

检查边界网关协议案例研究

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[规则](#)

[BGP 案例分析 1](#)

[BGP 如何工作？](#)

[eBGP 和iBGP](#)

[启用 BGP 路由](#)

[形成 BGP 邻居](#)

[BGP 和环回接口](#)

[eBGP 多跳](#)

[eBGP 多跳 \(负载平衡\)](#)

[路由映射](#)

[match 和 set 配置命令](#)

[示例 1](#)

[示例 2](#)

[network 命令](#)

[重分发](#)

[静态路由和重分配](#)

[iBGP](#)

[BGP 决策算法](#)

[BGP 案例分析 2](#)

[AS_PATH 属性](#)

[源属性](#)

[BGP 下一跳属性](#)

[BGP 下一跳 \(多路访问网络\)](#)

[BGP 下一跳 \(NBMA\)](#)

[next-hop-self 命令](#)

[BGP 后门](#)

[同步](#)

[禁用同步](#)

[权重属性](#)

[本地优先级属性](#)

[度量属性](#)

[社区属性](#)

[BGP 案例分析 3](#)

[BGP过滤器](#)

[路由过滤器](#)

[路径过滤器](#)

[AS 正则表达式](#)

[BGP 社区过滤器](#)

[BGP 邻居和路由映射](#)

[使用 set as-path prepend 命令](#)

[BGP 对等体组](#)

[BGP 案例分析 4](#)

[CIDR 和聚合地址](#)

[聚合命令](#)

[CIDR 示例 1](#)

[CIDR 示例 2 \(as-set\)](#)

[BGP 联盟](#)

[路由反射器](#)

[群集内的多个路由反射器](#)

[路由反射器和常规 BGP 扬声器](#)

[避免路由信息环路](#)

[路由抖动减弱](#)

[BGP 如何选择路径](#)

[BGP 案例分析 5](#)

[实用设计示例](#)

[相关信息](#)

简介

本文档介绍五个边界网关协议 (BGP) 案例研究。

先决条件

要求

本文档没有任何特定的要求。

使用的组件

本文档不限于特定的软件和硬件版本。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始 (默认) 配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

规则

有关文档规则的详细信息，请参阅 Cisco 技术提示规则。

BGP 案例分析 1

使用由 RFC 1771 定义的 BGP，可以在自治系统 (AS) 之间创建无环路域间路由。AS 是同处一个技术管理之下的一组路由器。AS 中的路由器可以使用多个内部网关协议 (IGP) 在 AS 内部交换路由

信息。这些路由器可以使用外部网关协议将数据包路由至 AS 外部。

BGP 如何工作？

BGP 在端口 179 上使用 TCP 作为传输协议。两个 BGP 路由器互相之间形成 TCP 连接。这些路由器是对等路由器。对等路由器交换消息以公开和确认连接参数。

BGP 路由器交换网络可达性信息。这些信息主要指示路由为达到目标网络而必须通过的完整路径。路径以 BGP AS 编号形式表示。此信息有助于构建无环路 AS 的图形。该图形还显示在何处应用路由策略来对路由行为强制实施某些限制。

任何两个形成 TCP 连接以交换 BGP 路由信息的路由器都是“对等体”（或称“邻居”）。BGP 对等体最初交换完整的 BGP 路由表。在此交换以后，对等体在路由表更改时发送增量更新。BGP 保存 BGP 路由表的版本号。所有 BGP 对等体的版本号都是相同的。每当 BGP 通过更改的路由信息更新路由表时，版本号就会更改。发送保持连接数据包可确保 BGP 对等体之间的连接处于活动状态。出现错误或特殊情况时将发送通知数据包。

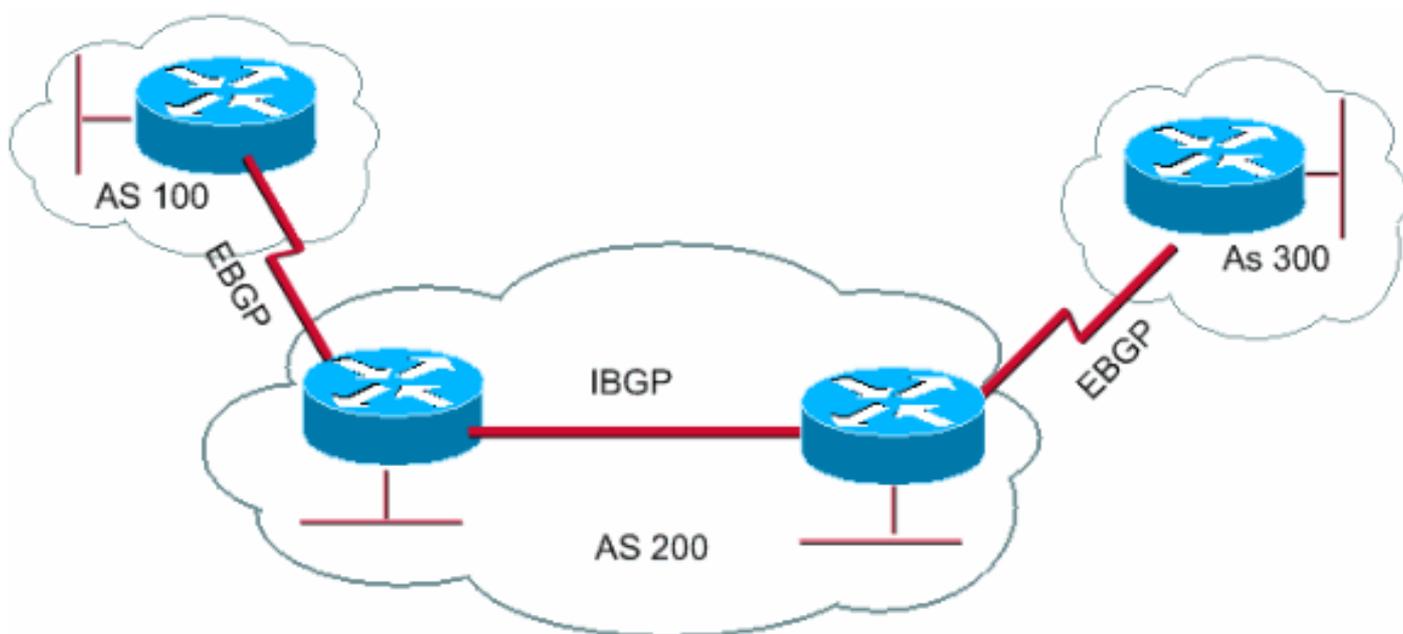
eBGP 和 iBGP

如果一个 AS 有多个 BGP 扬声器，则该 AS 可以充当其他 AS 的传输服务。如本部分中的下图所示，AS200 是 AS100 和 AS300 的传输 AS。

要将信息发送到外部 AS，必须确保网络的可达性。为了保证网络可达性，将进行如下过程：

- 在 AS 内的路由器之间进行内部 BGP (iBGP) 对等
- 将 BGP 信息重分配到 AS 中运行的 IGP

当 BGP 在分属两个不同 AS 的路由器之间运行时，该 BGP 称为外部 BGP (eBGP)。当 BGP 在同一 AS 内的路由器间运行时，该 BGP 称为 iBGP。



BGP在同一AS中的路由器之间运行

启用 BGP 路由

完成以下步骤，以便启用和配置 BGP。

假设您希望使用两个通过 BGP 进行通信的路由器，即 RTA 和 RTB。在第一个示例中，RTA 和 RTB 在不同的 AS 之中。在第二个示例中，两个路由器属于同一个 AS。

1. 定义路由器进程和路由器所属的 AS 编号。

发出以下命令以在路由器上启用 BGP：

```
<#root>  
  
router bgp <autonomous-system>  
  
RTA#  
router bgp 100  
  
RTB#  
router bgp 200
```

这些语句指示 RTA 运行 BGP，并且属于 AS100。RTB 运行 BGP，并且属于 AS200。

2. 定义 BGP 邻居。

BGP 邻居的形成表明路由器尝试通过 BGP 进行通信。下一部分将介绍此过程。

形成 BGP 邻居

两个 BGP 路由器在建立彼此间的 TCP 连接之后互称邻居。为使两个对等路由器启动路由更新交换，TCP 连接是必需的。

TCP 连接启动后，路由器发送开放消息，以便交换各自的值。路由器交换的值包括 AS 编号、路由器所运行的 BGP 的版本、BGP 路由器 ID 以及连接持续时间。在确认并接受这些值之后将建立邻居连接。除 Established 之外的任何状态都表示两个路由器未成为邻居，因而无法交换 BGP 更新。

发出以下 `neighbor` 命令以建立 TCP 连接：

```
<#root>
```

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

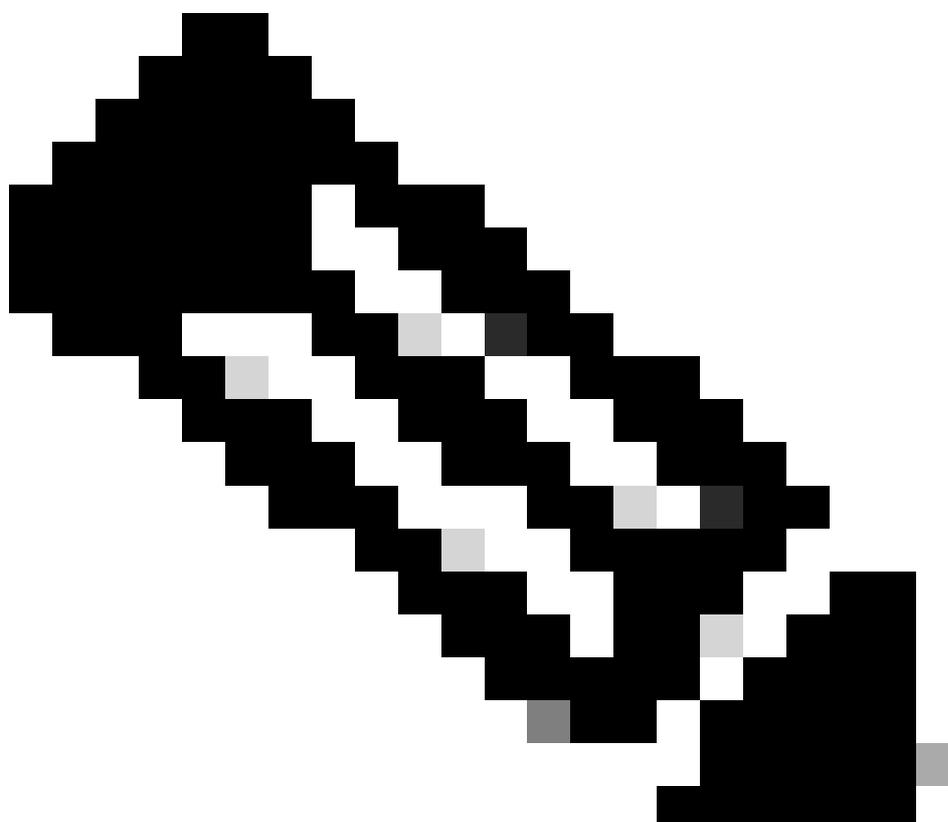
命令中的 number 是希望通过 BGP 进行连接的路由器的 AS 编号。对于 eBGP 来说，ip-address 是直接连接的下一跳地址。对于 iBGP 来说，ip-address 是另一个路由器上的任何 IP 地址。

在对等路由器的 neighbor 命令中使用的两个 IP 地址（子网）必须能够相互访问。验证这两个 IP 地址之间的可达性的一种方法是在它们之间执行扩展 ping 操作。扩展 ping 强制发出 ping 操作的路由器使用 neighbor 命令所指定的 IP 地址作为源。路由器必须使用此地址，而不是发出数据包的接口的 IP 地址。

如果有任何 BGP 配置发生更改，则必须 重置邻居连接以使新参数生效。

•

```
clear ip bgp address
```



注意：地址是邻居地址

-

```
clear ip bgp *
```

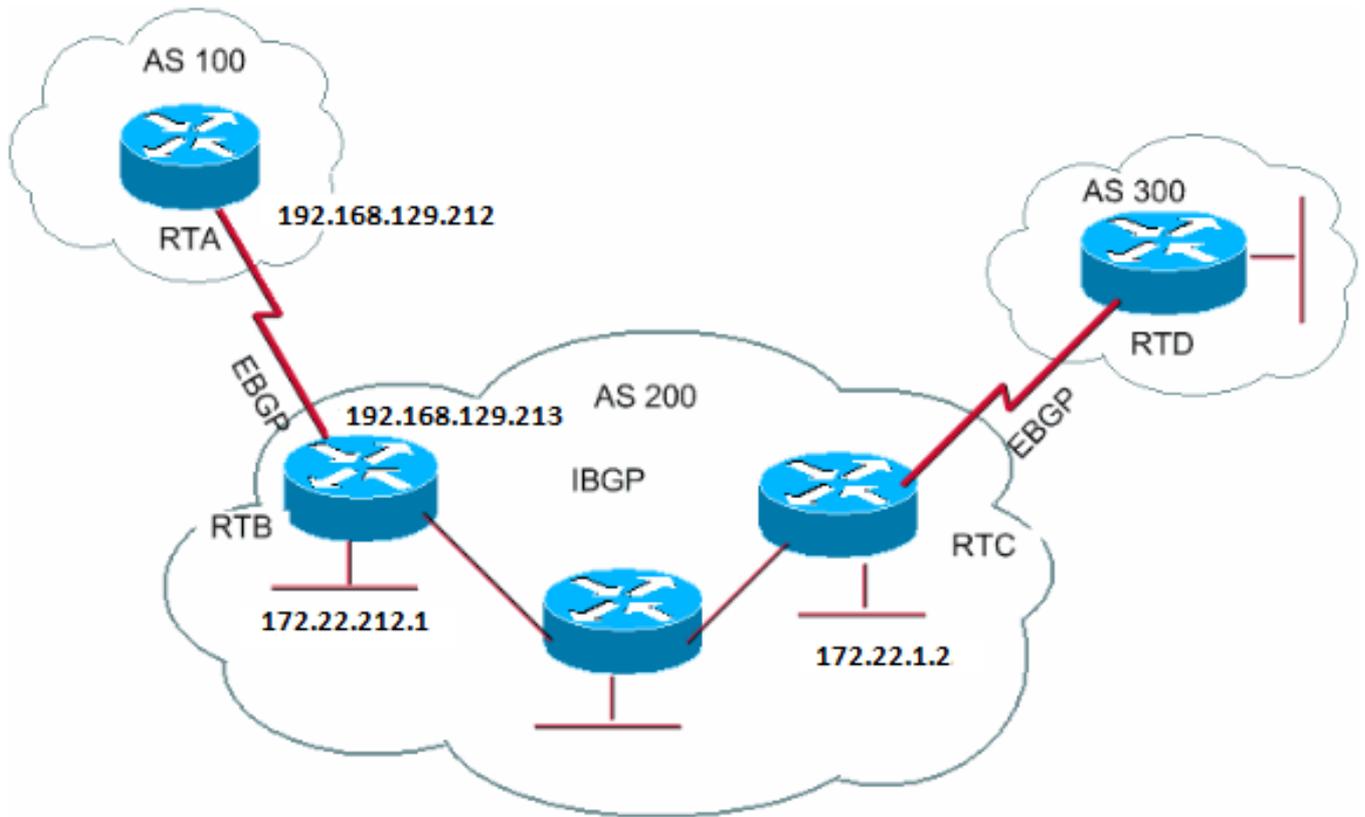
此命令清除所有邻居连接。

默认情况下，将使用 BGP 4 开始 BGP 会话，如有必要可以向下协商到早期版本。您可以禁止协商，强制指定路由器与邻居通信时使用的 BGP 版本。在路由器配置模式下发出以下命令：

```
<#root>
```

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

以下是 neighbor 命令配置的示例：



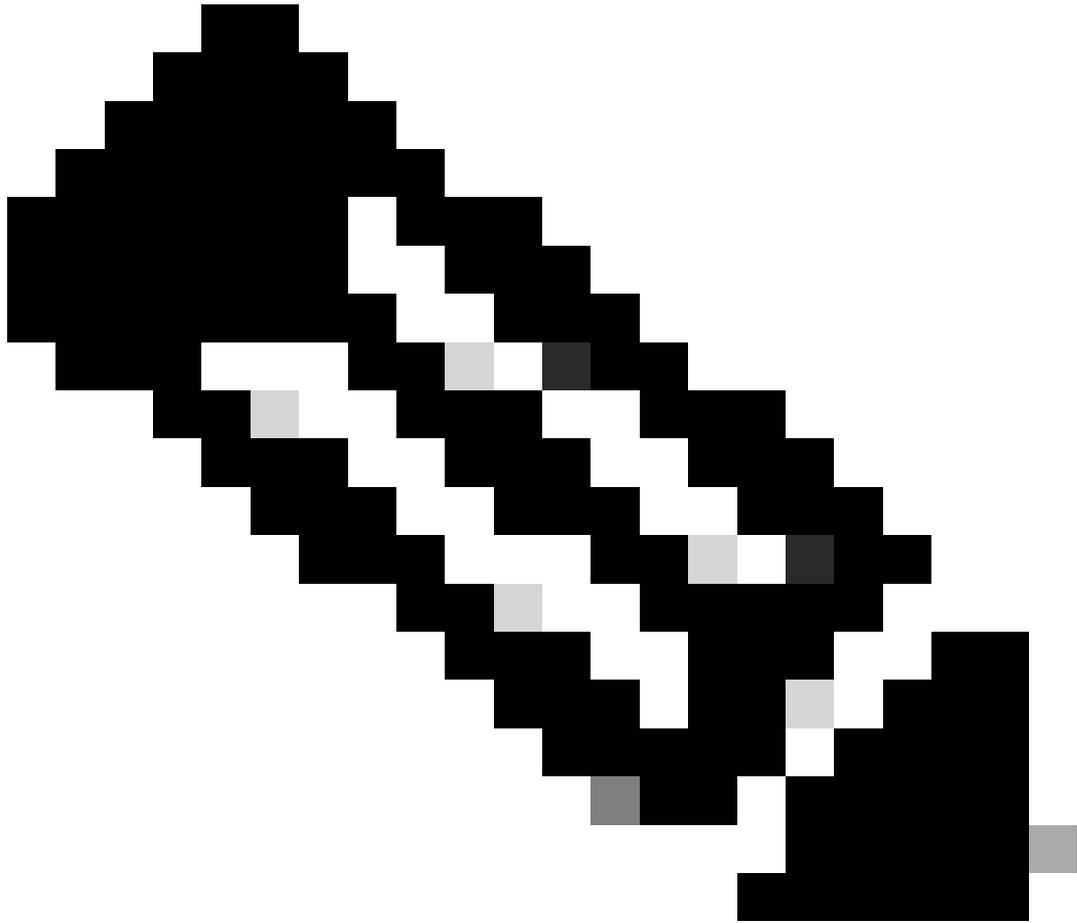
```
RTA#
router bgp 100
neighbor 192.168.129.213 remote-as 200
```

```
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
```

在本示例中，RTA 和 RTB 运行 eBGP。RTB 和 RTC 运行 iBGP。远程 AS 编号指向外部或内部 AS，表示使用 eBGP 或 iBGP。此外，eBGP 对等体有直接连接，但 iBGP 对等体没有直接连接。iBGP 路由器不需要有直接连接。但是，必须运行某些 IGP 并允许两个邻居相互通信。

此部分提供 show ip bgp neighbors 命令所显示信息的一个示例。



注意：请特别注意BGP状态。除Established之外的任何状态都表示对等体未启动。此外，请注意以下项目：

-
- BGP version 4
 - remote router ID

此编号是路由器上的最高 IP 地址或是最高环回接口（如果存在）。

-

table version

table version 提供表的状态。每当新信息传入时，表版本号就会增加。如果版本号持续增加，则表明存在路由抖动导致路由出现持续更新。

```
<#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 和环回接口

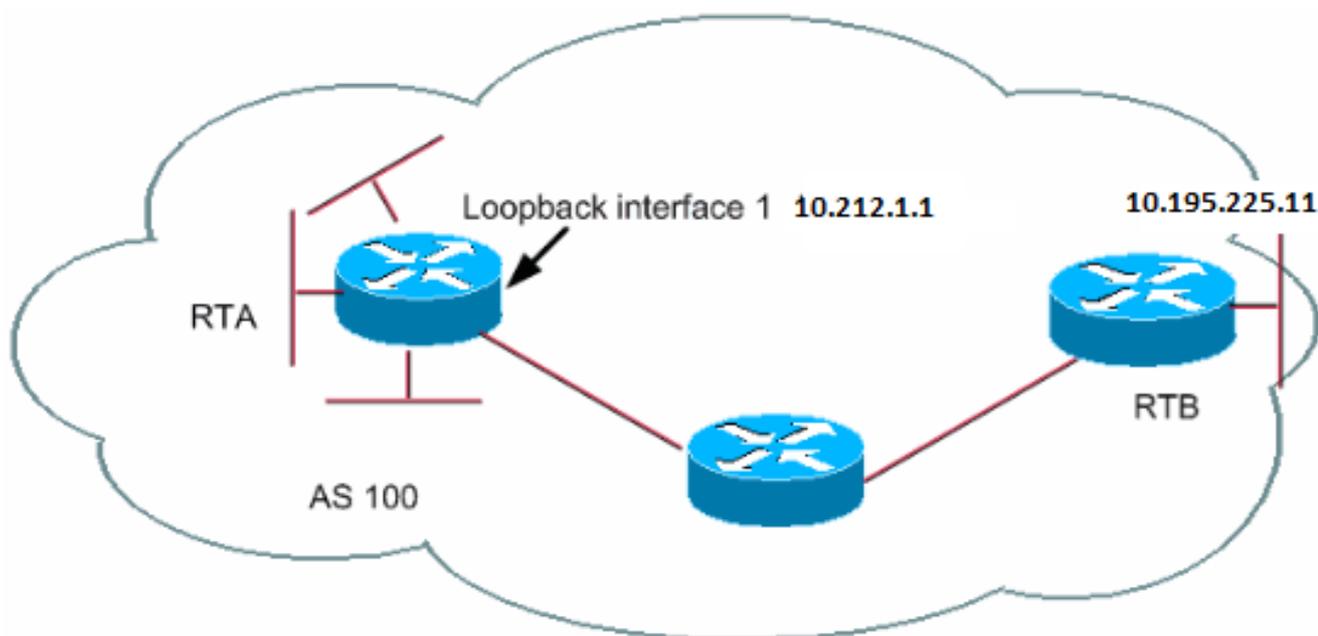
在iBGP中通常使用环回接口定义邻居，但在eBGP中并不常见。通常，使用环回接口可确保邻居的 IP 地址保持启动，并使其不依赖硬件而正常工作。如果是 eBGP，则对等路由器常常直接连接，而不使用环回。

如果在 neighbor 命令中使用环回接口的IP地址，则需要对邻居路由器进行某些额外配置。邻居路由器需要通知 BGP 启用 BGP 邻居 TCP 连接时使用的是环回接口而不是物理接口。为指示使用的是环回接口，可发出以下命令：

```
<#root>
```

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

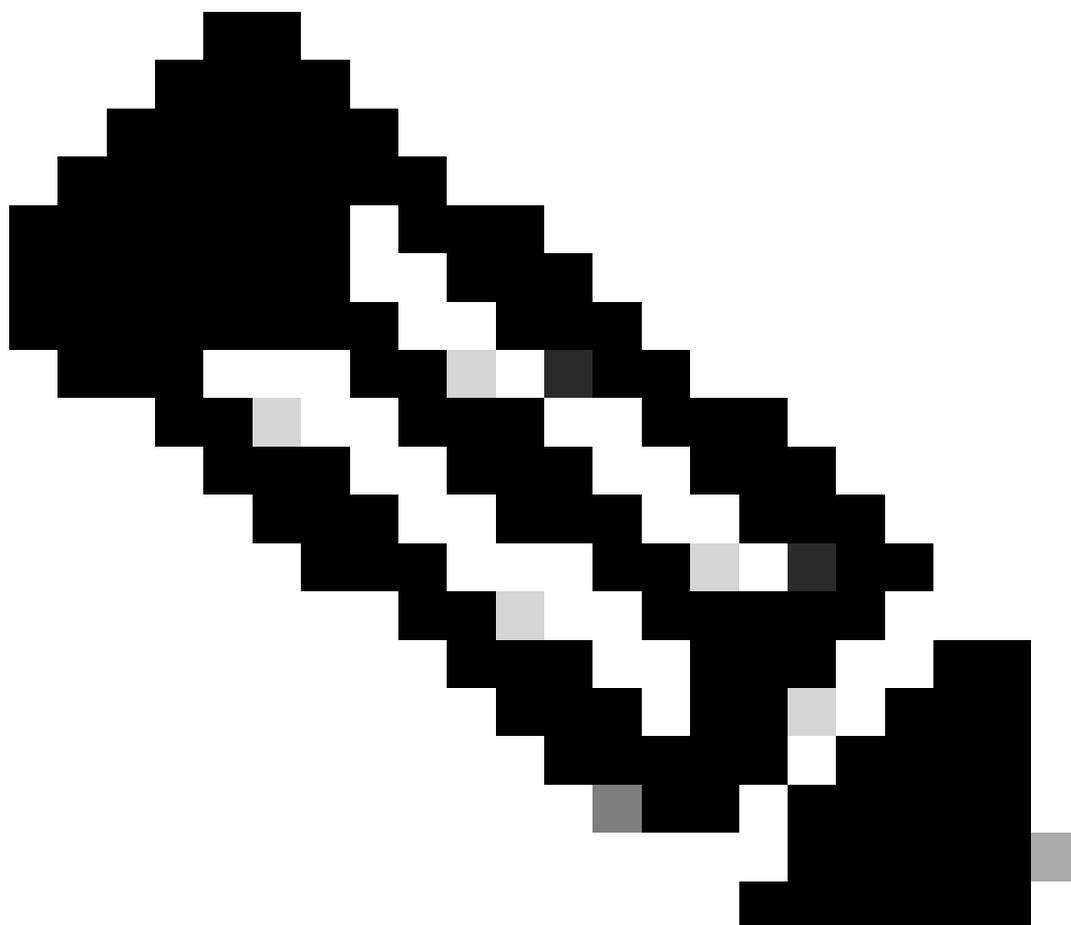
以下示例演示了此命令的用法：



```
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
```

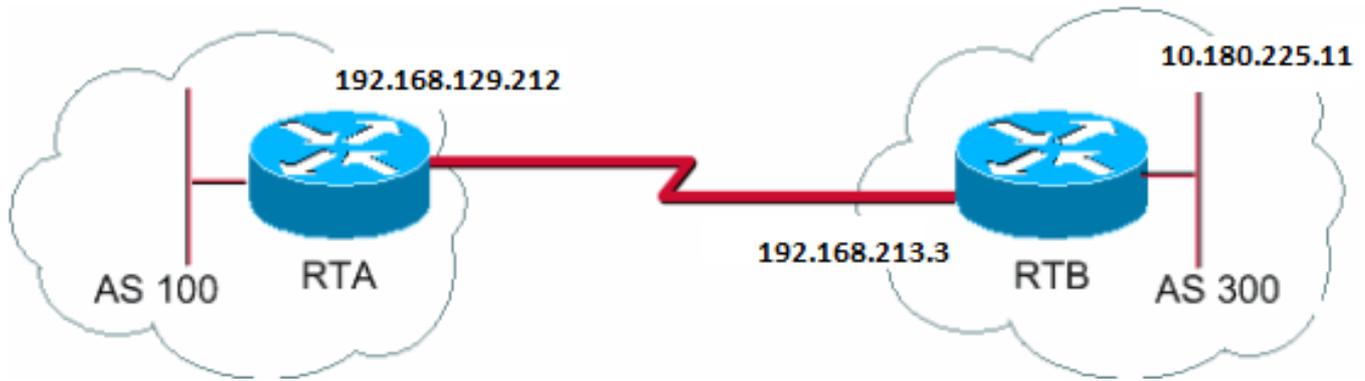
在本示例中，RTA 和 RTB 在 AS100 内运行 iBGP。在 neighbor 命令中，RTB 使用 RTA 的环回接口 10.212.1.1。在这种情况下，RTA 必须强制 BGP 使用环回 IP 地址作为 TCP 邻居连接的源。为强制实施此操作，RTA 添加 `update-source interface-type interface-number` 命令为 `neighbor 10.195.225.11 update-source loopback 1`。此语句将强制 BGP 在 BGP 与邻居 10.195.225.11 通信时使用环回接口的 IP 地址。



注意： RTA 已使用 RTB 的物理接口 IP 地址 10.195.225.11 作为邻居。因为使用的是此 IP 地址，所以无需为 RTB 进行特殊配置。有关完整的网络应用方案示例配置，请参阅使用或不使用环回地址的 iBGP 和 eBGP 示例配置。

eBGP 多跳

有时，Cisco 路由器与不允许直接连接两个外部对等体的第三方路由器之间可以运行 eBGP。可以使用 eBGP 多跳实现连接。eBGP 多跳允许在没有直接连接的两个外部对等体之间形成邻居连接。多跳只适用于 eBGP 而不适用于 iBGP。以下示例演示 eBGP 多跳：



```

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

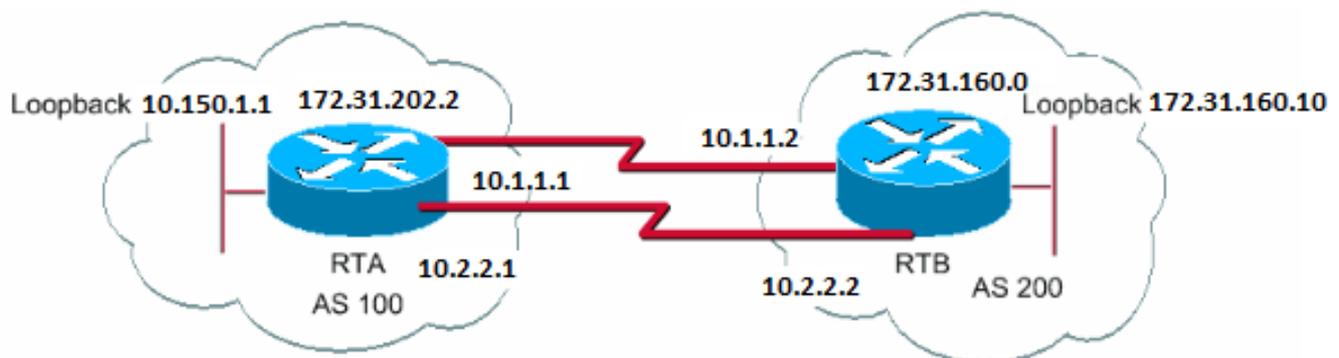
RTB#
router bgp 300
 neighbor 192.168.129.212 remote-as 100

```

RTA 表示没有直接连接的外部邻居。RTA需要指出自己使用neighbor ebgp-multihop命令。另一方面，RTB表示具有直接连接的邻居，即192.168.129.212。因为有此直接连接，RTB不需要neighbor ebgp-multihop 命令。您还必须配置IGP或静态路由以允许没有连接的邻居相互访问。

BGP多跳（负载均衡）部分的示例演示了在并行线路上使用eBGP的情况下如何实现BGP负载均衡。

eBGP 多跳（负载均衡）



```

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

```

此示例演示环回接口、update-source和ebgp-multihop的使用。此示例是使两个 eBGP 扬声器在并行串联线路上实现负载平衡的一种变通方法。在正常情况下，BGP 选取某一条线路发送数据包，此时负载不会平衡。引入环回接口之后，eBGP 的下一跳为环回接口。使用静态路由或 IGP 引入两个开销相等的路径来到达目标。RTA 具有到达下一跳 172.31.160.10 的两个选择：一条路径通过 10.1.1.2，另一条路径通过 10.2.2.2。RTB 同样有两种选择。

路由映射

BGP 大量使用路由映射。在 BGP 上下文中，路由映射是一种用于控制和修改路由信息的方法。通过定义路由从一个路由协议重分配到另一个路由协议的条件，可对路由信息进行控制和修改。或者，可在路由信息传入或传出 BGP 时对其进行修改。路由映射的格式如下：

```
<#root>
```

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

映射标记只是您为路由映射指定的名称。您可以定义同一路由映射或同一名称标签的多个实例。序号只是表明新路由映射在您已用同一名称配置的路由映射的列表中所处的位置。

本例定义了两个名称为 MYMAP 的路由映射实例。第一个实例的序号为 10，第二个实例的序号为 20。

•

```
route-map MYMAP permit 10 (此处为第一组条件。)
```

•

```
route-map MYMAP permit 20 (此处为第二组条件。)
```

将路由映射 MYMAP 应用到传入或传出路由时，将通过实例 10 应用第一组条件。如果不满足第一组条件，则继续使用序号更高的路由映射实例。

match 和 set 配置命令

每个路由映射包含一个 match 和 set 配置命令列表。match 指定条 match 件，set 在满足 match 命令强制实施的条件时指定 set 操作。

例如，您可以定义一个路由映射来检查传出的更新。如果有与 IP 地址 10.1.1.1 匹配的项，则将该更新的度量设置为 5。以下命令演示了该示例：

```
<#root>
```

```
match ip address 10.1.1.1
```

```
set metric 5
```

现在，如果满足匹配条件并且得到 permit，则会像设置操作所指定的那样重分配或控制路由。您将跳出列表。

如果满足匹配条件并且得到 deny，则不会重分配或控制路由。您将跳出列表。

如果不满足匹配条件，并且得到 permit 或 deny，则检查路由映射的下一个实例。例如，检查实例 20。这种对下一个实例的检查将持

续下去，直到跳出列表或检查完路由映射的所有实例。如果完成列表检查，并且未找到匹配项，则路由为not accepted nor forwarded。

在早于Cisco IOS软件版本11.2的Cisco IOS®软件版本中，当使用路由映射过滤BGP更新，而不是在协议之间进行重分配时，您无法在IP地址上使用**match** 命令时进行入站过滤。可接受针对出站更新的过滤。Cisco IOS 软件版本 11.2 以及更高版本没有此限制。

match 的相关命令有：

-

match as-path

-

match community

-

match cllns

-

match interface

-

match ip address

-

match ip next hop

-

match ip route-source

-

match metric

-

match route-type

-

match tag

set

的相关命令有：

-

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

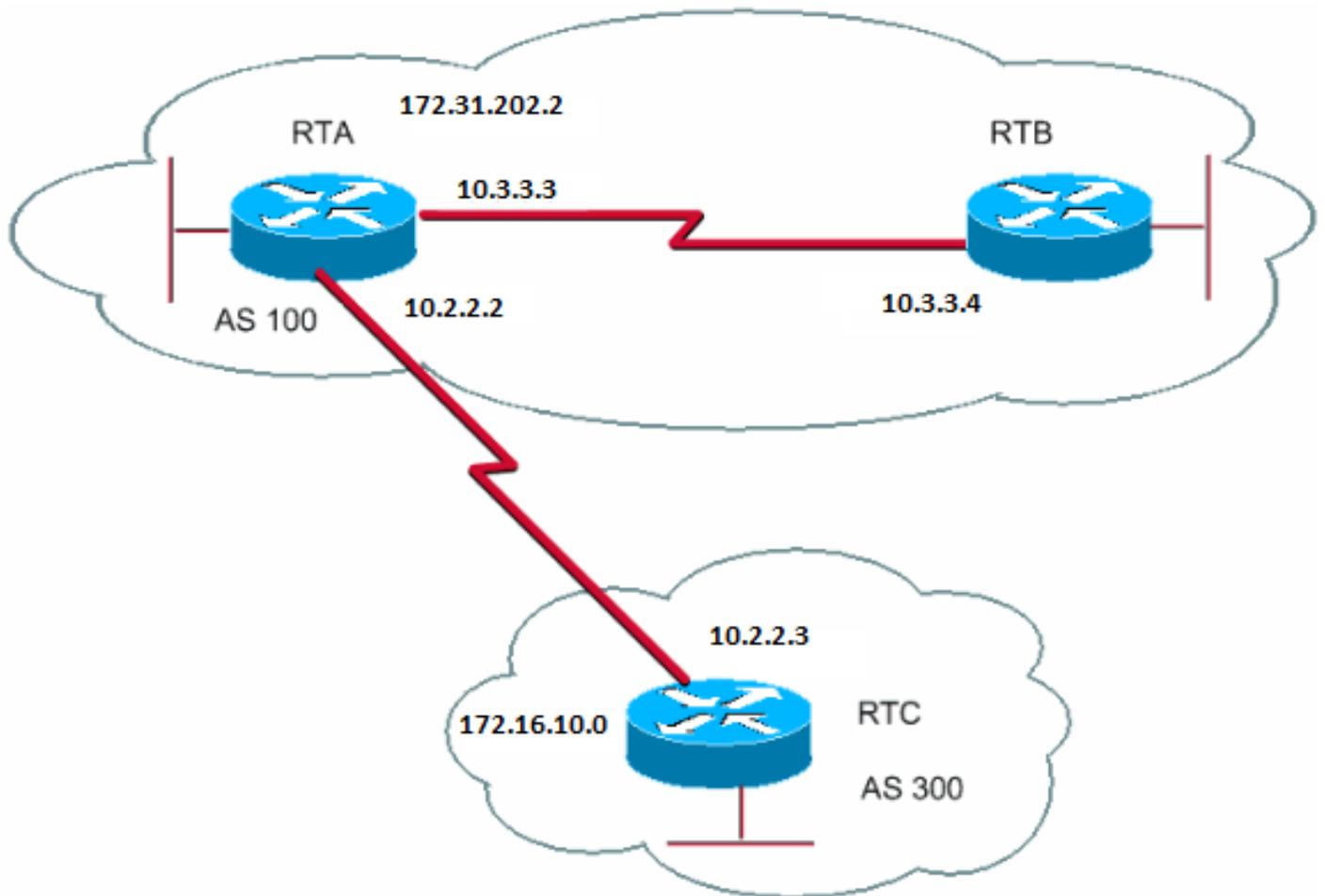
-

set tag

-

set weight

以下是一些路由映射示例：



路由映射示例

示例 1

假设 RTA 和 RTB 运行路由信息协议 (RIP)，RTA 和 RTC 运行 BGP。RTA 通过 BGP 获取更新并将更新重分配到 RIP。假设 RTA 要将度量为 2 的约 172.16.10.0 路由和度量为 5 的所有其他路由重分配到 RTB 路由。在这种情况下，您可以使用此配置：

```

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

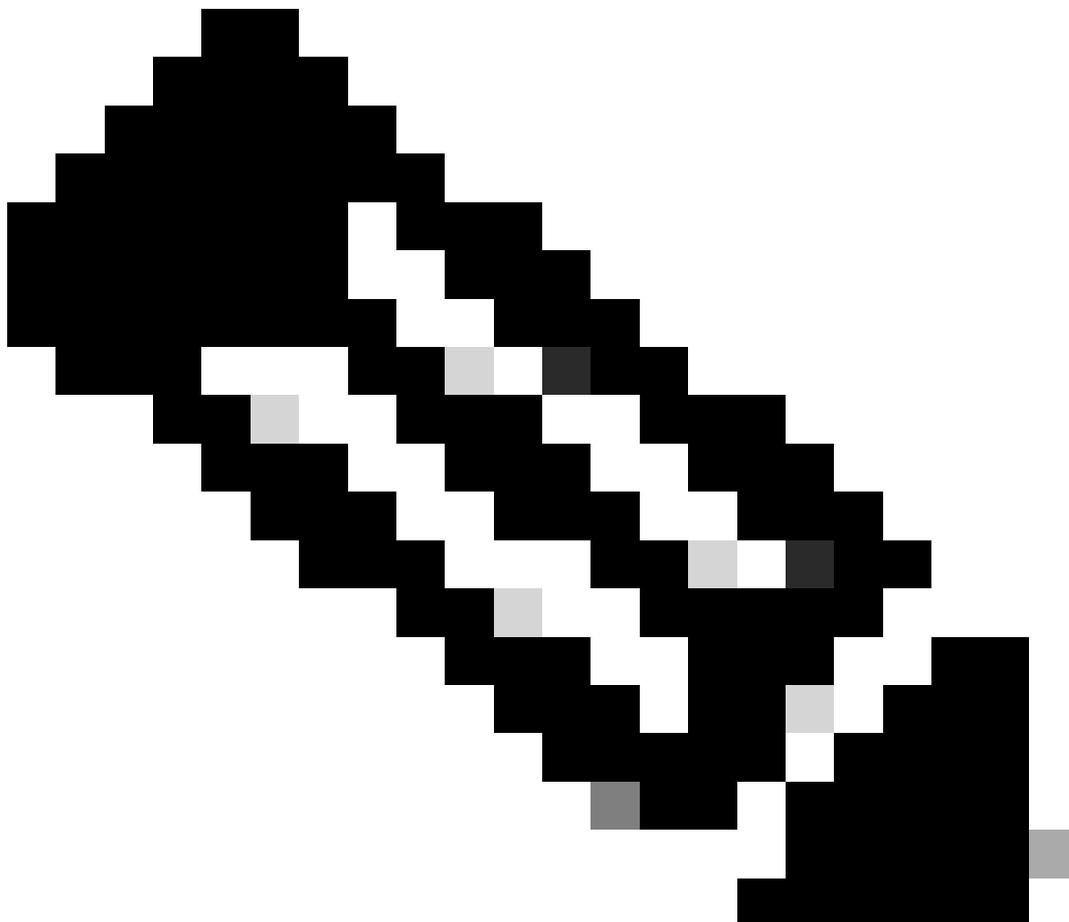
route-map SETMETRIC permit 10
match ip-address 1
set metric 2

route-map SETMETRIC permit 20
set metric 5

access-list 1 permit 172.16.10.0 0.0.255.255

```

在本示例中，如果路由与 IP 地址 172.16.10.0 相匹配，则该路由将具有度量 2。然后跳出路由映射列表。如果没有匹配项，则继续路由映射列表，表明其他所有项均设置为度量5。



注意：请始终询问“不匹配任何匹配语句的路由会发生什么情况？”默认情况下，将丢弃这些路由。

示例 2

假设在示例1中，您不希望AS100接受有关172.16.10.0的更新。当根据 IP 地址进行匹配时，无法针对入站更新应用路由映射。因此，必须对 RTC 使用出站路由映射：

```
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
```

至此，您已经比较熟悉如何启动 BGP 以及如何定义邻居，现在看一下如何开始进行网络信息交换。

使用 BGP 发送网络信息有多种途径。以下几部分内容逐一介绍了这些方法：

-

network 命令

-

重分发

-

静态路由和重分配

network 命令

用于标识网络地址的命令如下：network

<#root>

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

network 命令控制源于此的网络。此概念不同于使用 Interior Gateway Routing Protocol (IGRP) 和 RIP 的常用配置。此命令不尝试针对某个接口运行 BGP，相反，您尝试向 BGP 指示 BGP 必须源于此盒的网络。该命令使用掩码部分，因为 BGP 版本 4 (BGP4) 可以处理子网和超网。最多可以接受 200 条 network 命令。

如果路由器知道您尝试通告的网络（无论网络状态是已连接、静态，还是动态获知），network 命令就会起作用。

以下是一个 network 命令示例：

```
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
```

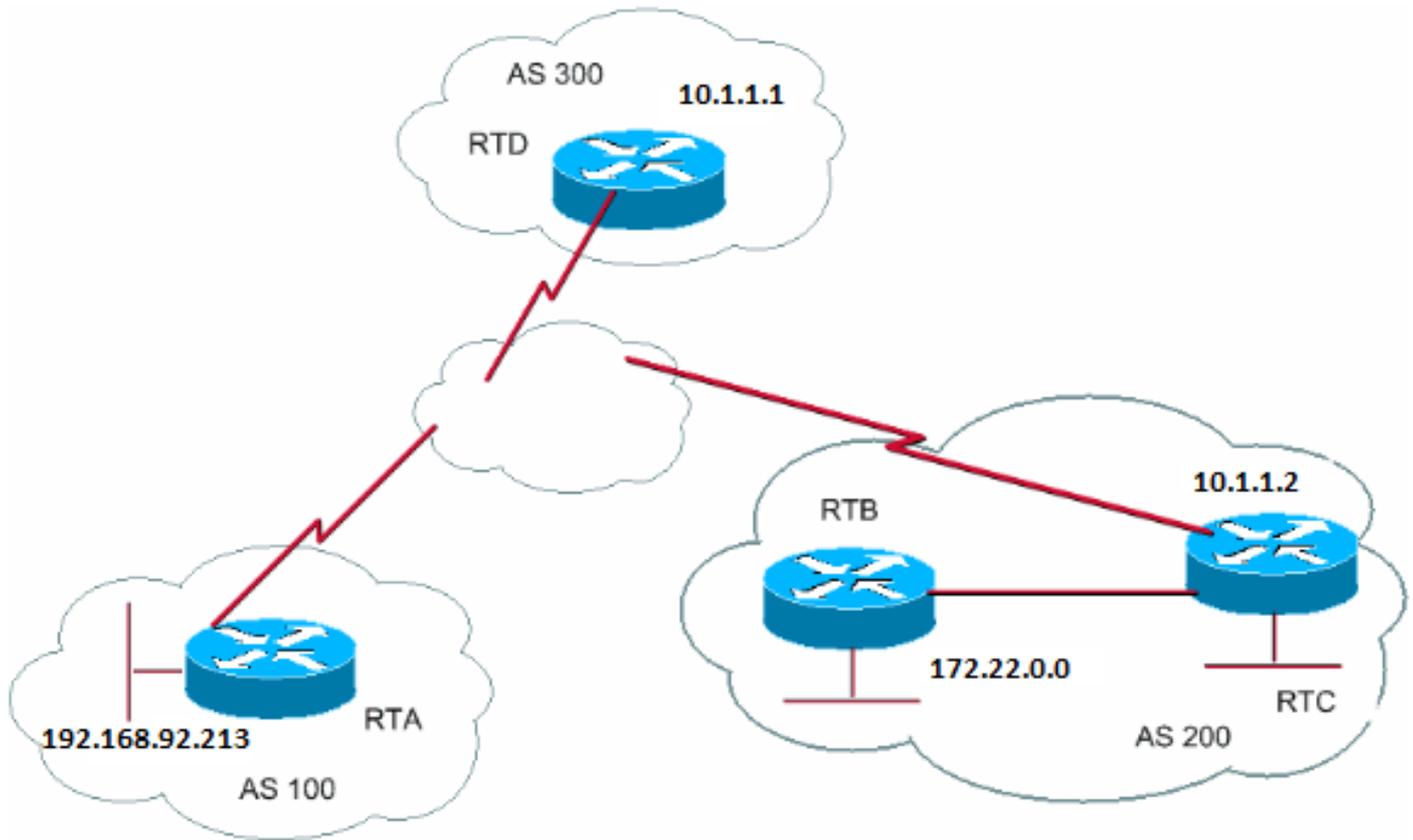
此示例指示路由器 A 为 192.168.213.0/16 生成一个网络条目。/16 表示使用 C 类地址的超网，通告前两个八进制数，即前 16 位。

注意：您需要静态路由来使路由器生成192.168.213.0，因为静态路由在路由表中放置了一个匹配的条目。

重分发

`network` 命令是通过BGP通告网络的一种方式。另一种方式是将IGP重分配到BGP中。IGP可以是IGRP、开放最短路径优先(OSPF)协议、RIP、增强型内部网关路由协议(EIGRP)或其他协议。此重分发看起来很可怕，因为现在您将所有内部路由转储到BGP中；其中一些路由可以通过BGP获知，您不需要再次发送它们。在过滤时请小心，确保将您想通告的仅互联网路由发送到，而不是发送到您拥有的所有路由。下面是一个示例。

RTA 公告 192.168.92.213，RTC 公告 172.22.0.0。查看 RTC 配置：



如果发出network命令，将得到如下结果：

```
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.

如果改用重分配，则将得到如下结果：

```
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.
```

此重分配将导致从您的 AS 发源 192.168.92.213。您不是192.168.92.213的源；AS100是源。因此，您必须使用过滤器阻止AS将源地地址从该网络中移除。正确的配置如下：

```
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
```

使用access-list命令控制源于AS200的网络。

将 OSPF 重分配到 BGP 与对其他 IGP 的重分配略有不同。redistribute ospf 1 under router bgp这一简单问题不起作用。重分配各自的路由需要使用特定关键字如internal、external和nssa-external 等。有关详细信息，请参阅[了解将OSPF路由重分配到BGP](#)。

静态路由和重分配

始终可以使用静态路由生成网络或子网。唯一的区别是 BGP 会将这些路由的源视为不完整的或未知的。可以使用如下方法获得与 Redistribution部分中的示例相同的结果：

```
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

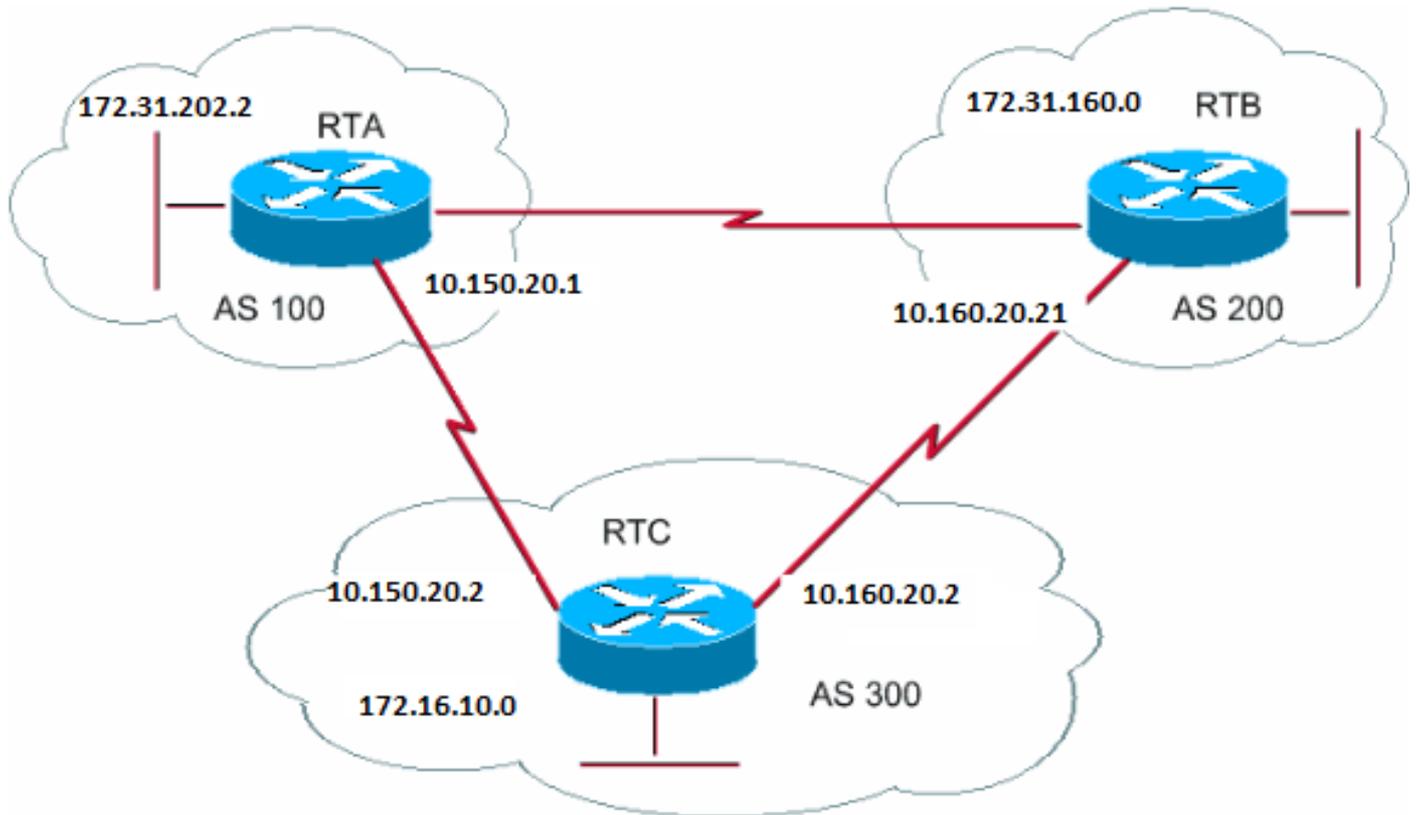
ip route 172.22.0.0 255.255.255.0 null0
```

接null0 口表示忽略该数据包。因此，如果您获得了数据包，并且存在比172.22.0.0更精确的匹配，则路由器会将数据包发送到特定的匹配。否则，路由器将忽略该数据包。这是通告超网的一个好方法。

本文讨论如何使用不同的方法从您的 AS 生成路由。请记住，生成这些路由时，还有其他一些 BGP 路由是 BGP 通过邻居（内部或外部）获知的。BGP 将 BGP 获知的信息从一个对等体传送到另一个对等体。区别在于源自 network 命令、重分配或静态路由的路由表明 AS 是这些网络的源。

重分配始终可以作为将 BGP 注入 IGP 的方法。

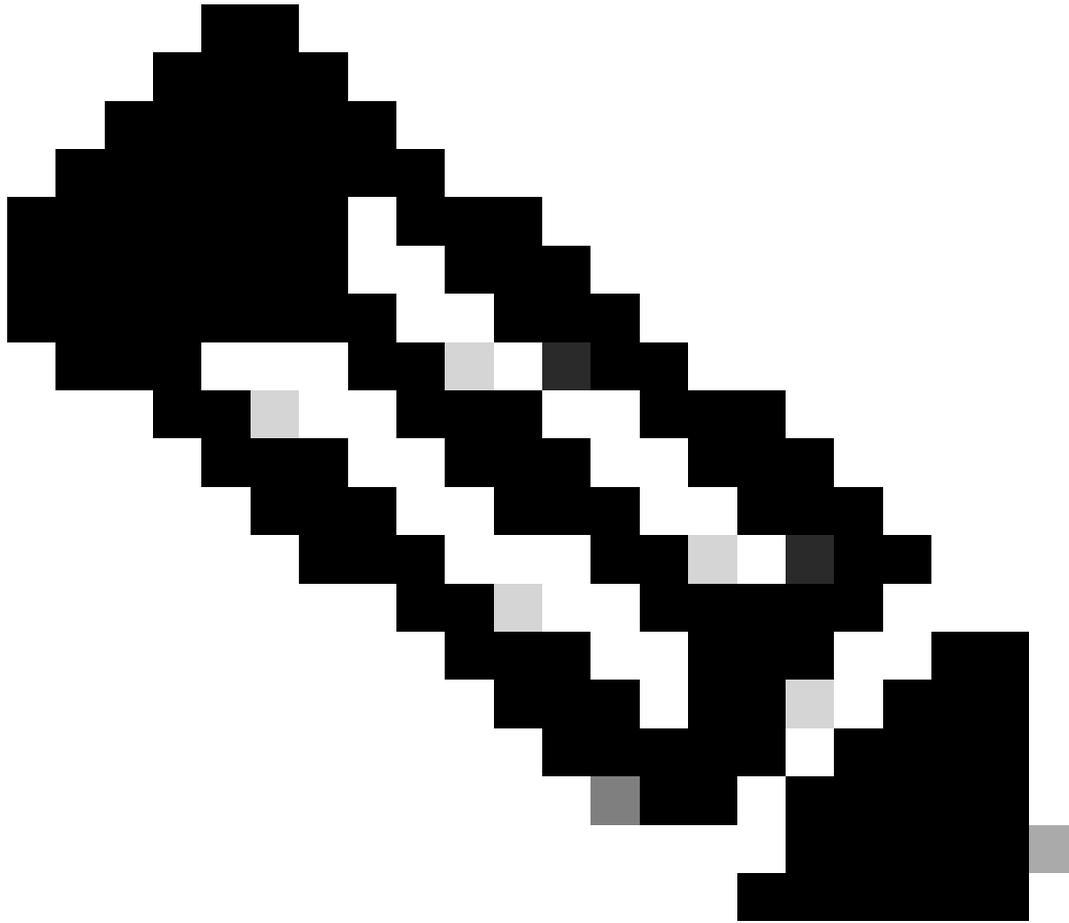
例如：



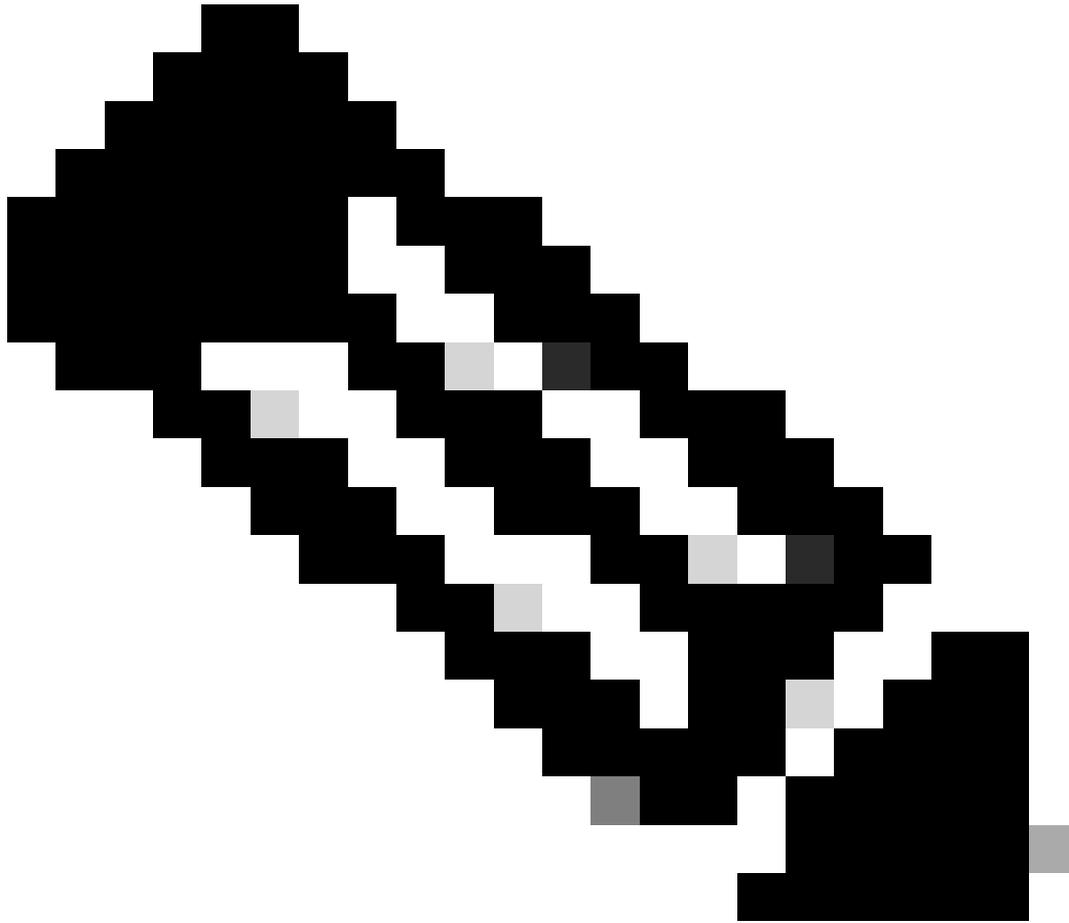
```
RTA#  
router bgp 100  
neighbor 10.150.20.2 remote-as 300  
network 172.31.202.2
```

```
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
```



注意：在RTC中不需要网络172.31.202.2或网络172.31.160.0，除非您希望RTC生成这些网络，并在这些网络从AS100和AS200传入时传递这些网络。区别同样是network命令为这些网络添加一个额外的通告，表明AS300也是这些路由的源。

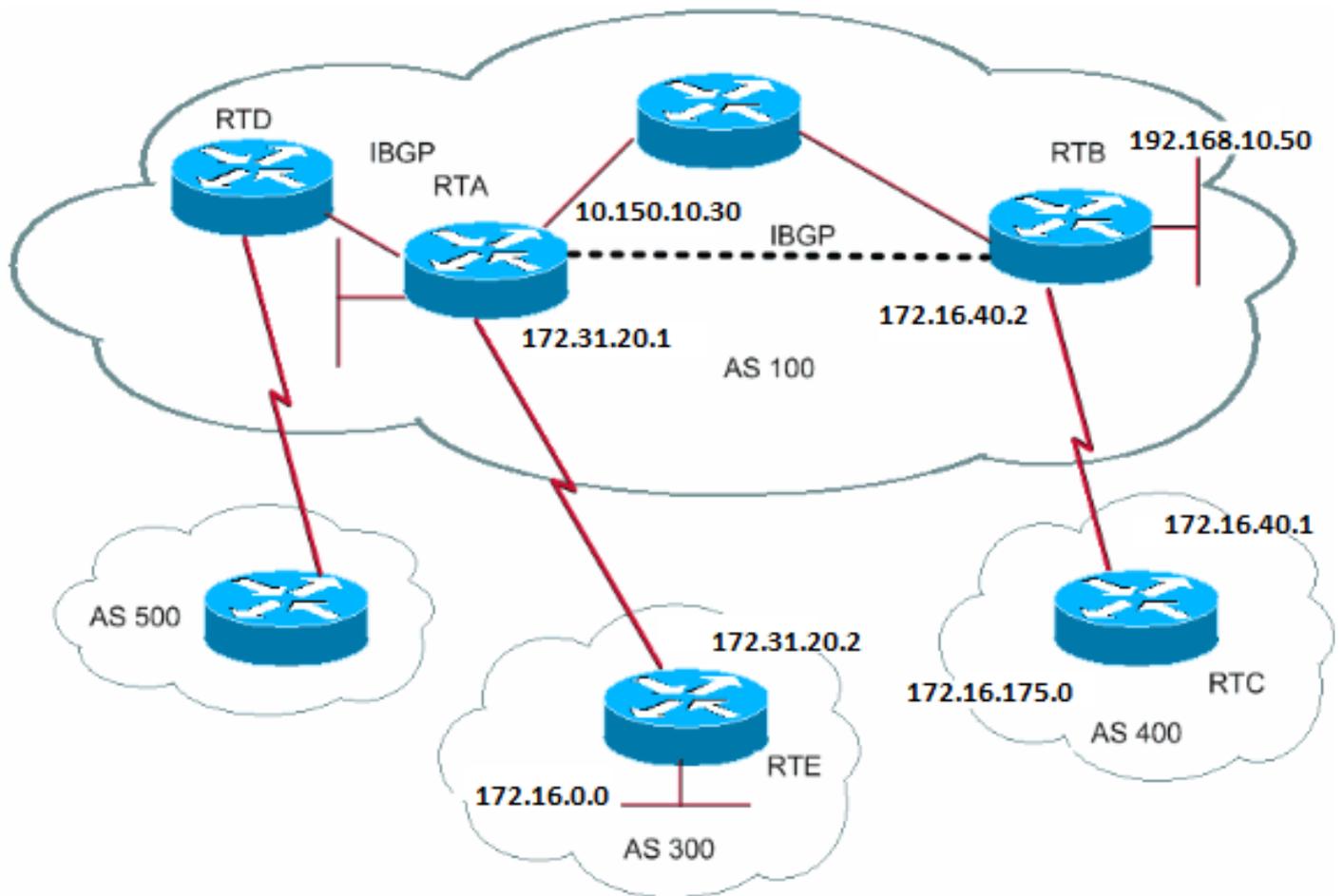


注意：请记住，BGP不接受源于其自身AS的更新。拒绝这样的更新可为无环路域间拓扑提供保证。

例如，假设本部分示例中的 AS200 与 AS100 有直接 BGP 连接。RTA生成路由172.31.202.2并将该路由发送到AS300。然后，RTC 将该路由由传递到 AS200，源保持为 AS100。RTB将172.31.202.2传递到AS100，而原点仍为AS100。RTA 注意到更新源于其自身的 AS，并且忽略该更新。

iBGP

如果AS要充当到其他AS的传输系统，则使用iBGP。如果通过eBGP学习，您可以执行相同操作，重新分发到IGP，然后再重新分发到另一个AS。但iBGP提供了在AS内交换信息的更灵活、更有效的方式。例如，iBGP 可以使用本地优先级控制退出 AS 的最佳退出点。“本地优先级属性”部分提供有关本地优先级的更多信息。



```

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

```

```

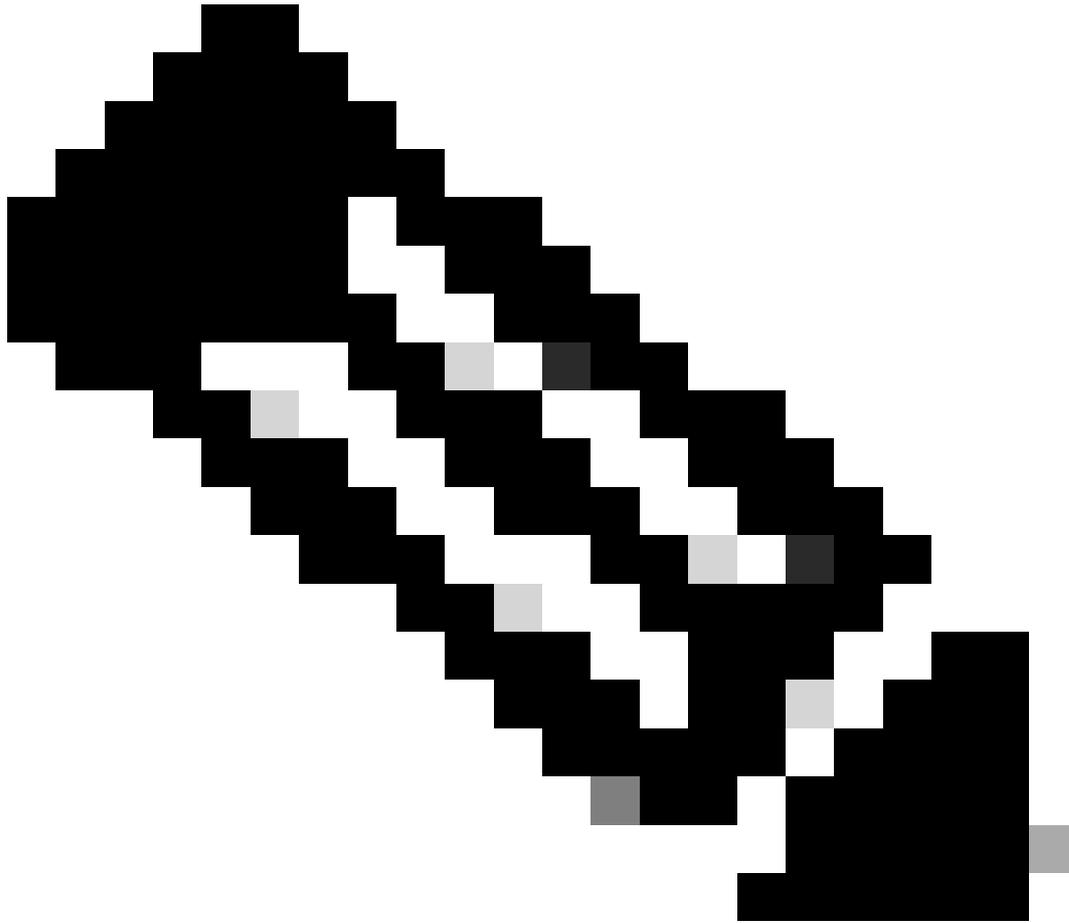
RTB#
router bgp 100
neighbor 10.150.10.30 remote-as 100
neighbor 172.16.40.1 remote-as 400
network 192.168.10.150

```

```

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

```



注意：请记住，当BGP扬声器收到来自其自身AS (iBGP)中的其他BGP扬声器的更新时，接收更新的BGP扬声器不会将该信息重新分发到其自身AS中的其他BGP扬声器。收到更新的 BGP 扬声器会将该信息重分配到自身 AS 之外的其他 BGP 扬声器。因此，AS 内的 iBGP 之间保持了完整的网格。

RTA和RTB运行iBGP。RTA 和 RTD 也运行 iBGP。从 RTB 到 RTA 的 BGP 更新传输到该 AS 之外的 RTE。该更新不传输到该 AS 内的 RTD。因此，应在 RTB 和 RTD 之间建立 iBGP 对等，以免中断更新流。

BGP 决策算法

BGP 从不同的自治系统接收到关于不同目的地的更新之后，该协议必须选择到达特定目的地的路径。BGP 只选择一条到达特定目的地的路径。

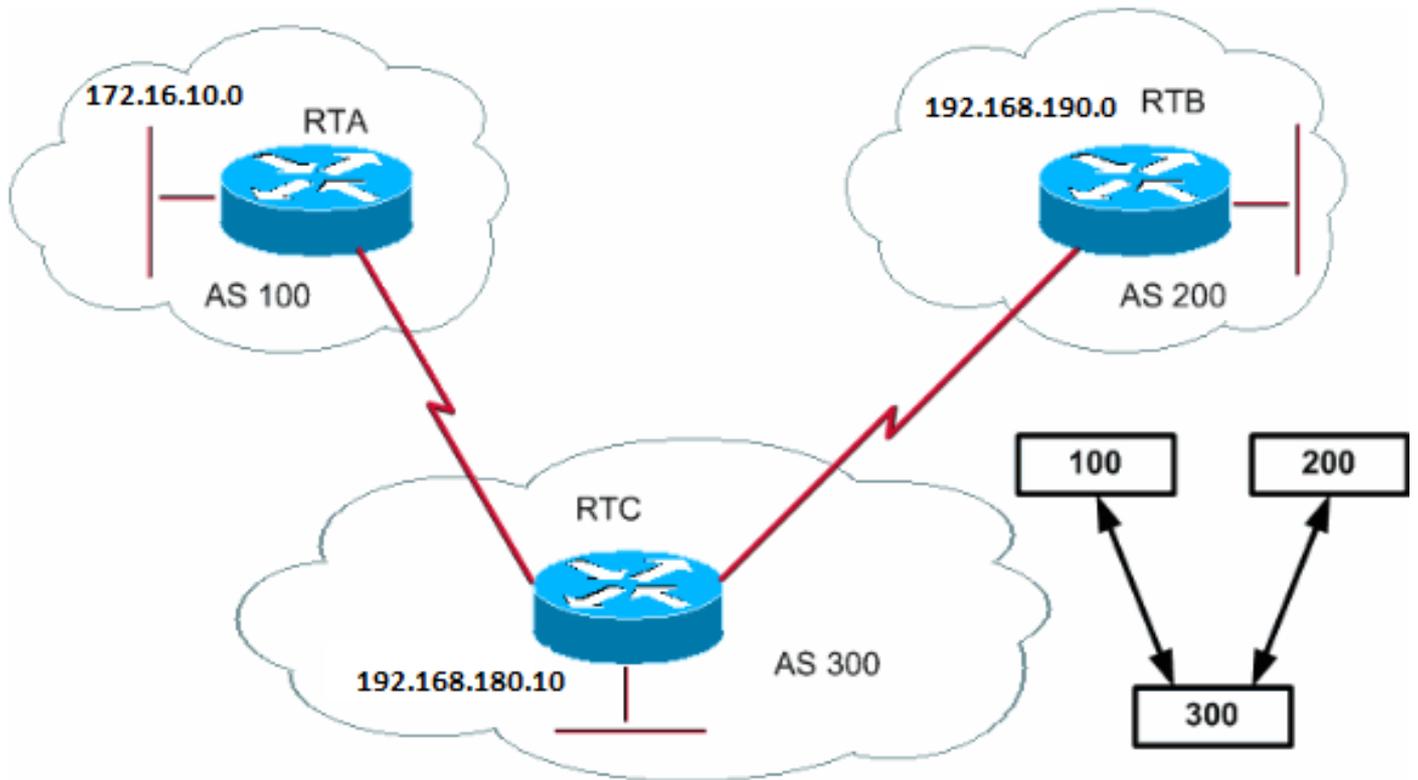
BGP根据不同的attributes做出决策，例如下一跳、管理权重、本地优先级、路由源、路径长度、源代码、度量以及其他属性。

BGP总是将最佳路径传播给邻居。有关详细信息，请参阅[BGP最佳路径选择算法](#)。

下一节将介绍这些属性及其用法。

BGP 案例分析 2

AS_PATH 属性



每当某路由更新传过一个 AS 时，该更新前面便会附加该 AS 编号。AS_PATH 属性实际上是路由为到达目的地而遍历过的 AS 的编号列表。AS_SET 是已遍历的所有 AS 的有序算法集 {}。本文档的CIDR示例2 (as-set)部分提供了一个AS_SET示例。

在本部分的示例中，RTB在AS200中通告网络192.168.190.0。当该路由遍历 AS300 时，RTC 将其自身的 AS 编号附加到网络。当 192.168.190.0到达RTA时，网络附带有两个AS编号：第一个200，然后是300。对于 RTA，到达 192.168.190.0 的路径是 (300, 200)。

相同的过程也应用于 172.16.10.0 和 192.168.180.10。RTB必须采用路径(300, 100)；RTB遍历AS300，然后遍历AS100才能到达 172.16.10.0。为到达 192.168.190.0，RTC 必须遍历路径 (200)；为到达 172.16.10.0，必须遍历路径 (100)。

源属性

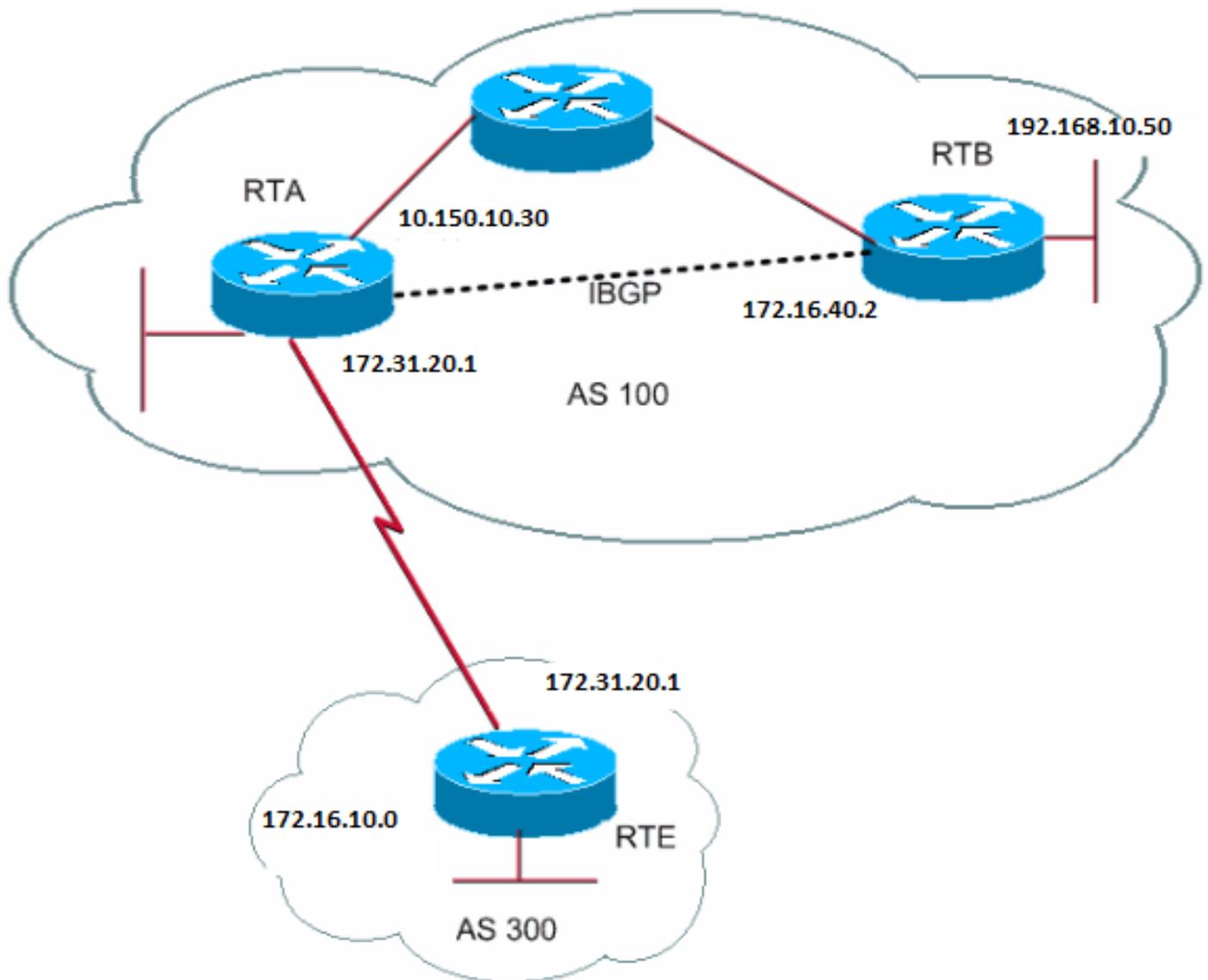
源是定义路径信息的源的必需属性。源属性可以采用以下三个值：

•

IGP — 网络层可达性信息 (NLRI) 是源 AS 的内部信息。发出 `bgp network`命令时通常使用该值。BGP表中的AnyIn表示IGP。

• EGP — NLRI 通过外部网关协议 (EGP) 获知。BGP表中的另一个表示EGP。

• INCOMPLETE — NLRI 为未知或通过其他途径获知。当将路由从其他路由协议重分配到 BGP，并且路由的源不完整时，通常会产生 INCOMPLETE。BGP表中的 ? 表示不完整。



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

```
ip route 192.168.190.0 255.255.0.0 null0
```

RTB#

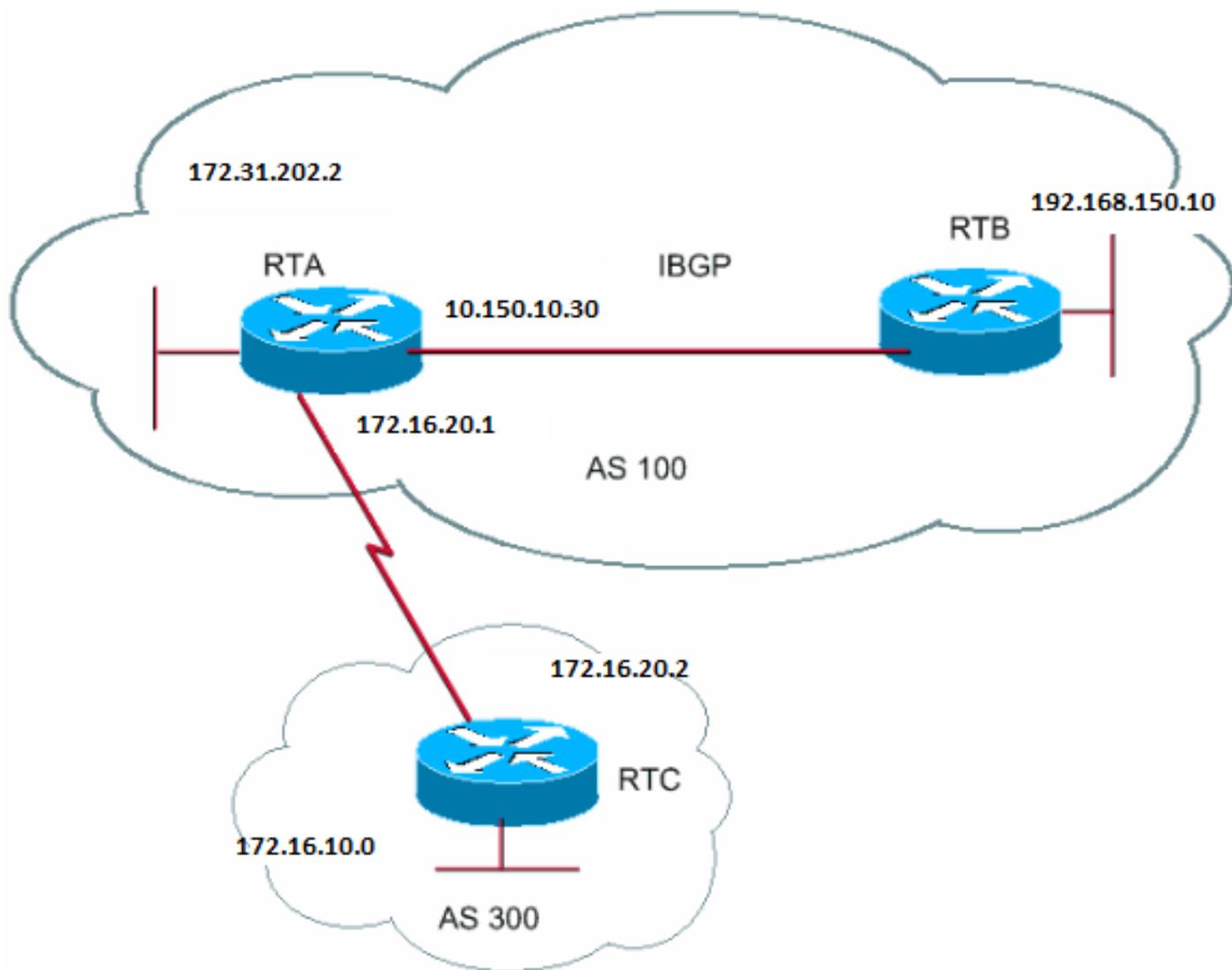
```
router bgp 100  
neighbor 10.150.10.30 remote-as 100  
network 192.168.10.150
```

RTE#

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

RTA 经由 300 i 到达 172.16.10.0。“300 i”表示下一个 AS 路径为 300，路由的源为 IGP。RTA 也通过 i 到达 192.168.10.150。这个“i”表示该项位于同一个 AS 中，且源为 IGP。RTE 经由 100 i 到达 172.31.202.2。“100 i”表示下一个 AS 为 100，且源为 IGP。RTE 也经由 100 ? 到达 192.168.190.0。“100 ?”表示下一个 AS 为 100，且来源不完整且来自静态路由。

BGP 下一跳属性



BGP 下一跳属性

BGP 下一跳属性是为到达特定目的地而使用的下一跳 IP 地址。

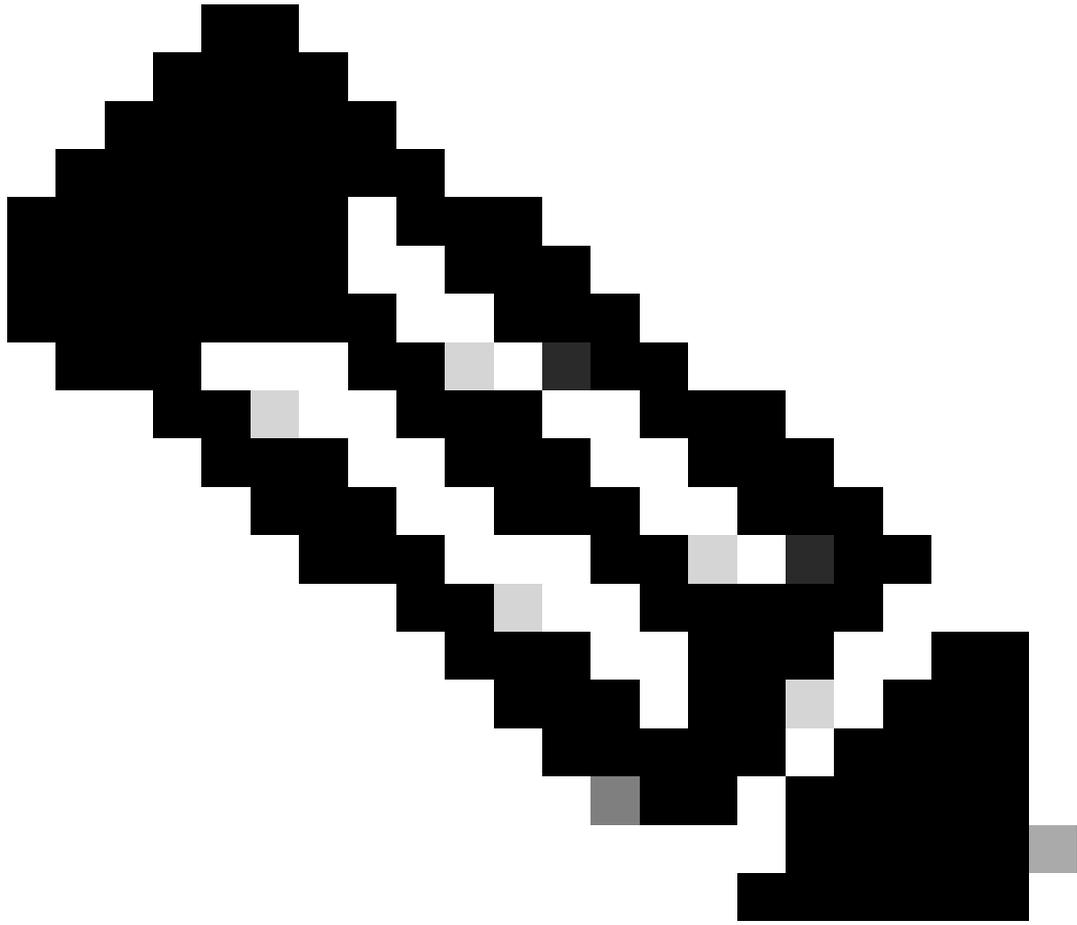
对于eBGP，下一跳总是neighbor命令指定的邻居的IP地址。在本部分的示例中，RTC将172.16.10.0通告给下一跳为172.31.20.2的RTA。RTA将172.31.20.2通告给下一跳为172.31.20.1的RTC。对于iBGP，协议规定eBGP通告的下一跳必须传输到iBGP中。因为有此规则，RTA将172.16.10.0通告到下一跳为172.31.20.2的iBGP对等体RTB。根据RTB，到达172.16.10.0的下一跳为172.31.20.2，而不是10.150.10.30。

应确保RTB可以通过IGP到达172.31.20.2。否则，RTB将丢弃目的地为172.16.10.0的数据包，因为下一跳地址不可访问。例如，如果RTB运行iGRP，还可以针对RTA网络172.16.10.0运行iGRP。您希望在到RTC的链路中将iGRP设为被动，以便只交换BGP。

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

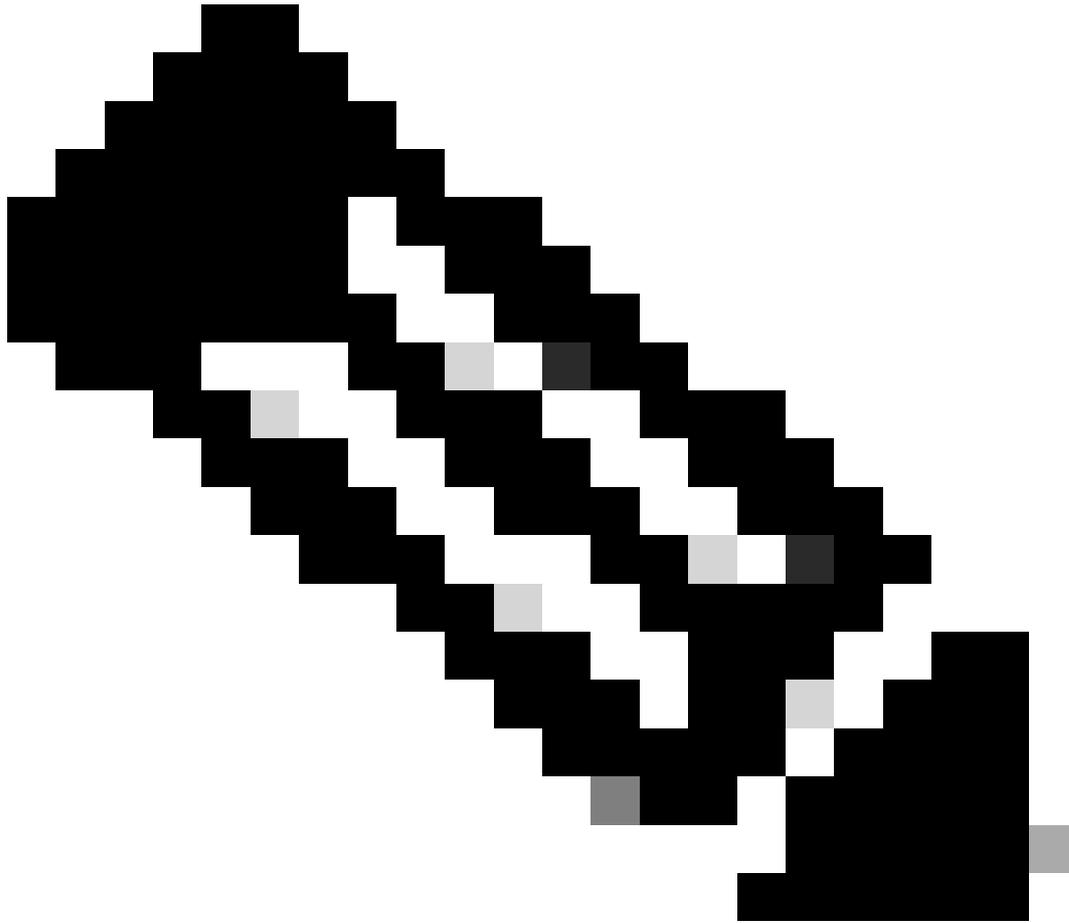
```
RTB#
router bgp 100
  neighbor 10.150.10.30 remote-as 100
```

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



注意：RTC将172.16.10.0通告给RTA，下一跳等于172.31.20.2。

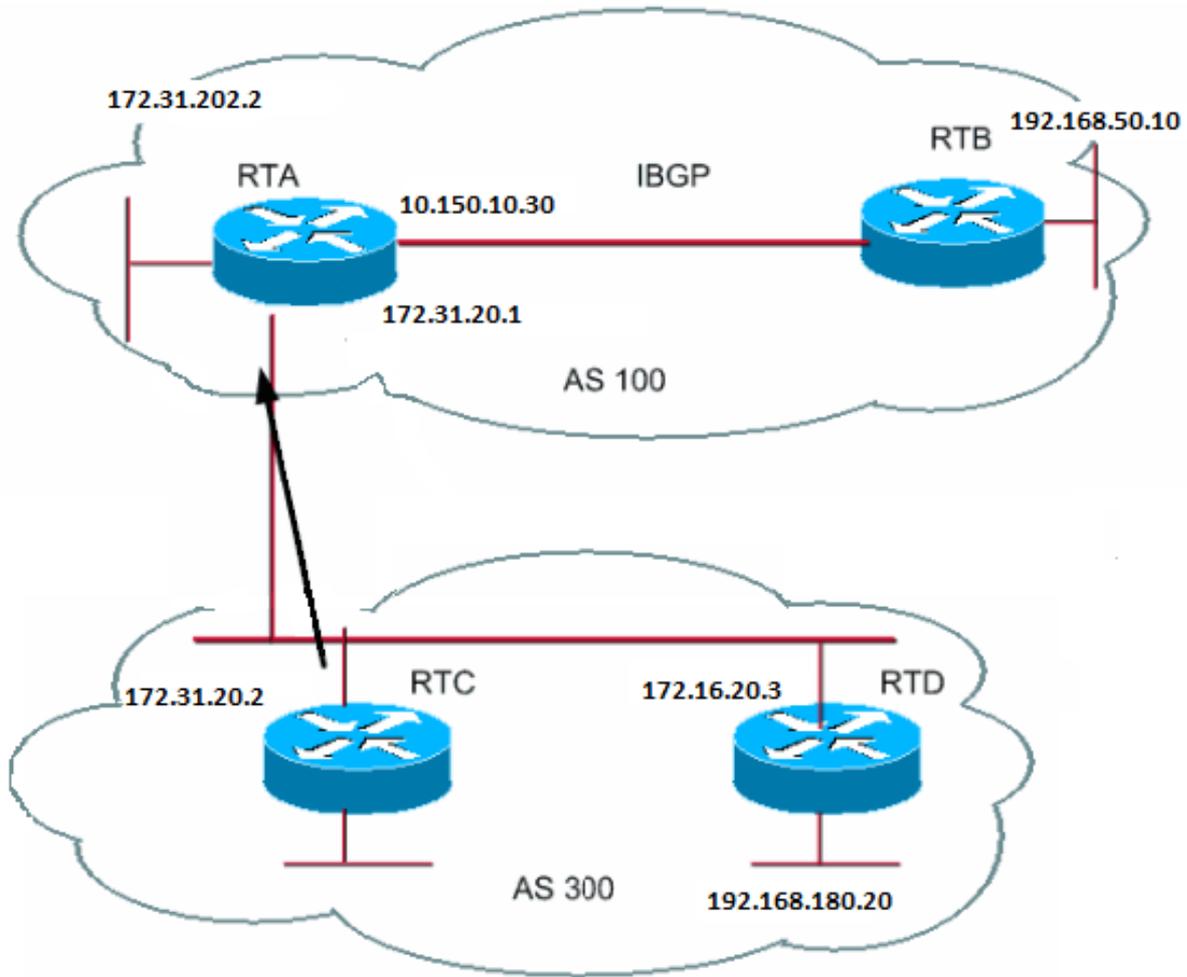




注意：RTA将172.16.10.0通告给RTB，下一跳等于172.31.20.2。eBGP 下一跳在 iBGP 中传递。

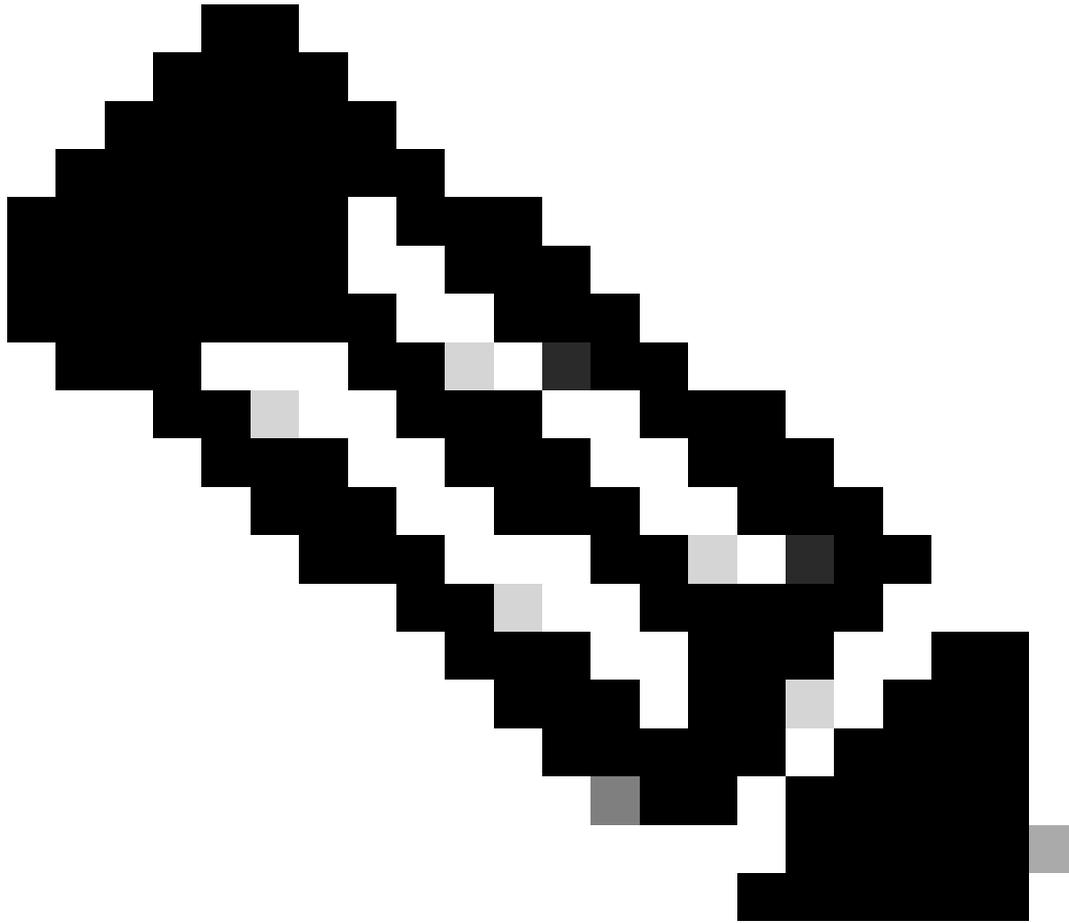
处理多路访问和非广播多路访问(NBMA)网络时，请特别小心。BGP下一跳（多路访问网络）和BGP下一跳(NBMA)部分提供了更多详细信息。

BGP 下一跳（多路访问网络）



此示例演示以太网等多路网络中下一跳的活动方式。

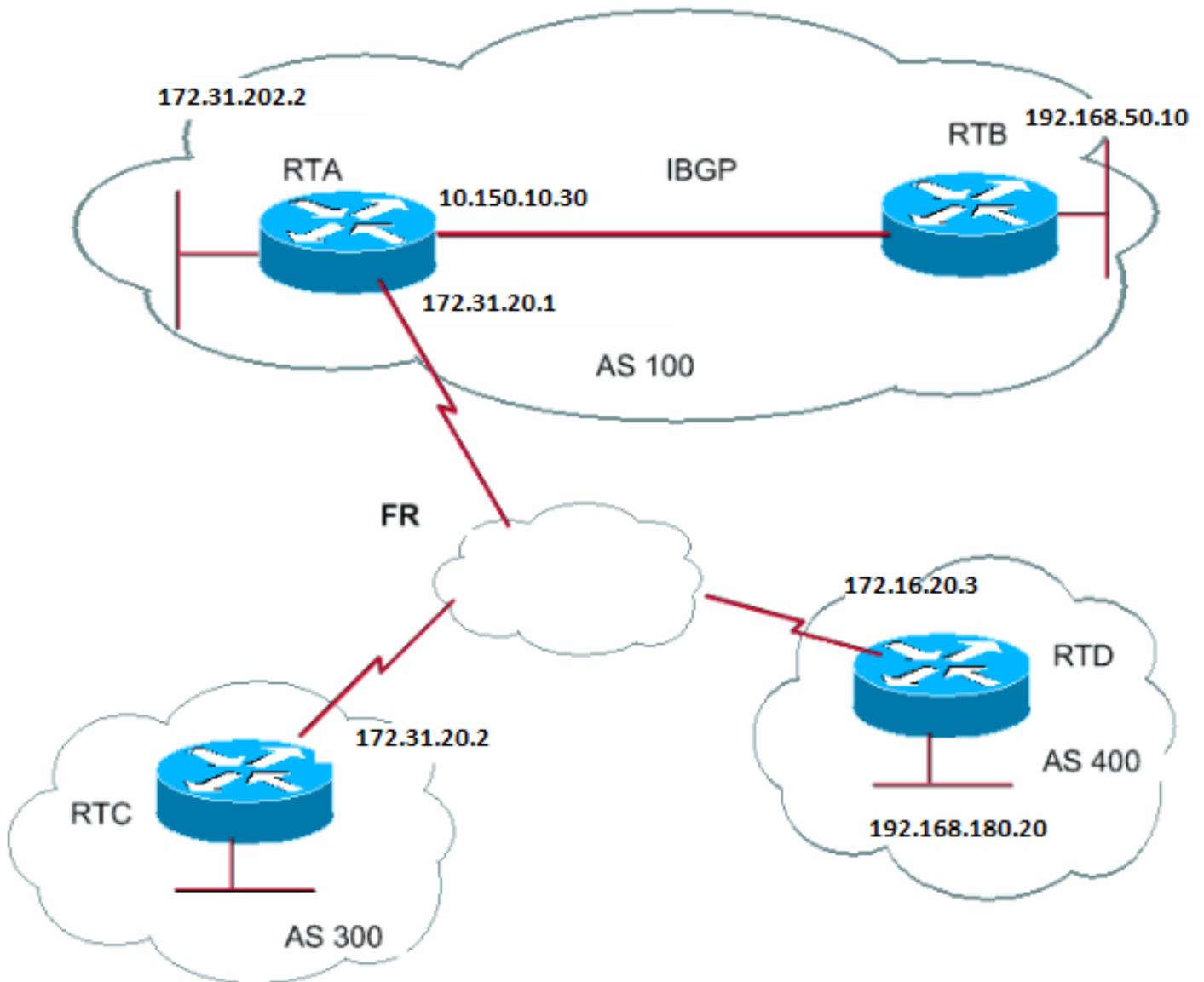
假设 AS300 中的 RTC 和 RTD 运行 OSPF。RTC 与 RTA 运行 BGP。RTC 可以经由 172.16.20.3 到达网络 192.168.180.20。当 RTC 将有关 192.168.180.20 的 BGP 更新发送到 RTA 时，RTC 将 172.16.20.3 用作下一跳。RTC 不使用自身的 IP 地址 172.31.20.2。RTC 使用此地址是因为 RTA、RTC 和 RTD 之间的网络是多路网络。RTA 使用 RTD 作为到达 192.168.180.20 的下一跳比经由 RTC 使用额外跃点更加有效。



注意：RTC将192.168.180.20通告给下一跳为172.16.20.3的RTA。

如果 RTA、RTC 和 RTD 的共同介质不是多路访问，而是 NBMA，则情况会更加复杂。

BGP 下一跳 (NBMA)



共同介质在图中显示为云状。如果共同介质为帧中继或任何 NBMA 云，则其表现和通过以太网进行连接是一样的。RTC 将 192.168.180.20 通告给下一跳为 172.16.20.3 的 RTA。

问题在于 RTA 没有到 RTD 的直接永久虚拟电路 (PVC)，不能到达下一跳。这时路由将失败。

next-hop-self 命令可解决此问题。

next-hop-self 命令

对于 BGP 下一跳 (NBMA) 示例中所示的下一跳问题，可以使用 next-hop-self 命令。语法为：

```
<#root>
```

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

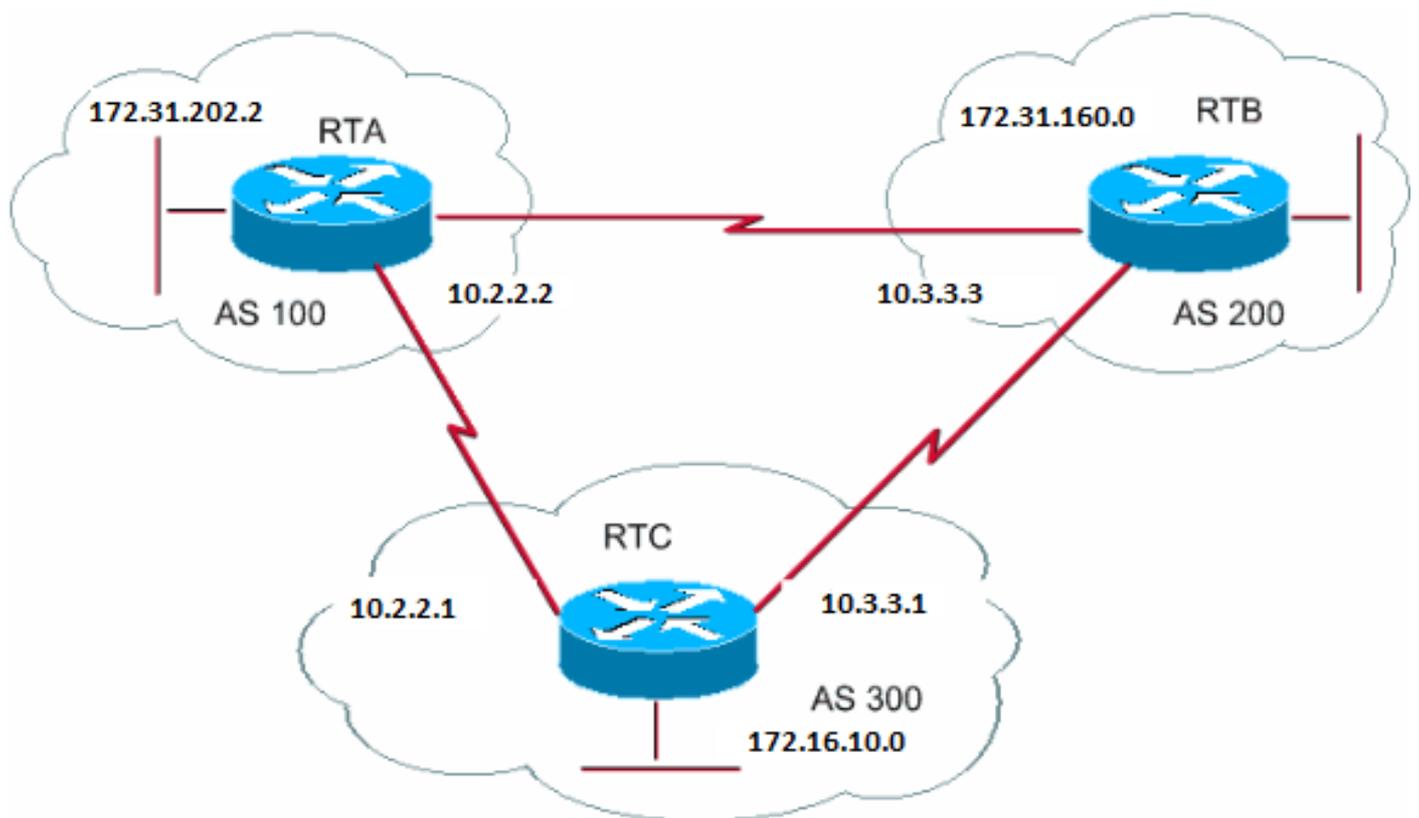
next-hop-self命令允许您强制BGP使用特定IP地址作为下一跳。

对于 BGP 下一跳 (NBMA) 示例，如下配置可解决该问题：

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

RTC 通告下一跳等于 172.31.20.2 的 192.168.180.20。

BGP 后门



在上图中，RTA和RTC运行eBGP。RTB 和 RTC 运行 eBGP。RTA 和 RTB 运行某种 IGP，即 RIP、IGRP 或其他协议。为 eBGP 更新定义的距离为 20，小于 IGP 距离。默认距离是：

-

对于 RIP 为 120

-

对于 IGRP 为 100

-

对于 EIGRP 为 90

-

对于 OSPF 为 110

RTA 通过两个路由协议接收有关 172.31.160.0 的更新：

-

距离为 20 的 eBGP

-

距离大于 20 的 IGP

默认情况下，BGP 距离如下：

-

外部距离 — 20

-

内部距离 — 200

-

本地距离 — 200

但可以使用distance 命令更改默认距离：

```
<#root>
```

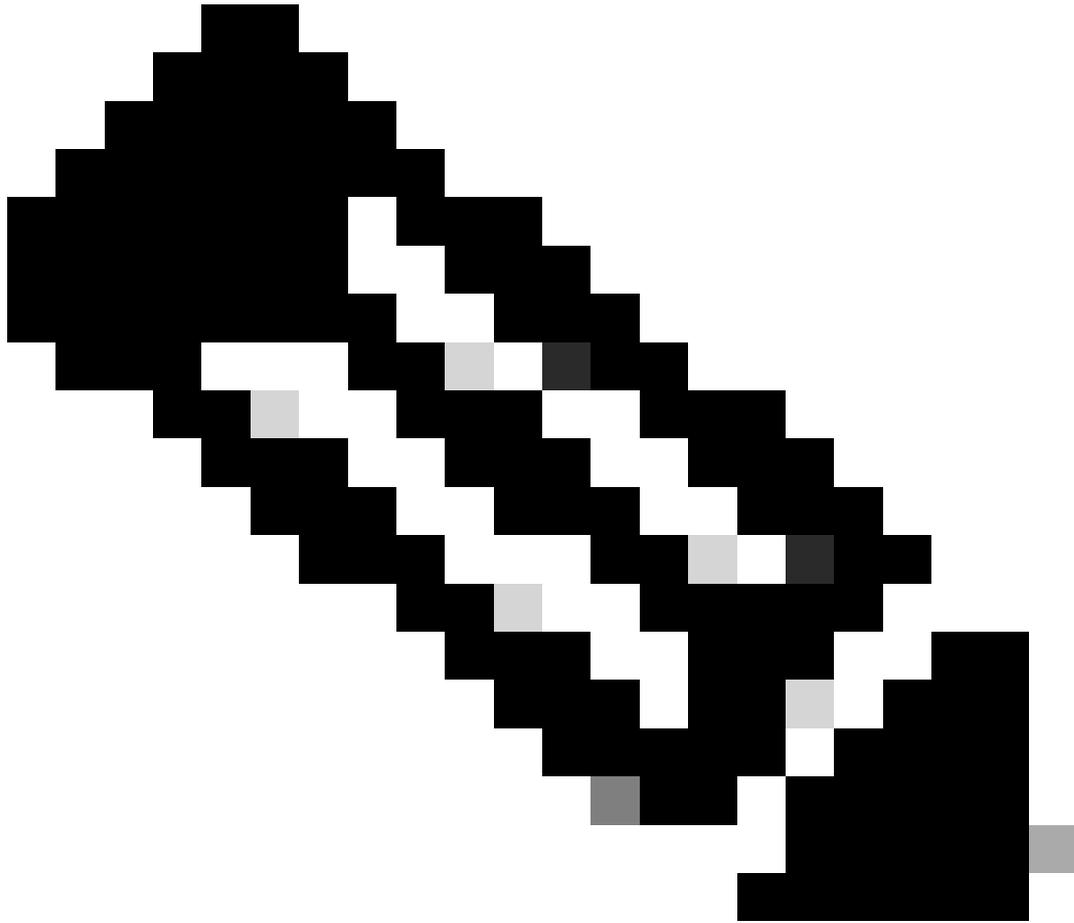
```
distance bgp <external-distance> <internal-distance> <local-distance>
```

RTA 选择使用 eBGP 连接的 RTC，因为这样距离更短。

如果希望 RTA 通过 RTB（使用 IGP）获知 172.31.160.0，则有两种选择：

-

更改 eBGP 的外部距离或者更改 IGP 距离。



注意：不建议进行此更改。

-

请使用 BGP 后门。

BGP 后门使 IGP 路由成为首选路由。

发出[networkaddressbackdoor](#)命令。

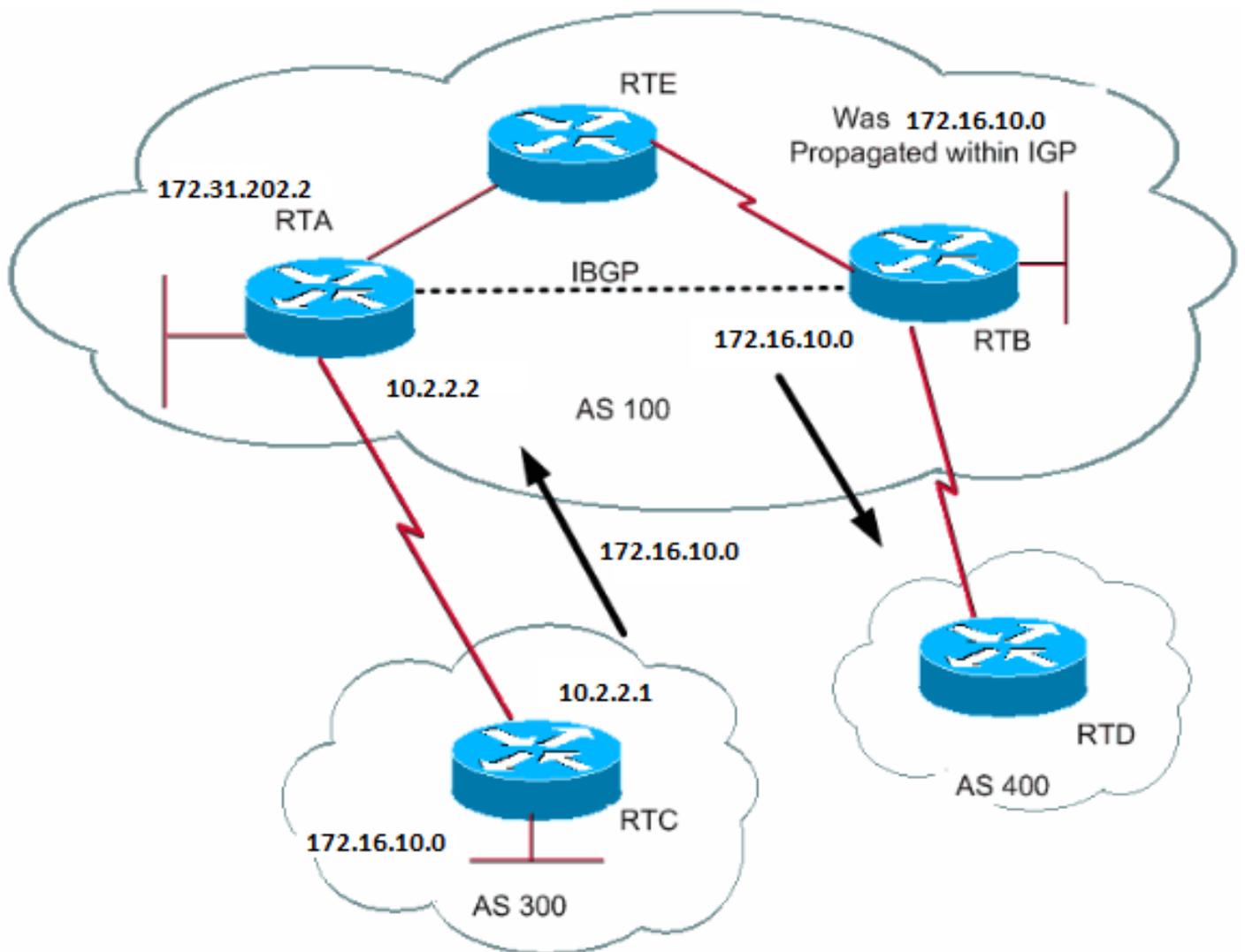
配置的网络是希望通过 IGP 到达的网络。对于 BGP 来说，此网络与本地分配的网络得到同等对待，除非 BGP 更新不通告此网络。

```
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
```

网络172.31.160.0被视为本地条目，但不会作为正常网络条目进行通告。

RTA 从距离为 90 的 RTB 处通过 EIGRP 获知 172.31.160.0。RTA 也通过 eBGP 从距离为 20 的 RTC 处获知了该地址。通常eBGP是首选项，但因为发出了network backdoor 命令，EIGRP成为首选项。

同步



讨论同步之前请先看下面的场景。AS300中的RTC发送有关172.16.10.0的更新。RTA 和 RTB 运行 iBGP，因此，RTB 获取更新并能

够经由下一跳 10.2.2.1 到达 172.16.10.0。请记住下一跳是通过 iBGP 传递的。RTB 必须将流量发送给 RTE 才能到达下一跳。

假设 RTA 未将网络 172.16.10.0 重分配给 IGP。这时，RTE 甚至不知道 172.16.10.0 的存在。

如果 RTB 开始向 AS400 通告 RTB 可以达到 172.16.10.0，则从 RTD 到 RTB 且目标为 172.16.10.0 的流量将进入 RTE 并在 RTE 处丢弃。

同步规定，如果您的 AS 将流量从另一个 AS 传递到第三个 AS，则 BGP 在您的 AS 中的所有路由器通过 IGP 获知该路由之前，不得通告该路由。BGP 应等到 IGP 在 AS 内传播该路由。然后，BGP 才将该路由通告给外部对等体。

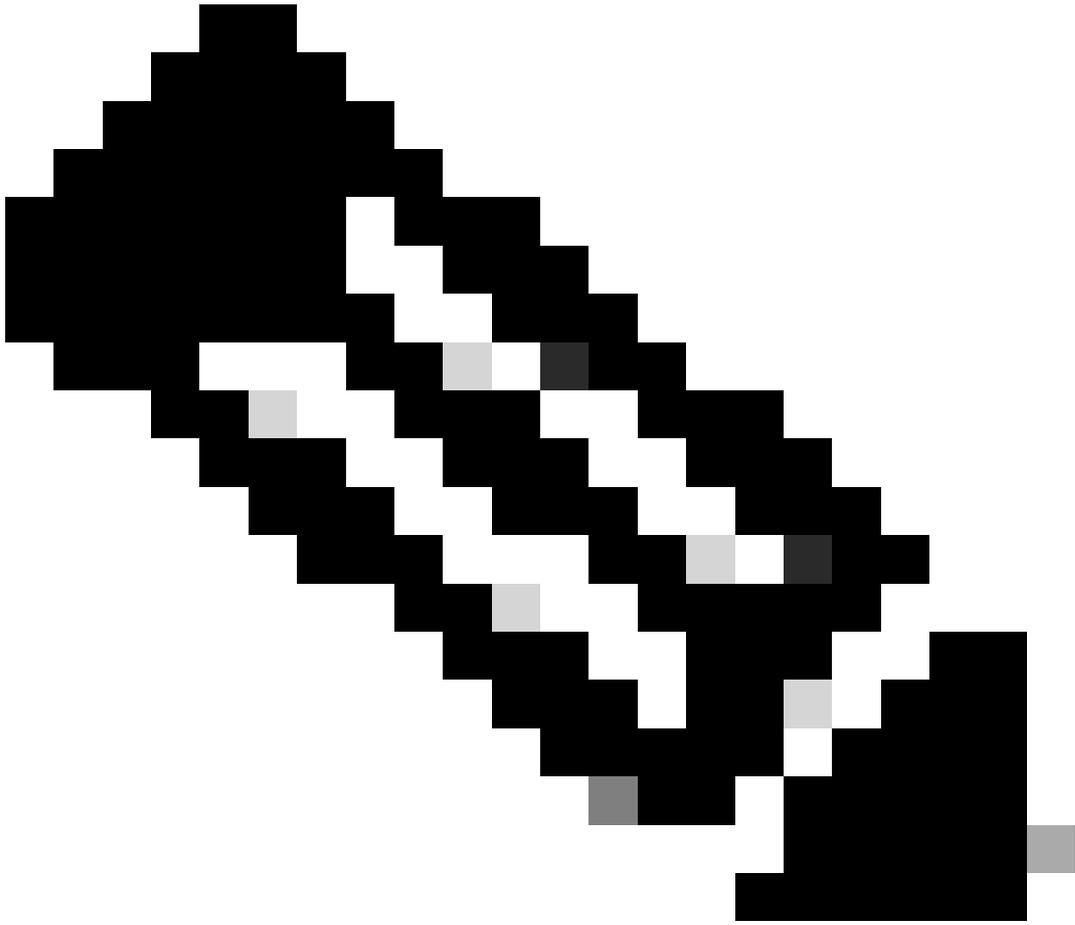
在此部分的示例中，RTB 等待通过 IGP 获知 172.16.10.0。然后，RTB 开始将更新发送到 RTD。通过在 RTB 中添加指向 172.16.10.0 的静态路由，可以使 RTB 认为 IGP 已经传播信息。确保其他路由器可以到达 172.16.10.0。

禁用同步

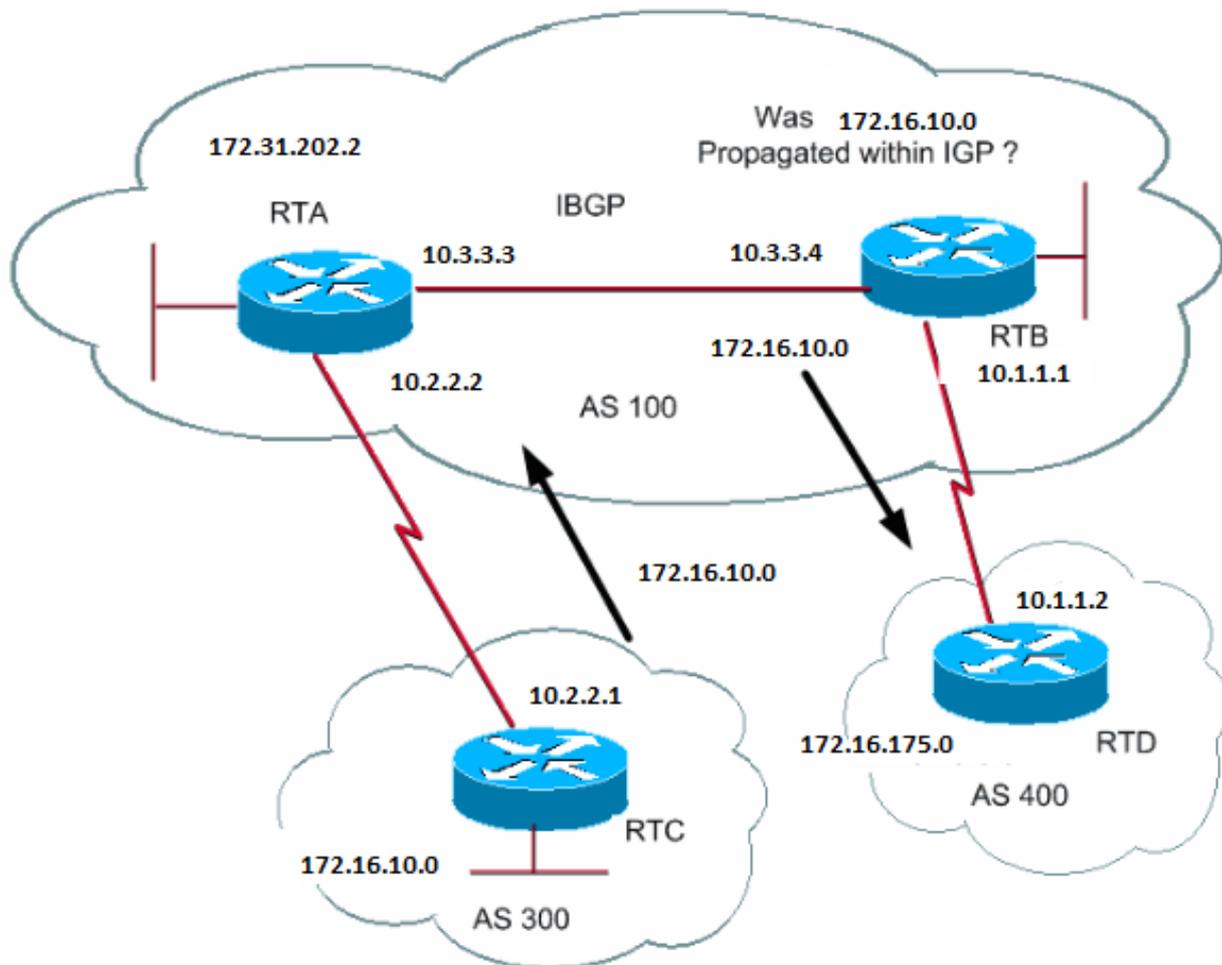
某些情况下并不需要同步。如果不通过您的 AS 传递来自其他 AS 的流量，则可以禁用同步。如果您的 AS 中的所有路由器都运行 BGP，则也可以禁用同步。禁用此功能可使您在 IGP 中传递的路由数减少，并使 BGP 更快地聚合。

禁用同步不是自动功能。如果 AS 中的所有路由器都运行 BGP，而完全不运行 IGP，路由器并不会知道已禁用同步。您的路由器将无限期等待有关某个路由的 IGP 更新，以将该路由发送到外部对等体。这时必须手动禁用同步，以便使路由正常工作：

```
router bgp 100
no synchronization
```



注意：请确保发出clear ip bgp address命令重置会话。



```

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

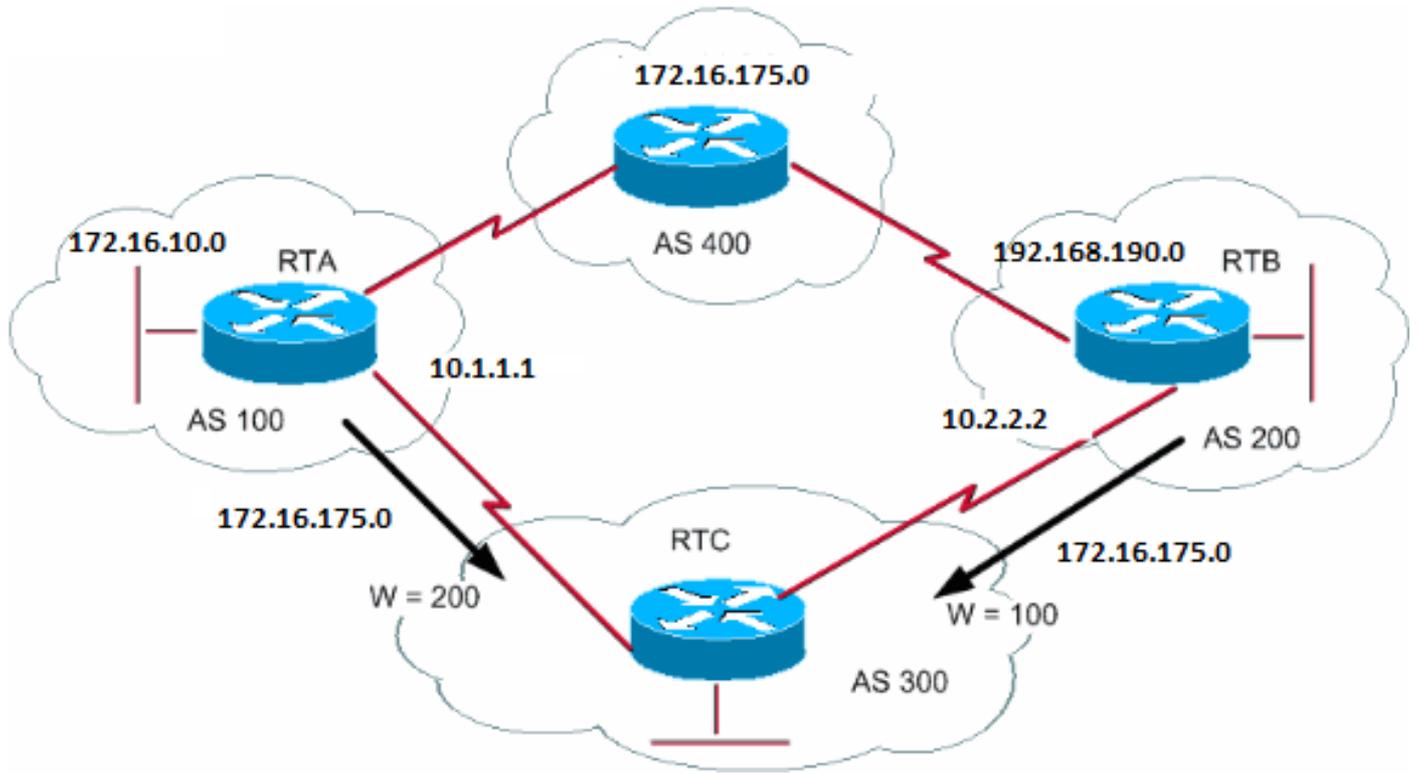
```

```

RTA#
router bgp 100
network 172.31.202.2
neighbor 10.3.3.4 remote-as 100

```

权重 属性



权重属性是由 Cisco 定义的属性。此属性使用权重来选择最佳路径。权重在本地分配到路由器。该值只对特定路由器有意义。任何路由更新都不会传播或传递该值。权重可以是 0 到 65,535 之间的数字。默认情况下，由路由器生成的路径权重值为 32,768，其他路径权重值则为 0。

当存在到达同一目的地的多个路由时，权重值较高的路由优先。请看本部分中的示例。RTA从AS4获知了网络172.16.0.0。RTA 将更新传播到 RTC。RTB还从AS4获知了网络172.16.0.0。RTB 将更新传播给 RTC。RTC 有两条途径到达 172.16.0.0，必须确定采取哪一条。如果在 RTC 上设置来自 RTA 的更新的权重，使该权重大于来自 RTB 的更新的权重，则强制 RTC 使用 RTA 作为到达 172.16.0.0 的下一跳。有多种方法可以设置此权重：

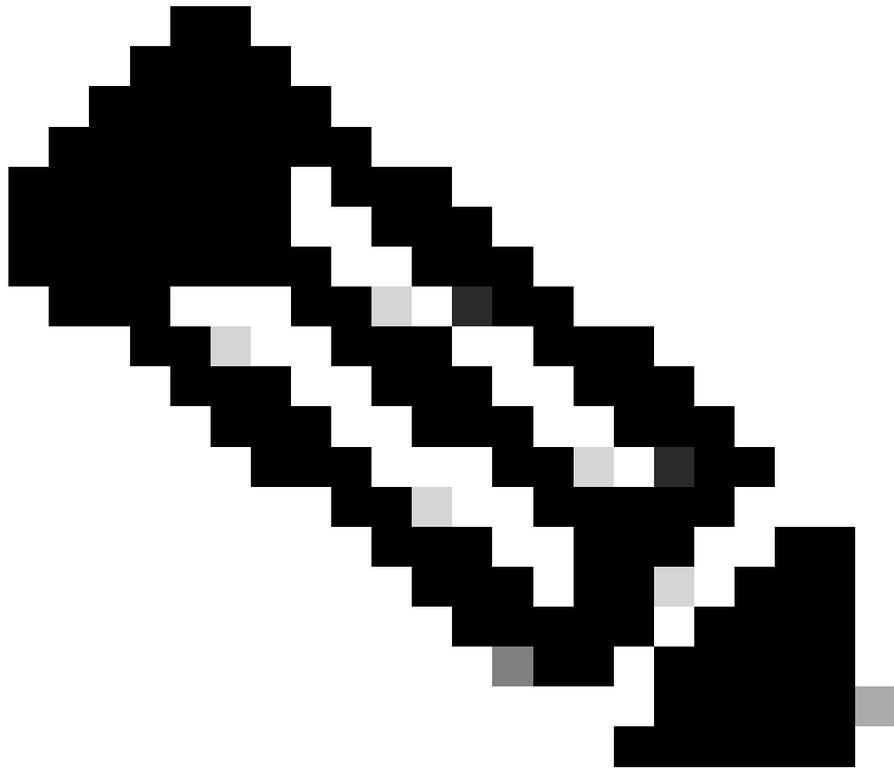
- 使用 neighbor 命令。

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

- 使用 AS_PATH 访问列表。

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

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



注意：在某些情况下，可能只有极少数命令在某些软件版本中不可用。

使用路由映射。

```
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.
```

具有较高权重值的 RTA 具有作为下一跳的优先权。

使用 IP AS_PATH 和过滤器列表也可以产生同样的结果。

```
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$
...
```

也可以使用路由映射获得相同的结果。

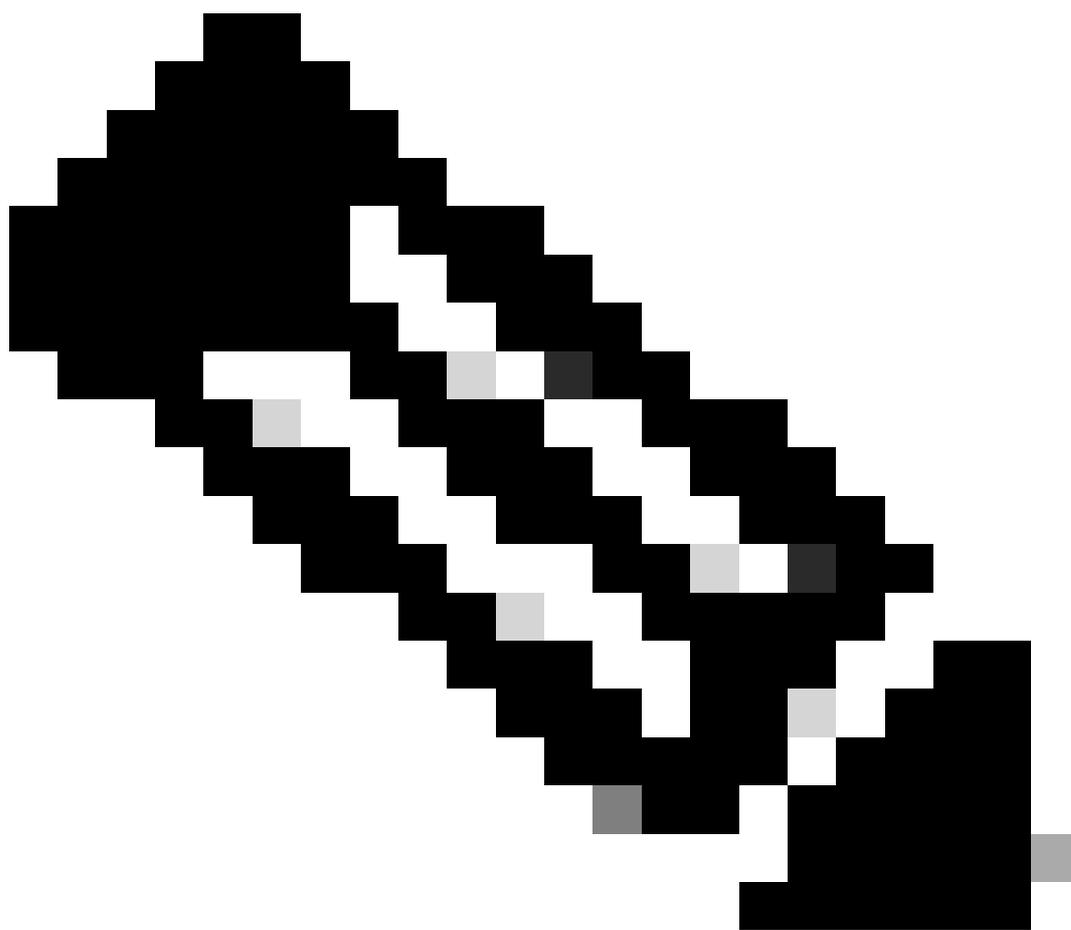
```
RTC#
router bgp 300
  neighbor 10.1.1.1 remote-as 100
  neighbor 10.1.1.1 route-map setweightin in
  neighbor 10.2.2.2 remote-as 200
  neighbor 10.2.2.2 route-map setweightin in
  ...
ip as-path access-list 5 permit ^100$
...

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.

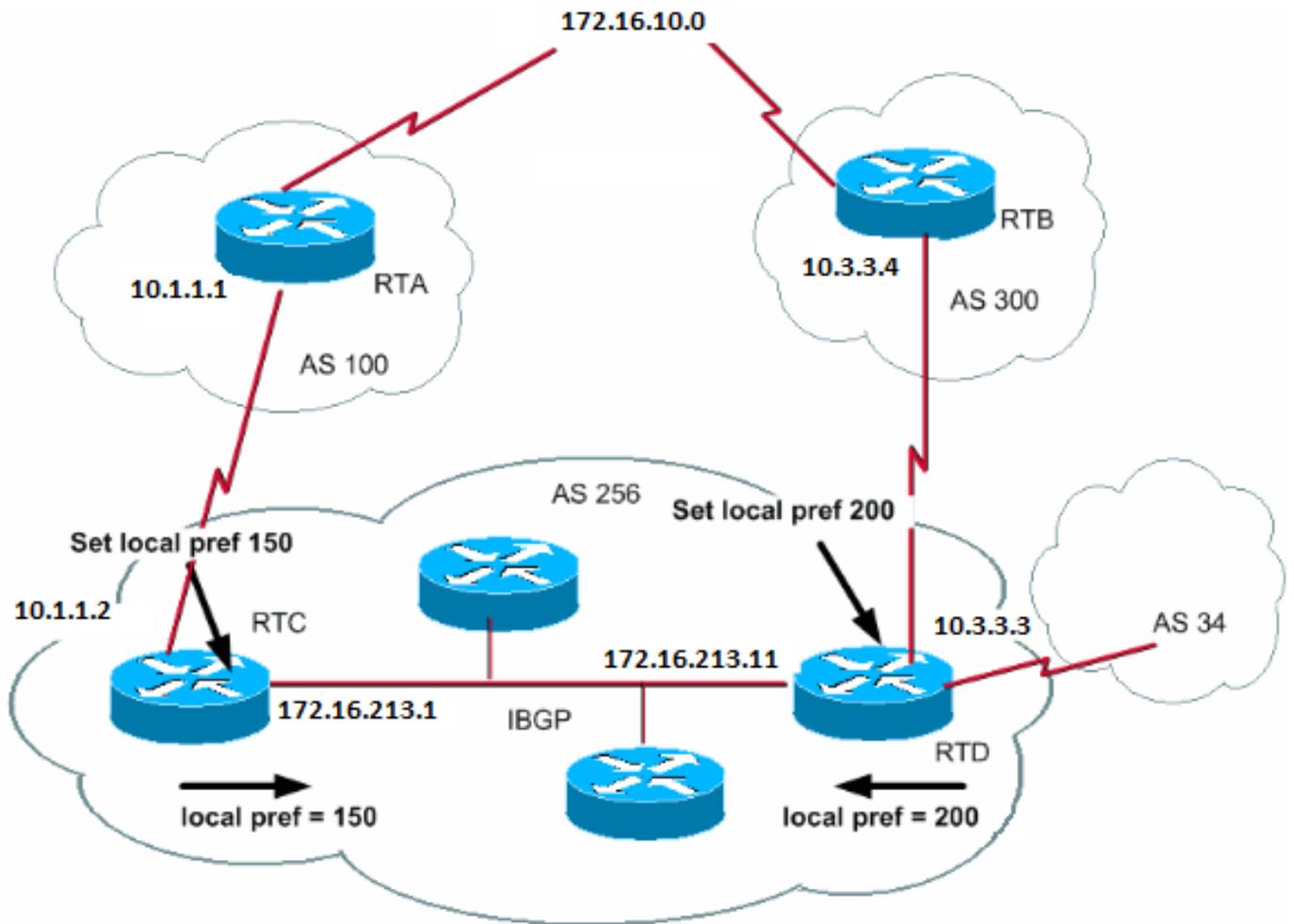
```
route-map setweightin permit 20  
  set weight 100
```

!--- Anything else has weight 100.



注意：您可以修改权重以首选MPLS VPN BGP路径，并将IGP路径作为备份。

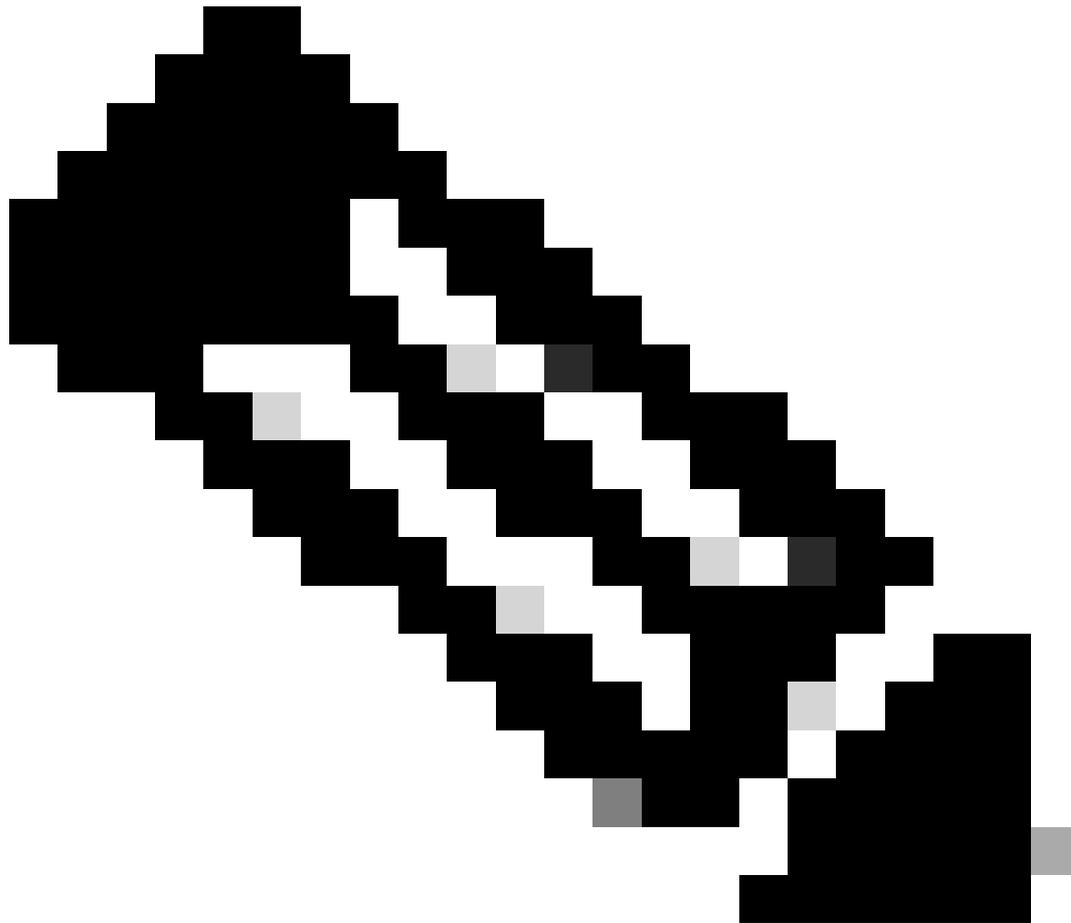
本地优先级属性



本地优先级指示 AS 以哪条路径作为首选来退出 AS，到达特定网络。路径的本地优先级越高，越优先选择该路径。本地优先级的默认值是 100。

与权重属性（只与本地路由器有关）不同，本地优先级属性是路由器在同一 AS 内交换的属性。

通过发出 `bgp default local-preference value` 命令可以设置本地优先级。也可以如本部分示例所示，使用路由映射来设置本地优先级：



注意：必须执行软重置（即，清除路由器上的bgp进程）才能将更改考虑在内。要清除bgp进程，请使用clear ip bgp [soft][in/out]命令，其中soft表示软重置，不中断会话，并且[in/out]指定入站或出站配置。如果未指定in/out，则会重置入站和出站会话。

bgp default local-preference 命令为从路由器传出到同一 AS 内的对等端的更新设置本地优先级。在本部分的图中，AS256从组织的两个不同方面接收有关172.16.10.0的更新。本地优先级可帮助您确定退出 AS256 以到达该网络的路径。假设 RTD 是首选的退出点。以下配置为从 AS300 到 200 以及从 AS100 到 150 传递的更新设置本地优先级：

```
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
```

在此配置中，RTC 将所有更新的本地优先级设置为 150。同样，RTD 将所有更新的本地优先级设置为 200。在 AS256 内存在本地优先级的交换。因此，当更新来自AS300而不是AS100时，RTC和RTD都意识到网络172.16.10.0具有更高的本地优先级。AS256 中以该网络为目的地的所有流量都会使用 RTD 作为退出点进行传输。

路由映射的使用提供了更多灵活性。在此部分的示例中，RTD 收到的所有更新在到达 RTD 时都将本地优先级标记为 200。来自 AS34 的更新也标记为具有本地优先级 200。此标记可能是不必要的。因此，可以使用路由映射指定需要做特定本地优先级标记的特定更新。例如：

```
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$
...

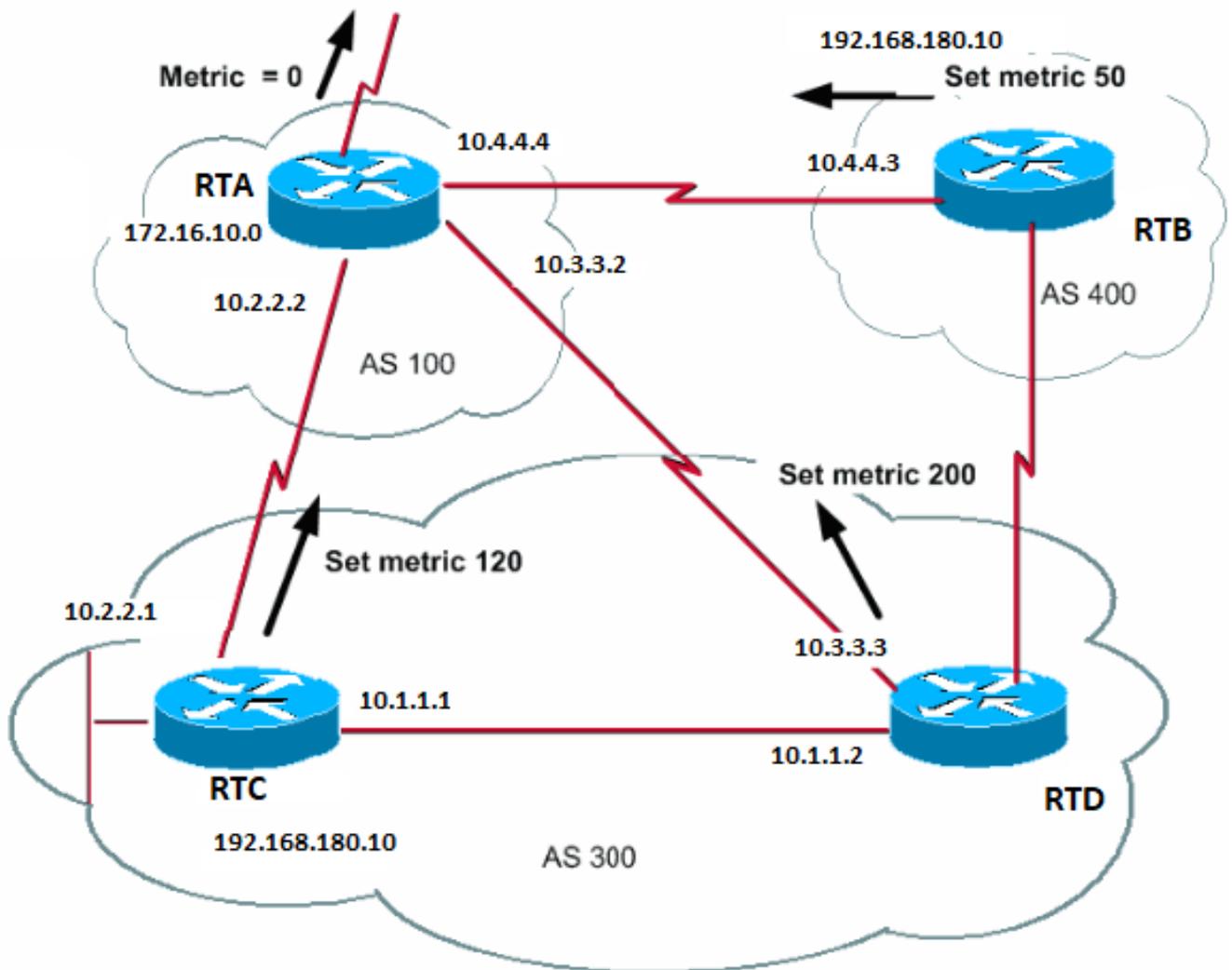
route-map setlocalin permit 10
match as-path 7
set local-preference 200

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

使用此配置，来自 AS300 的所有更新都会有本地优先级 200。所有其他更新（如来自 AS34 的更新）的本地优先级值都为 150。

度量属性

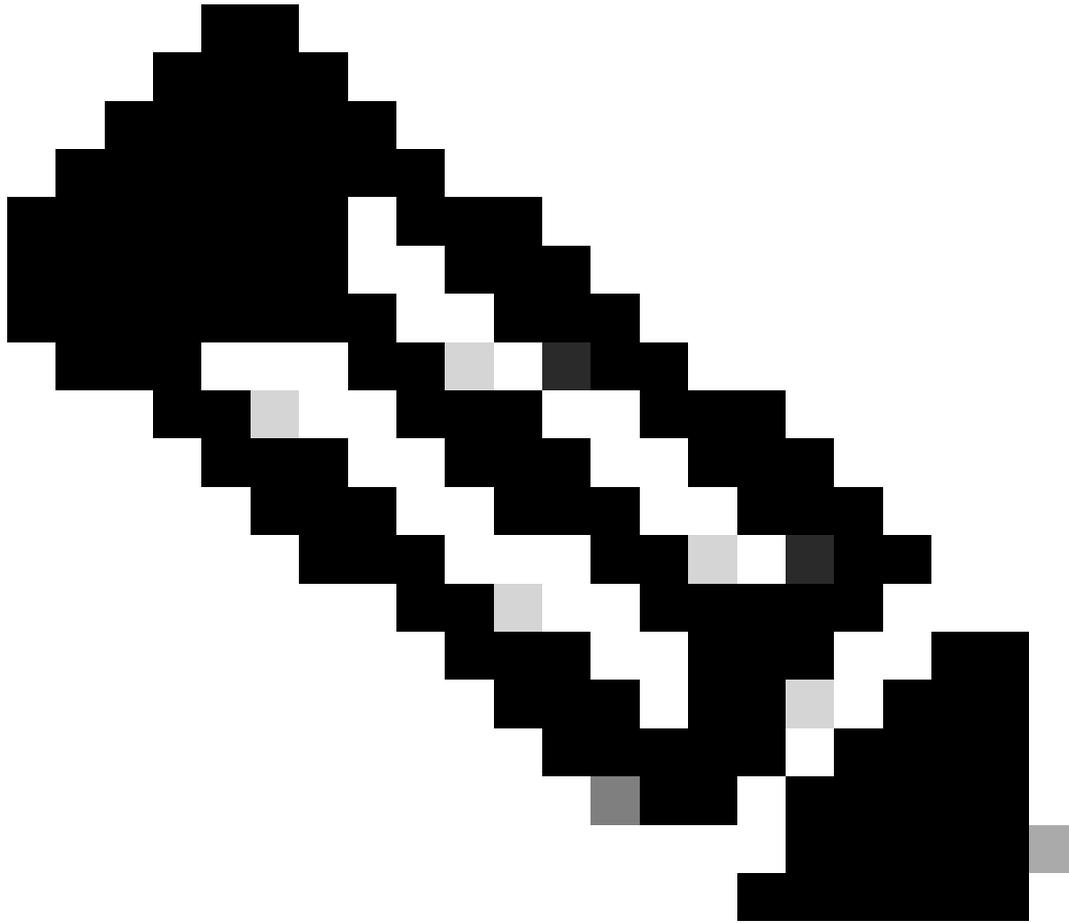
METRIC (MULTI_EXIT_DISC) (INTER_AS)



度量属性也具有名称 MULTI_EXIT_DISCRIMINATOR、MED (BGP4) 或 INTER_AS (BGP3)。此属性提示外部邻居到达 AS 的路径优先级。如果另一个 AS 有多个进入点，该属性提供了一种在到达路由的途中影响该 AS 的动态方式。优先选择较低的度量值。

不同于本地优先级，度量在 AS 之间进行交换。度量传递到 AS 中，但不离开该 AS。当某个更新以特定的度量进入 AS 时，将使用该度量在该 AS 内作出决策。当同一更新继续传递到第三个 AS 时，该度量返回为 0。本部分中的图显示了一组度量。度量默认值是 0。

除非某路由器接到其他指令，否则该路由器将比较同一 AS 内邻居的度量以选择路径。要使路由器比较来自其他 AS 的邻居的度量，需要对路由器发出特殊配置命令 `bgp always-compare-med`。



注意：有两个BGP配置命令可影响基于多出口标识符(MED)的路径选择。bgp deterministic-med 命令和 bgp always-compare-med 命令。发出 bgp deterministic-med 命令将确保在同一 AS 内通告不同对等体时，选择路由时会比较 MED 变量。发出 bgp always-compare-med 命令确保比较不同 AS 内的邻居的 MED 以选择路径。bgp always-compare-med 命令适用于多个服务提供商或企业同意采用统一策略设置 MED 的情况。要了解这些命令如何影响 BGP 路径选择，请参阅 bgp deterministic-med 命令和 bgp always-compare-med 命令的区别。

在本部分的图中，AS100通过三个不同的路由器获取有关网络192.168.180.10的信息：RTC、RTD和RTB。RTC 和 RTD 在 AS300 中，RTB 在 AS400 中。

在本示例中，将忽略通过命令 **bgp bestpath as-path ignore** 对RTA进行AS路径比较。它配置为强制BGP转到下一个属性以进行路由比较（本例中为度量或MED）。如果省略此命令，则BGP可以从具有最短AS路径的路由器RTC安装路由192.168.180.10。

假设将来自 RTC 的度量设置为 120，来自 RTD 的度量设置为 200，来自 RTB 的度量设置为 50。默认情况下，路由器比较来自同一

AS 内的邻居的度量。因此，RTA 只比较来自 RTC 的度量和来自 RTD 的度量。RTA 选择 RTC 作为最佳下一跳，因为 120 小于 200。当 RTA 从度量为 50 的 RTB 处获取更新时，RTA 不能将此度量与 120 相比较，因为 RTC 和 RTB 处于不同的 AS 中。RTA 必须根据某些其他属性进行选择。

要强制 RTA 比较这些度量，必须针对 RTA 发出 `bgp always-compare-med` 命令。以下配置阐明了此过程：

```
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
```

使用这些配置，RTA 选择 RTC 作为下一跳，因为所有其他属性都相同。要在度量比较中包含 RTB，必须如下配置 RTA：

```
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
```

在这种情况下，RTA 选取 RTB 作为到达网络 192.168.180.10 的最佳下一跳。

如果发出`default-metricnumber`命令，也可以在将路由重分配到BGP时设置度量。

本部分的示例中假设 RTB 通过静态路由将网络注入 AS100。配置如下：

```
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.
```

社区属性

社区属性是可转变的可选属性，范围是 0 到 4,294,967,200。社区属性是对特定社区中的目标进行分组并应用与这些社区匹配的路由决策的一种方式。路由决策包括接受、优选、重新分配等等。

可以使用路由映射设置社区属性。路由映射 `set` 命令语法如下：

```
<#root>
```

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

此命令中用到的几个预定义的常用社区如下：

-

no-export — 不通告给 eBGP 对等体。将此路由保留在 AS 内部。

-

no-advertise — 不将此路由通告给内部或外部的任何对等体。

-

internet — 将此路由通告到 Internet 社区。所有路由器都属于此社区。

-

local-as — 在联盟情况下使用，避免将数据包传输到本地 AS 之外。

以下是设置社区的路由映射的两个示例：

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

或

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

如果不设置 **additive** 关键字，200 将替换任何已存在的旧社区。如果使用关键字 **additive**，则会将 200 添加到该社区。即使设置了社区属性，此属性也不会默认传输到邻居。要将此属性传输到邻居，必须使用如下命令：

<#root>

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

例如：

```
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
```

在Cisco IOS软件版本12.0及更高版本中，可以使用三种不同的格式配置社区：十进制、十六进制和AA：NN。默认情况下，Cisco IOS软件使用较旧的十进制格式。要在AA：NN中配置和显示，请发出**ip bgp-community new-global configuration format**命令。

AA：NN的第一部分代表AS编号，第二部分代表2字节编号。

例如：

如果全局配置中不含 `ip bgp-community new-format` [命令](#)，则发出 `show ip bgp 10.6.0.0` [命令将以十进制格式显示社区属性值](#)。在本示例中，社区属性值显示为6553620。

<#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
```

现在，针对此路由器全局发出 ip bgp-community new-format 命令。

```
<#root>
```

```
Router#
```

```
configure terminal
```

```
Enter configuration commands, one per line. End with CNTL/Z.  
Router(config)#
```

```
ip bgp-community new-format
```

```
Router(config)#
```

```
exit
```

使用ip bgp-community new-format global configuration命令，社区值以AA : NN格式显示。在如下示例的show ip bgp 10.6.0.0命令的输出中，该值显示为100:20：

```
<#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
```

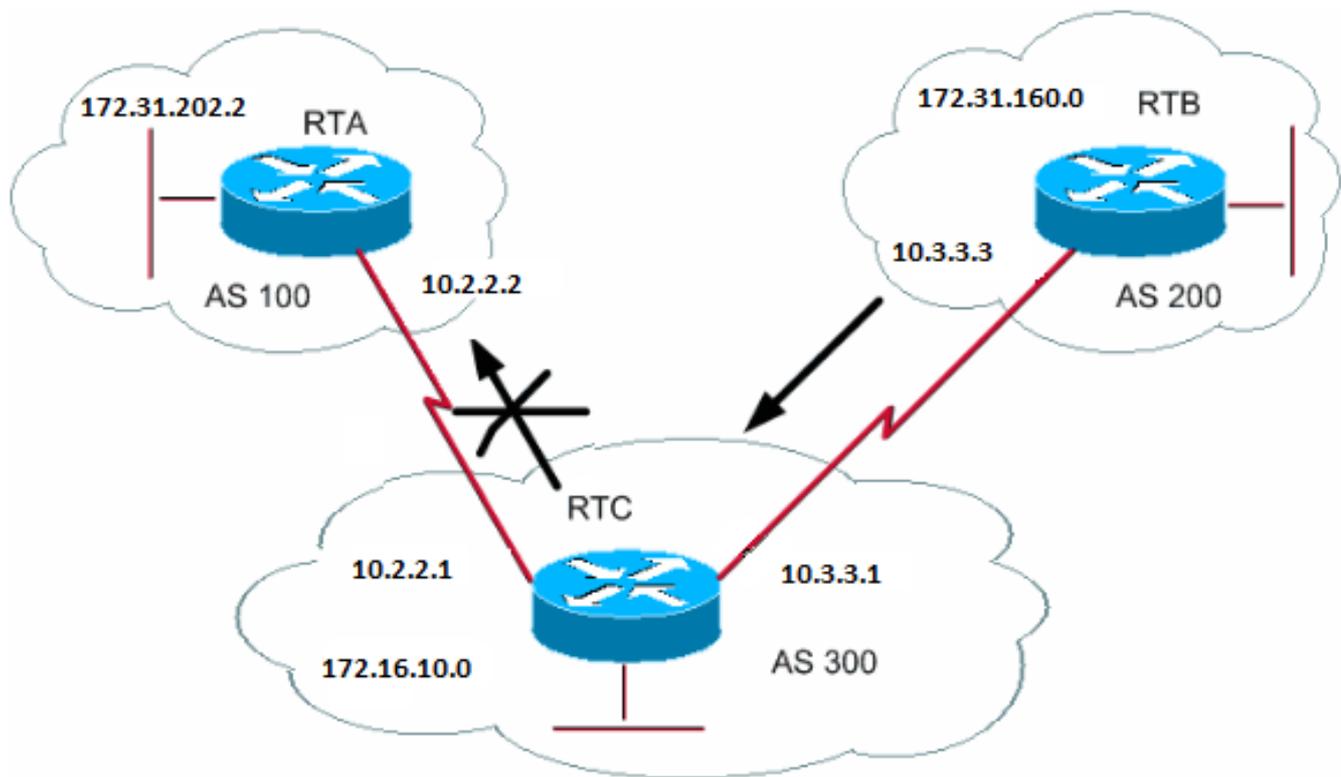
```
Community: 100:20
```

BGP 案例分析 3

BGP过滤器

使用各种不同的过滤器方法可以控制 BGP 更新的发送和接收。您可以根据路由信息，或根据路径信息或社区过滤 BGP 更新。所有方法都能获得同样的结果。要根据特定网络配置在不同方法中做出选择。

路由过滤器



要限制路由器获知或通告的路由信息，可以使用传出或传入特定邻居的路由更新过滤 BGP。您将定义访问列表并将该列表应用到传入或传出邻居的更新。以路由器配置模式发出如下命令：

```
<#root>
```

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

在本示例中，RTB 是网络 172.31.160.0 的源并将更新发送到 RTC。如果希望 RTC 停止向 AS100 传播更新，则必须定义访问列表，以过滤这些更新，并在与 RTA 进行通信时应用该访问列表：

```
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
```

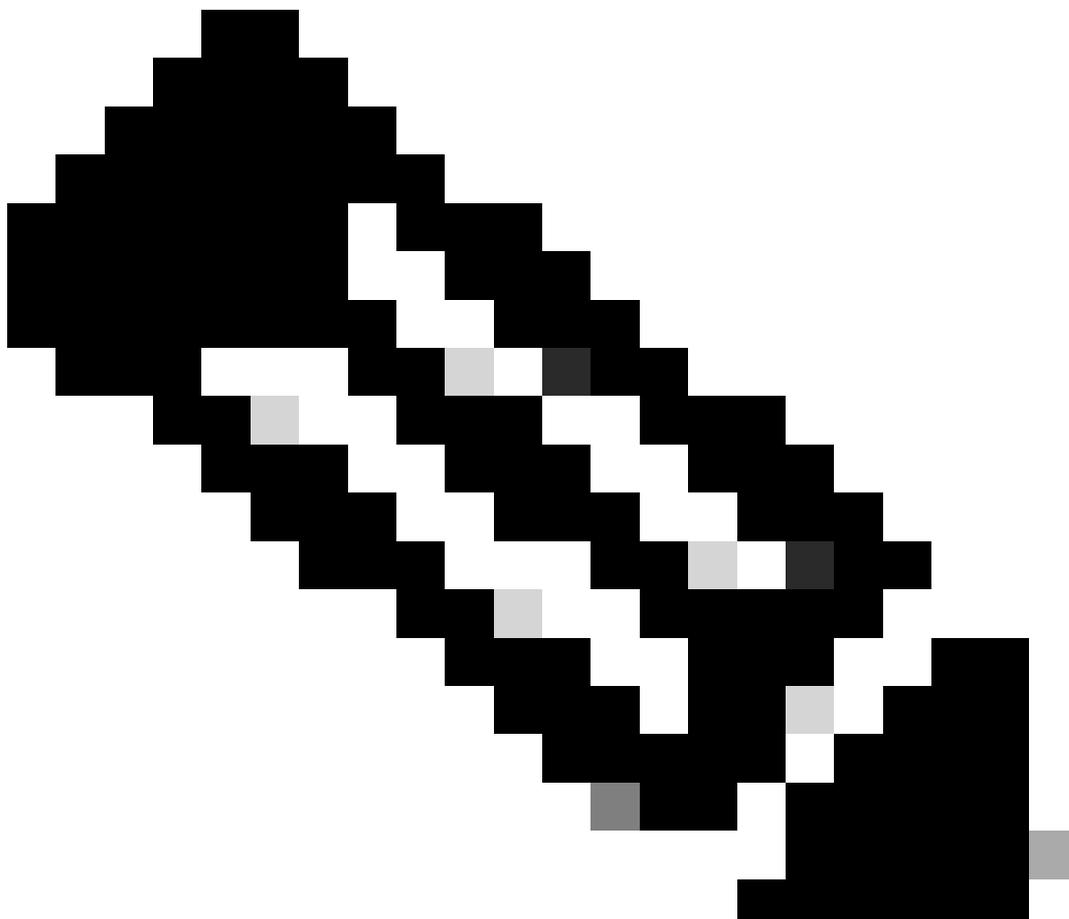
```
access-list 1 deny 172.31.160.0 0.0.255.255
```

```
access-list 1 permit 0.0.0.0 255.255.255.255
```

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

在处理可能产生某些冲突的超网时使用访问列表会有点难以处理。

假设在此部分的示例中，RTB 有 160.10.x.x 的不同子网。您的目标是过滤更新，并且只通告 192.168.160.0/8。



注意： /8表示法表示您使用8位子网掩码，这些位从IP地址的最左侧开始。此地址等同于 192.168.160.0 255.0.0.0。

```
access-list 1 permit 192.168.160.0 0.255.255.25 5
```

命令允许192.168.160.0/8、192.168.160.0/9等。要将更新只限制到 192.168.160.0/8，则必须使用如下格式的扩展访问列表：

```
<#root>
```

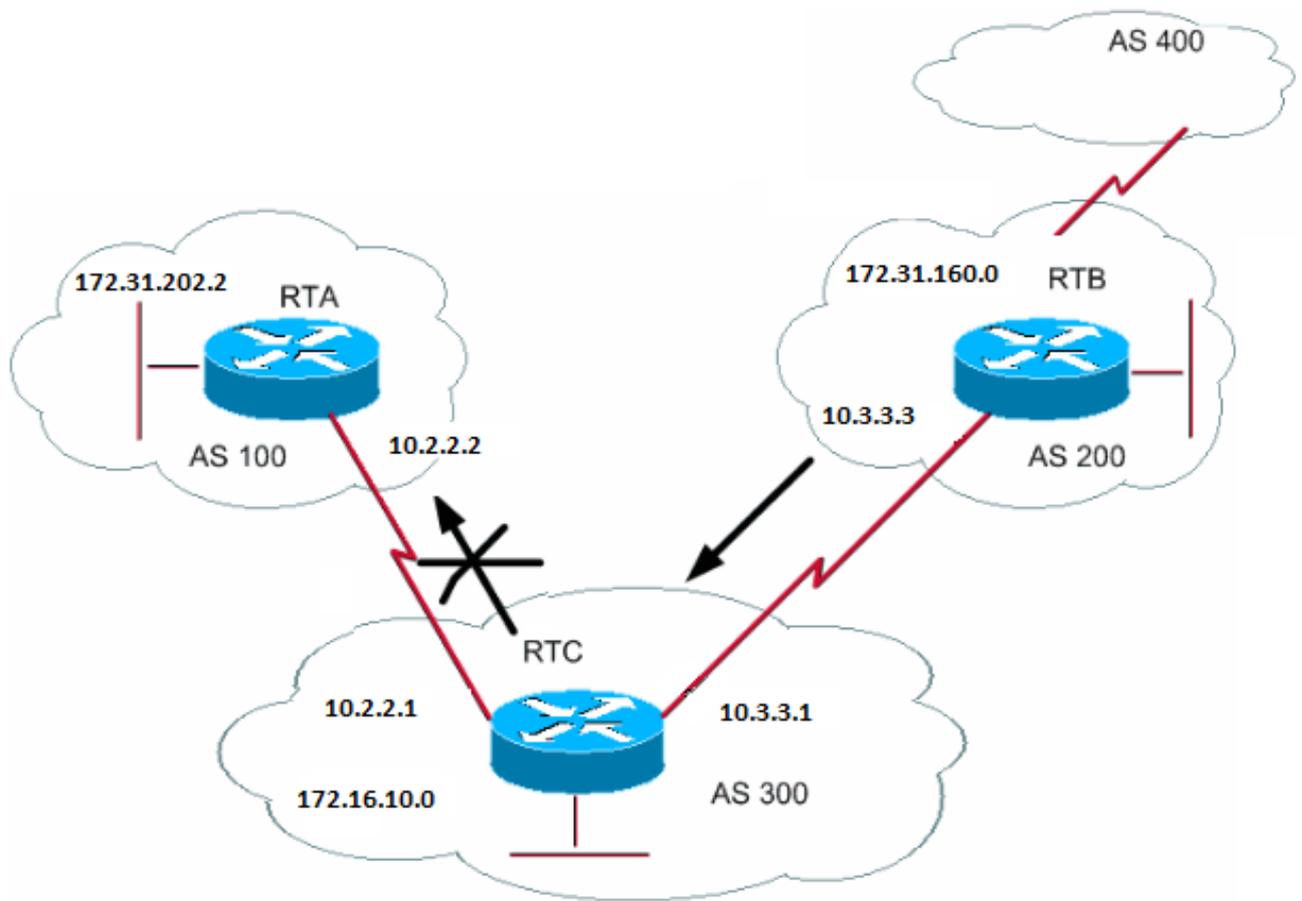
```
access-list 101 permit ip 192.168.160.0 0.255.255.255 255.0.0.0 0.0.0.0.
```

此列表仅允许 192.168.160.0/8。

有关如何过滤来自BGP对等体的网络的示例配置，请参阅[阻止来自BGP对等体的一个或多个网络](#)。该方法使用**distribute-list** 命令，采用标准和扩展访问控制列表(ACL)，并能够过滤前缀列表。

路径过滤器

还可以过滤路径。



您可以使用 BGP AS 路径信息同时针对传入和传出更新指定访问列表。在本部分的图中，您可以阻止有关 172.31.160.0 的更新，以便它们不会转到 AS100。要阻止这些更新，可以针对 RTC 定义一个访问列表，阻止源自 AS200 的所有更新传输到 AS100。发出以下命令：

```
<#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}
```

此示例阻止 RTC 将有关 172.31.160.0 的更新发送到 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 .*
```

本示例中的 `access-list 1` 命令强制拒绝包含以 200 开始并以 200 结尾的路径信息的任何更新。命令中的 `^200$` 为“正则表达式”，其中 `^` 表示“始于”，`$` 表示“止于”。由于 RTB 发送有关 172.31.160.0 的更新，路径信息以 200 开头并以 200 结尾，因此更新与访问列表匹配。访问列表将拒绝这些更新。

`.*` 是另一个正则表达式，其中 `.` 表示“任意字符”，`*` 表示“该字符的重复”。所以 `.*` 表示任何路径信息，这对于传输所有其他更新是必要的。

如果不使用 `^200$`，而使用 `^200?` 以此示例图中的 AS400 为例，源于 AS400 的更新路径信息的格式为 (200, 400)。在此路径信息中，200 在先 400 在后。这些更新与访问列表 `^200` 相匹配，因为路径信息以 200 开头。该访问列表会阻止将这些更新传送到 RTA，但这并不是我们的要求。

要检查使用的正则表达式是否正确，可以发出 [show ip bgp regexpregular-expression](#) 命令。此命令显示所有与正则表达式配置相匹配的路径。

AS 正则表达式

此部分说明如何创建正则表达式。

正则表达式是一种与输入字符串进行匹配的模式。构建正则表达式时，指定输入必须与之匹配的字符串。如果使用 BGP，则指定一个字符串，该字符串包含输入必须匹配的路径信息。

在 [路径过滤器](#) 部分的示例中，指定了字符串 `^200$`。您希望来自更新内部的路径信息与字符串相匹配才能做出决定。

正则表达式包括：

•

范围

范围由左右方括号内的字符序列表示。示例如 [abcd]。

•

凌动

原子是一个单个字符。例如：

•

◦

. 匹配任意单个字符。

^

◦

^ 与输入字符串的开头相匹配。

\$

◦

\$ 与输入字符串的结尾相匹配。

\

。

\与字符相匹配。

-

。

_匹配以下项：逗号(,)、左大括号({)、右大括号(})、输入字符串的开头、输入字符串的结尾或空格。

•

片段

片段是以下符号之一，它位于原子之后：

*

。

*匹配零个或多个原子序列。

+

◦
+匹配一个或多个原子序列。

?

◦
?与原子或空字符串匹配。

•
分支机构

分支是 0 个或多个串联的片段。

以下是一些正则表达式的示例：

a*

•
此表达式表示字母“a”的任何出现次数，包括 0 次。

a+

•

此表达式表示字母“a”至少出现一次。

ab?a

•

此表达式匹配“aa”或“aba”。

100

•

此表达式表示经由 AS100。

_100\$

•

此表达式表示源于 AS100。

^100 .*

•
此表达式表示从 AS100 传输。

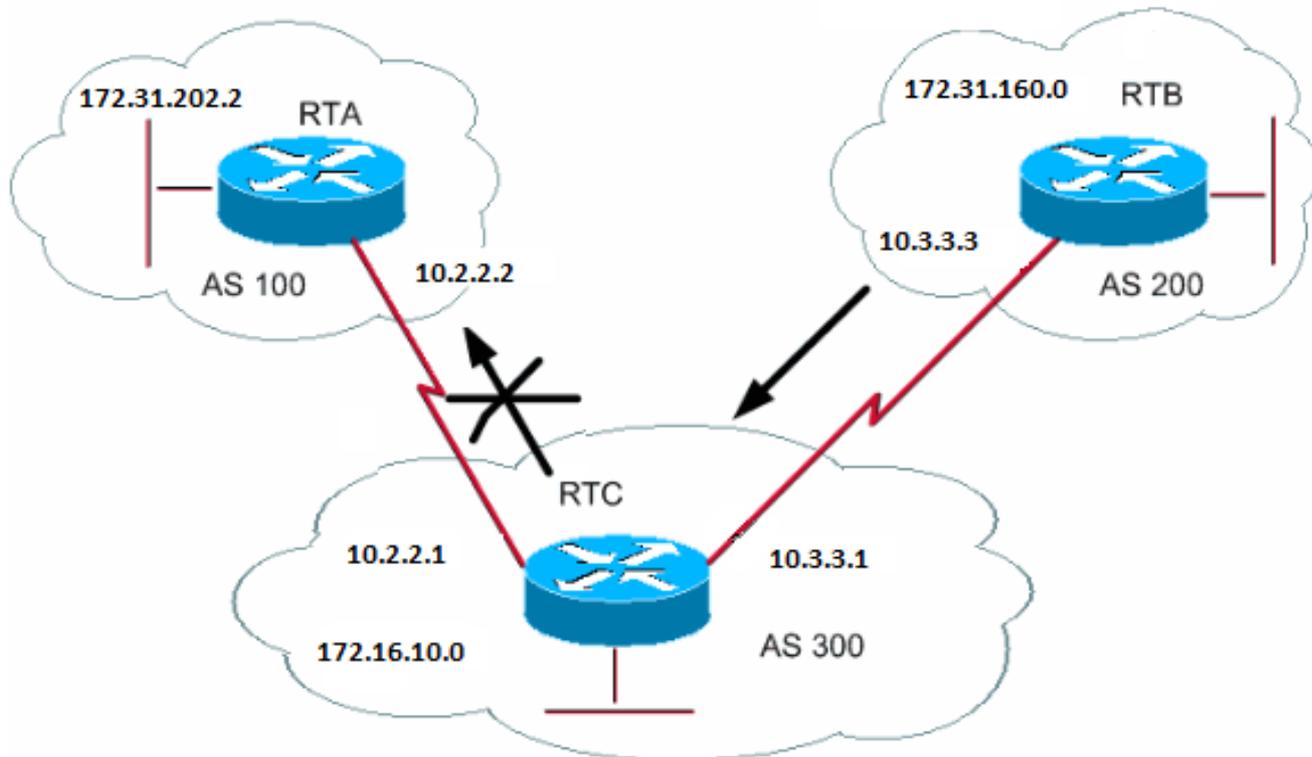
^\$

•
此表达式表示源于此 AS。

有关正则表达式过滤的示例配置，请参阅[在BGP中使用正则表达式](#)。

BGP社区过滤器

本文介绍路由过滤和 AS 路径过滤。另一种方法是社区过滤。社区属性部分讨论了社区，此部分提供了几个有关如何使用社区的示例

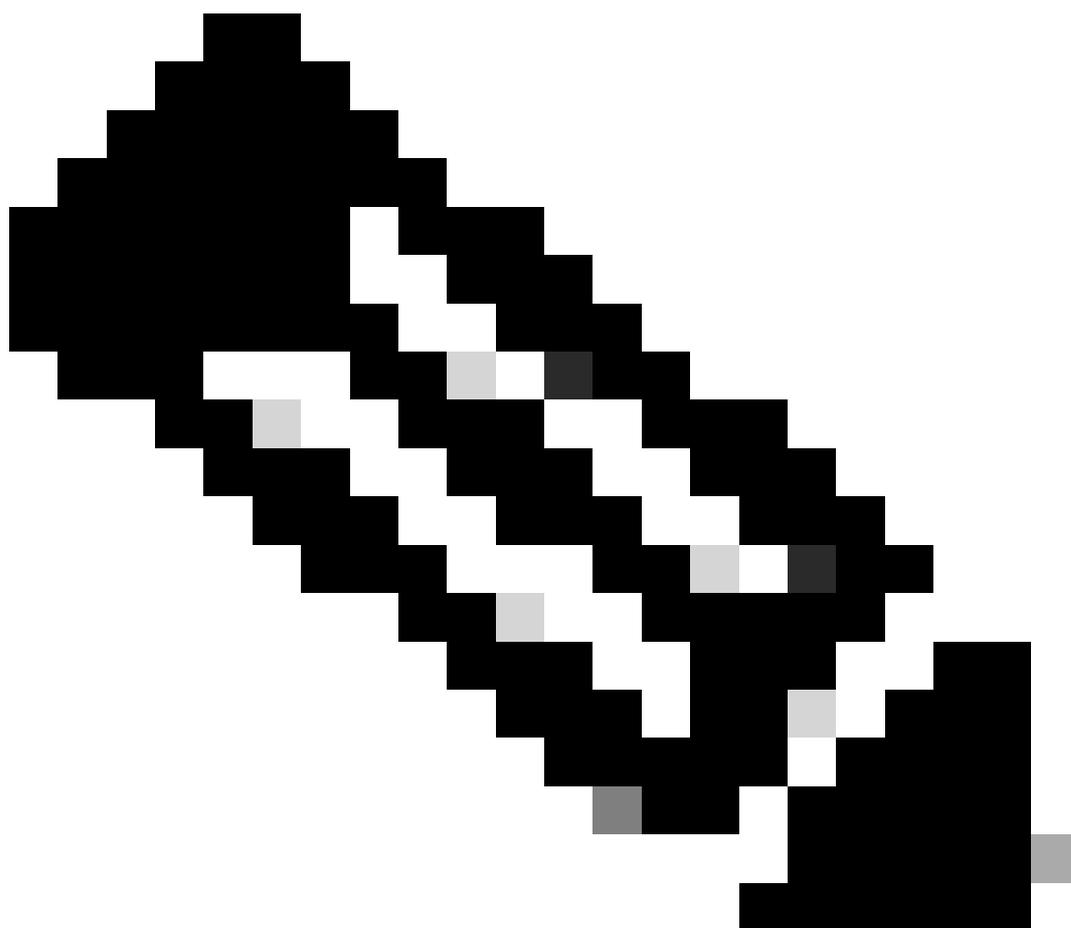


在本示例中，您希望 RTB 为其通告的 BGP 路由设置社区属性，这样 RTC 不会将这些路由传播到外部对等体。使用no-export社区属性。

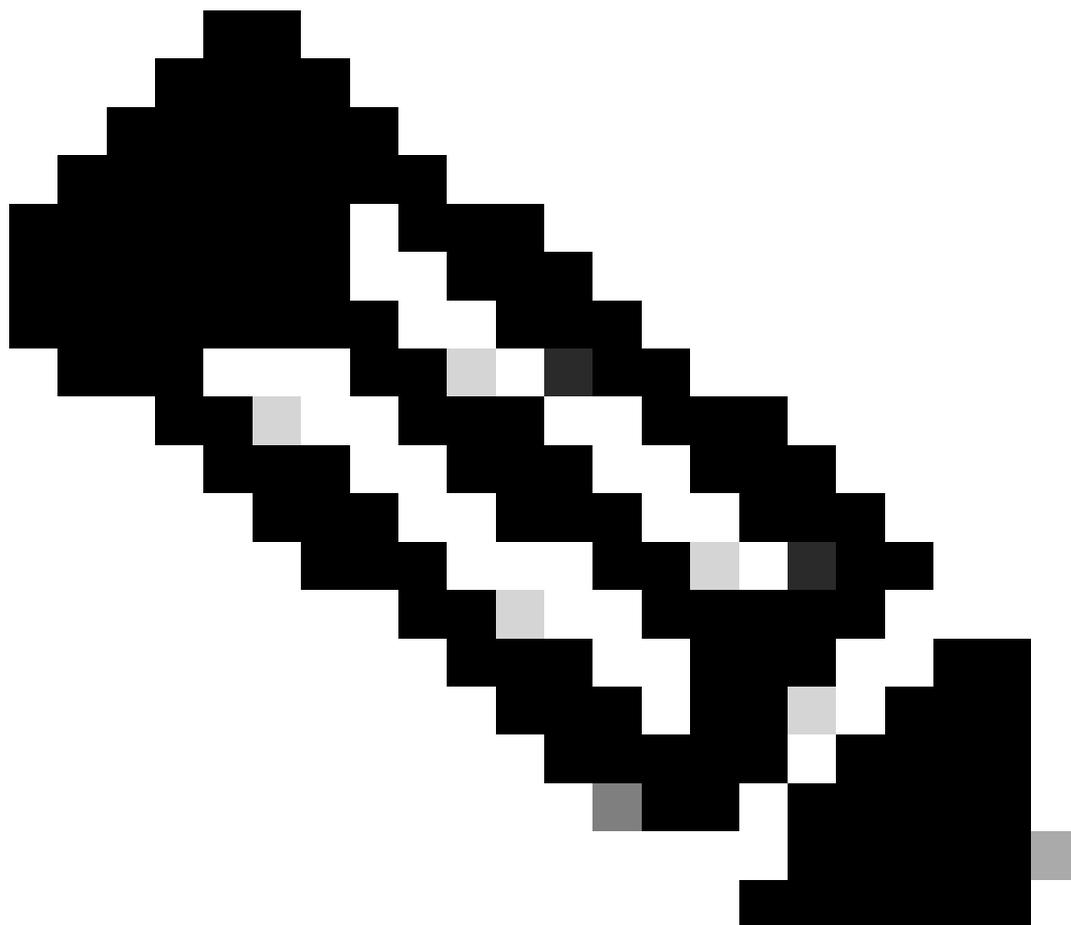
```
RTB#
router bgp 200
 network 172.31.160.0
 neighbor 10.3.3.1 remote-as 300
 neighbor 10.3.3.1 send-community
 neighbor 10.3.3.1 route-map setcommunity out

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

access-list 1 permit 0.0.0.0 255.255.255.255
```



注意：此示例使用route-map setcommunity命令，以便将社区设置为无输出。



注意：必须使用neighbor send-community 命令才能将此属性发送到RTC。

当 RTC 使用 NO_EXPORT 属性获取更新时，RTC 不会将更新传播到外部对等 RTA。

在本示例中，RTB将社区属性设置为**100 200 additive**。此操作会在传输到RTC之前将值100 200添加到任何当前社区值。

```
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
```

社区列表是在路由映射的 match 子句中使用的社区组。使用社区列表可以根据不同的社区编号列表过滤或设置属性。

```
<#root>
```

```
ip community-list <community-list-number> {permit | deny} <community-number>
```

例如，您可以定义路由映射 match-on-community：

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

!--- The community list number is 10.

 set weight 20
 ip community-list 10 permit 200 300

!--- The community number is 200 300.
```

您可以使用社区列表，根据社区值在某些更新中过滤或设置某些参数，如权重和度量。在此部分的第二个示例中，RTB 将更新发送到社区为 100 200 的 RTC。如果 RTC 希望根据这些值设置权重，可以执行如下操作：

```
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

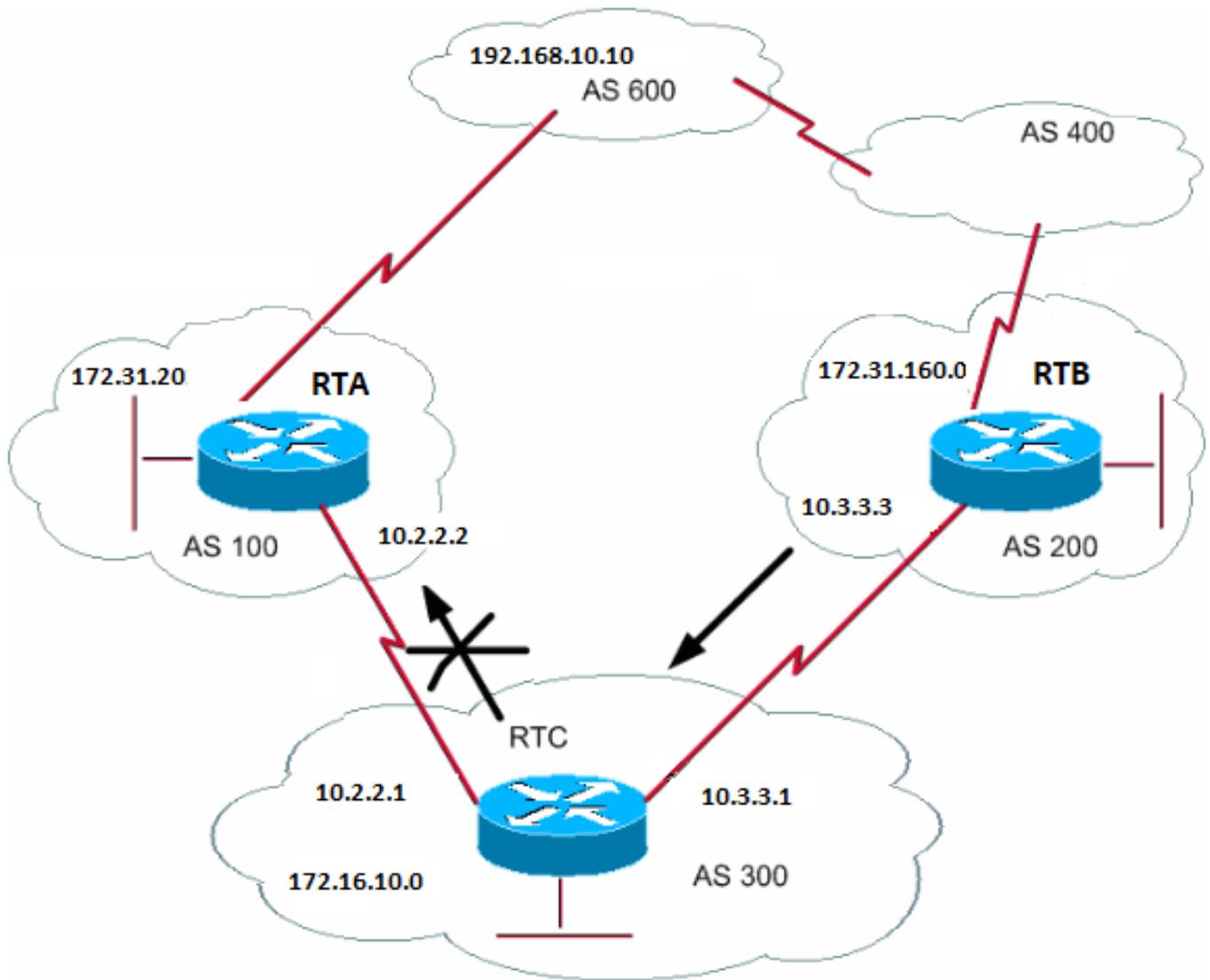
route-map check-community permit 30
  match community 3

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

在本示例中，社区属性为 100 的任何路由都与列表 1 相匹配。此路由的权重设置为 20。任何社区只包含 200 的路由都与列表 2 相匹配，并且权重为 20。关键字 `exact` 表示社区只包含 200，不包含任何其他内容。此处的最后一个社区列表是为了确保不丢弃其他更新。请记住，默认情况下，会丢弃任何与列表不匹配的内容。关键字 `internet` 表示所有路由，因为所有路由都是 Internet 社区的成员。

有关详细信息，请参阅[使用BGP社区值配置和控制上游供应商网络](#)。

BGP 邻居和路由映射



可以将 neighbor 命令与路由映射一起使用，以针对传入或传出的更新过滤或设置参数。

当根据 IP 地址进行匹配时，与 neighbor 语句相关的路由映射对传入的更新无效：

```
<#root>
```

```
neighbor <ip-address> route-map <route-map-name>
```

假设在此部分的图中，希望 RTC 从 AS200 处获知 AS200 内的网络，而不必获知其他网络。并且，希望将接受的路由的权重设置为 20。请联合使用 neighbor 和 as-path 访问列表：

```
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$
```

源于 AS200 的任何更新都包含以 200 开头和结尾的路径信息。将允许这些更新。而将丢弃任何其他更新。

假设您希望：

- 接受源于 AS200 且权重为 20 的更新

- 丢弃源于 AS400 的更新

- 其他更新的权重为 10

```
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 .*
```

此语句将 AS200 内的更新的权重设置为 20。该语句还将AS400之后的更新的权重设置为10，并丢弃来自AS400的更新。

使用 set as-path prepend 命令

在某些情况下，必须对路径信息进行处理，以便影响 BGP 决策过程。对路由映射使用的命令如下：

```
<#root>
```

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

假设在BGP邻居和路由映射部分的图中，RTC将自身的网络172.16.10.0通告给两个不同的AS，即AS100和AS200。当信息传播到AS600时，AS600中的路由器通过两个不同的路由拥有有关172.16.10.0的网络可达性信息。第一个路由是经由 AS100，路径为 (100, 300)，第二个路由是经由 AS400，路径为 (400, 200, 300)。如果所有其他属性都相同，AS600 将选择最短的路径，即选择经由 AS100 的路由。

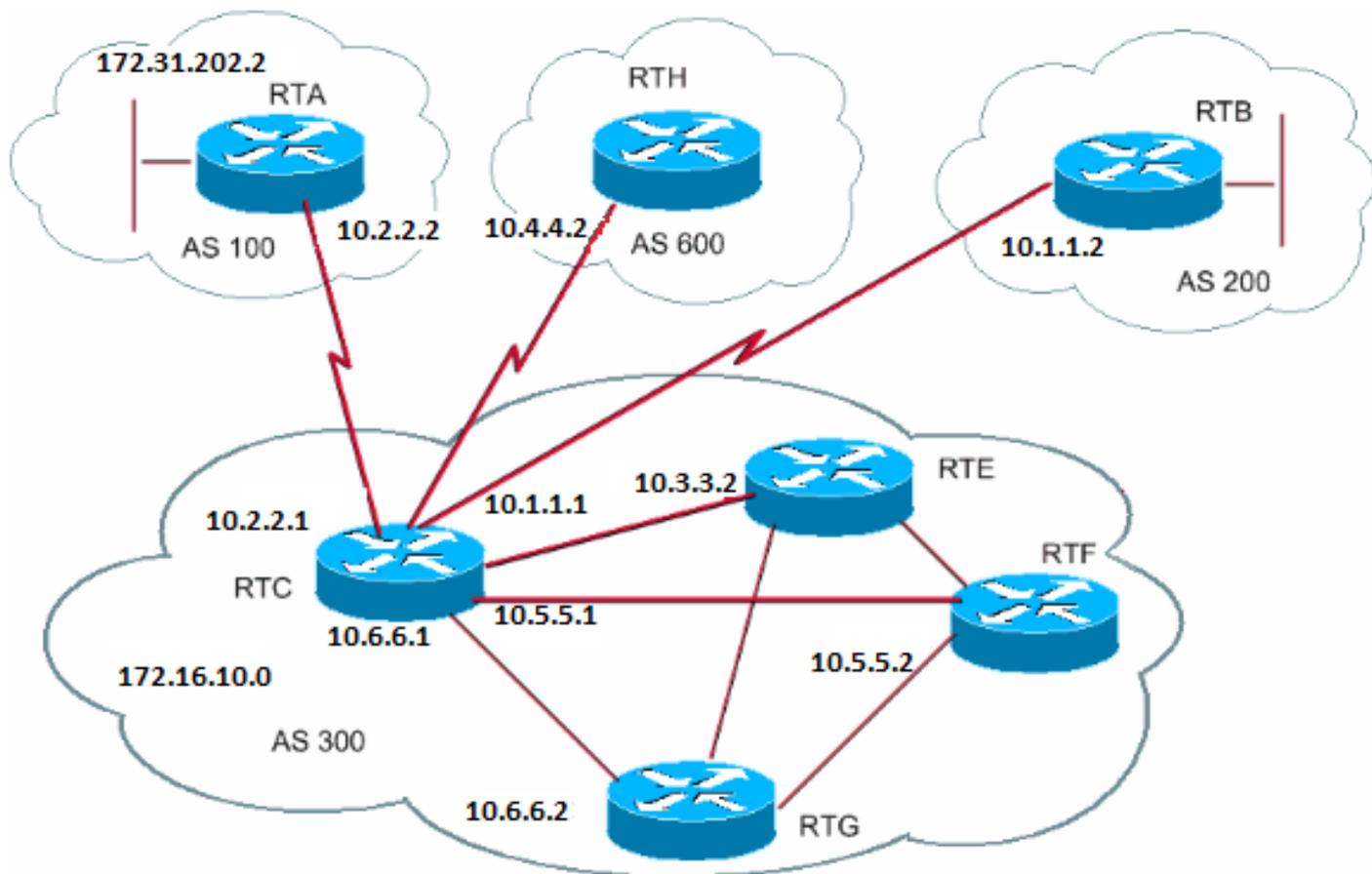
AS300 通过 AS100 获取所有流量。如果希望从 AS300 处影响此路由决策，可以使经由 AS100 的路径看起来比经由 AS400 的路径长一些。如果将AS编号附加到通告给AS100的当前路径信息，则可以这样做。常规做法是以如下方式重复 AS 编号：

```
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
```

由于此配置，AS600通过AS100接收有关172.16.10.0的更新，路径信息为：(100、300、300、300)。此路径信息比 AS600 从 AS400 接收更新的路径 (400, 200, 300) 长。

BGP

对等体组



BGP 对等体组是采用相同更新策略的一组 BGP 邻居。通常由路由映射、分配列表、过滤器列表设置更新策略。您未为每个单独的邻居定义相同的策略；相反，您定义对等体组名称并将这些策略分配给对等体组。

对等体组的成员继承对等体组的所有配置选项。如果这些选项对于出站更新无效，您也可以将成员配置为覆盖这些选项。您只能覆盖针对入站更新设置的选项。

要定义对等体组，可以发出如下命令：

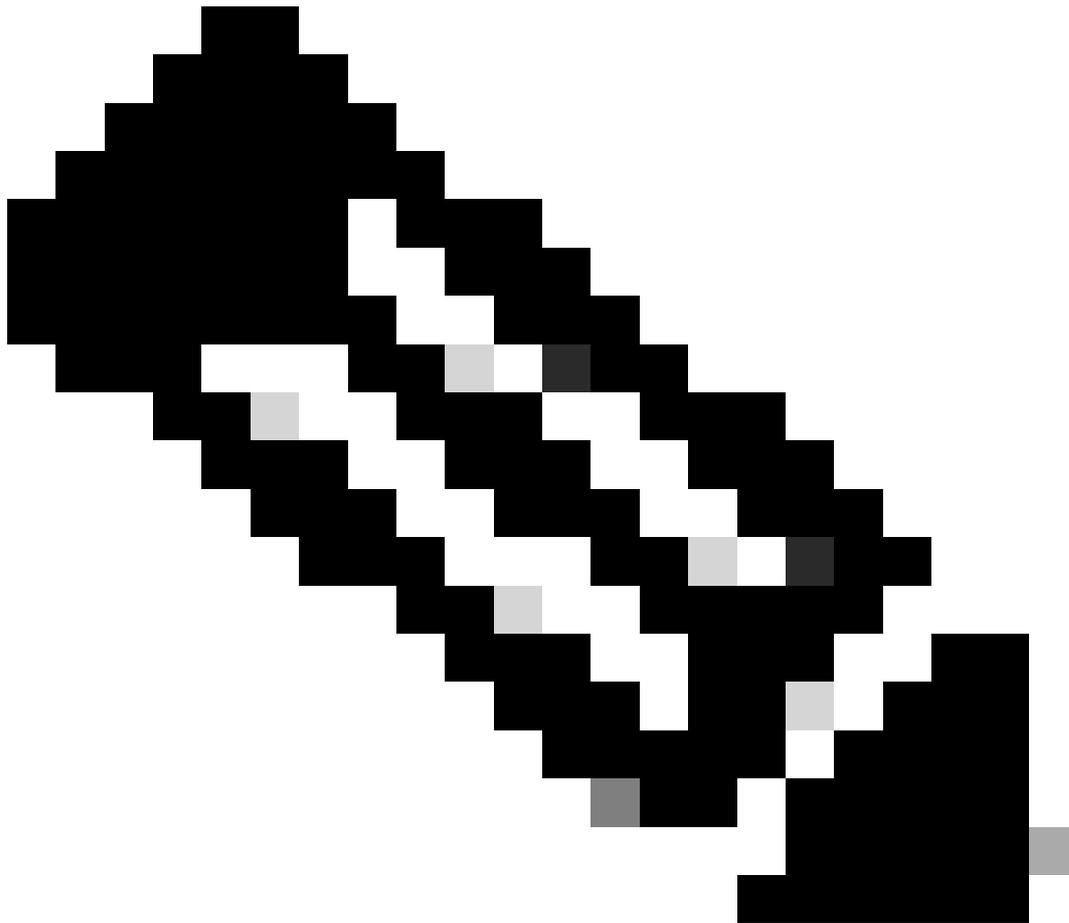
```
<#root>
```

```
neighbor peer-group-name peer-group
```

此示例将对等体组应用到内部和外部 BGP 邻居：

```
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
```

此配置定义一个名称为 internalmap 的对等体组。该配置为对等体组定义某些策略，如定义路由映射 SETMETRIC 以将度量设置为 5，并定义两个不同的过滤器列表 1 和 2。该配置将对等体组应用于所有内部邻居，即 RTE、RTF 和 RTG。并且，该配置为邻居 RTE 定义一个独立的过滤器列表 3。此过滤器列表覆盖对等体组内部的过滤器列表 2。

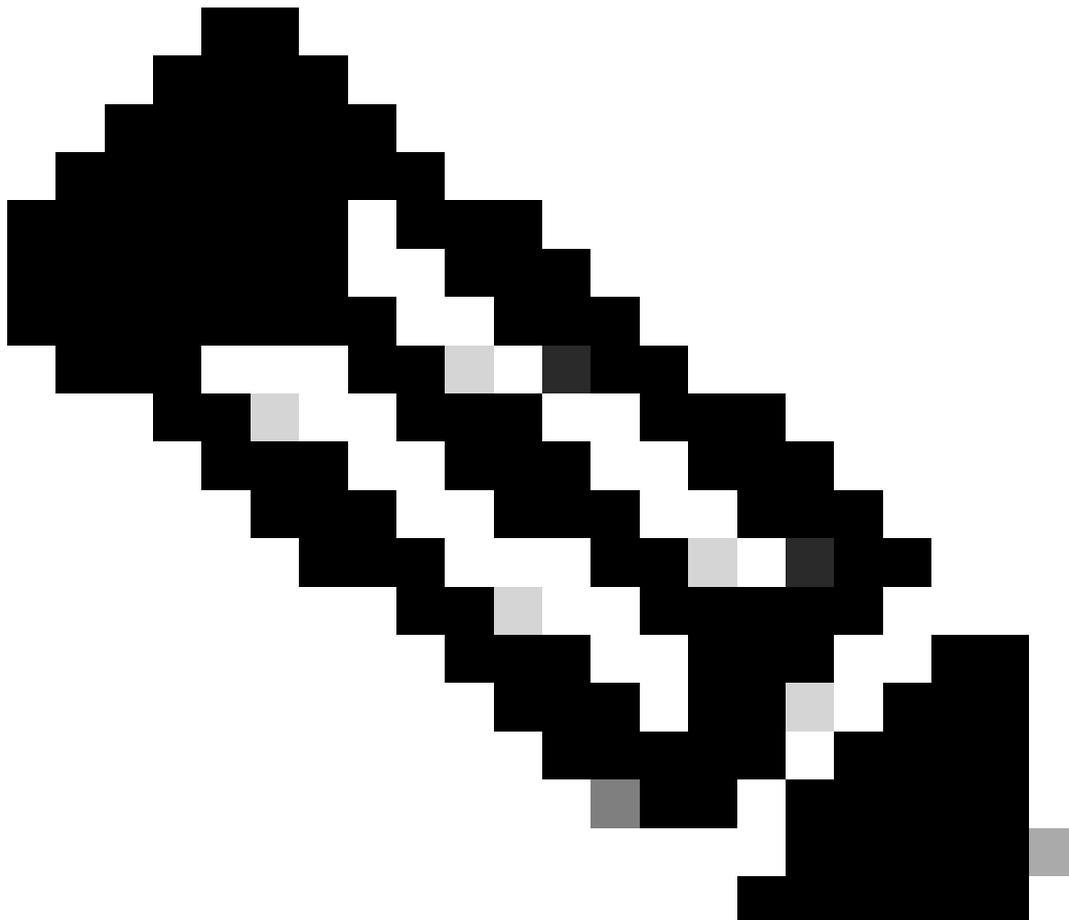


注意：您只能覆盖影响入站更新的选项。

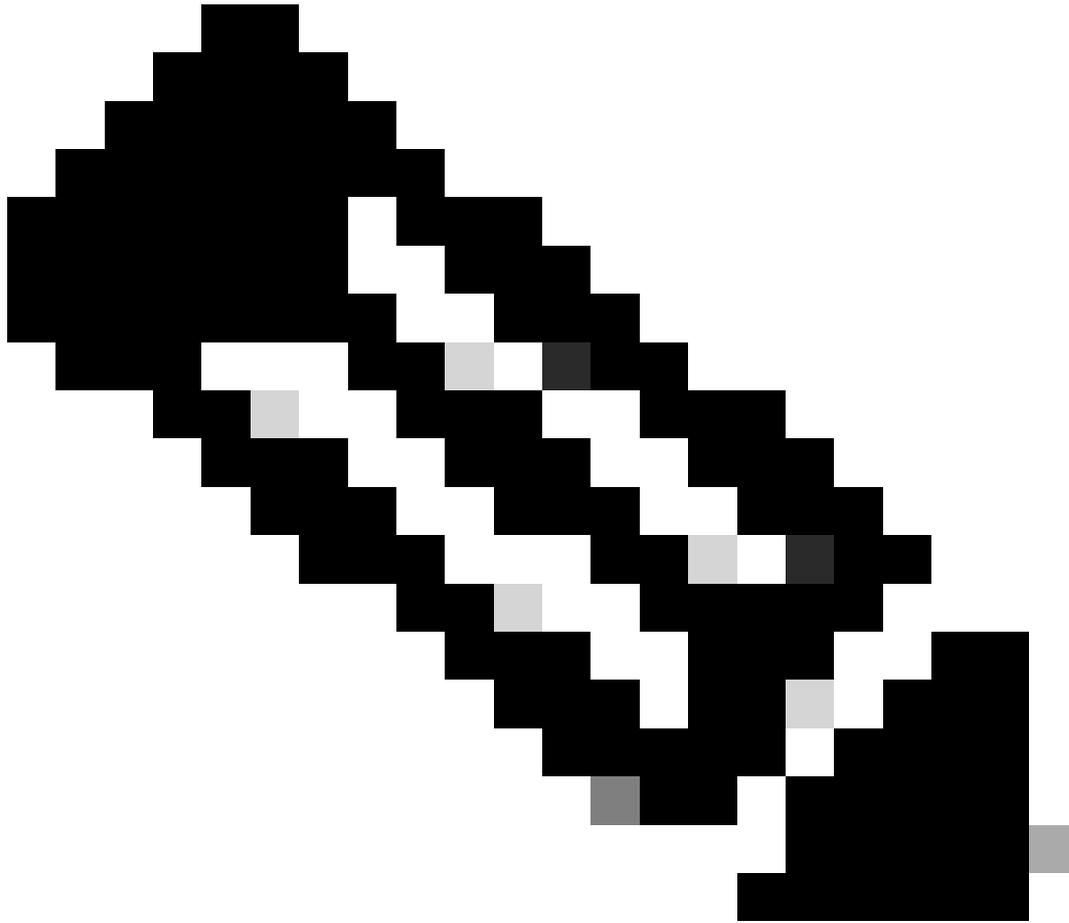
现在，看一看如何针对外部邻居使用对等体组。在本部分的同一个图中，使用对等体组 externalmap 配置 RTC，将该对等体组应用到外部邻居。

```
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
```



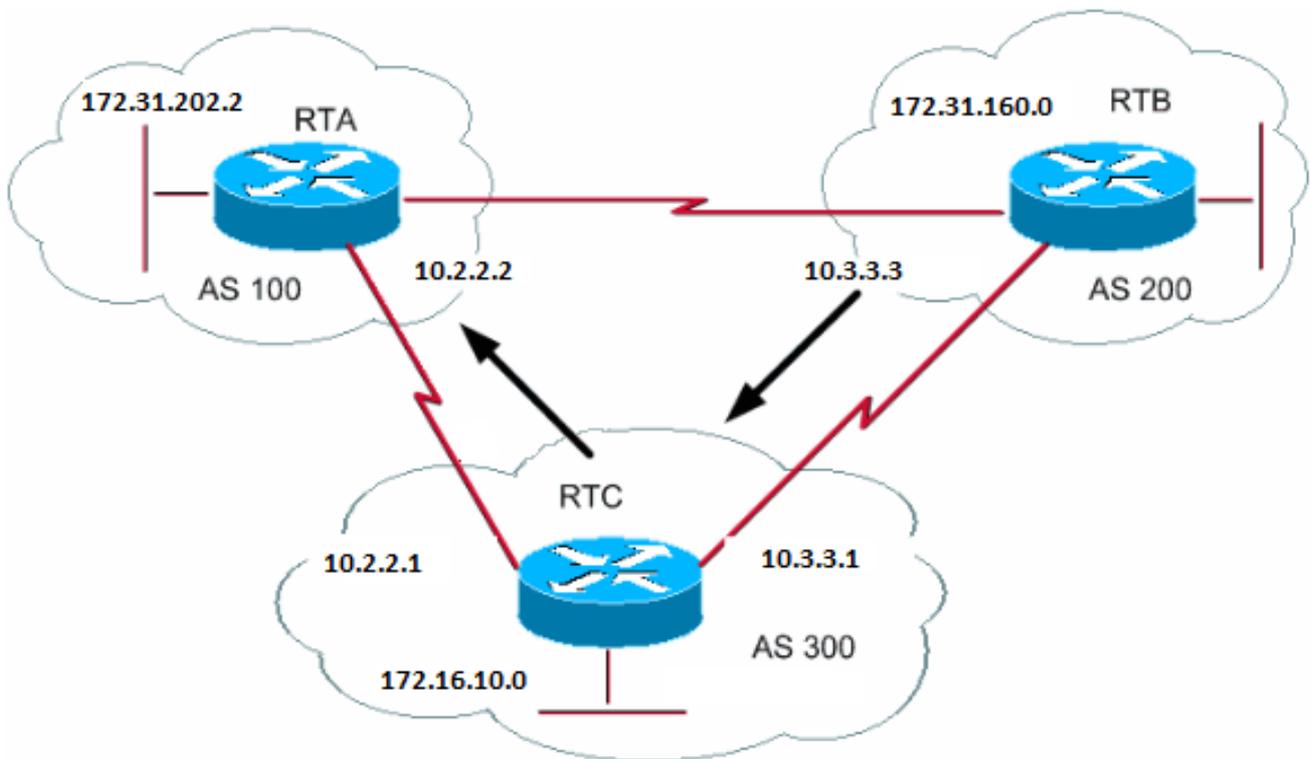
注意：在这些配置中，在对等体组之外定义remote-as语句，因为您必须定义不同的外部AS。并且，通过分配过滤器列表3，将覆盖邻居10.1.1.2的入站更新。有关对等体组的详细信息，请参阅BGP对等体组。



注意：在Cisco IOS软件版本12.0(24)S中，Cisco引入了BGP动态更新对等体组功能。更高的Cisco IOS软件版本中也提供该功能。该功能引入一个新算法，该算法动态计算并优化共享相同出站策略的邻居的更新组。这些邻居可以共享相同的更新消息。在Cisco IOS软件的更早期版本中，BGP更新消息组基于对等体组配置。这种组更新方法对出站策略和特定的会话配置做了限制。BGP动态更新对等体组功能将更新组复制从对等体组配置中分离出来。这种分离缩短了收敛时间，提高了邻居配置的灵活性。有关详细信息，请参阅BGP动态更新对等体组。

BGP 案例分析 4

CIDR 和聚合地址



无类别域间路由 (CIDR) 是 BGP4 相对于 BGP3 的主要改进之一。CIDR 或超网是查看 IP 地址的新途径。使用 CIDR 时，不存在类 (如 A 类、B 类或 C 类) 的概念。例如，网络 192.168.213.0 曾经是非法的 C 类网络。现在，该网络是合法的超网 192.168.213.0/16。16 表示从 IP 地址最左侧开始计算时，子网掩码中的位数。此表示法类似于 192.168.213.0 255.255.0.0。

可以使用聚合最小化路由表的大小。聚合是合并一些不同路由的特征的过程，该过程使通告单个路由成为可能。在本示例中，RTB 生成网络 172.31.160.0。配置 RTC，使其将路由 192.168.160.0 的超网传播到 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

```

RTC 将聚合地址 192.168.160.0 传播给 RTA。

聚合命令

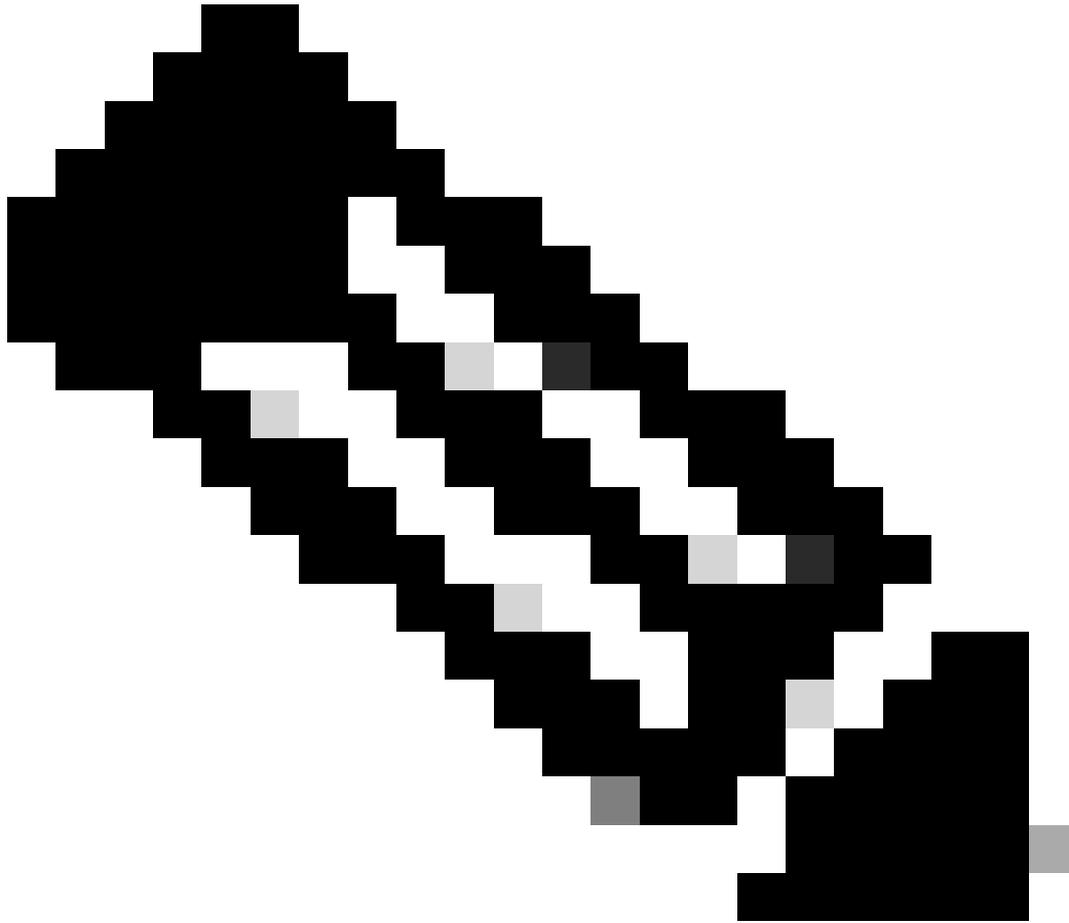
有很多种聚合命令。必须了解每个命令的作用，以便按需要进行聚合。

第一个命令来自CIDR和聚合地址部分的示例：

```
<#root>
```

```
aggregate-address address-mask
```

此命令通告前缀路由以及所有更加具体的路由。命令**aggregate-address 192.168.160.0** 传播附加网络192.168.160.0，但不阻止将172.31.160.0传播到RTA。结果是将网络 192.168.160.0 和 172.31.160.0 都传播到 RTA，既通告了前缀又通告了更加具体的路由。



注意：如果您在BGP路由表中没有该地址的更具体的路由，则无法聚合该地址。

例如，如果RTB在BGP表中没有更具体的条目192.168.160.0，则RTB无法为192.168.160.0生成聚合。可以将更具体的路由注入 BGP 表。通过以下项注入路由：

- 来自其他 AS 的传入更新

-

将 IGP 或静态路由重分配到 BGP

-

network 命令，如 network 172.31.160.0

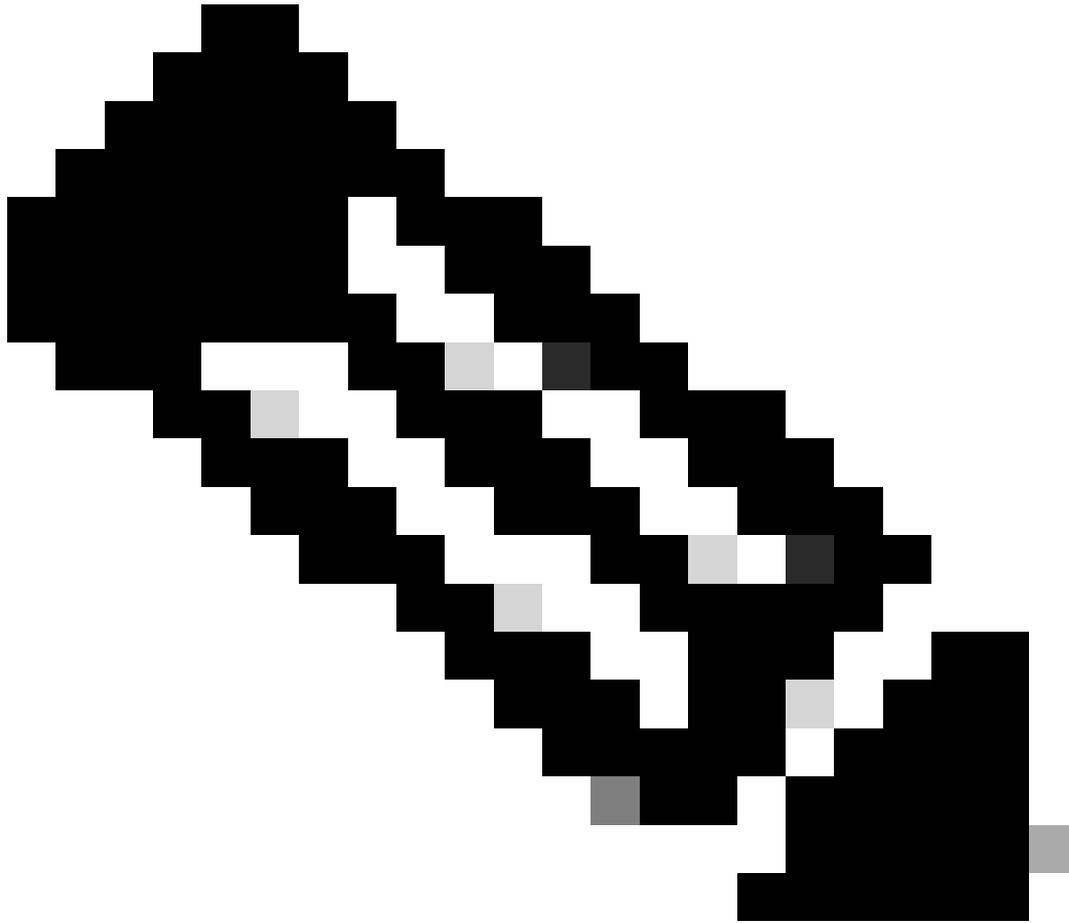
如果希望RTC只传播网络192.168.160.0，而不传播更具体的路由，请发出以下命令：

```
<#root>
```

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

此命令只通告前缀。该命令抑制所有更具体的路由。

命令**aggregate 192.168.160.0 255.0.0.0 summary-only** 传播网络192.168.160.0，抑制更具体的路由172.31.160.0。



注意：如果聚合一个通过network语句注入BGP的网络，则该网络项始终注入BGP更新中。即使使用 aggregate summary-only 命令，此注入仍会进行。CIDR示例1部分中的示例讨论了此种情况。

<#root>

aggregate-address <address> <mask> as-set

此命令通告前缀和更具体的路由。但此命令在路由更新的路径信息中包含 as-set 信息。

```
<#root>
```

```
aggregate 192.168.0.0 255.0.0.0 as-set
```

CIDR 示例2 (as-set)部分讨论了此命令。

聚合时如果希望抑制更具体的路由，可以定义一个路由映射，并将路由映射应用到聚合中。使用该操作可以更方便地选择抑制哪些更具体的路由。

```
<#root>
```

```
aggregate-address <address> <mask> suppress-map <map-name>
```

此命令通告前缀和更具体的路由。但是此命令根据路由映射抑制通告。假设在 CIDR 和聚合地址部分的图中，希望聚合 192.168.160.0，抑制更具体的路由 192.168.160.20，允许传播 172.31.160.0。请使用此路由映射：

```
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
```

通过定义 suppress-map，会抑制访问列表所允许的所有数据包的更新。

然后，将路由映射应用到 aggregate 语句。

```
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
```

以下是另一种变化方式：

<#root>

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

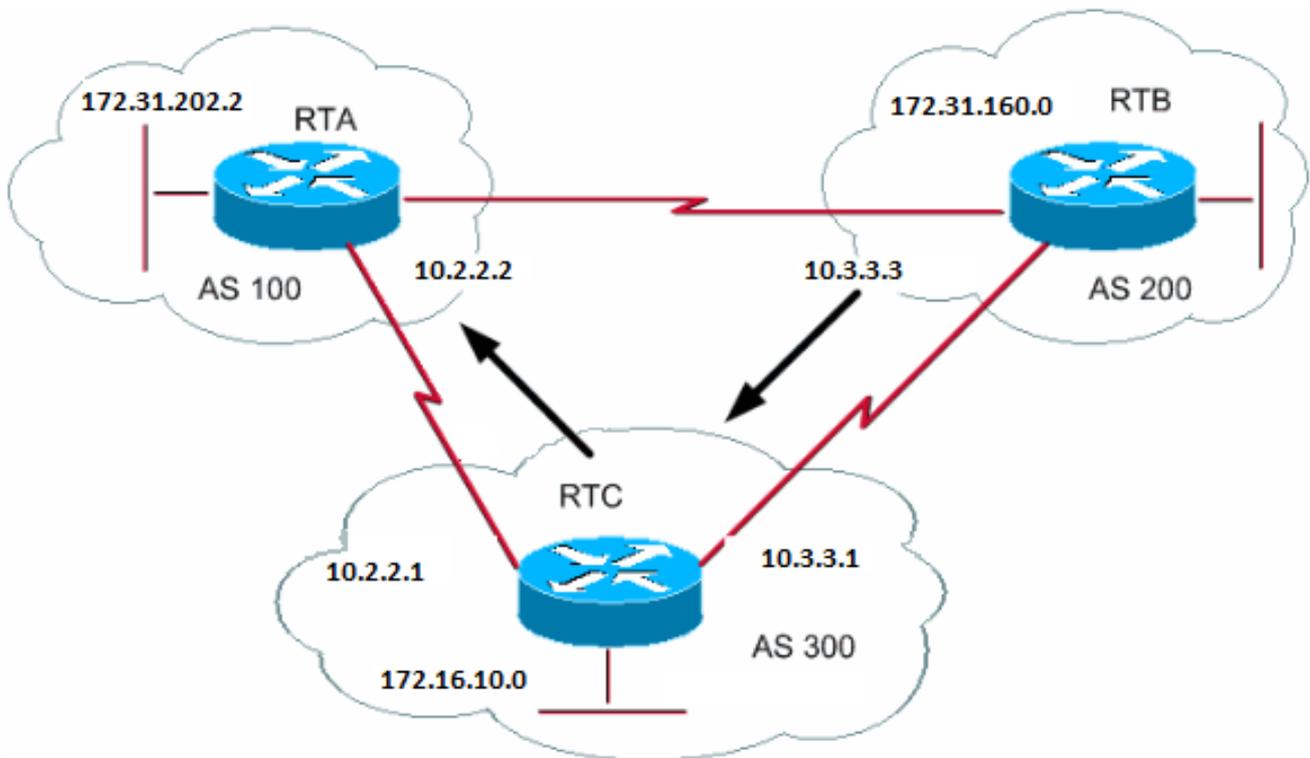
使用此命令可以在发送聚合时设置属性，如度量。要将聚合的源设置为 IGP，可以将此路由映射应用到 aggregate attribute-map 命令：

```
route-map SETMETRIC
 set origin igp

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

有关详细信息，请参阅[了解BGP中的路由聚合](#)。

CIDR 示例 1



请求：允许RTB通告前缀192.168.160.0并抑制所有更具体的路由。此请求的问题是网络172.31.160.0位于AS200本地，这意味着AS200是172.31.160.0的发起方。在不为172.31.160.0生成条目的情况下，RTB不能为192.168.160.0生成前缀，即使使用aggregate summary-only命令也是如此。RTB将生成这两个网络，因为RTB是172.31.160.0的源。此问题有两个解决方法。

第一个解决方法是使用静态路由并将其重分配到BGP。结果是RTB通告聚合时源是不完整的(?)。

```
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
```

在第二个解决方法中，除静态路由之外，还为network命令添加条目。该条目可以起到基本相同的作用，只是将更新的源设置为IGP。

```
RTB#
```

```

router bgp 200
 network 192.168.160.0 mask 255.0.0.0

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

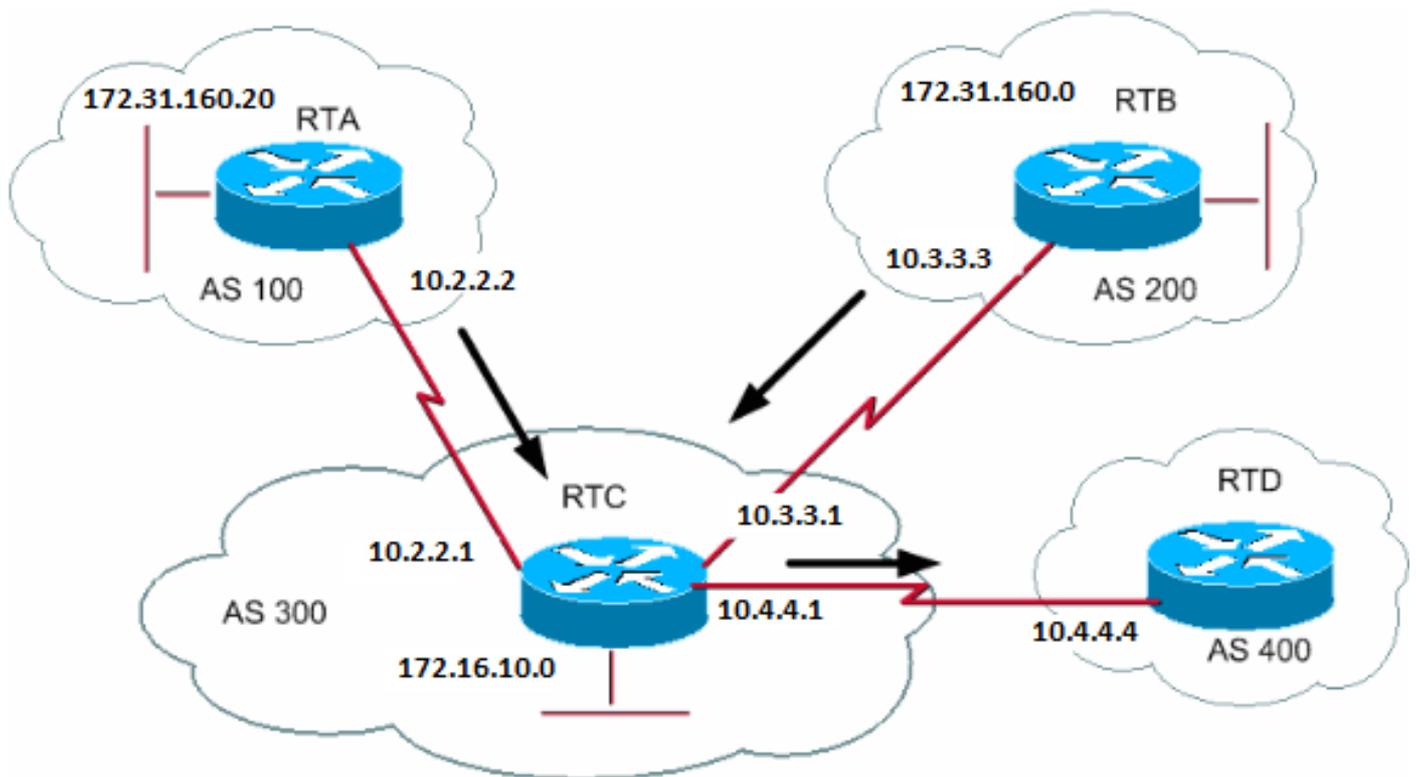
 neighbor 10.3.3.1 remote-as 300
 redistribute static

ip route 192.168.160.0 255.0.0.0 null0

```

CIDR 示例 2 (as-set)

在聚合中使用 as-set 语句以减小路径信息的大小。使用 as-set 语句时，AS 编号不管在被聚合的多条路径中显示过多少次，都只列出一一次。在信息聚合导致有关路径属性的信息丢失时，将使用 aggregate as-set 命令。在本示例中，RTC 从 RTA 处获取有关 192.168.160.20 的更新，从 RTB 处获取有关 172.31.160.0 的更新。假设 RTC 希望聚合网络 192.168.160.0/8 并将该网络发送至 RTD。RTD 不知道该路由的源。如果添加 aggregate as-set 语句，则强迫 RTC 以集合 {} 形式生成路径信息。该集合包含所有路径信息，而不论先经过哪条路径。



```

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

```

第 1 种情况：

RTC 没有 as-set 语句。RTC向RTD发送带有路径信息(300)的更新192.168.160.0/8，就像该路由源自AS300一样。

```
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.
```

第 2 种情况：

```
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}.
```

以下两个主题BGP联盟和路由反射器针对的是希望进一步控制AS内iBGP对等急剧增长的Internet服务提供商(ISP)。

BGP 联盟

实施 BGP 联盟可以减少 AS 内的 iBGP 网格。其中的技巧是将一个 AS 分为多个 AS，并将整个组分配给单一的联盟。每个 AS 内的 iBGP 已完全网格化，每个 AS 与联盟内的其他 AS 都有连接。虽然这些 AS 对于联盟内的 AS 有 eBGP 对等体，但这些 AS 交换路由时看起来像在使用 iBGP。联盟通过这种方式保留下一跳、度量和本地优先级信息。对于外界来说，联盟看起来像一个单一的 AS。

要配置 BGP 联盟，可以发出如下命令：

<#root>

`bgp confederation identifier <autonomous-system>`

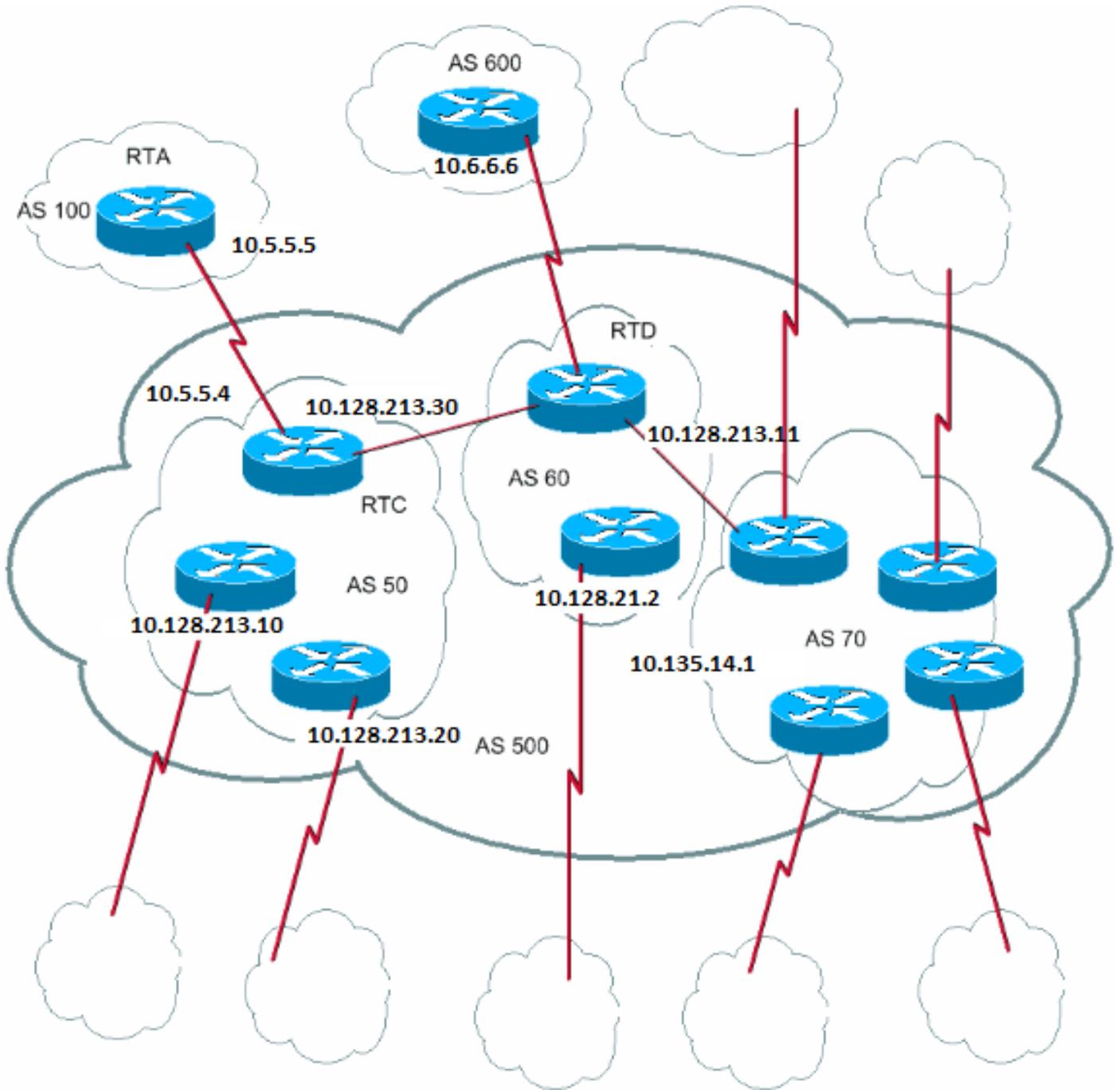
联盟标识符是联盟组的 AS 编号。

发出此命令将在联盟内的多个 AS 之间建立对等：

<#root>

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

以下是联盟的示例：



假设您有一个包含九个 BGP 扬声器的 AS500。其他非 BGP 扬声器也存在，但您只对与其他 AS 有 eBGP 连接的 BGP 扬声器有兴趣。如果希望在 AS500 内建立完整的 iBGP 网格，则每个路由器需要九个对等连接。您需要八个 iBGP 对等体以及外部 AS 的一个 eBGP 对等体。

如果使用联盟，可以将AS500划分为多个AS：AS50、AS60和AS70。您为 AS 设置的联盟标识符为 500。外界只看到一个 AS，即 AS500。对于AS50、AS60和AS70，您分别定义一个iBGP对等体的完整网格，并使用**bgp confederation peers** 命令定义联盟对等体列表。

以下是 RTC、RTD 和 RTA 路由器配置示例：

注意：RTA不了解AS50、AS60或AS70。RTA 只知晓 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)
```

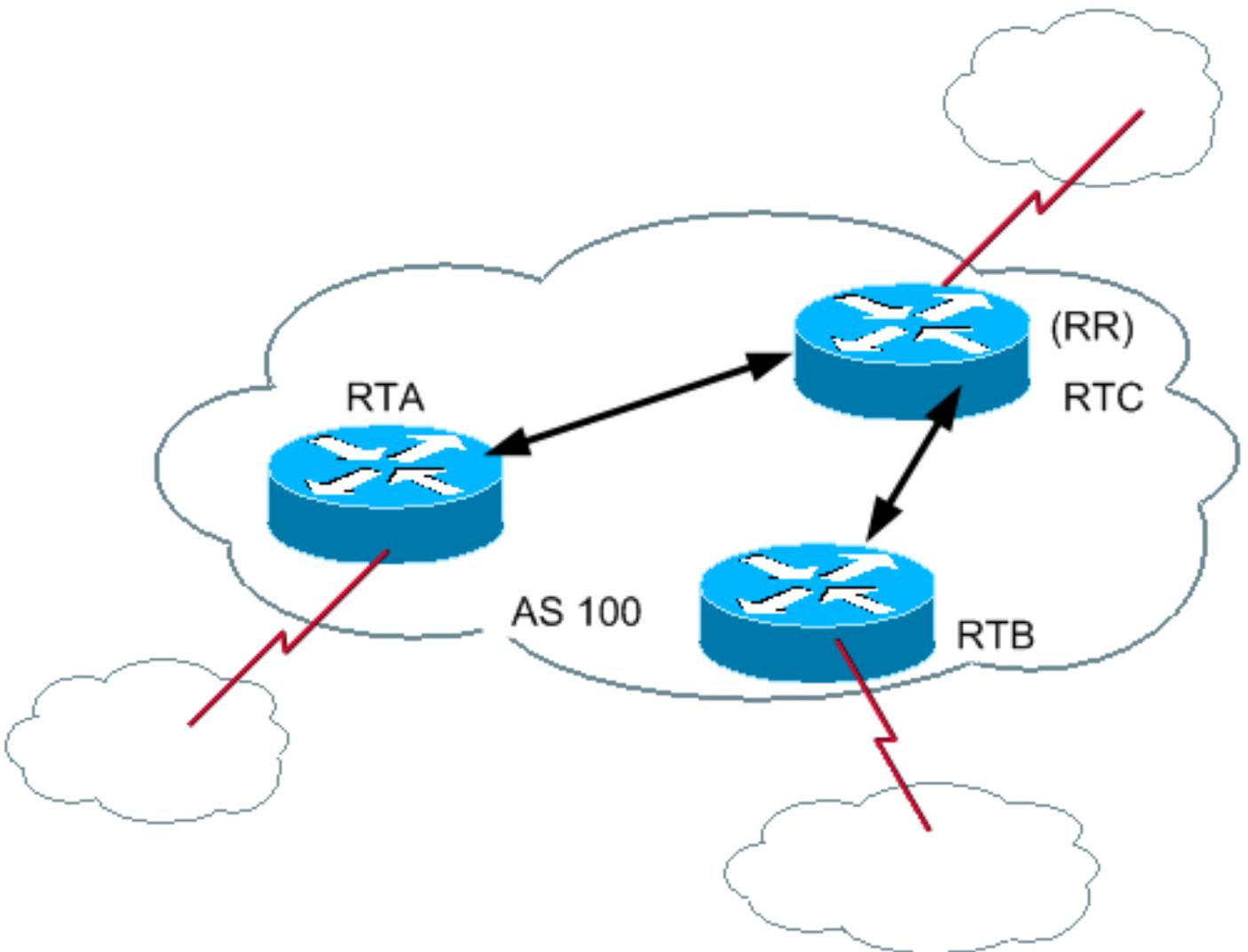
```
RTD#
router bgp 60
```

```
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)
```

路由反射器

应对 AS 内 iBGP 对等体急剧增长的另一个解决方法是使用路由反射器 (RR)。如 iBGP 部分所述，BGP 扬声器并不会将通过其他 iBGP 扬声器获知的路由通告给第三个 iBGP 扬声器。可以稍稍放宽此限制，提供额外的控制，允许路由器将通过 iBGP 获知的路由通告或反射给其他 iBGP 扬声器。此路由反射可减少 AS 内 iBGP 对等体的数量。



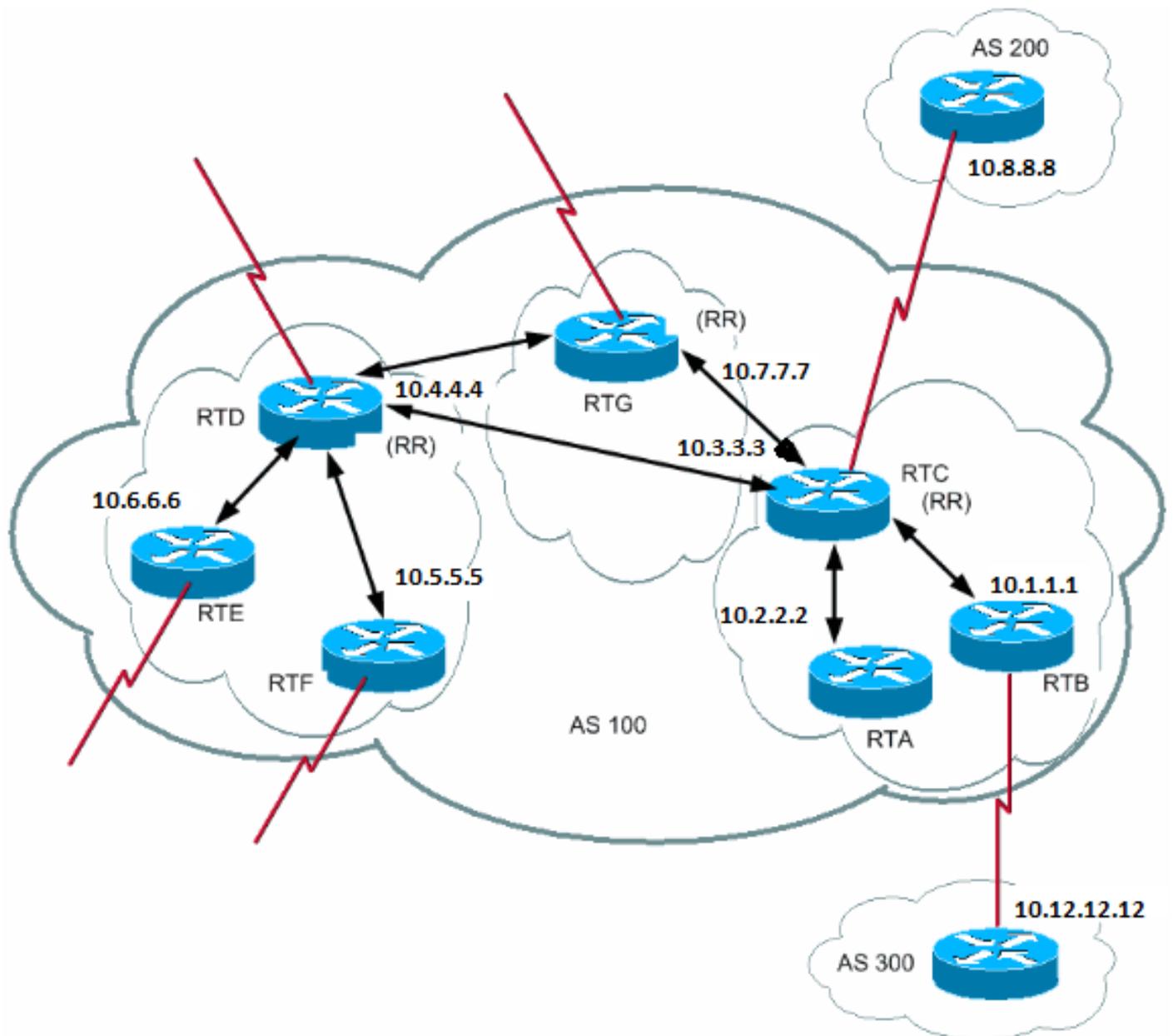
在正常情况下，将在 AS100 内的 RTA、RTB 和 RTC 之间保留完整的 iBGP 网络。如果利用路由反射器的概念，可选择 RTC 作为路由反射器。通过这种方式，RTC 与 RTA 和 RTB 之间有部分 iBGP 对等。RTA 和 RTB 之间的对等是不必要的，因为 RTC 是来自 RTA 和 RTB 的更新的路由反射器。

<#root>

[neighbor <ip address> route-reflector-client](#)

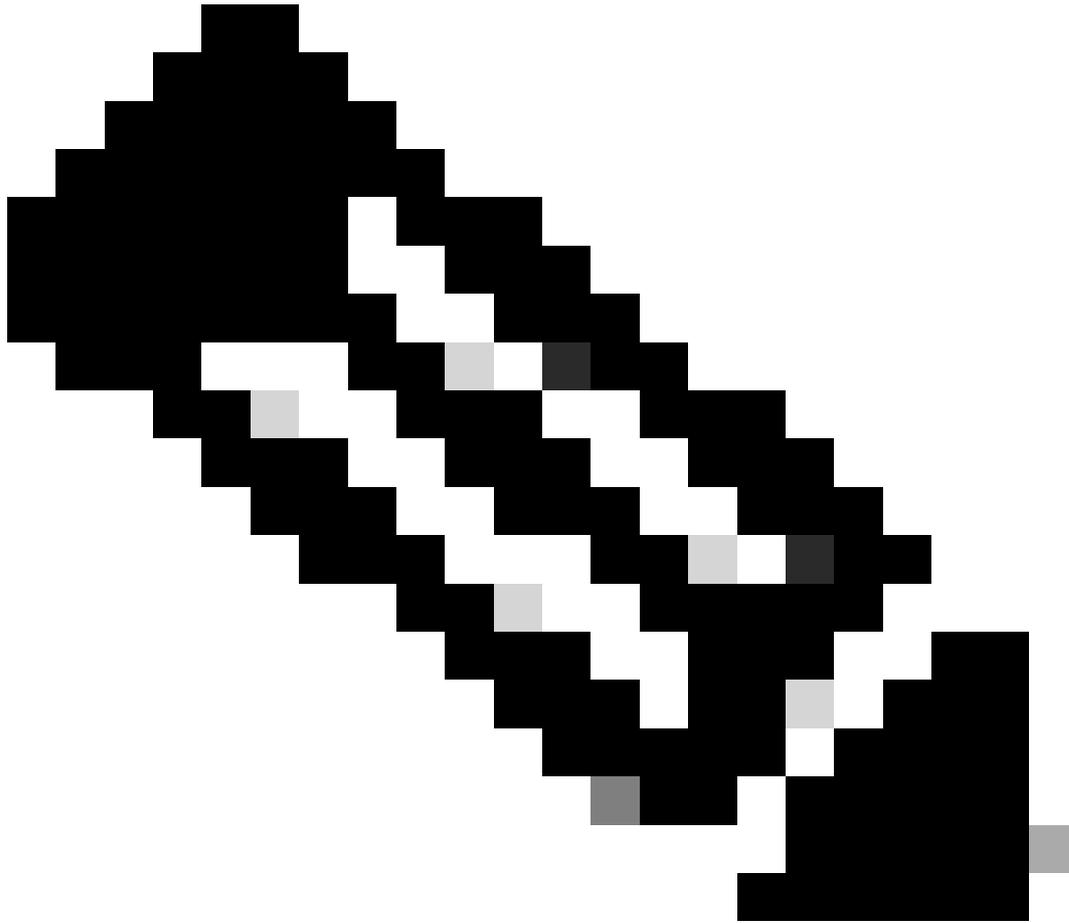
使用此命令的路由器即路由反射器，命令所指的邻居即该路由反射器的客户端。在本示例中，RTC 配置使用指向 RTA 和 RTB IP 地址的 `neighbor route-reflector-client` 命令。路由反射器和客户端的组合称作“群集”。在本示例中，RTA、RTB 和 RTC 形成一个群集，在 AS100 内有单个路由反射器。

非客户端的路由反射器的其他 BGP 对等体为非客户端。



一个 AS 可以有多个路由反射器。在此种情况下，路由反射器将其他路由反射器视为任何其他 iBGP 扬声器。其他路由反射器可以属于同一个群集（客户端组），也可以属于其他群集。在简单配置中，可以将 AS 分为多个群集。在完全网格化的拓扑中，配置每个路由反射器，并将其他路由反射器作为非客户端对等体。客户端不得与客户端群集外部的 iBGP 扬声器对等。

在上图中，RTA、RTB 和 RTC 组成了一个集群。RTC 是路由反射器。对于 RTC 来说，RTA 和 RTB 是客户端，其他路由器都是非客户端。请记住，`neighbor route-reflector-client` 命令指向路由反射器的客户端。客户端 RTE 和 RTF 的路由反射器同样是 RTD。RTG 是第三个群集中的路由反射器。



注意：RTD、RTC和RTG是全网状，但集群中的路由器不是全网状。

当一个路由反射器收到一个路由时，该路由反射器将按下表所示进行路由。然而，此行为取决于对等体类型：

-

从非客户端对等体进行路由 — 反射到群集内的所有客户端。

-

从客户端对等体进行路由 — 反射到所有非客户端对等体和客户端对等体。

-

从 eBGP 对等体进行路由 — 将更新发送到所有客户端和非客户端对等体。

以下是路由器 RTC、RTD 和 RTB 的相关 BGP 配置：

```
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
```

因为存在通过 iBGP 获知的路由的反射，所以可能会形成路由信息环路。路由反射器方案提供了避免这种环路的几种方法：

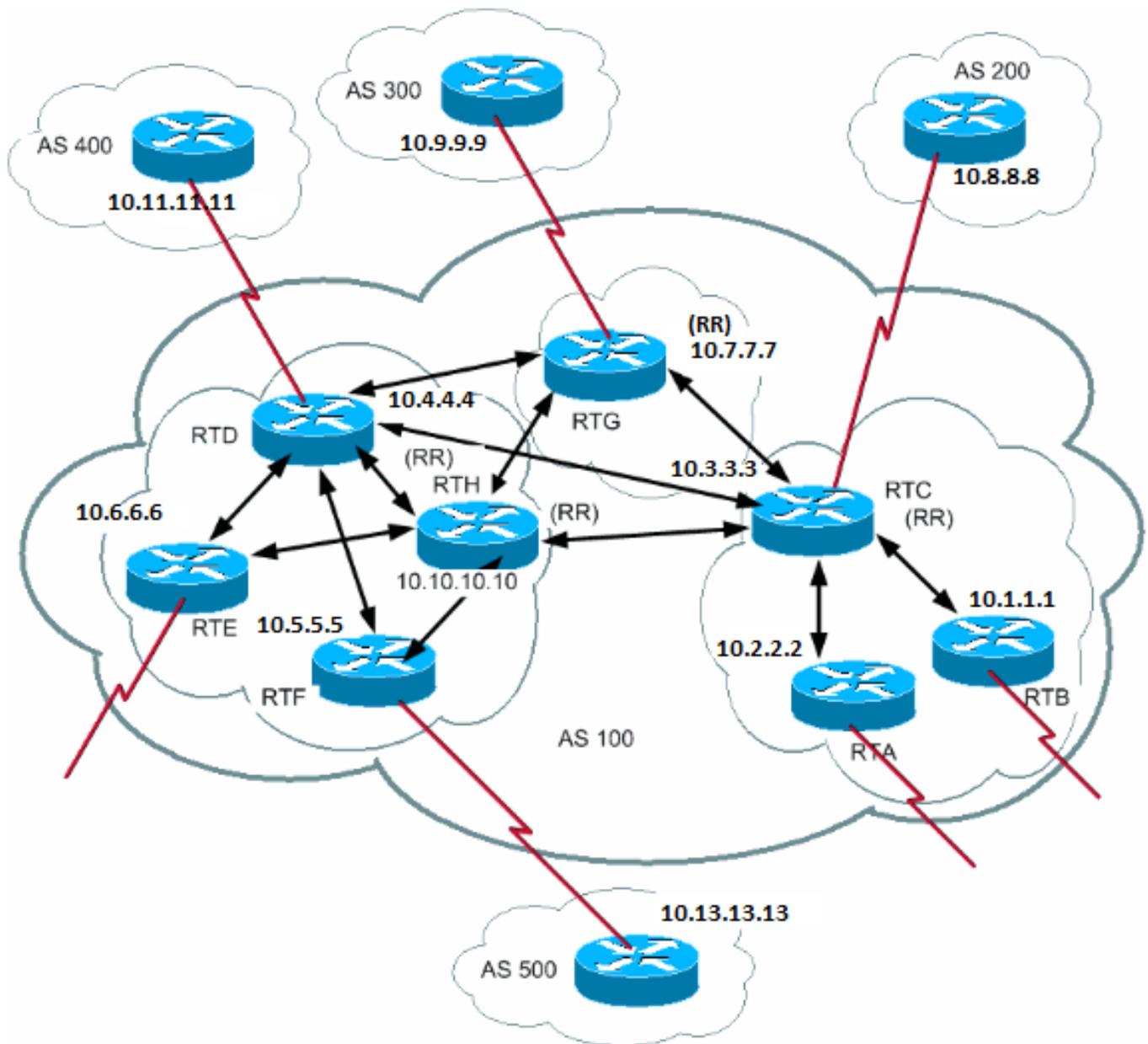
-

originator-id — 为可选的、非传递性的 BGP 属性，长度为 4 个字节。此属性由路由反射器创建。该属性在本地 AS 内传递路由的源的路由器 ID (RID)。如果配置不合适，路由信息将返回源并被忽略。

-

cluster-list — “群集内的多个路由反射器”部分介绍了群集列表。

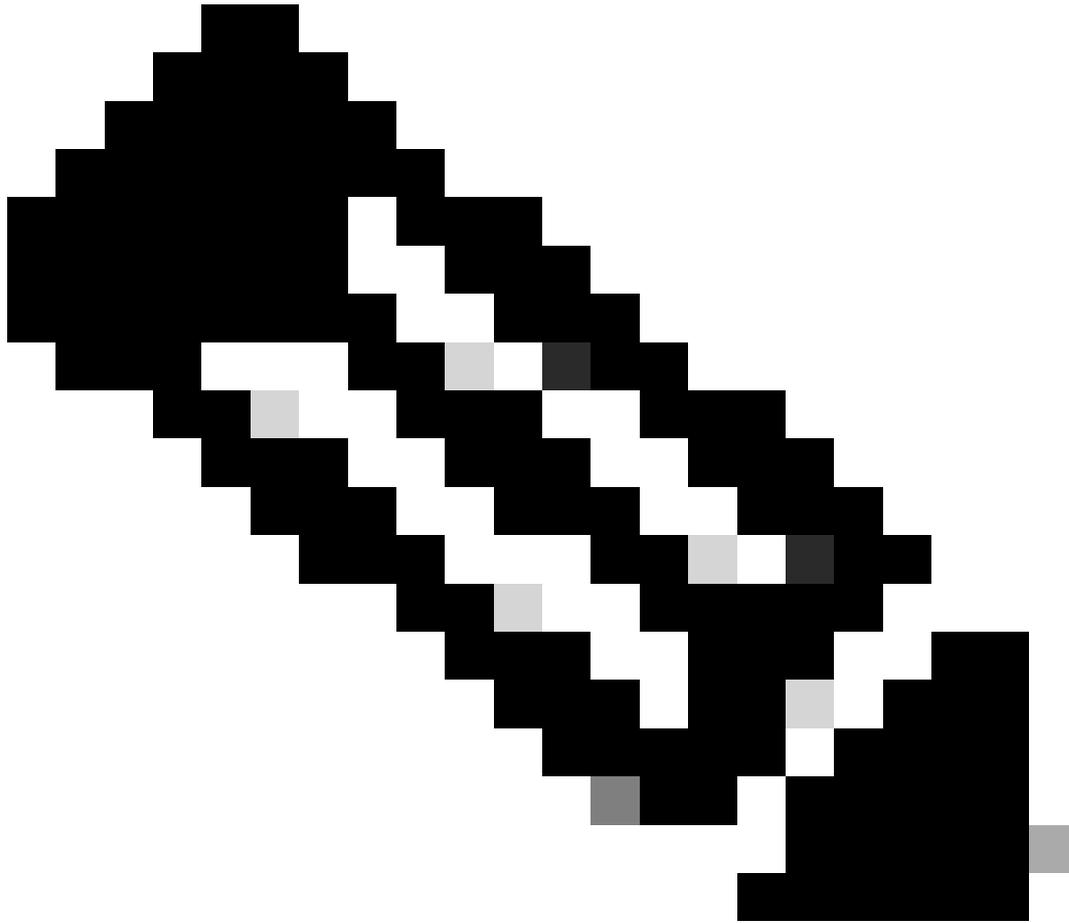
群集内的多个路由反射器



通常，客户端群集中只包含一个路由反射器。在这种情况下，路由反射器的路由器 ID 可以标识该群集。要增加冗余并避免单点故障，群集内可包含多个路由反射器。需要为同一群集内的所有路由反射器配置 4 个字节的群集 ID，这样路由反射器可以识别来自同一群集内的路由反射器的更新。

群集列表是路由所传递的群集 ID 的序列。当路由反射器反射从路由反射器客户端到群集外的非客户端的路由时，路由反射器将本地群集 ID 附加到群集列表。如果此更新包含空的群集列表，路由反射器将创建一个列表。使用此属性，路由反射器可以标识路由信息是否因为配置不合适而环回到同一群集。如果在群集列表中发现本地群集 ID，则忽略该通告。

在此部分的图中，RTD、RTE、RTF 和 RTH 属于同一个群集。RTD 和 RTH 都是同一个群集的路由反射器。



注意：由于RTH与所有RR均采用全网状对等，因此存在冗余。如果 RTD 失败，RTH 将替换 RTD。

RTH、RTD、RTF 和 RTC 的配置如下：

```
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#

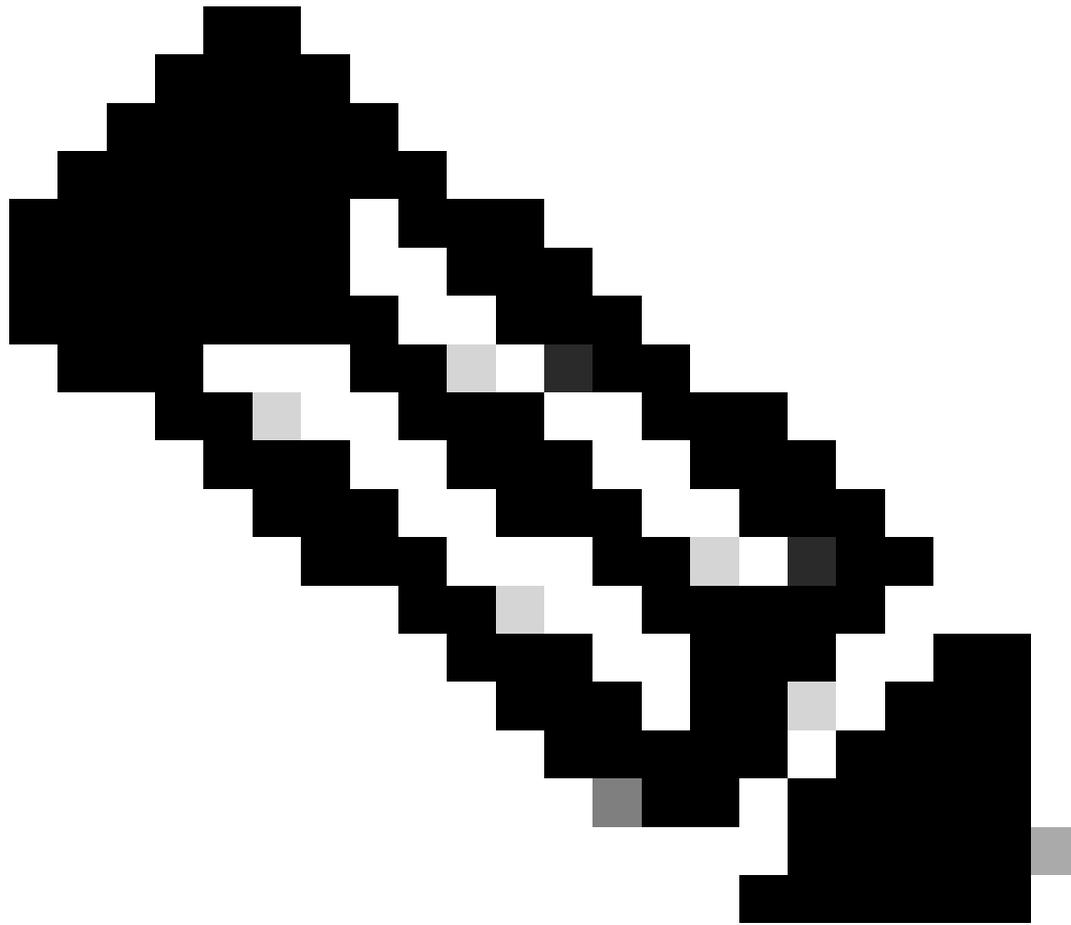
```
router bgp 100
neighbor 10.10.10.10 remote-as 100
neighbor 10.5.5.5 remote-as 100
neighbor 10.5.5.5 route-reflector-client
neighbor 10.6.6.16 remote-as 100
neighbor 10.6.6.16 route-reflector-client
neighbor 10.7.7.7 remote-as 100
neighbor 10.3.3.3 remote-as 100
neighbor 10.11.11.11 remote-as 400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

RTF#

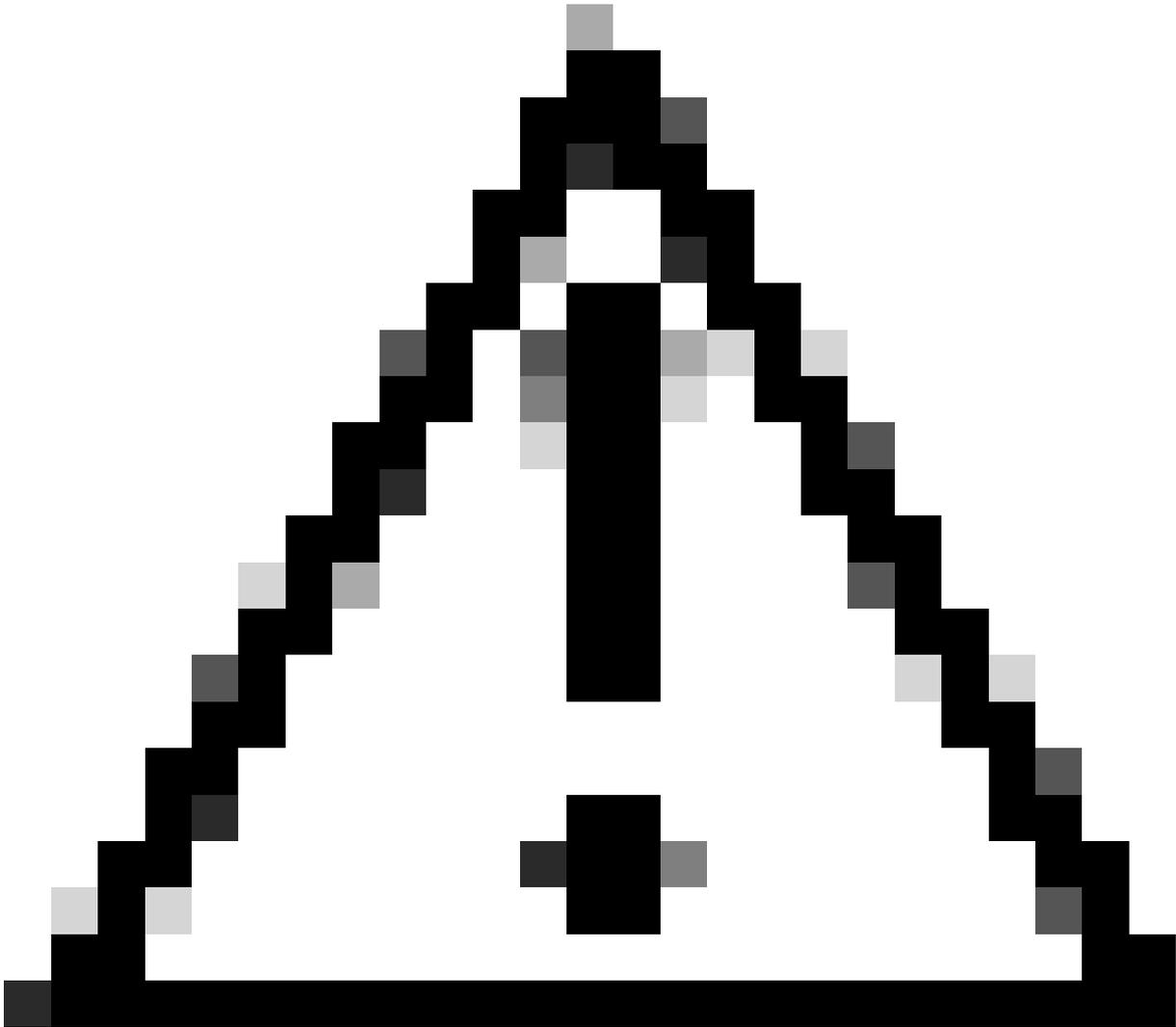
```
router bgp 100
neighbor 10.10.10.10 remote-as 100
neighbor 10.4.4.4 remote-as 100
neighbor 10.13.13.13 remote-as 500
```

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
```



注意：对于RTC，不需要bgp cluster-id命令，因为该群集中只存在一个路由反射器。



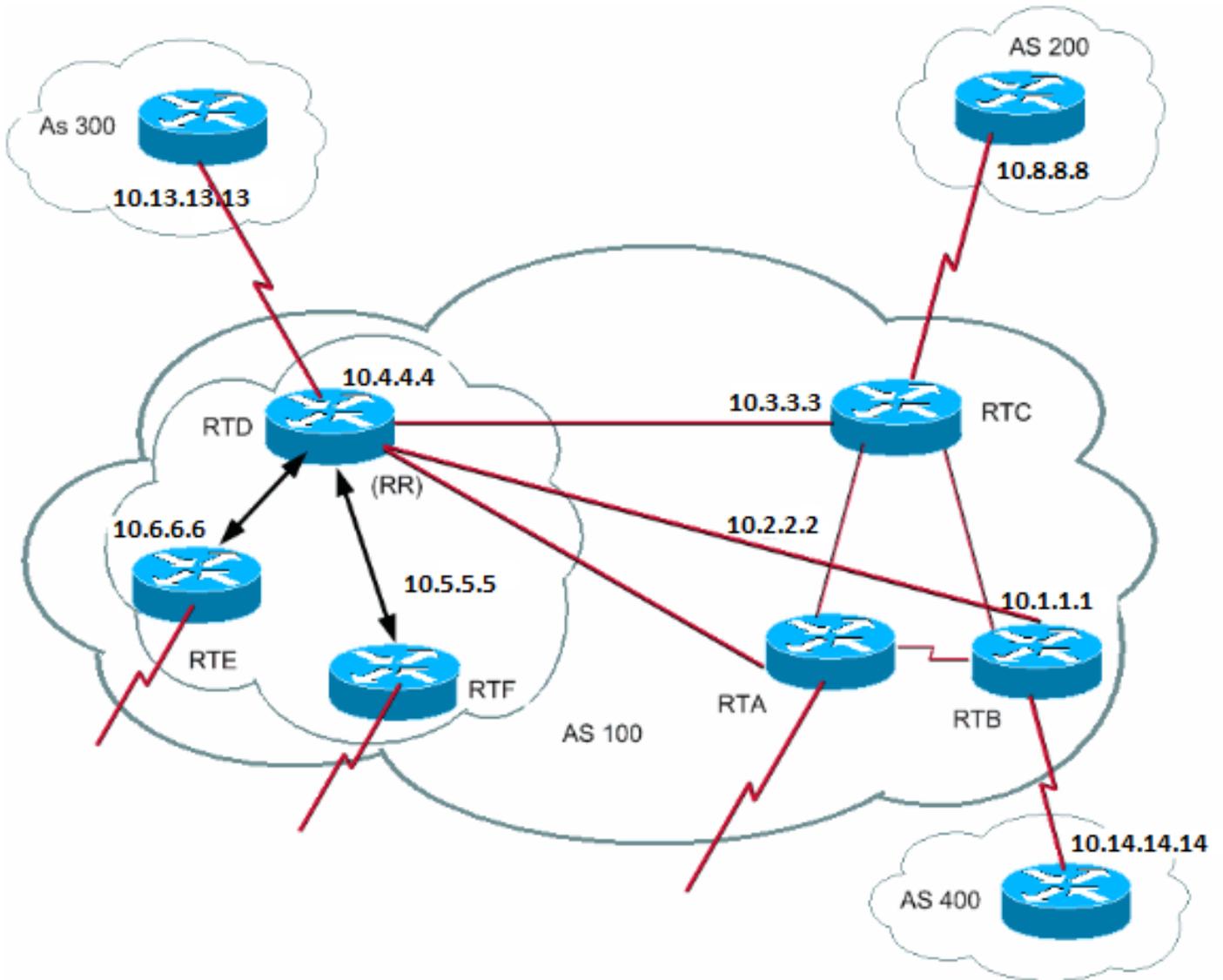
注意：此配置不使用对等体组。如果群集内的客户端相互之间没有直接 iBGP 对等体，并且客户端通过路由反射器交换更新，请勿使用对等体组。如果配置对等体组，在路由反射器上退回路由源的可能性将传送到群集内的所有客户端。此传输可能引起问题。

默认情况下，会在路由反射器上启用路由器命令 `bgp client-to-client reflection`。如果关闭路由反射器上 BGP 客户端到客户端的反射，并在客户端之间设置冗余 BGP 对等，就可以安全使用对等体组。有关详细信息，请参阅[对等体组的限制](#)。

路由反射器和常规 BGP 扬声器

AS 内可以存在不识别路由反射器概念的 BGP 扬声器。本文将这些路由器称为常规 BGP 扬声器。路由反射器方案允许这些常规 BGP 扬声器共存。这些路由器既可以是客户端组的成员，也可以是非客户端组的成员。这些路由器的存在可方便从当前 iBGP 模型逐步迁移到路由反射器模型。如果将单个路由器配置为路由反射器，并将其他路由反射器和路由反射器客户端配置为常规 iBGP 对等端，就

可以开始创建群集。然后可以逐渐创建更多群集。



在此图中，RTD、RTE 和 RTF 能进行路由反射。RTC、RTA和RTB是传统路由器。您不能配置将这些路由器配置为路由反射器。可以在这些路由器和 RTD 之间进行常规的 iBGP 网格化。以后，当准备好进行升级时，可以将 RTC 设置为路由反射器，将 RTA 和 RTB 设置为客户端。客户端不必了解路由反射方案；仅RR需要升级。

RTD 和 RTC 的配置如下：

```
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
```

当准备好升级 RTC 并将 RTC 作为路由反射器时，请删除 iBGP 完全网格，使 RTA 和 RTB 成为 RTC 的客户端。

避免路由信息环路

到目前为止，本文档已经提到了两个可用于防止潜在信息循环的属性：**originator-id**和**cluster-list**。

控制环路的另一个方法是对出站路由映射的 set 子句进行更多限制。出站路由映射的 set 子句不影响反射到 iBGP 对等体的路由。

也可以对**next-hop-self**进行更多限制，这是每个邻居的配置选项。当您在RR上使用**next-hop-self**时，该子句只影响eBGP获知的路由的下一跳，因为不能更改反射的路由的下一跳。

路由抖动减弱

Cisco IOS 软件 11.0 版引入了路由抖动惩罚功能。路由惩罚是将路由抖动导致的不稳定性减到最小的一种机制。路由惩罚还可减少网络震荡。您可以定义用以识别不佳路由的标准。抖动的路由每抖动一次就会受到值为 1000 的惩罚。一旦累积惩罚达到预定义的抑制限制，就会抑制路由通告。惩罚根据预配置的半衰期呈指数衰减。一旦惩罚在预定义的重复使用限制下减少，路由通告将不再被抑制。

路由惩罚不适用于 AS 外部或通过 iBGP 获知的路由。通过这种方式，路由惩罚可避免对 AS 外部路由的 iBGP 对等体进行更高的惩罚。

惩罚每 5 秒减弱一次。路由在10秒的粒度内未被抑制。路由器会保留阻尼信息，直到惩罚值小于重新使用限制的一半。此时路由器将清除该信息。

最初，路由惩罚默认处于关闭状态。如果需要，将来可对此功能提供默认启用。以下命令控制路由惩罚：

-

bgp dampening — 启用惩罚。

-

no bgp dampening — 关闭惩罚。

-

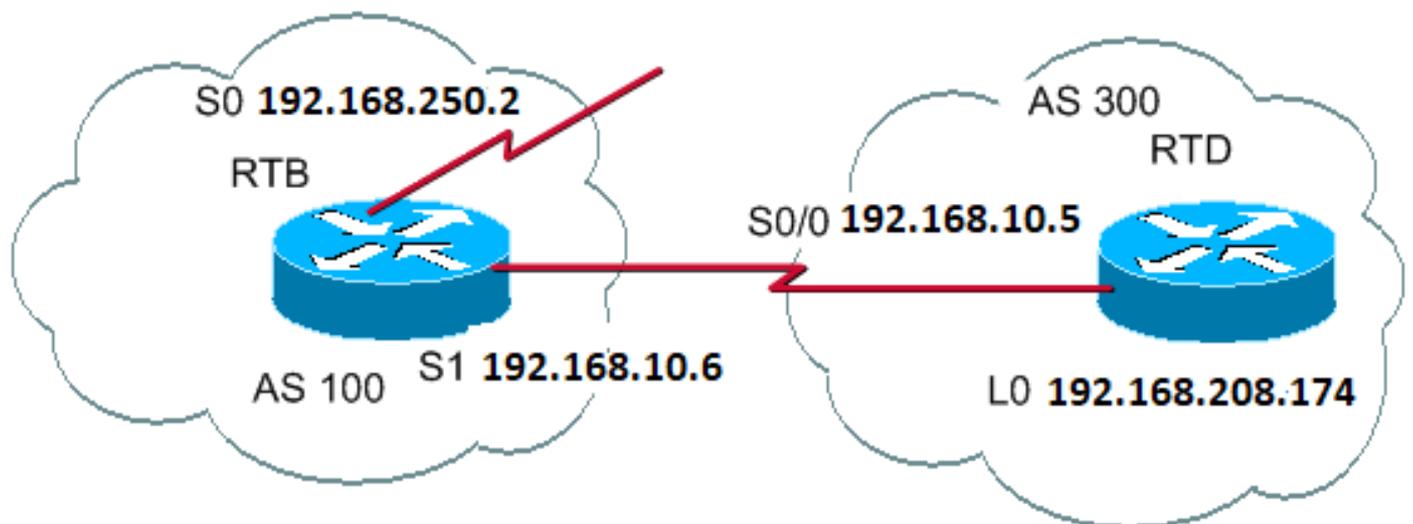
bgp dampeninghalf-life-time -更改半衰期时间。

以下是同时设置所有参数的命令：

- **bgp dampening half-life timer uses suppress maximum-suppress-time**

下表详细说明了语法：

- **half-life-time** — 范围是1-45分钟，当前默认值是15分钟。
- **reuse-value** — 范围是1-20,000，默认值是750。
- **suppress-value** — 范围是1-20,000，默认值为2000。
- **max-suppress-time** — 这是抑制路由的最长持续时间。范围是1-255分钟，默认值是半衰期时间的4倍。



```
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
```

对 RTB 的配置针对使用默认参数的路由惩罚。如果假设 eBGP 到 RTD 的连接是稳定的，则 RTB BGP 表应如下：

```
<#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

要模拟路由抖动，可以对 RTD 发出 clear ip bgp 192.168.10.6 命令。RTB BGP 表如下所示：

<#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

192.168.10.0的BGP条目处于ahistorystate (历史状态)。此设置表示到该路由没有最佳路径，但有关路由抖动的信息仍然存在。

<#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

路由已收到抖动惩罚，但该惩罚仍在suppress限制下。默认值是 2000。路由抑制尚未发生。如果路由抖动更多次，您将看到：

<#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

该路由已被阻尼或抑制。惩罚值达到“重新使用值”时重新使用路由。在这种情况下，重新使用值是默认的 750。惩罚值低于重新使用限制的一半时，将会清除惩罚信息。在本例中，惩罚值变为 375 (750/2=375) 时将发生清除。以下命令显示和清除抖动统计信息：

•

show ip bgp flap-statistics — 显示所有路径的抖动统计信息。

•

show ip bgp flap-statistics regexregular-expression —显示与正则表达式相匹配的所有路径的抖动统计信息。

•

show ip bgp flap-statistics filter-listlist— 显示通过过滤器的所有路径的抖动统计信息。

•

show ip bgp flap-statisticsA.B.C.D m.m.m.m— 显示单个条目的抖动统计信息。

•

show ip bgp flap-statisticsA.B.C.D m.m.m.mlonger-prefix — 显示更具体的条目的抖动统计信息。

•

show ip bgp neighbor [dampened-routes] | [flap-statistics] — 显示来自邻居的所有路径的抖动统计信息。

•

clear ip bgp flap-statistics — 清除所有路由的抖动统计信息。

•

clear ip bgp flap-statistics regexprregular-expression —清除与正则表达式相匹配的所有路径的抖动统计信息。

•

clear ip bgp flap-statistics filter-listlist— 清除通过过滤器的所有路径的抖动统计信息。

•

clear ip bgp flap-statisticsA.B.C.D m.m.m.m— 清除单个条目的抖动统计信息。

•

clear ip bgpA.B.C.Dflap-statistics — 清除来自某个邻居的所有路径的抖动统计信息。

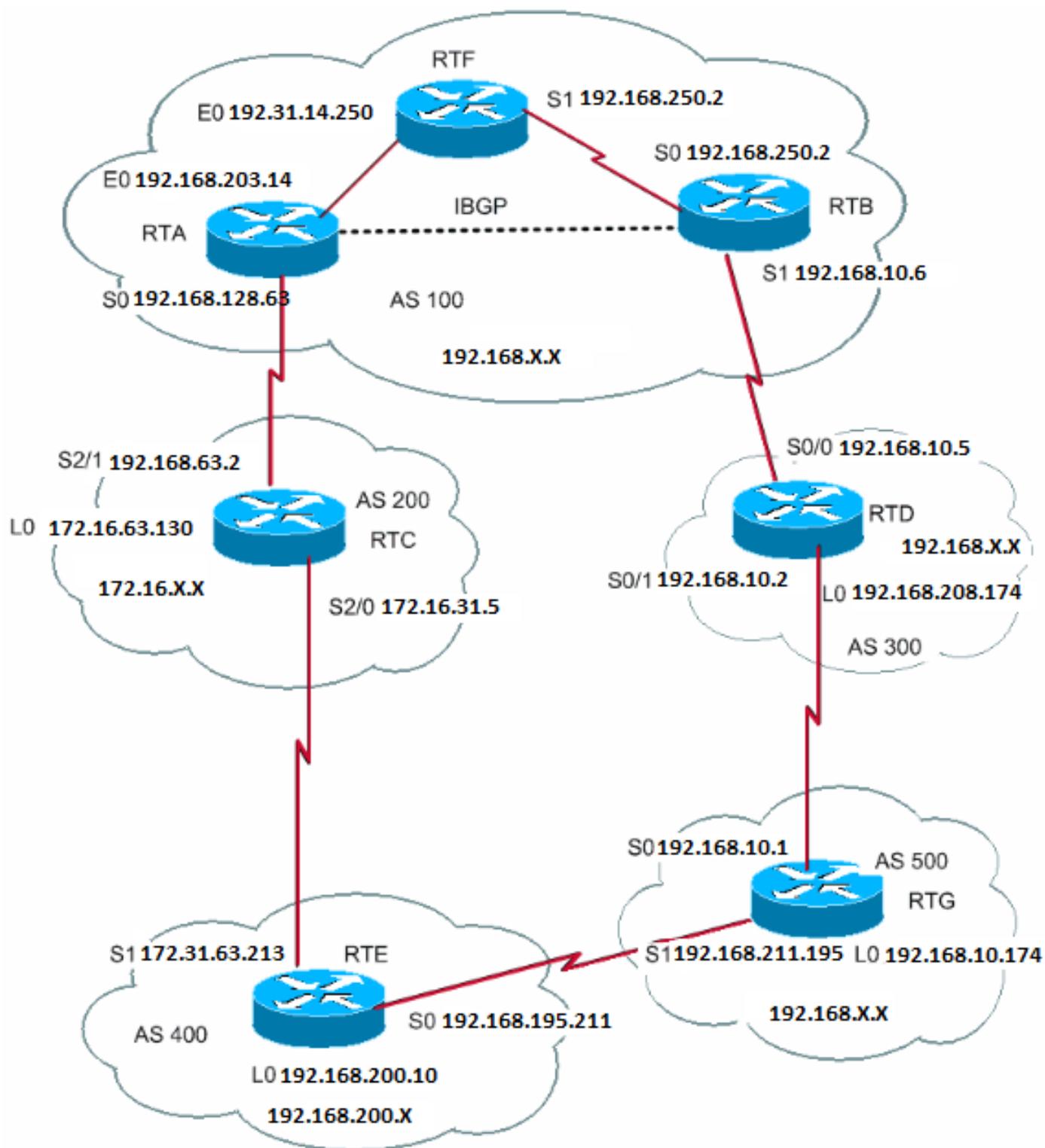
BGP 如何选择路径

现在您已经熟悉 BGP 属性和术语，接下来请参阅 BGP 最佳路径选择算法。

BGP 案例分析 5

实用设计示例

此部分包含一个设计示例，该示例显示的配置和路由表与 Cisco 路由器中实际显示的表相同。



此部分逐步演示如何构建此配置以及该过程中可能出现何种错误。每当您的 AS 通过 eBGP 与两个 ISP 连接时，始终应在 AS 内运行 iBGP，以便更好地控制您的路由。在本示例中，iBGP 在 AS100 内的 RTA 和 RTB 之间运行，OSPF 作为 IGP 运行。假设您连接到两个 ISP，即 AS200 和 AS300。以下是第一次运行所有路由器的配置：

注意：这些配置不是最终配置。

```
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
```

请始终使用network 命令，或将静态条目重分配到BGP以通告网络。此方法优于将 IGP 重分配到 BGP 的方法。此示例使用network 命令将网络注入BGP。

在此处，RTB 关闭时最开始使用 s1 接口，当做 RTB 和 RTD 之间的链接不存在。下面是 RTB BGP 表：

```
<#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   100     0 200 400 500
300 i
*i192.168.211.10   172.31.63.250          100   100     0 200 400 500 i
*i192.168.10.10    172.31.63.250          100   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
```

此表中的一些表示方法：

-

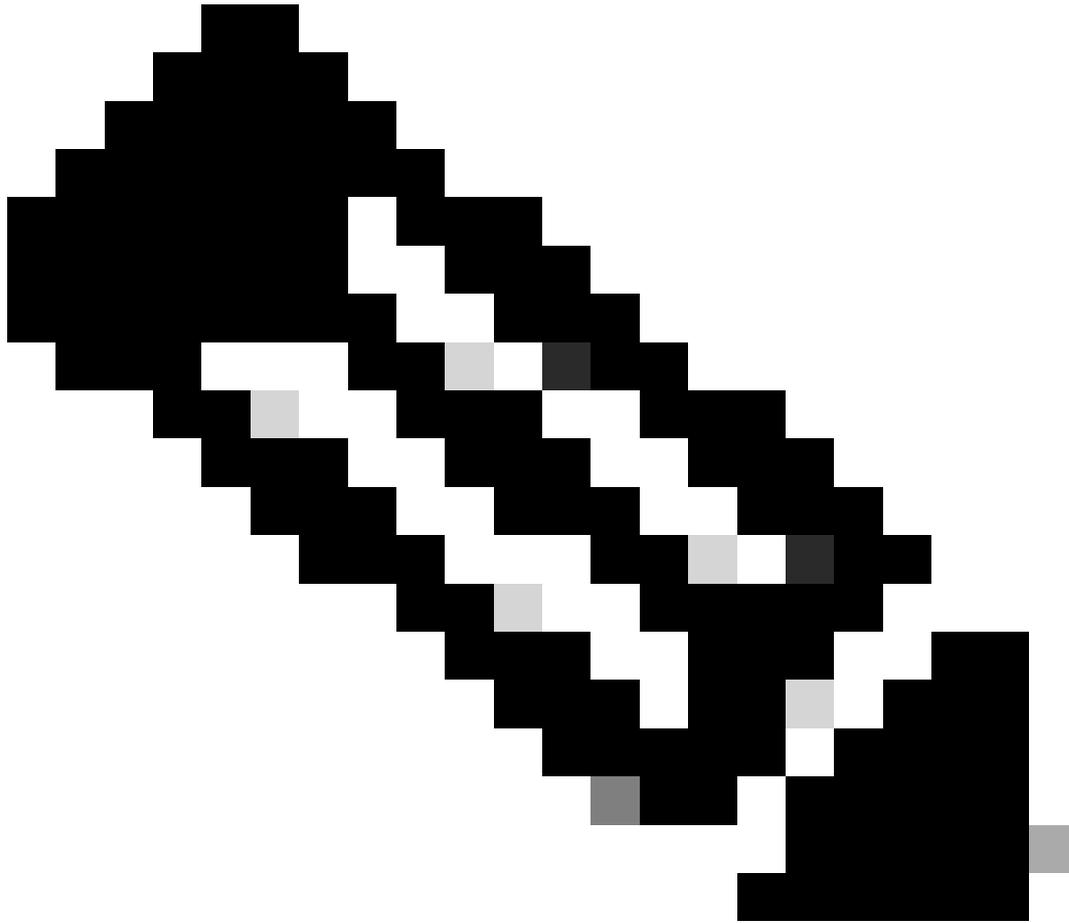
Aniat the beginning -表示通过iBGP对等体获知条目。

-

Aniat the end -指示路径信息的源是IGP。

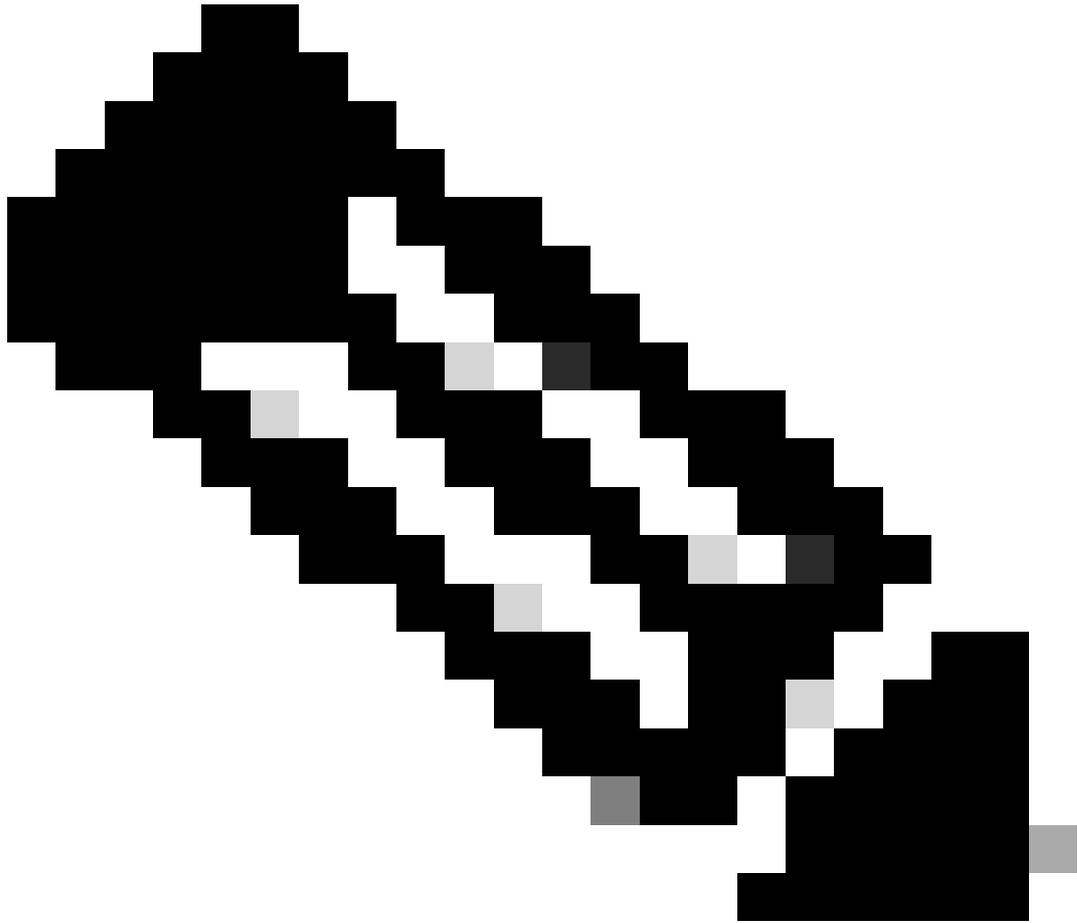
-

Pathinformation -此信息为直观信息。例如，通过下一跳为 172.31.63.250 的路径 200 获知网络 172.31.10.0。



注意：任何本地生成的条目（例如192.168.250.15）的下一跳为0.0.0.0。

-
- > 符号表示 BGP 已选择最佳路由。BGP 使用 BGP 最佳路径选择算法文档所列出的决策步骤。BGP 选择一个到达目的地的最佳路径，将该路径输入 IP 路由表中，并将其通告给其他 BGP 对等体。



注意：注意“下一跳”属性。RTB通过下一跳172.31.63.250获知有关172.31.10.0的信息，该下一跳是iBGP中传输的eBGP下一跳。

查看 IP 路由表：

```
<#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
```

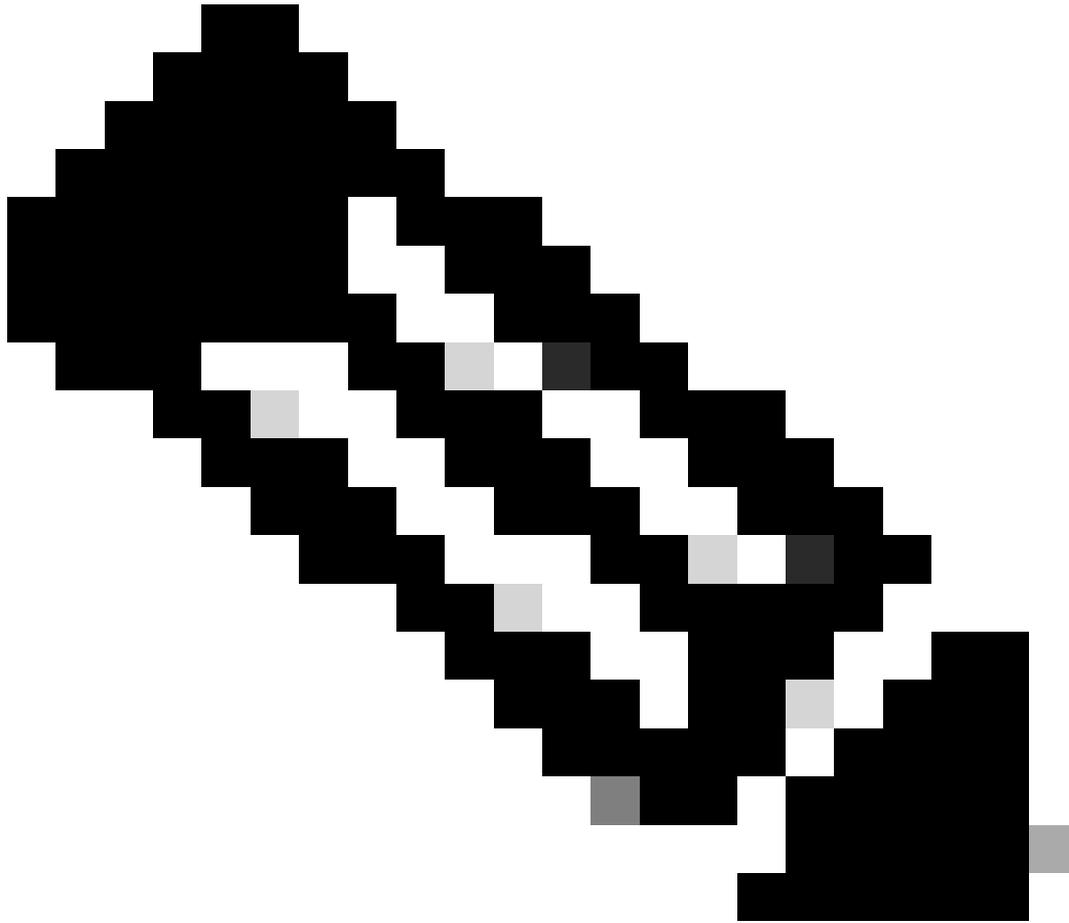
```
Gateway of last resort is not set
```

```
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
```

显然，没有任何 BGP 条目到达该路由表。这里存在两个问题。

第一个问题是无法到达这些条目的下一跳 172.31.63.250。没有通过此 IGP (这里为 OSPF) 到达该下一跳的途径。RTB 尚未通过 OSPF 获知 192.168.213.63。您可以在 RTA s0 接口上运行 OSPF 并使其成为被动接口；通过这种方式，RTB 知道如何到达下一跳 172.31.63.250。此 RTA 配置如下：

```
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
```



注意：您可以在RTA和RTB之间发出**bgp nexthop self**命令，以便更改下一跳。

RTB 上的新 BGP 表类似如下所示：

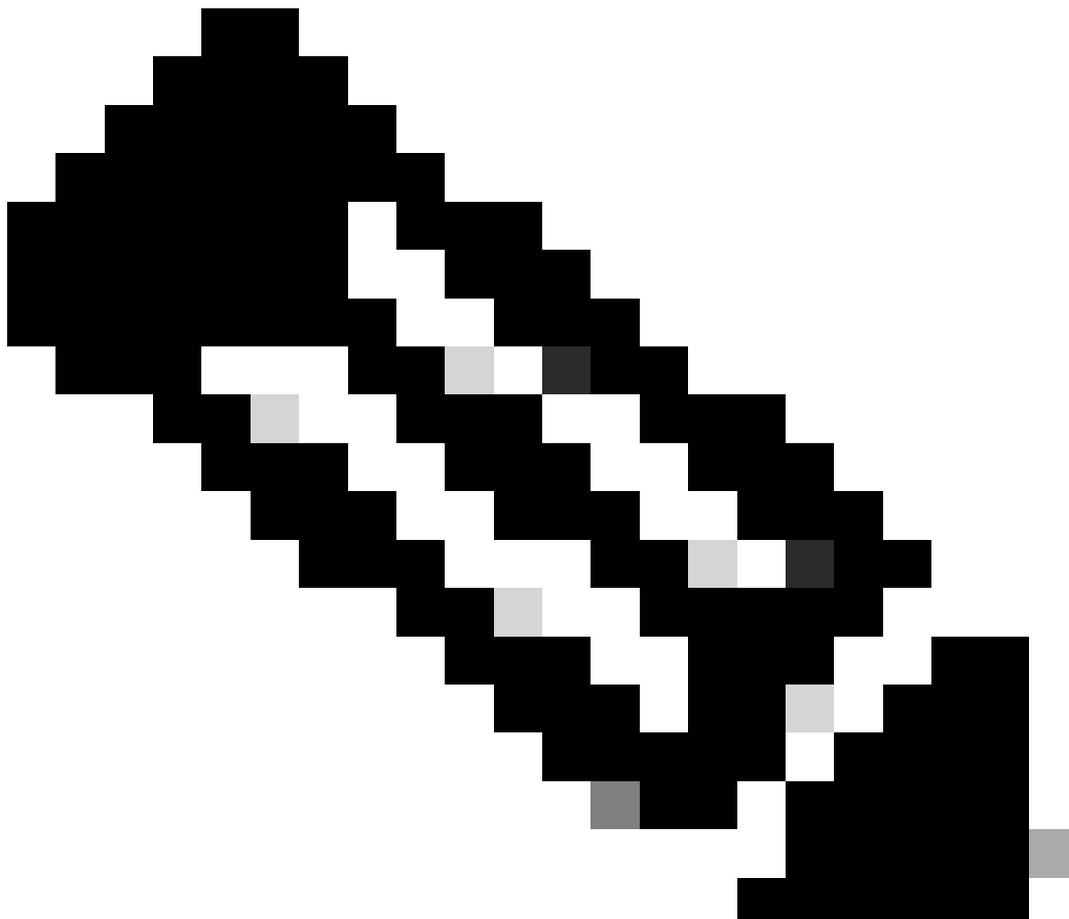
```
<#root>
```

```
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



注意：所有条目都包含>，这表示BGP可以到达下一跳。

查看路由表：

```
<#root>
```

```
RTB#
```

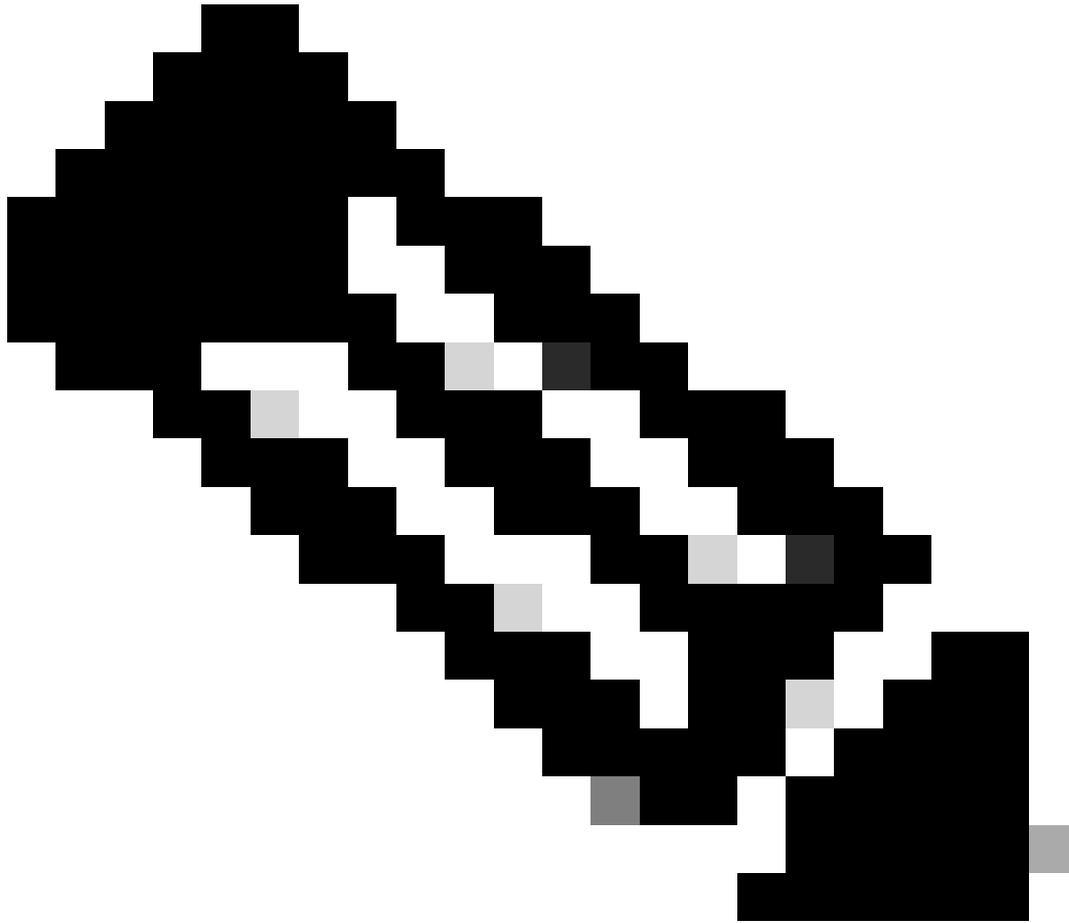
```
show ip route
```

```
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
```

第二个问题是仍然无法在路由表中看到 BGP 条目。唯一的区别是现在可以通过 OSPF 到达 192.168.213.63。此问题是同步问题。BGP 不会将这些条目输入到路由表中，也不会 BGP 更新中发送这些条目，因为未与 IGP 进行同步。



注意：RTF没有网络192.168.10.0和192.168.211.10的概念，因为您尚未将BGP重分配到OSPF中。

在此情况下，如果关闭同步，条目将出现在路由表中。但仍会中断连接。

如果关闭 RTB 上的同步，结果将如下所示：

```
<#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
```

路由表看起来没有异常，但是不能到达这些网络。中间的 RTF 不知道如何到达这些网络：

```
<#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
```

此时如果关闭同步，问题仍会存在。但稍后您在解决其他问题时需要同步。在度量为 2000 的 RTA 上，将 BGP 重分配到 OSPF：

```
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
```

路由表如下所示：

```
<#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
```

BGP 条目消失，因为 OSPF 比 iBGP 距离更短。OSPF 距离为 110，而 iBGP 距离为 200。

关闭 RTA 上的同步，以便 RTA 能够通告 192.168.250.15。由于 RTA 因掩码差异未与 OSPF 同步，所以此操作是必须的。保持关闭 RTB 上的同步，以便 RTB 能够通告 192.168.203.13。出于上述相同的原因，此操作对于 RTB 是必须的。

现在引入 RTB s1 接口，以查看路由状况。并且启用 RTB 串行接口 1 上的 OSPF 以使其处于被动状态。使用此步骤，RTA 通过 IGP 获知下一跳 192.168.10.5。如果不采取此步骤，则将发生环路，因为要到达下一跳 192.168.10.5，需要通过 eBGP 采用其他路径。

以下是 RTA 和 RTB 的新配置：

```
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
```

BGP 表如下所示：

<#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

通过多种网络设计方法都可以实现与两个不同 ISP (AS200 和 AS300) 的通信。一种方法是设置主 ISP 和备份 ISP。您能够从其中一个 ISP 获知部分路由，并获知到两个 ISP 的默认路由。在本示例中，您从 AS200 接收部分路由，从 AS300 只接收本地路由。RTA 和 RTB 都能生成到 OSPF 的默认路由，但 RTB 优先，因为 RTB 度量较低。采用这种方式，您可以平衡两个 ISP 之间的传出流量。

如果从 RTA 传出的流量通过 RTB 传回，则可能会出现非对称的情况。如果在与两个 ISP 进行通信时使用同一个 IP 地址池 (同一个主网) ，则可能会发生此种情况。由于进行了聚合，您的全部 AS 在外界上来像一个完整的实体。可以通过 RTA 或 RTB 产生网络的进入点。您会发现，所有传入您的 AS 的流量都是通过一个单一的点进入，即使您有进入 Internet 的多个点时也是如此。在本示例中，与两个 ISP 进行通信时有两个不同的主网。

发生非对称情况的另一个潜在原因是所通告的到达 AS 的路径长度不同。可能某一个服务提供商与某个目的地的距离近于另一个提供商。在本示例中，从 AS400 传输至您的网络的流量总是经由 RTA 传入，因为此路径距离较短。您可以设法影响该决策。可以使用 set as-path prepend 命令，以便将路径编号附加到您的更新的前面，使路径看起来更长一些。但是，因为存在本地优先级、度量或权重等属性，AS400 可能已将退出点设置为 AS200。如果是这样，您将无法进行更改。

下面是所有路由器的最终配置：

```

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

```

在 RTA 上，来自 AS200 的路由的本地优先级设置为 200。并且，网络 172.31.200.200 也作为默认候选网络。使用 ip default-network 命令可以选择默认网络。

此示例还对 OSPF 使用 default-information originate 命令，将默认路由注入 OSPF 域。此示例还对 Intermediate System-to-Intermediate System Protocol (IS-IS Protocol) 和 BGP 使用了此命令。对于 RIP，无需额外配置，即会将 0.0.0.0 自动重分配到 RIP。对于 IGRP 和 EIGRP，将 BGP 重分配到 IGRP 和 EIGRP 之后，会将默认信息注入 IGP 域。同样，对于 IGRP 和 EIGRP，可以将到 0.0.0.0 的静态路由重分配到 IGP 域。

```

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

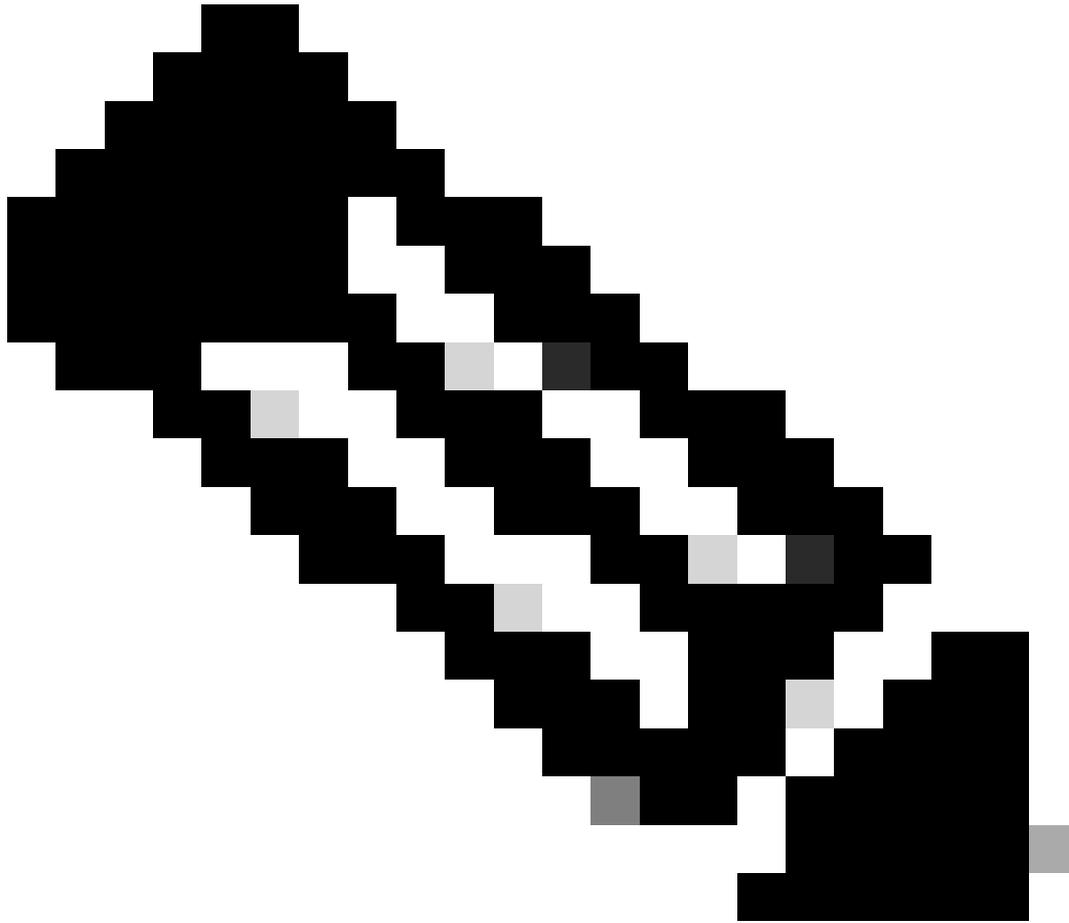
interface Serial0
 ip address 192.168.250.2 255.255.255.252
!
interface Serial1
 ip address 192.168.10.6 255.255.255.252

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

```

对于 RTB，来自 AS300 的更新的本地优先级设置为 300。该值高于来自 RTA 的 iBGP 更新的本地优先级值。采用此种方法，AS100 选择 RTB 作为 AS300 的本地路由。如果存在 RTB 的其他路由，则所有这些路由将以本地优先级 100 进行内部传输。该值低于 RTA 的本地优先级 200。首选使用 RTA。



注意：您仅通告AS300本地路由。与 ^300\$ 不匹配的任何路由都将丢弃。如果希望通告作为 ISP 客户的本地路由和邻居路由，应使用 ^300_[0-9]*。

以下是表示 AS300 本地路由的正则表达式的输出：

<#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
```

在RTC上，聚合172.31.10.0/16并指示要注入到AS100的特定路由。如果ISP拒绝此项任务，则您必须在AS100的传入端进行过滤。

```
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

```

有关如何使用社区过滤的演示位于RTG上。将 no-export community 添加到192.168.211.0更新以获取RTD。这样，RTD 不将该路由导出到 RTB。然而，在此种情况下，RTB 无论如何都不会接受这些路由。

```

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
```

RTE 聚合 172.31.200.200/16。以下是 RTA、RTF 和 RTB 的最终 BGP 和路由表：

```
<#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
0 E2 192.168.10.0 255.255.255.0
      [110/1000] via 172.31.14.250, 00:41:25, Ethernet0
0    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

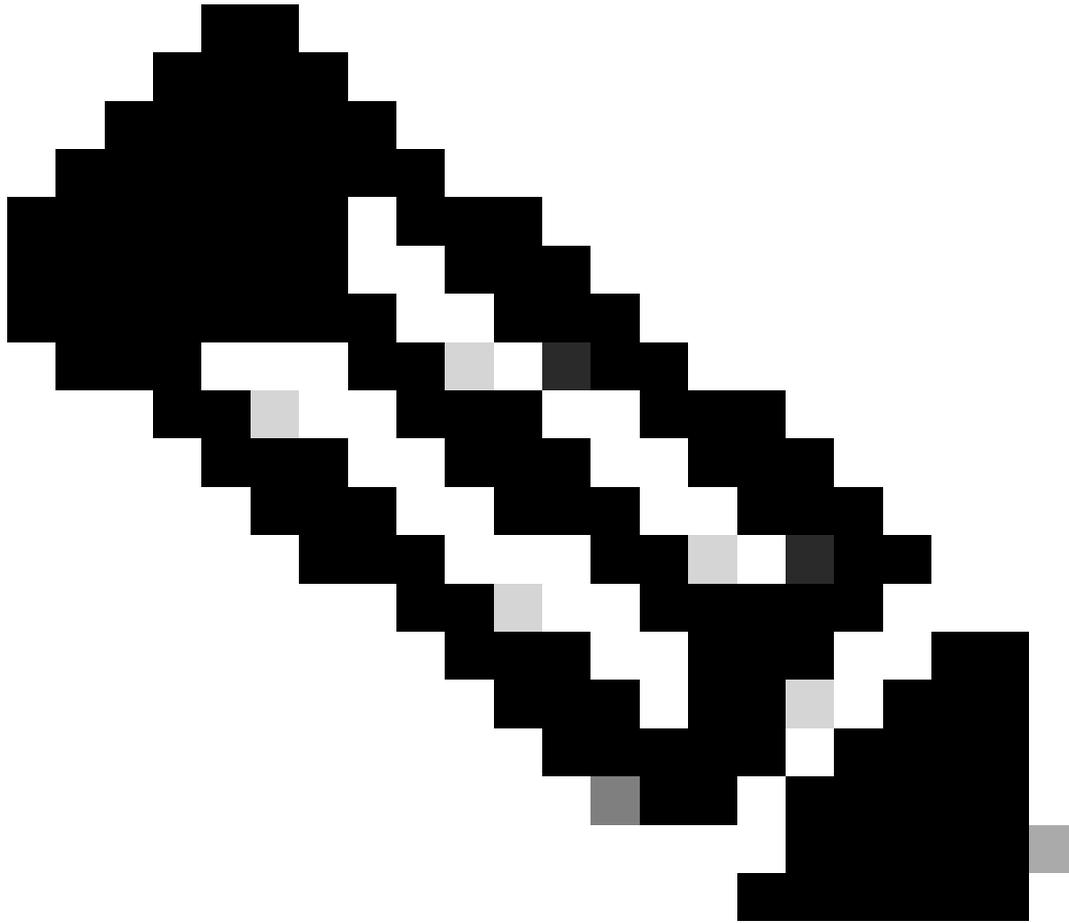
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
 D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
 E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
 i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * -
 candidate default

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

```



注意：RTF路由表指示到达AS300本地网络（例如192.168.10.0）的方法是通过RTB。到达其他已知网络（172.31.200.200）的路径要经过RTA。最后选用网关设为RTB。如果RTB和RTD之间的连接出现问题，则RTA所通告的默认路由将以度量2000参与进来。

<#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
```

相关信息

- [BGP：常见问题](#)
- [跨越PIX 防火墙的 BGP 配置示例](#)
- [如何使用 HSRP 在多宿主 BGP 网络中提供冗余](#)
- [在Cat6000 MSFC上配置单路由器模式冗余和BGP](#)
- [实现最优路由和减少 BGP 内存消耗](#)
- [排除常见BGP问题](#)
- [对BGP扫描程序或路由器进程导致的CPU使用率过高进行故障排除](#)
- [了解单宿主和多宿主环境中与BGP的负载共享](#)
- [BGP 支持页](#)
- [Cisco技术支持和下载](#)

关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。