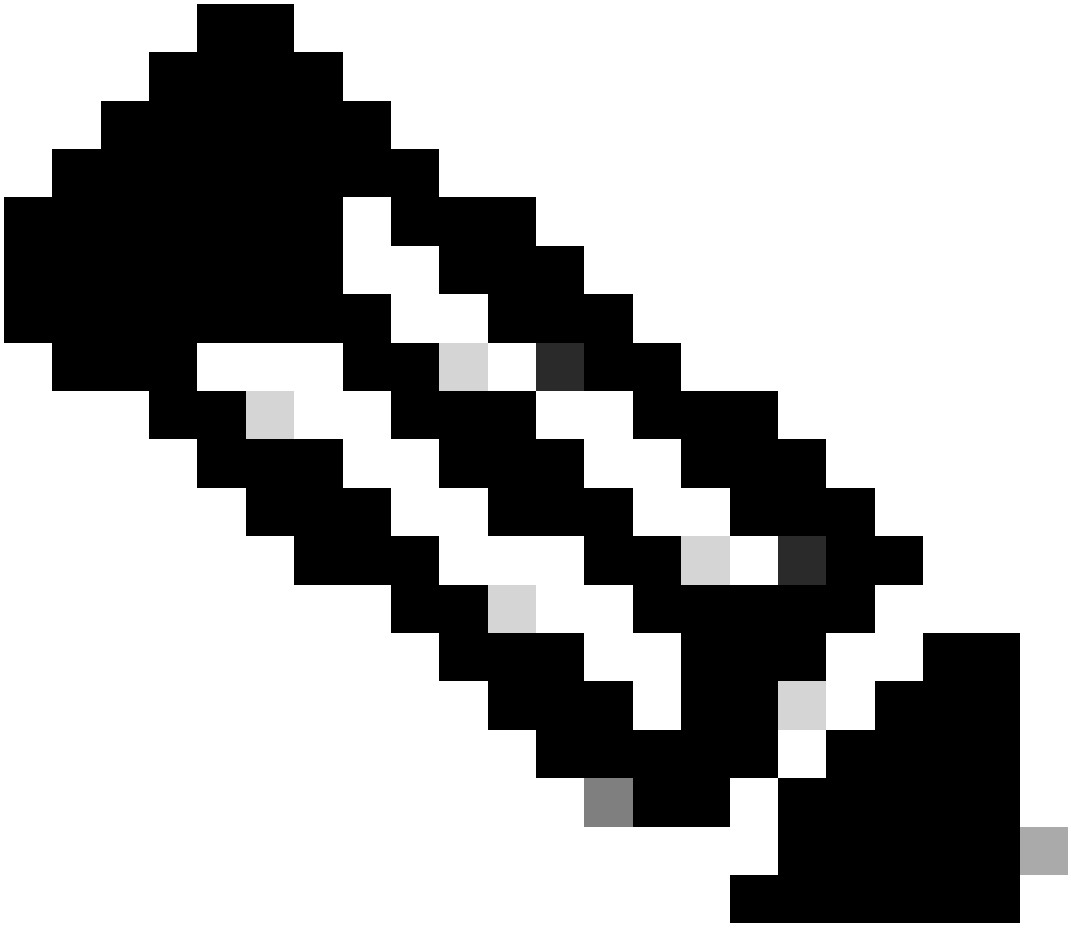
show ip route

```
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

     192.168.203.13 255.255.255.255 is subnetted, 1 subnets
O       192.168.203.250 [110/75] via 172.16.15.250, 02:50:45, Serial0
     192.168.250.15 255.255.255.252 is subnetted, 1 subnets
C       192.168.250.15 is directly connected, Serial0
O    192.168.250.14 [110/74] via 172.16.15.250, 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, 172.31.63.250, é inacessível. Não há nenhuma maneira de alcançar esse salto seguinte através deste IGP, que é OSPF. O RTB não aprendeu sobre 192.168.213.63 através do OSPF. Você pode executar o OSPF na interface s0 do RTA e torná-lo passivo; dessa forma, o RTB sabe como alcançar o próximo salto 172.31.63.250. Esta configuração RTA aparece aqui:

```
RTA#
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 192.168.203.250 255.255.255.0

interface Ethernet0
ip address 192.168.203.14 255.255.255.0

interface Serial0
 ip address 192.168.128.63 255.255.255.252

router ospf 10
 passive-interface Serial0
 network 192.168.203.25 0.0.255.255 area 0
 network 172.31.10.0 0.0.255.255 area 0

router bgp 100
 network 192.168.203.25 mask 255.255.0.0
 neighbor 172.31.63.250 remote-as 200
 neighbor 192.168.250.2 remote-as 100
 neighbor 192.168.250.2 update-source Loopback0
```



Observação: você pode emitir o comandobgp nexthop self entre RTA e RTB a fim alterar o salto seguinte.

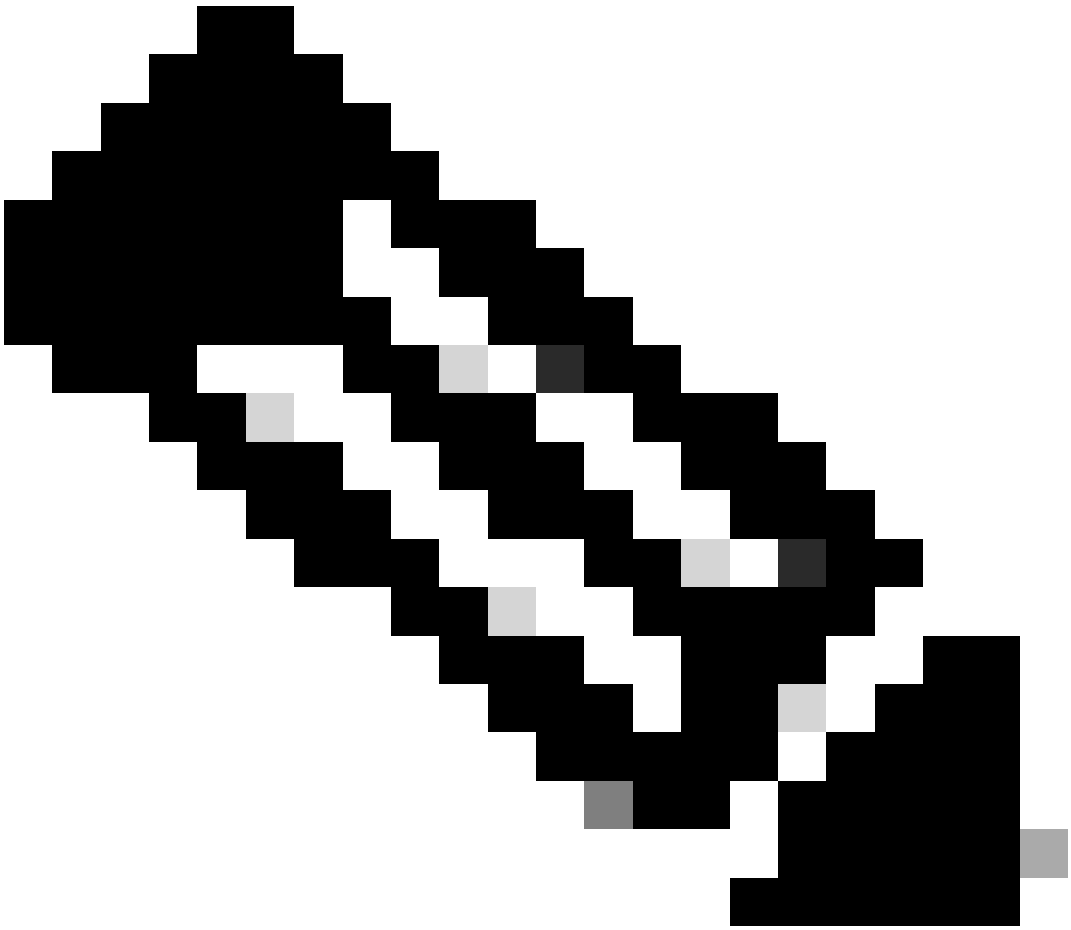
A tabela de BGP nova no RTB olha como esta:

<#root>

RTB#

show ip bgp

```
BGP table version is 10, local router ID is 192.168.250.2
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
*>i172.31.10.0      172.31.63.250           0    100      0 200 i
*>i192.168.10.0     172.31.63.250                100      0 200 400 500
300 i
*>i192.168.211.10     172.31.63.250                100      0 200 400 500 i
*>i192.168.10.10     172.31.63.250                100      0 200 400 i
*>i192.168.203.13     192.168.203.250          0    100      0 i
*>i192.168.250.14     192.168.203.250          0    100      0 i
*> 192.168.250.15     0.0.0.0                  0         32768 i
```



Observação: todas as entradas têm >, o que significa que o BGP pode alcançar o próximo salto.

Olhe a tabela de roteamento:

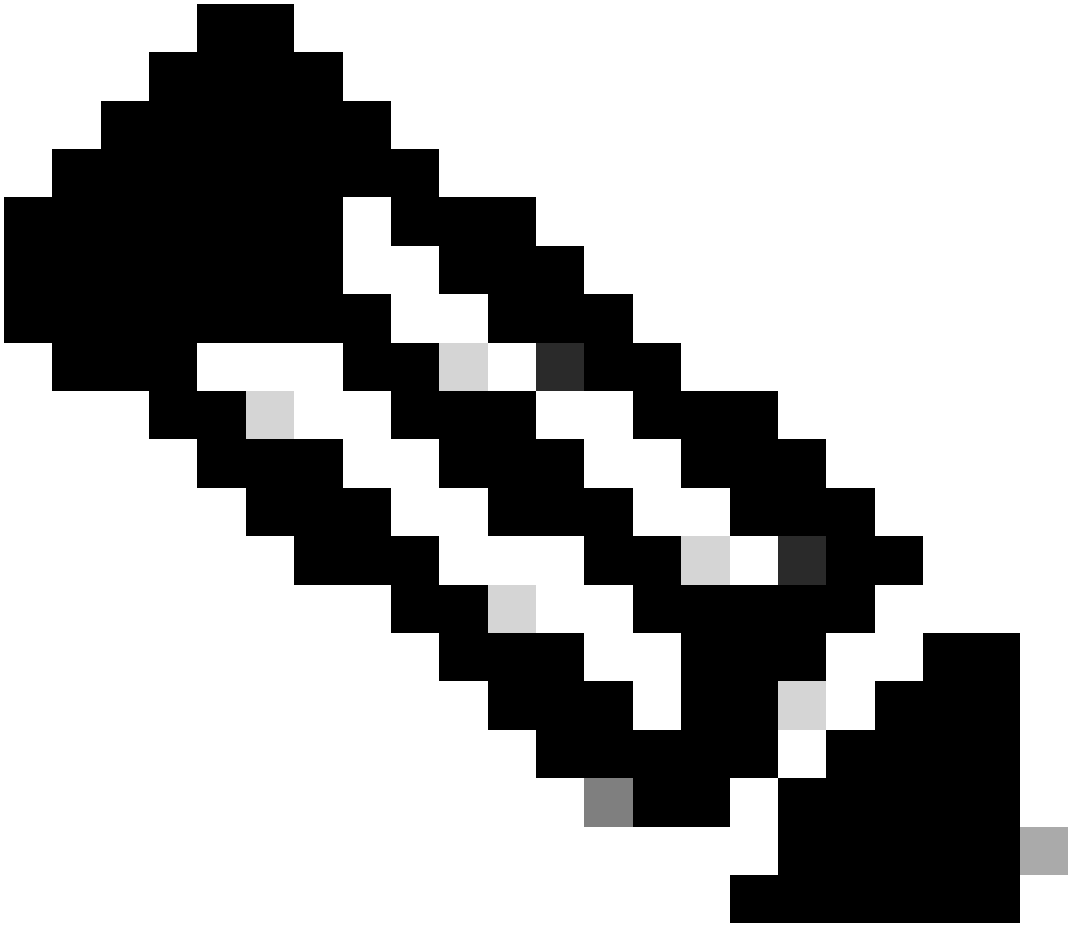
<#root>

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

     192.168.203.13 255.255.255.255 is subnetted, 1 subnets
O       192.168.203.250 [110/75] via 172.16.15.250, 00:04:46, Serial0
     192.168.250.15 255.255.255.252 is subnetted, 1 subnets
C       192.168.250.15 is directly connected, Serial0
O    192.168.250.14 [110/74] via 172.16.15.250, 00:04:46, Serial0
     172.31.10.0 255.255.255.252 is subnetted, 1 subnets
O       192.168.213.63 [110/138] via 172.16.15.250, 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 192.168.213.63 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.



Observação: o RTF não tem noção das redes 192.168.10.0 e 192.168.211.10 porque você ainda não redistribuiu o BGP no OSPF.

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:

<#root>

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    192.168.10.10 [200/0] via 172.31.63.250, 00:01:07
B    192.168.211.10 [200/0] via 172.31.63.250, 00:01:07
B    192.168.10.0 [200/0] via 172.31.63.250, 00:01:07
     192.168.203.13 is variably subnetted, 2 subnets, 2 masks
O       192.168.203.250 255.255.255.255
           [110/75] via 172.16.15.250, 00:12:37, Serial0
B       192.168.203.13 255.255.255.0 [200/0] via 192.168.203.250, 00:01:08
     192.168.250.15 255.255.255.252 is subnetted, 1 subnets
C       192.168.250.15 is directly connected, Serial0
O    192.168.250.14 [110/74] via 172.16.15.250, 00:12:37, Serial0
     172.31.10.0 is variably subnetted, 2 subnets, 2 masks
B       172.31.10.0 255.255.0.0 [200/0] via 172.31.63.250, 00:01:08
O       192.168.213.63 255.255.255.252
           [110/138] via 172.16.15.250, 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:

<#root>

```
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

     192.168.203.13 255.255.255.255 is subnetted, 1 subnets
```

```
O       192.168.203.250 [110/11] via 192.168.203.14, 00:14:15, Ethernet0
     192.168.250.15 255.255.255.252 is subnetted, 1 subnets
C       192.168.250.15 is directly connected, Serial1
C    192.168.250.14 is directly connected, Ethernet0
     172.31.10.0 255.255.255.252 is subnetted, 1 subnets
O       192.168.213.63 [110/74] via 192.168.203.14, 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 192.168.203.250 255.255.255.0

interface Ethernet0
ip address 192.168.203.14 255.255.255.0

interface Serial0
 ip address 192.168.128.63 255.255.255.252

router ospf 10
 redistribute bgp 100 metric 2000 subnets
 passive-interface Serial0
 network 192.168.203.25 0.0.255.255 area 0
 network 172.31.10.0 0.0.255.255 area 0

router bgp 100
 network 192.168.203.25 mask 255.255.0.0
 neighbor 172.31.63.250 remote-as 200
 neighbor 192.168.250.2 remote-as 100
 neighbor 192.168.250.2 update-source Loopback0
```

A tabela de roteamento olha como esta:

<#root>

```
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 192.168.10.10 [110/2000] via 172.16.15.250, 00:00:14, Serial0
O E2 192.168.211.10 [110/2000] via 172.16.15.250, 00:00:14, Serial0
O E2 192.168.10.0 [110/2000] via 172.16.15.250, 00:00:14, Serial0
     192.168.203.13 is variably subnetted, 2 subnets, 2 masks
O       192.168.203.250 255.255.255.255
           [110/75] via 172.16.15.250, 00:00:15, Serial0
O E2    192.168.203.13 255.255.255.0
           [110/2000] via 172.16.15.250, 00:00:15, Serial0
     192.168.250.15 255.255.255.252 is subnetted, 2 subnets
C       172.31.250.8 is directly connected, Loopback1
C       192.168.250.15 is directly connected, Serial0
O    192.168.250.14 [110/74] via 172.16.15.250, 00:00:15, Serial0
     172.31.10.0 is variably subnetted, 2 subnets, 2 masks
O E2    172.31.10.0 255.255.0.0 [110/2000] via 172.16.15.250,
00:00:15,Serial0
O       192.168.213.63 255.255.255.252
           [110/138] via 172.16.15.250, 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 192.168.250.15. 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 192.168.203.13. 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 192.168.10.5 através do IGP. Se você não toma esta etapa, os loop de roteamento ocorrem porque, a fim alcançar o salto seguinte 192.168.10.5, 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 192.168.203.250 255.255.255.0

interface Ethernet0
ip address 192.168.203.14 255.255.255.0
```

```
interface Serial0
 ip address 192.168.128.63 255.255.255.252

router ospf 10
 redistribute bgp 100 metric 2000 subnets
 passive-interface Serial0
 network 192.168.203.25 0.0.255.255 area 0
 network 172.31.10.0 0.0.255.255 area 0

router bgp 100
 no synchronization
 network 192.168.203.13
 network 192.168.250.14
 neighbor 172.31.63.250 remote-as 200
 neighbor 192.168.250.2 remote-as 100
 neighbor 192.168.250.2 update-source Loopback0

RTB#
hostname RTB

ip subnet-zero

interface Serial0
 ip address 192.168.250.2 255.255.255.252

interface Serial1
 ip address 192.168.10.6 255.255.255.252

router ospf 10
 redistribute bgp 100 metric 1000 subnets
 passive-interface Serial1
 network 192.168.203.25 0.0.255.255 area 0
 network 192.168.208.0 0.0.255.255 area 0

router bgp 100
 no synchronization
 network 192.168.250.15
 neighbor 192.168.10.5 remote-as 300
 neighbor 192.168.203.250 remote-as 100
```

As tabelas de BGP olham como esta:

<#root>

```
RTA#
```

show ip bgp

```
BGP table version is 117, local router ID is 192.168.203.250
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
*> 172.31.10.0      172.31.63.250           0             0 200 i
*>i192.168.10.0     192.168.10.5            0    100      0 300 i
*>i192.168.211.10     192.168.10.5                100      0 300 500 i
*                   172.31.63.250                         0 200 400 500 i
*> 192.168.10.10     172.31.63.250                        0 200 400 i
*> 192.168.203.13     0.0.0.0                 0         32768 i
*> 192.168.250.14     0.0.0.0                 0         32768 i
*>i192.168.250.15     192.168.250.2             0    100      0 i

RTB#
```

show ip bgp

```
BGP table version is 12, local router ID is 172.16.15.2500
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
*>i172.31.10.0      172.31.63.250           0    100      0 200 i
*                   192.168.10.5                          0 300 500 400
200 i
*> 192.168.10.0     192.168.10.5            0             0 300 i
*> 192.168.211.10     192.168.10.5                          0 300 500 i
*>i192.168.10.10     172.31.63.250                100      0 200 400 i
*                   192.168.10.5                          0 300 500 400 i
*>i192.168.203.13     192.168.203.250           0    100      0 i
*>i192.168.250.14     192.168.203.250           0    100      0 i
*> 192.168.250.15     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:

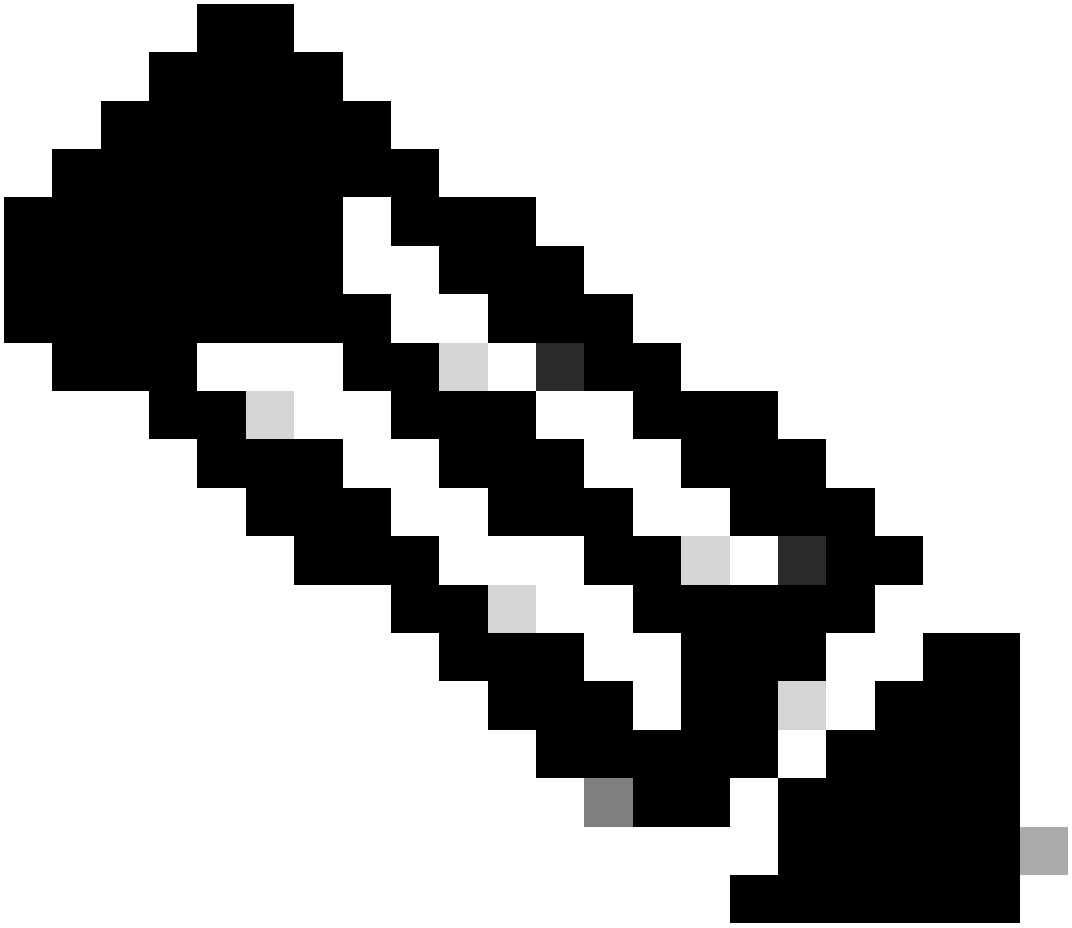
```
RTA#
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 192.168.203.250 255.255.255.0

interface Ethernet0
 ip address 192.168.203.14 255.255.255.0

interface Serial0
 ip address 192.168.128.63 255.255.255.252

router ospf 10
 redistribute bgp 100 metric 2000 subnets
 passive-interface Serial0
 network 192.168.203.25 0.0.255.255 area 0
 network 172.31.10.0 0.0.255.255 area 0
 default-information originate metric 2000

router bgp 100
 no synchronization
 network 192.168.203.13
 network 192.168.250.14
 neighbor 172.31.63.250 remote-as 200
 neighbor 172.31.63.250 route-map setlocalpref in
 neighbor 192.168.250.2 remote-as 100
 neighbor 192.168.250.2 update-source Loopback0

ip classless
ip default-network 172.31.200.200

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 172.31.200.200 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 172.31.14.250 255.255.255.0

interface Serial1
 ip address 172.16.15.250 255.255.255.252

router ospf 10
 network 192.168.203.25 0.0.255.255 area 0

ip classless

RTB#
hostname RTB

ip subnet-zero

interface Loopback1
 ip address 172.16.15.2500 255.255.255.252

interface Serial0
 ip address 192.168.250.2 255.255.255.252
!
interface Serial1
 ip address 192.168.10.6 255.255.255.252

router ospf 10
 redistribute bgp 100 metric 1000 subnets
 passive-interface Serial1
 network 192.168.203.25 0.0.255.255 area 0
 network 192.168.10.6 0.0.0.0 area 0
 default-information originate metric 1000
!
router bgp 100
 no synchronization
 network 192.168.250.15
 neighbor 192.168.10.5 remote-as 300
 neighbor 192.168.10.5 route-map localonly in
 neighbor 192.168.203.250 remote-as 100
!
ip classless
ip default-network 192.168.10.0
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. O RTA é a preferência.



Observação: você anunciou apenas as rotas locais 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:

<#root>

RTB#

show ip bgp regexp ^300$

```
BGP table version is 14, local router ID is 172.16.15.2500
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
*> 192.168.10.0     192.168.10.5           0    300      0 300

RTC#
hostname RTC

ip subnet-zero

interface Loopback0
 ip address 192.168.128.6330 255.255.255.192

interface Serial2/0
 ip address 172.16.31.5 255.255.255.252
!
interface Serial2/1
 ip address 172.31.63.250 255.255.255.252

router bgp 200
 network 172.31.10.0
 neighbor 192.168.128.63 remote-as 100
 neighbor 192.168.128.63 distribute-list 1 out
 neighbor 172.31.63.213 remote-as 400

ip classless
access-list 1 deny   192.168.211.0 0.0.255.255
access-list 1 permit any
```

No RTC, você agrega 172.31.10.0/16 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 192.168.208.174 255.255.255.192
!
interface Serial0/0
 ip address 192.168.10.5 255.255.255.252
!
interface Serial0/1
 ip address 192.168.10.2 255.255.255.252
```

```
router bgp 300
 network 192.168.10.0
 neighbor 192.168.10.1 remote-as 500
 neighbor 192.168.10.6 remote-as 100

RTG#
hostname RTG

ip subnet-zero

interface Loopback0
 ip address 192.168.211.19574 255.255.255.192

interface Serial0
 ip address 192.168.10.1 255.255.255.252

interface Serial1
 ip address 192.168.211.195 255.255.255.252

router bgp 500
 network 192.168.211.10
 aggregate-address 192.168.211.0 255.255.0.0 summary-only
 neighbor 192.168.10.2 remote-as 300
 neighbor 192.168.10.2 send-community
 neighbor 192.168.10.2 route-map setcommunity out
 neighbor 192.168.195.211 remote-as 400
!
ip classless
access-list 1 permit 192.168.211.0 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 de como usar a filtragem de comunidade está no RTG. Você adiciona uma no-export comunidade às atualizações de 192.168.211.0 em direção ao 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 192.168.200.10 255.255.255.0

interface Serial0
 ip address 192.168.195.211 255.255.255.252

interface Serial1
 ip address 172.31.63.213 255.255.255.252

router bgp 400
```

```
 network 192.168.10.10
 aggregate-address 172.31.200.200 255.255.0.0 summary-only
 neighbor 172.16.31.5 remote-as 200
 neighbor 192.168.211.195 remote-as 500

ip classless
```

o RTE agrega 172.31.200.200/16. Estão aqui o BGP e as tabelas de roteamento para o RTA, o RTF, e o RTB finais:

```
    <#root>


RTA#
```

show ip bgp

```
BGP table version is 21, local router ID is 192.168.203.250
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
*> 172.31.10.0      172.31.63.250           0    200      0 200 i
*>i192.168.10.0     192.168.10.5            0    300      0 300 i
*> 172.31.200.200/16   172.31.63.250               200      0 200 400 i
*> 192.168.203.13     0.0.0.0                  0         32768 i
*> 192.168.250.14     0.0.0.0                  0         32768 i
*>i192.168.250.15     192.168.250.2             0    100      0 i

RTA#
```

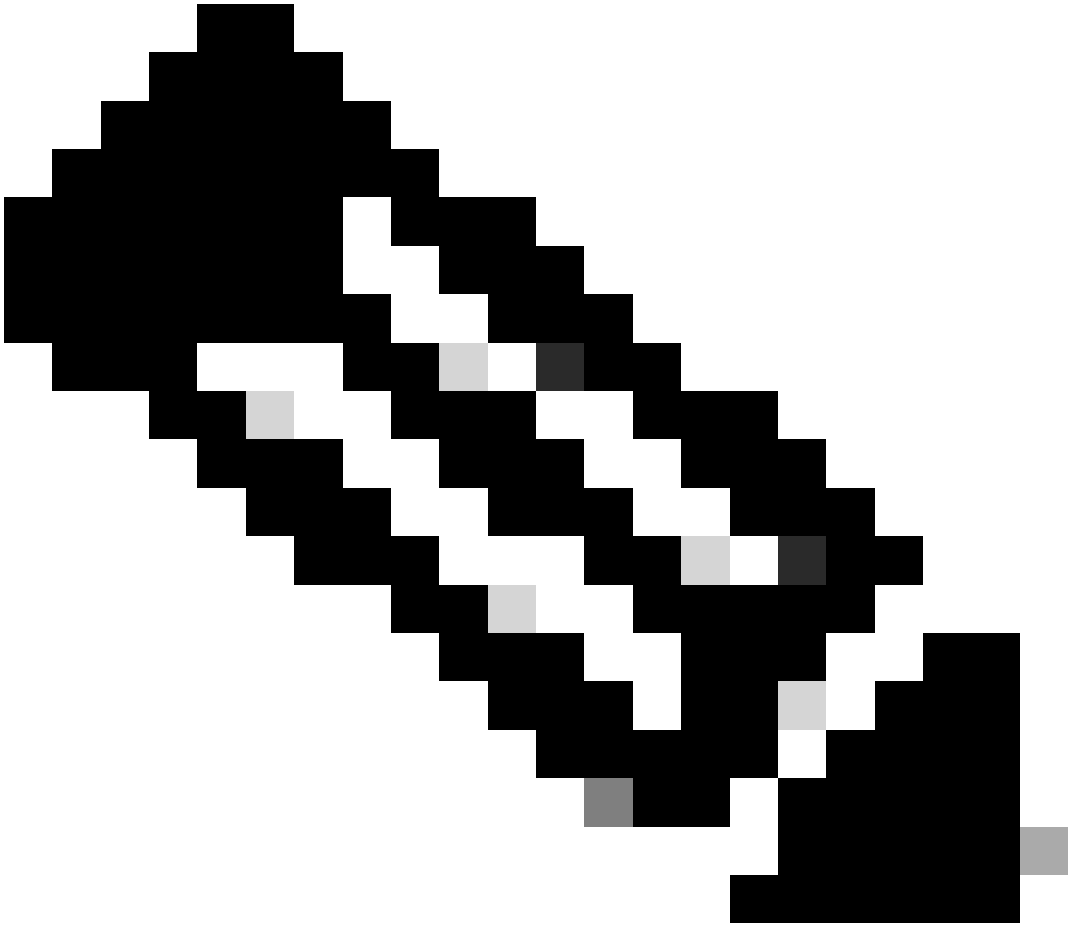
show ip route

```
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 172.31.63.250 to network 172.31.200.200

     192.168.10.0 is variably subnetted, 2 subnets, 2 masks
```

```
O E2    192.168.10.0 255.255.255.0
           [110/1000] via 172.31.14.250, 00:41:25, Ethernet0
O       192.168.10.4 255.255.255.252
           [110/138] via 172.31.14.250, 00:41:25, Ethernet0
C    192.168.203.13 is directly connected, Loopback0
     192.168.250.15 is variably subnetted, 3 subnets, 3 masks
O       172.16.15.2500 255.255.255.255
           [110/75] via 172.31.14.250, 00:41:25, Ethernet0
O       192.168.250.15 255.255.255.252
           [110/74] via 172.31.14.250, 00:41:25, Ethernet0
B       192.168.250.15 255.255.255.0 [200/0] via 192.168.250.2, 00:41:25
C    192.168.250.14 is directly connected, Ethernet0
     172.31.10.0 is variably subnetted, 2 subnets, 2 masks
B       172.31.10.0 255.255.0.0 [20/0] via 172.31.63.250, 00:41:26
C       192.168.213.63 255.255.255.252 is directly connected, Serial0
O*E2 0.0.0.0/0 [110/1000] via 172.31.14.250, Ethernet0/0
B*   172.31.200.200 255.255.0.0 [20/0] via 172.31.63.250, 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 192.168.250.2 to network 0.0.0.0

     192.168.10.0 is variably subnetted, 2 subnets, 2 masks
O E2    192.168.10.0 255.255.255.0
           [110/1000] via 192.168.250.2, 00:48:50, Serial1
O       192.168.10.4 255.255.255.252
           [110/128] via 192.168.250.2, 01:12:09, Serial1
     192.168.203.13 is variably subnetted, 2 subnets, 2 masks
O       192.168.203.250 255.255.255.255
           [110/11] via 192.168.203.14, 01:12:09, Ethernet0
O E2    192.168.203.13 255.255.255.0
           [110/2000] via 192.168.203.14, 01:12:09, Ethernet0
     192.168.250.15 is variably subnetted, 2 subnets, 2 masks
O       172.16.15.2500 255.255.255.255
           [110/65] via 192.168.250.2, 01:12:09, Serial1
C       192.168.250.15 255.255.255.252 is directly connected, Serial1
C    192.168.250.14 is directly connected, Ethernet0
     172.31.10.0 is variably subnetted, 2 subnets, 2 masks
O E2    172.31.10.0 255.255.0.0
           [110/2000] via 192.168.203.14, 00:45:01, Ethernet0
O       192.168.213.63 255.255.255.252
           [110/74] via 192.168.203.14, 01:12:11, Ethernet0
O E2 172.31.200.200 255.255.0.0 [110/2000] via 192.168.203.14, 00:03:47, Ethernet0
O*E2 0.0.0.0 0.0.0.0 [110/1000] via 192.168.250.2, 00:03:33, Serial1
```



Observação: a tabela de roteamento RTF indica que a maneira de acessar redes locais para o AS300, como 192.168.10.0, é através do RTB. A maneira de alcançar outras redes conhecidas, tais como 172.31.200.200, 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.

<#root>

RTB#

show ip bgp

```
BGP table version is 14, local router ID is 172.16.15.2500
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
*>i172.31.10.0      172.31.63.250           0    200      0 200 i
*> 192.168.10.0     192.168.10.5            0    300      0 300 i
*>i172.31.200.200/16   172.31.63.250               200      0 200 400 i
*>i192.168.203.13     192.168.203.250          0    100      0 i
*>i192.168.250.14     192.168.203.250          0    100      0 i
*> 192.168.250.15     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 192.168.10.5 to network 192.168.10.0

 *   192.168.10.0 is variably subnetted, 2 subnets, 2 masks
B*      192.168.10.0 255.255.255.0 [20/0] via 192.168.10.5, 00:50:46
C       192.168.10.4 255.255.255.252 is directly connected, Serial1
     192.168.203.13 is variably subnetted, 2 subnets, 2 masks
O       192.168.203.250 255.255.255.255
           [110/75] via 172.16.15.250, 01:20:33, Serial0
O E2    192.168.203.13 255.255.255.0
           [110/2000] via 172.16.15.250, 01:15:40, Serial0
     192.168.250.15 255.255.255.252 is subnetted, 2 subnets
C       172.31.250.8 is directly connected, Loopback1
C       192.168.250.15 is directly connected, Serial0
O    192.168.250.14 [110/74] via 172.16.15.250, 01:20:33, Serial0
     172.31.10.0 is variably subnetted, 2 subnets, 2 masks
O E2    172.31.10.0 255.255.0.0 [110/2000] via 172.16.15.250, 00:46:55, Serial0
O       192.168.213.63 255.255.255.252
           [110/138] via 172.16.15.250, 01:20:34, Serial0
O*E2 0.0.0.0/0 [110/2000] via 172.16.15.250, 00:08:33, Serial0
O E2 172.31.200.200 255.255.0.0 [110/2000] via 172.16.15.250, 00:05:42, Serial0
```

Informações Relacionadas

- **BGP: perguntas frequentes**
- **Configurações de amostra do BGP através de um PIX Firewall**
- **Como usar o HSRP para fornecer redundância em uma rede BGP multihomed**
- **Configurar a redundância de modo de roteador único e o BGP em um MSFC Cat6000**
- **Alcance um Roteamento Ideal e Reduza o Consumo de Memória BGP**
- **Identificar e Solucionar Problemas Comuns do BGP**
- **Solucionar problemas de alta utilização da CPU causados pelo scanner BGP ou pelo processo do roteador**
- **Entender o compartilhamento de carga com BGP em ambientes únicos e multihomed**
- **Página de suporte de BGP**
- **Suporte técnico e downloads da Cisco**

Sobre esta tradução
A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês (link fornecido) seja sempre consultado.