ボーダー ゲートウェイ プロトコルのケーススタディの確認

内容

はじめに

このドキュメントでは、5 つのボーダー ゲートウェイ プロトコル (BGP) のケーススタディについて説明します。

前提条件

要件

このドキュメントに関する固有の要件はありません。

使用するコンポーネント

このドキュメントの内容は、特定のソフトウェアやハードウェアのバージョンに限定されるものではありません。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな (デフォルト) 設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

表記法

表記法の詳細については、『シスコ テクニカル ティップスの表記法』を参照してください。

BGP ケース スタディ 1

RFC 1771 で定義されている BGP を使用すると、Autonomous System（AS; 自律システム）間にループのないドメイン間ルーティングを作成できます。AS は、単一の技術管理に基づくルータのまとまりです。AS 内のルータは、複数の内部ゲートウェイ プロトコル（IGP）を使用して AS 内のルーティング情報を交換できます。これらのルータは、外部ゲートウェイ プロトコルを使用して AS の外部にパケットをルーティングできます。

BGP の動作

BGP はトランスポート プロトコルとして TCP（ポート 179）を使用します。2 台の BGP ルータは相互に TCP 接続を形成します。これらのルータはピア ルータです。ピア ルータはメッセージを交換し、接続パラメータを開いて確認します。

BGP ルータはネットワーク到達可能性情報を交換します。主にこの情報は、宛先ネットワークに到達するルートで経由する必要のあるフル パスを示します。これらのパスは BGP AS 番号です。この情報は、ループフリーな AS のグラフの作成に役立ちます。このグラフでは、ルーティング動作を制限するためにルーティング ポリシーを適用すべき場所もわかります。

BGP ルーティング情報を交換するために TCP 接続を確立している 2 台のルータは、「ピア」または「ネイバー」と呼ばれます。BGP ピアは最初に完全な BGP ルーティング テーブルを交換します。この交換以降、ピアはルーティング テーブルが変更されるたびに差分更新を送信します。BGP には BGP テーブルのバージョン番号が保持されます。バージョン番号はすべての BGP ピアで同一です。ルーティング情報の変更によって BGP がテーブルを更新するたびに、バージョン番号は変更されます。キープアライブ パケットを送信することで、BGP ピア間の接続が有効であるかどうかが確認されます。エラーまたは特殊な状況が発生すると、通知パケットが送信されます。

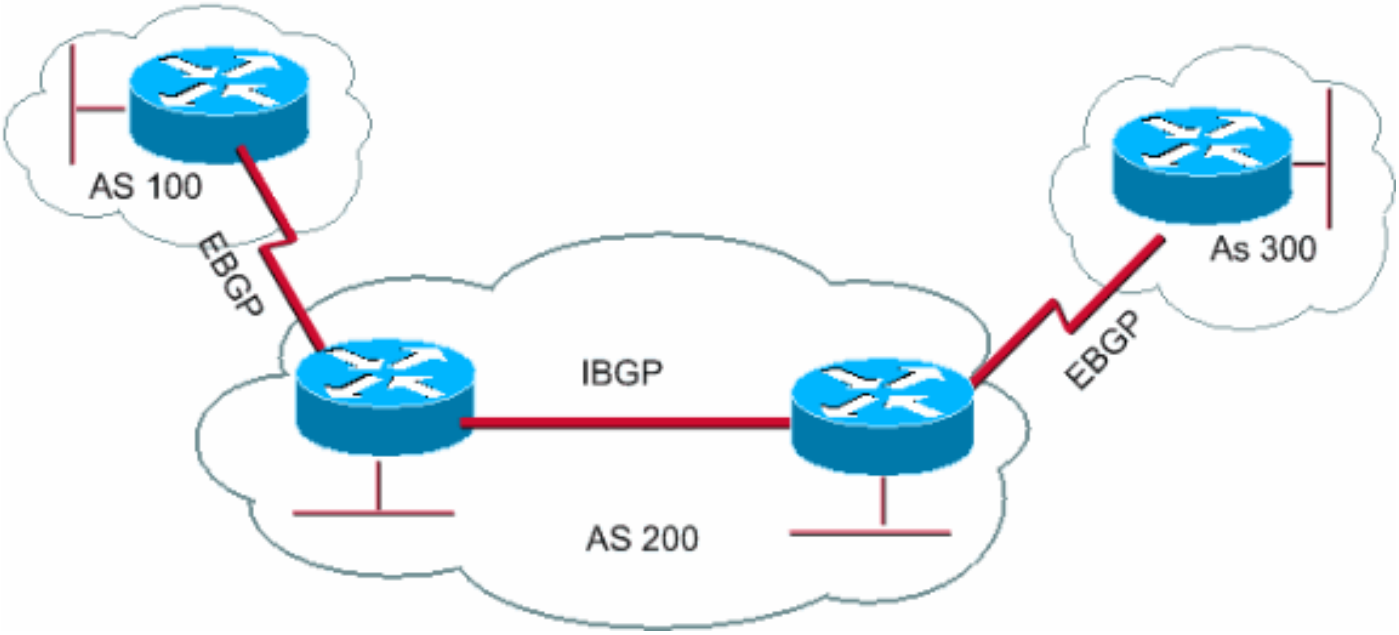
eBGP と iBGP

複数の BGP スピーカを含む AS は、他の AS の中継サービスとして機能できます。このセクションの次の図に示すように、AS200はAS100およびAS300の中継ASです。

外部 AS に情報を送信するには、ネットワークの到達可能性が確保されている必要があります。到達可能性を確保するために必要な処理は次のとおりです。

- AS 内のルータ間の内部 BGP（iBGP）ピアリング
- AS 内で動作している IGP への BGP 情報の再配布

2 つの異なる AS に属するルータ間で動作する BGP は、外部 BGP（eBGP）と呼ばれます。BGP が同じ AS 内のルータ間で動作している場合は、iBGP と呼ばれます。



同じAS内のルータ間で稼働するBGP

BGP ルーティングの有効化

BGP を有効化および設定するには、次の手順を実行します。

2 台のルータ (RTA と RTB) が BGP を使用して通信すると仮定します。最初の例では、RTA と RTB は別の AS に属しています。2 番目の例では、両方のルータが同じ AS に属しています。

1. ルータ プロセスと、ルータが属する AS 番号を定義します。

   次のコマンドを発行して、ルータで BGP を有効にします。

   <#root>

   router bgp <autonomous-system>

   ```
   RTA#
   router bgp 100

   RTB#
   router bgp 200
   ```

   これらのステートメントは、RTA が BGP を実行し、AS100 に属すること、そして RTB は BGP を実行し、AS200 に属することを示します。

2. BGP ネイバーを定義します。

   BGP ネイバーを形成することで、BGP を使用した通信を試行するルータを示します。次のセクションでは、このプロセスについて説明します。

BGP ネイバーの形成

2 台の BGP ルータは、相互に TCP 接続を確立することでネイバーになります。2 台のピア ルータがルーティング アップデートの交換を開始するには、TCP 接続が不可欠です。

TCP 接続が確立されると、ルータはオープン メッセージを送信して値を交換します。ルータが交換する値には、AS 番号、ルータが実行する BGP バージョン、BGP ルータ ID、キープアライブ ホールド時間が含まれます。これらの値の確認と承認が完了すると、ネイバー接続が確立されます。状態が Established 以外である場合、2 台のルータはネイバーになっておらず、BGP アップデートを交換できないことを意味します。

TCP接続を確立するには、次の neighbor コマンドを発行します。

<#root>

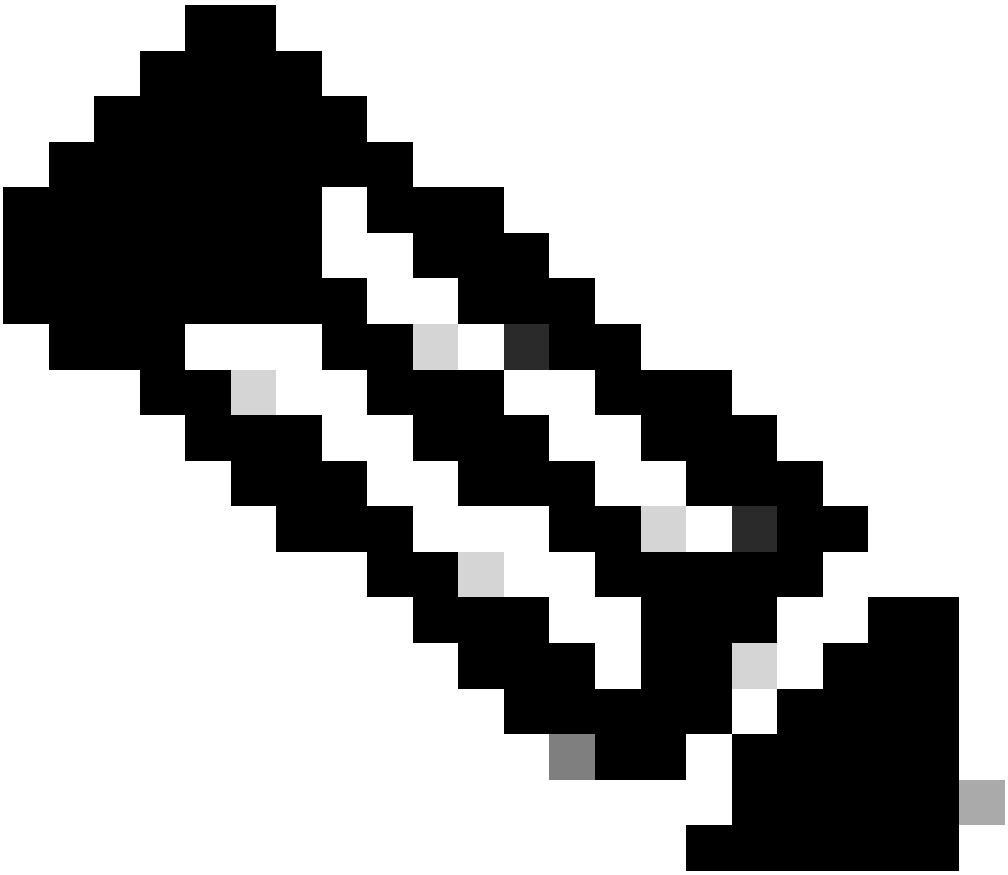
neighbor <ip-address> remote-as <number>

このコマンドの number には、BGP を使用して接続させるルータの AS 番号を指定します。ip-address には、eBGP の直接接続しているネクスト ホップ アドレスを指定します。iBGP の場合、ip-address はもう一方のルータの IP アドレスです。

ピアルータの neighbor コマンドで使用する2つのIPアドレスは、相互に到達可能である必要があります。到達可能性を確認する 1 つの方法は、2 つの IP アドレス間で拡張 ping を実行することです。拡張pingは、 neighbor コマンドで指定されたIPアドレスを送信元として使用するようにpingを実行するルータを強制します。ルータは、パケットの送信元となるインターフェイスの IP アドレスではなく、このアドレスを使用する必要があります。

BGP 設定が変更された場合は、新しいパラメータを有効にするためにネイバー接続をリセットする必要があります。 .

- clear ip bgp address



注:addressは、ネイバーアドレスです

•

clear ip bgp *

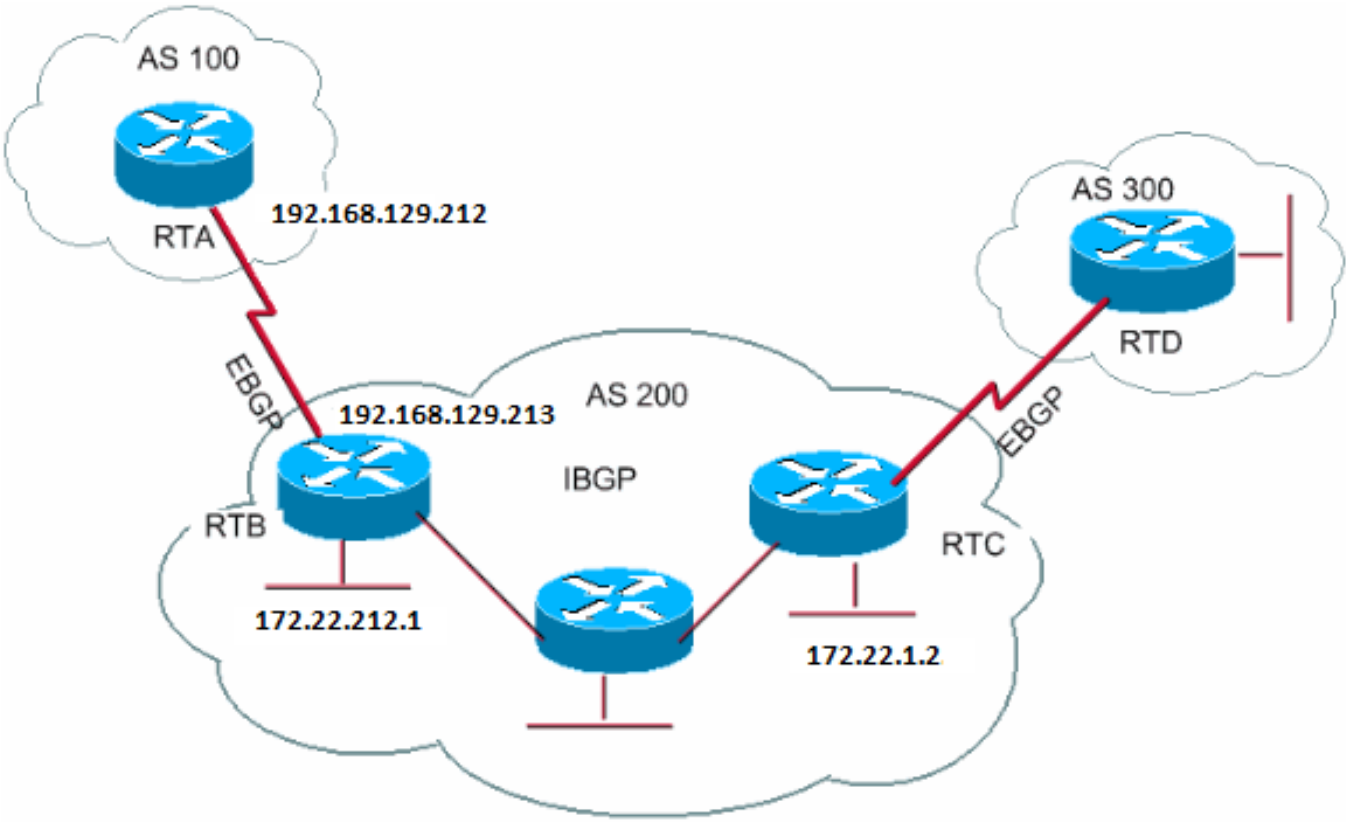
このコマンドを実行するとすべてのネイバー接続がクリアされます。

デフォルトでは、BGP セッションは BGP バージョン 4 を使用して開始され、必要に応じて以前のバージョンへの下方調整がネゴシエートされます。ネゴシエーションを回避して、ネイバーとの通信にルータが使用する BGP バージョンを指定することができます。ルータ設定モードで次のコマンドを発行します。

<#root>

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

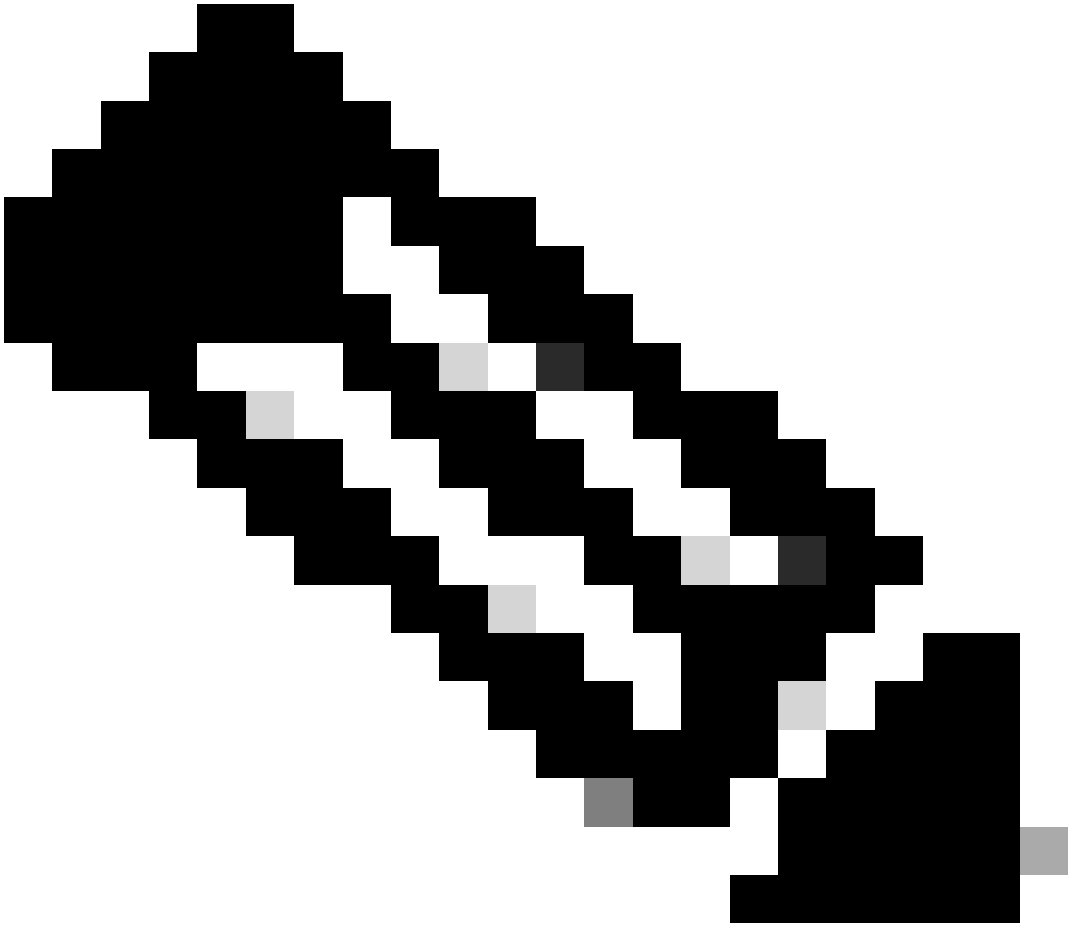
neighbor コマンドの設定例を次に示します。



```
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
```

この例では、RTA と RTB は eBGP を実行します。RTB と RTC は iBGP を実行します。リモート AS 番号は外部または内部 AS を指し、eBGP または iBGP のどちらであるかを示します。また、eBGP ピアは直接接続されていますが、iBGP ピアは直接接続されてはいません。iBGP ルータは、直接接続する必要がありません。ただし、何らかのIGPが稼働していて、2つのネイバーが相互に到達できるようにする必要があります。

ここでは、show ip bgp neighbors コマンドによって表示される情報の例を示します。



注：特にBGPの状態に注意してください。状態がEstablished以外の場合は、ピアが確立されていません。 また、次の項目にも注意してください。

- BGP version（4です）
- remote router ID

この数値は、ルータの最上位 IP アドレスまたは最上位ループバック インターフェイスです (存在する場合) 。

- table version

table version はテーブルの状態を示します。新しい情報が追加されるたびに、テーブルのバージョンが上がります。バージョンが増え続ける場合は、ルートの継続的な更新を引き起こすルート フラップが発生しています。

<#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 とループバック インターフェイス

ネイバーを定義するためにループバックインターフェイスを使用することは、iBGPでは一般的ですが、eBGPでは一般的ではありません。通常、ループバック インターフェイスは、ネイバーの IP アドレスが有効で、正常に機能しているハードウェアに依存し

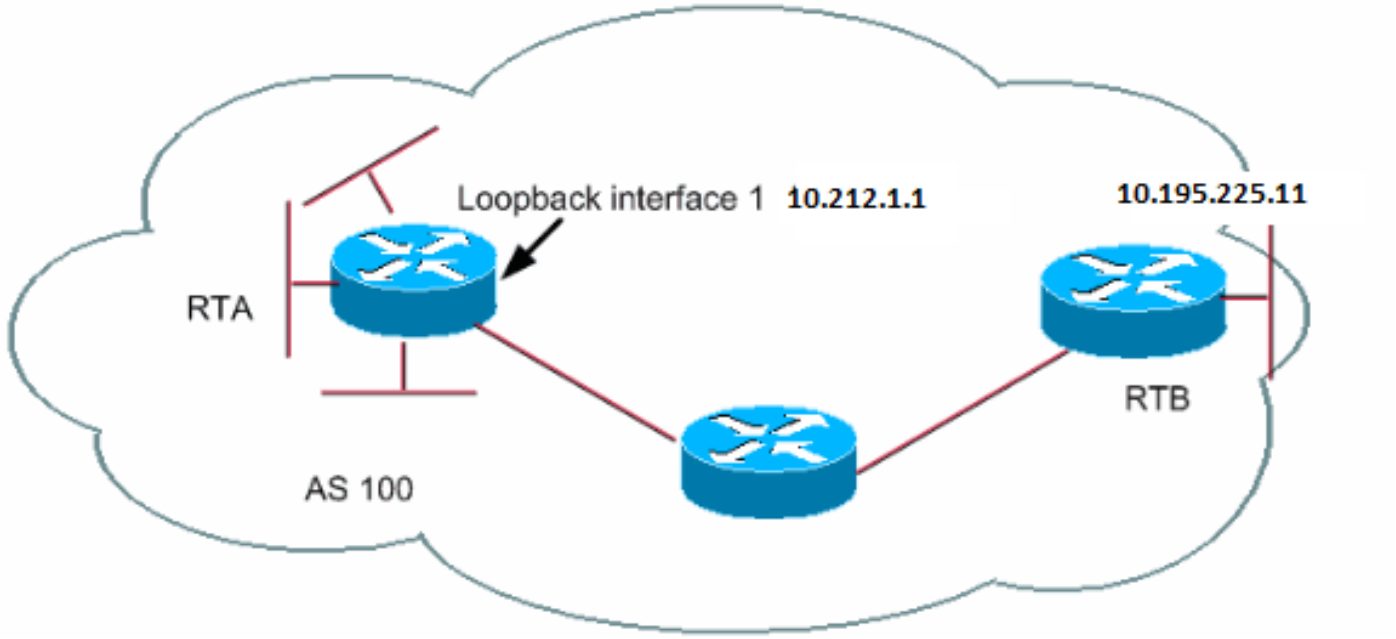
ていないことを確認するために使用されます。eBGP の場合はピア ルータが直接接続されていることが多く、ループバックは適用されません。

ループバックインターフェイスのIPアドレスを neighbor コマンドで使用する場合は、ネイバールータでいくつか追加の設定が必要になります。ネイバー ルータは、物理インターフェイスではなくループバック インターフェイスを使用して BGP ネイバー TCP 接続を開始することを BGP に通知する必要があります。ループバック インターフェイスを示すには、次のコマンドを発行します。

<#root>

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

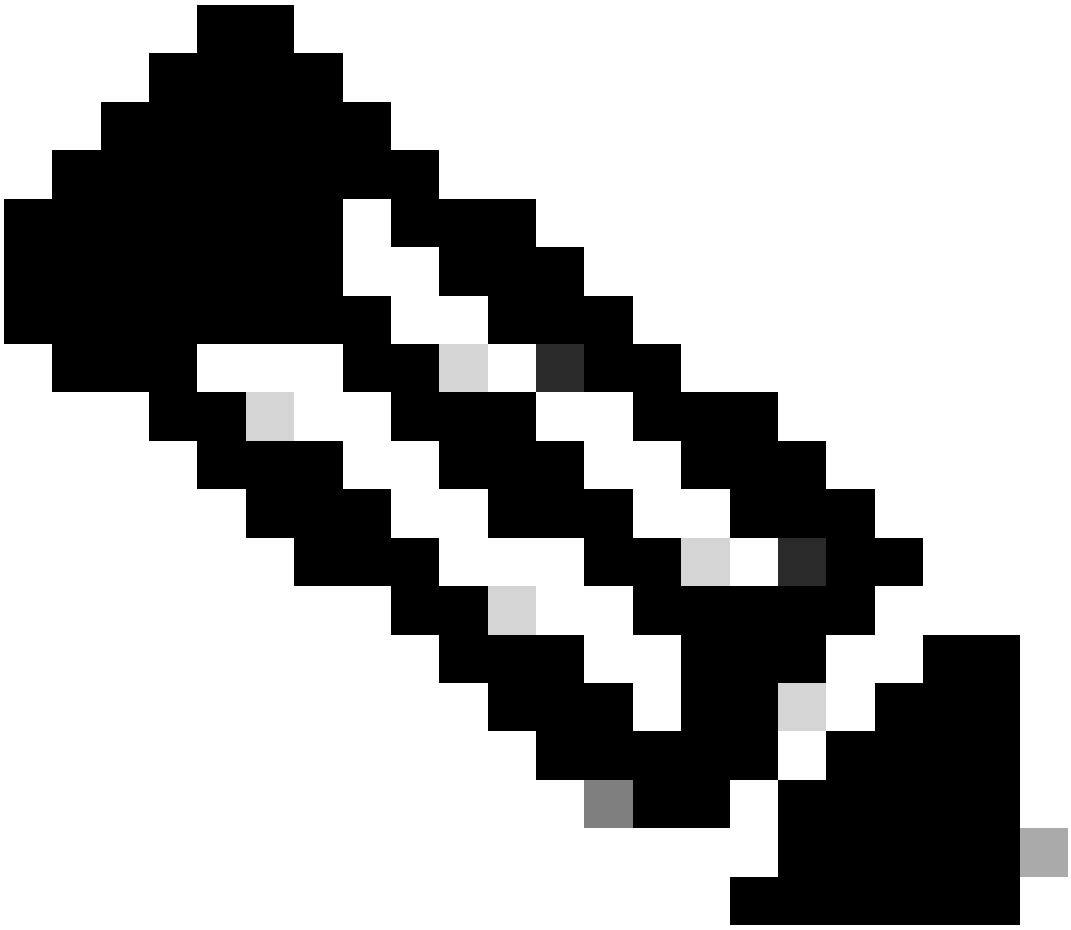
次に、このコマンドの使用例を示します。




```
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
```

この例では、RTA と RTB は AS100 内で iBGP を実行します。 neighbor コマンドでは、RTBはRTAのループバックインターフェイス(10.212.1.1)を使用します。この場合、RTA は TCP ネイバー接続の送信元としてループバック IP アドレスを使用することを BGP に強制する必要があります。この動作を強制するために、RTAは**update-source interface-type interface-number** を追加して、コマンドがneighbor 10.195.225.11 update-source loopback 1になるようにします。この文は、BGPがネイバー10.195.225.11と通信するときに、ループバックインターフェイスのIPアドレスを使用するように強制します。

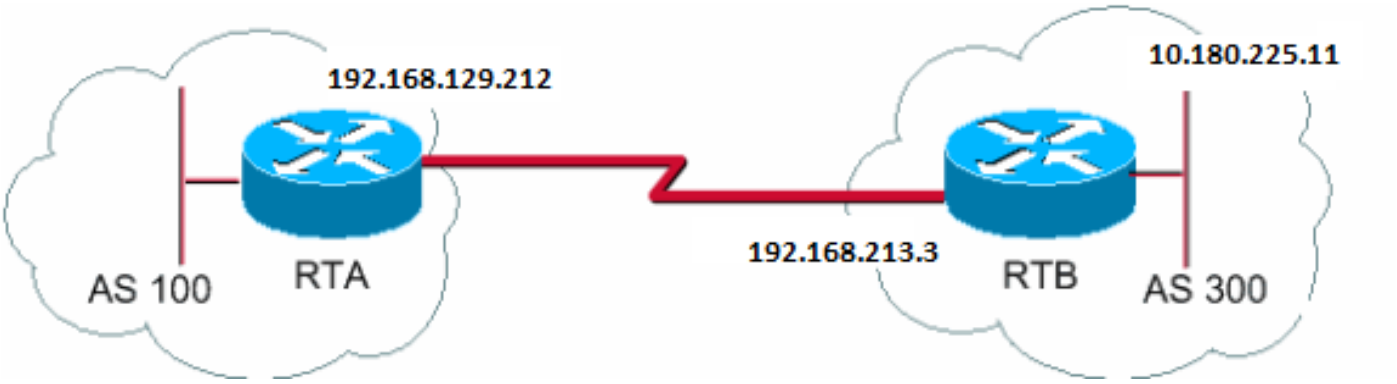


注:RTAはネイバーとして、RTBの物理インターフェイスIPアドレス(10.195.225.11)を使用しています。この IP アドレスが使用されるため、RTB では特別な設定が必要ありません。完全なネットワーク シナリオの設定例については、『ループバック アドレスを使用する場合と使用しない場合の iBGP と eBGP の設定例』を参照してください。

---

eBGP マルチホップ

場合によっては、シスコ ルータは 2 つの外部ピアの直接接続を許可しないサード パーティ製ルータとの eBGP を実行できます。この接続を実現するには、eBGP マルチホップを使用します。eBGP マルチホップを使用すると、直接接続されていない 2 つの外部ピアをネイバー接続できます。マルチホップは eBGP のみを対象としており、iBGP では使用されません。次の例で eBGP マルチホップについて説明します。

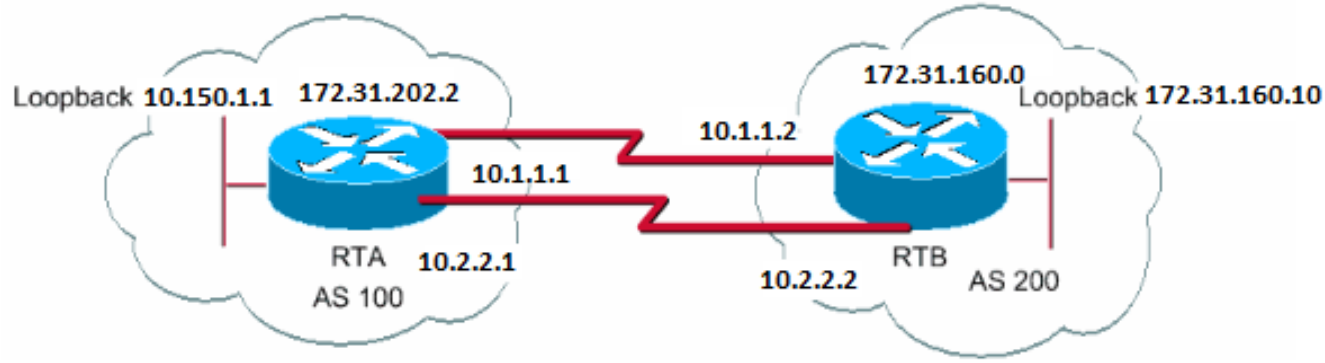



```
RTA#
router bgp 100
 neighbor 10.180.225.11 remote-as 300
 neighbor 10.180.225.11 ebgp-multihop

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

RTA は直接接続されていない外部ネイバーを示しています。RTA では、neighbor ebgp-multihop コマンドの使用が示される必要があります。一方、RTBは直接接続されているネイバー(192.168.129.212)を示しています。この直接接続により、RTBではneighbor ebgp-multihop コマンドが不要です。また、接続されていないネイバーが相互に到達できるように、IGPまたはスタティックルーティングを設定する必要があります。

「BGPマルチホップ（ロードバランシング）」セクションの例は、パラレル回線上にBGPが存在する場合に、BGPを使用してロードバランシングを実現する方法を示しています。

eBGP マルチホップ（ロード バランシング）




```
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
```

次の例は、ループバックインターフェイス、update-source、およびebgp-multihopの使用を示しています。この例は、パラレル シリアル回線上の 2 つの eBGP スピーカ間でロード バランシングを実現するための回避策を示しています。通常はパケットを送信する回線を BGP が 1 つ選択するため、ロード バランシングは実行されません。ループバック インターフェイスを使用することで、eBGP のネクスト ホップはループバック インターフェイスになります。スタティック ルートまたは IGP を使用して、宛先に到達する 2 つの等コスト パスを導入します。RTAがネクストホップ172.31.160.10に到達するには、2つの選択肢があります。1つのパスは10.1.1.2を経由し、もう1つのパスは10.2.2.2を経由します。RTB にも同じ選択肢があります。

ルート マップ

BGP ではルート マップが多用されます。BGP において、ルート マップはルーティング情報を制御および変更するためのメソッドです。ルーティング情報の制御と変更は、1 つのルーティング プロトコルから別のルーティング プロトコルへのルート再配布の条件を定義することで行われます。または、BGP に対するインジェクトおよび取り出しによってもルーティング情報を制御でき

ます。ルートマップの形式は次のとおりです。

<#root>

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

マップ タグはルート マップに指定する単なる名前です。同じルート マップ、つまり同じ名前タグの複数のインスタンスを定義できます。シーケンス番号は、同一の名前ですでに設定されているルート マップのリスト内で新しいルート マップが配置される位置を示します。

この例では、MYMAP という名前のルート マップのインスタンスが 2 つ定義されています。最初のインスタンスのシーケンス番号は 10 で、2 番目のインスタンスのシーケンス番号は 20 です。

- **oute-map MYMAP permit 10（最初の条件セットが入ります）**
- **route-map MYMAP permit 20**（2番目の条件セットがここに入ります）

着信または発信ルートにルート マップ MYMAP を適用すると、最初の条件セットはインスタンス 10 によって適用されます。最初の条件セットが満たされない場合は、ルート マップの上位のインスタンスに進みます。

match および set 設定コマンド

各ルートマップは、match およびset 設定コマンドのリストから構成されます。matchでは基match 準を指定し、setでは、コmatch マンドが適用する基準が満たされた場合のset アクションを指定します。

たとえば、発信アップデートをチェックするルート マップを定義できます。IP アドレス 10.1.1.1 との一致が見つかった場合、そのアップデートのメトリックは 5 に設定されます。これらのコマンドの例を示します。

<#root>

match ip address 10.1.1.1

set metric 5

ここで一致基準が満たされた場合、permitが指定されていると、setアクションで指定されたとおりにルートの再配布または制御が行われます。ここでリストから抜けます。

一致基準が満たされた場合、denyが指定されていると、ルートの再配布または制御は行われません。ここでリストから抜けます。

一致基準が満たされず、permit またはdeny が指定されている場合は、ルートマップの次のインスタンスがチェックされます。たとえば、インスタンス 20 がチェックされます。次のインスタンスのチェックは、リストから抜けるか、ルート マップのすべてのインスタンスが終了するまで続きます。一致しないままリストが終了した場合、ルートはnot accepted nor forwardedです。

Cisco IOSソフトウェアリリース11.2よりも前のCisco IOS®ソフトウェアリリースでは、プロトコル間での再配布ではなくBGPアップデートのフィルタリングにルートマップを使用する場合、IPアドレスで**match**コマンドを使用した際に着信でのフィルタリングはできません。発信でのフィルタリングは可能です。Cisco IOS ソフトウェア リリース 11.2 以降のリリースでは、この制限はありません。

match の関連コマンドは次のとおりです。

- matchas-path
- match community
- matchclns

- match interface
- matchip address
- matchip nexthop
- matchip route-source
- matchmetric
- match route-type
- match tag
- set

の関連コマンドは次のとおりです。

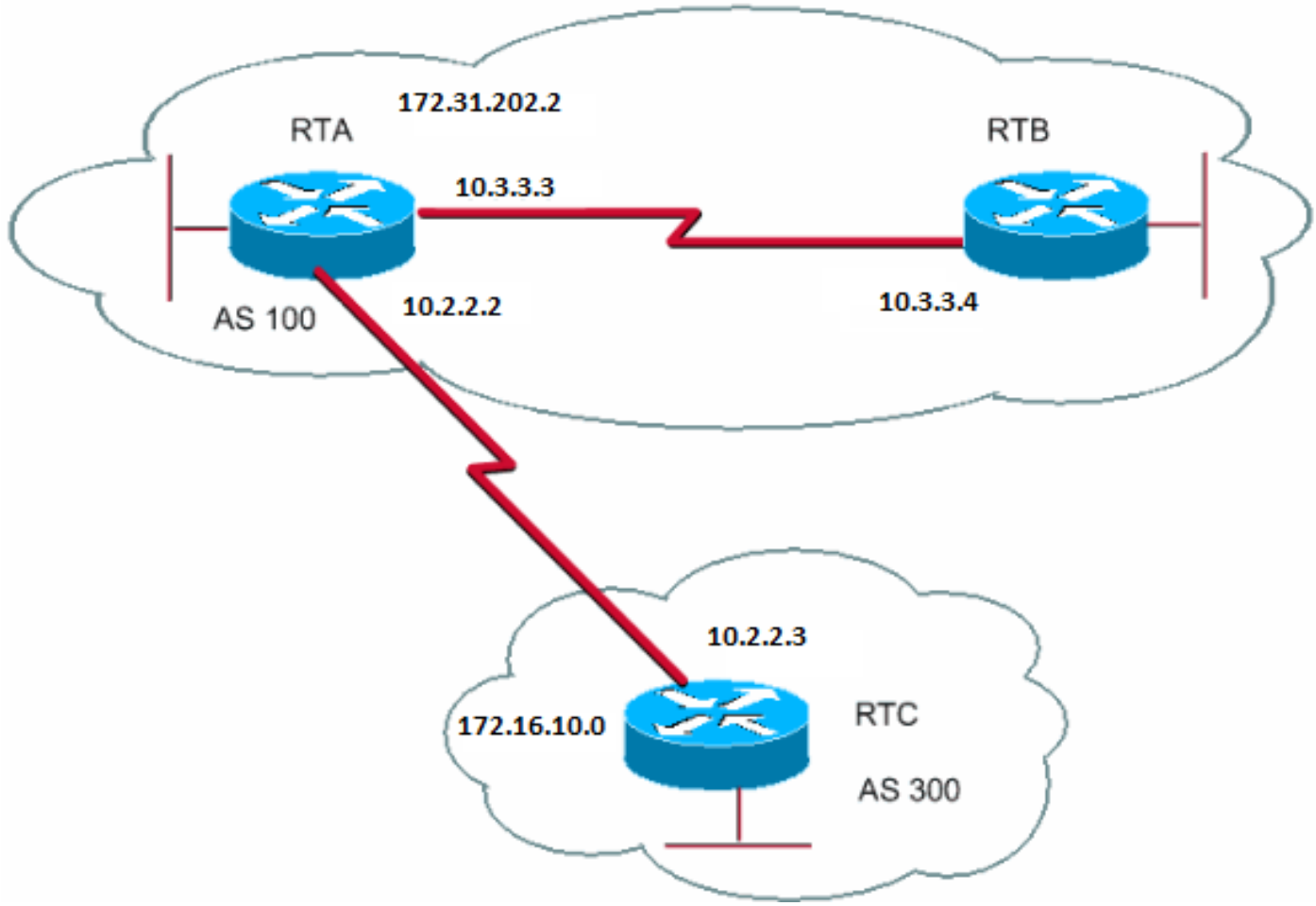
- set as-path
- set clns
- set automatic-tag

- 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

ルートマップの例をいくつか紹介します。



ルートマップの例

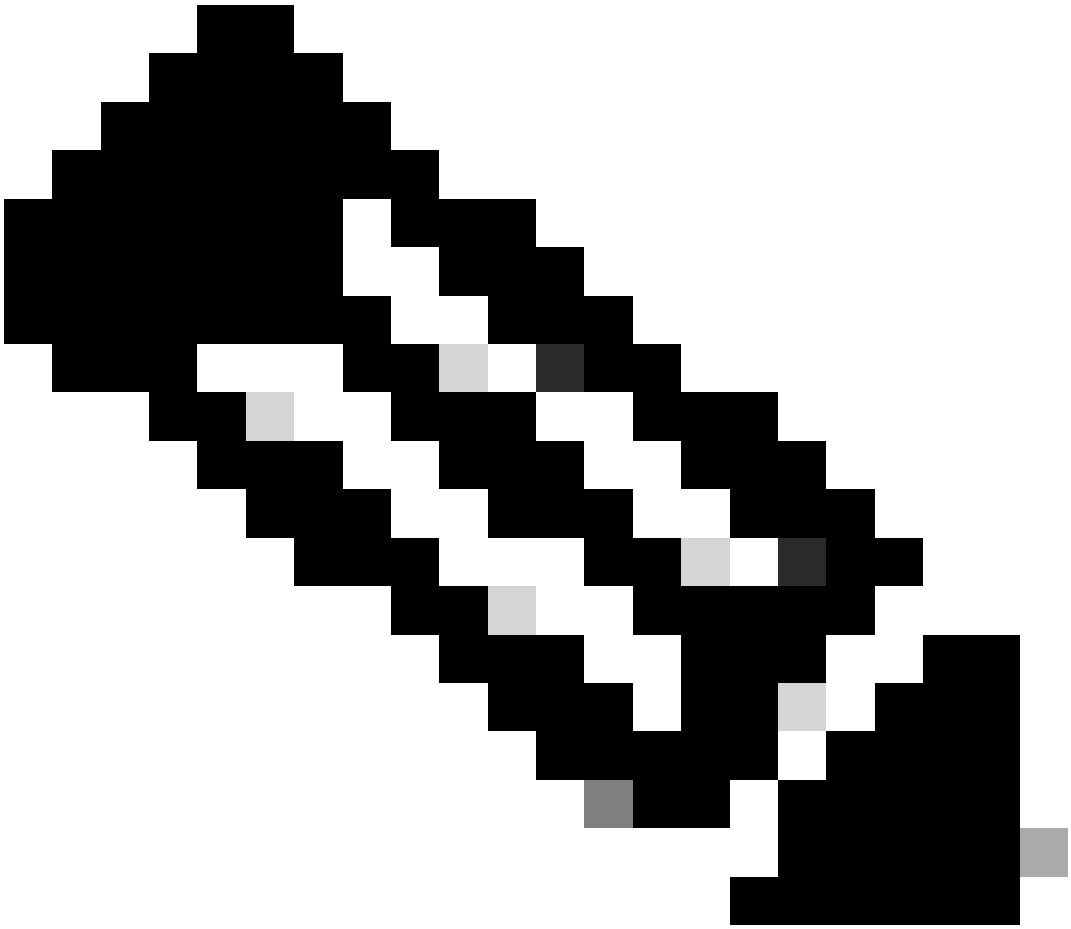
例 1

RTA と RTB は Routing Information Protocol（RIP）を実行し、RTA と RTC は BGP を実行すると仮定します。RTA は BGP 経由でアップデートを取得し、RIP に再配布します。RTA が RTB に、メトリック 2 で 172.16.10.0 に関するルートを再配布し、メトリック 5 で他のすべてのルートを再配布するとします。この場合、次の設定を使用できます。

```
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
```

この例では、IP アドレス 172.16.10.0 に一致するルートはメトリック 2 に設定されます。ここでルート マップ リストから抜けます。一致するものがなければ、ルートマップリストを下に進みます。これは、他のすべてがメトリック5に設定されていることを示します。



注:「match文のいずれにも一致しないルートはどうなるか」という質問を必ず行ってください。 デフォルトでドロップされます。

例 2

例1で、172.16.10.0に関するアップデートをAS100に受け入れさせたくなかったとします。IP アドレスに基づいて照会する場合、着信にはルート マップを適用できません。したがって、RTC で発信ルート マップを使用する必要があります。

```
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
```

BGP の開始方法とネイバーの定義方法について理解できたところで、次にネットワーク情報の交換を開始する方法を説明します。

BGP を使用してネットワーク情報を送信するには、いくつかの方法があります。次の項で 1 つずつ説明していきます。

- network コマンド
- 再配布
- スタティック ルートと再配布

network コマンド

network コマンドの形式は次のとおりです。

<#root>

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

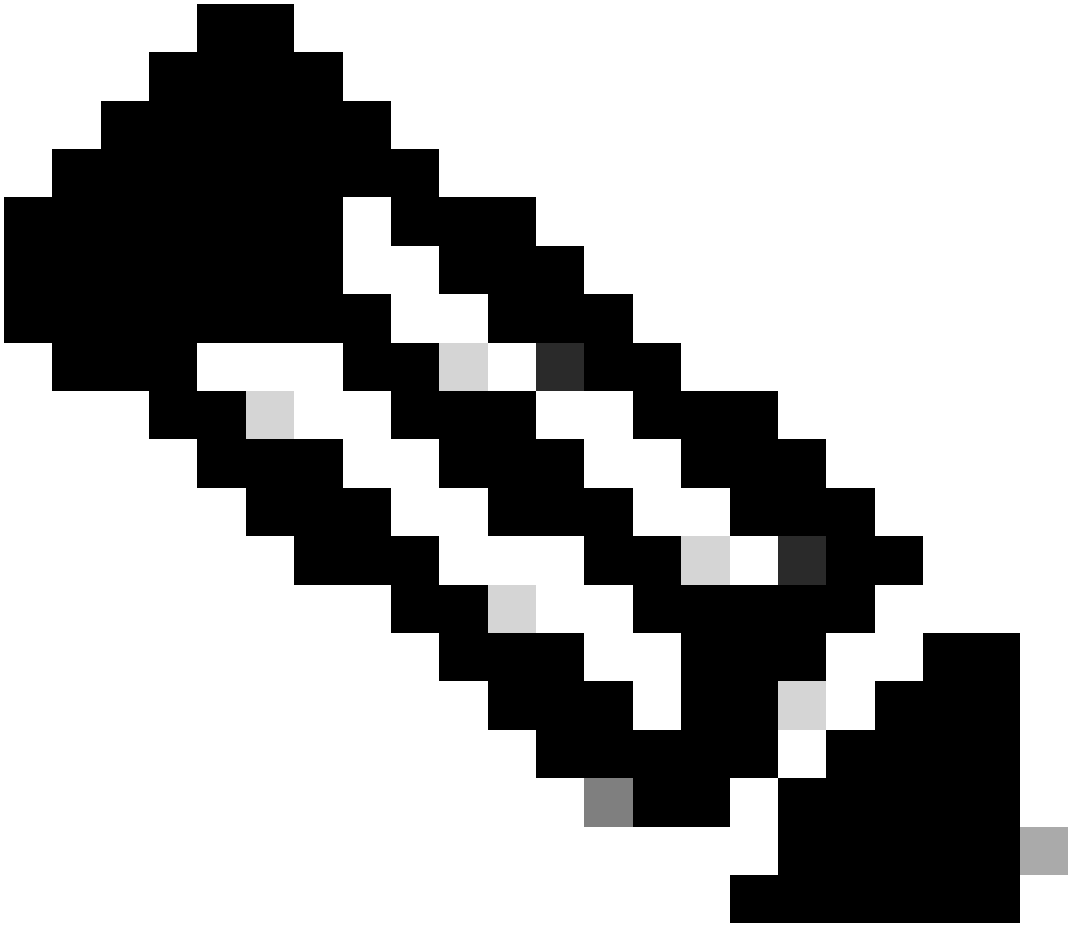
network コマンドは、このルータから発信されるネットワークを制御します。この概念は、Interior Gateway Routing Protocol (IGRP) および RIP を使用したよく知られている設定とは異なります。このコマンドは、特定のインターフェイス上で BGP を実行するために使用するのではなく、その代わりに、BGPがこのルータから発信する必要があるネットワークをBGPに指示します。BGP バージョン 4 (BGP4) はサブネット化およびスーパーネット化を処理できるため、このコマンドはマスク部分を使用します。network コマンドのエントリは最大200まで使用できます。

network コマンドは、アドバタイズの対象となるネットワークが、接続済み、スタティック、またはダイナミックに学習済みとしてルータで認識されている場合に機能します。

network コマンドの例を示します。

```
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
```

この例では、ルータ A が 192.168.213.0/16 のネットワーク エントリを生成します。/16 は、クラス C アドレスのスーパーネットを使用して、最初の 2 つのオクテット (最初の 16 ビット) をアドバタイズすることを意味します。

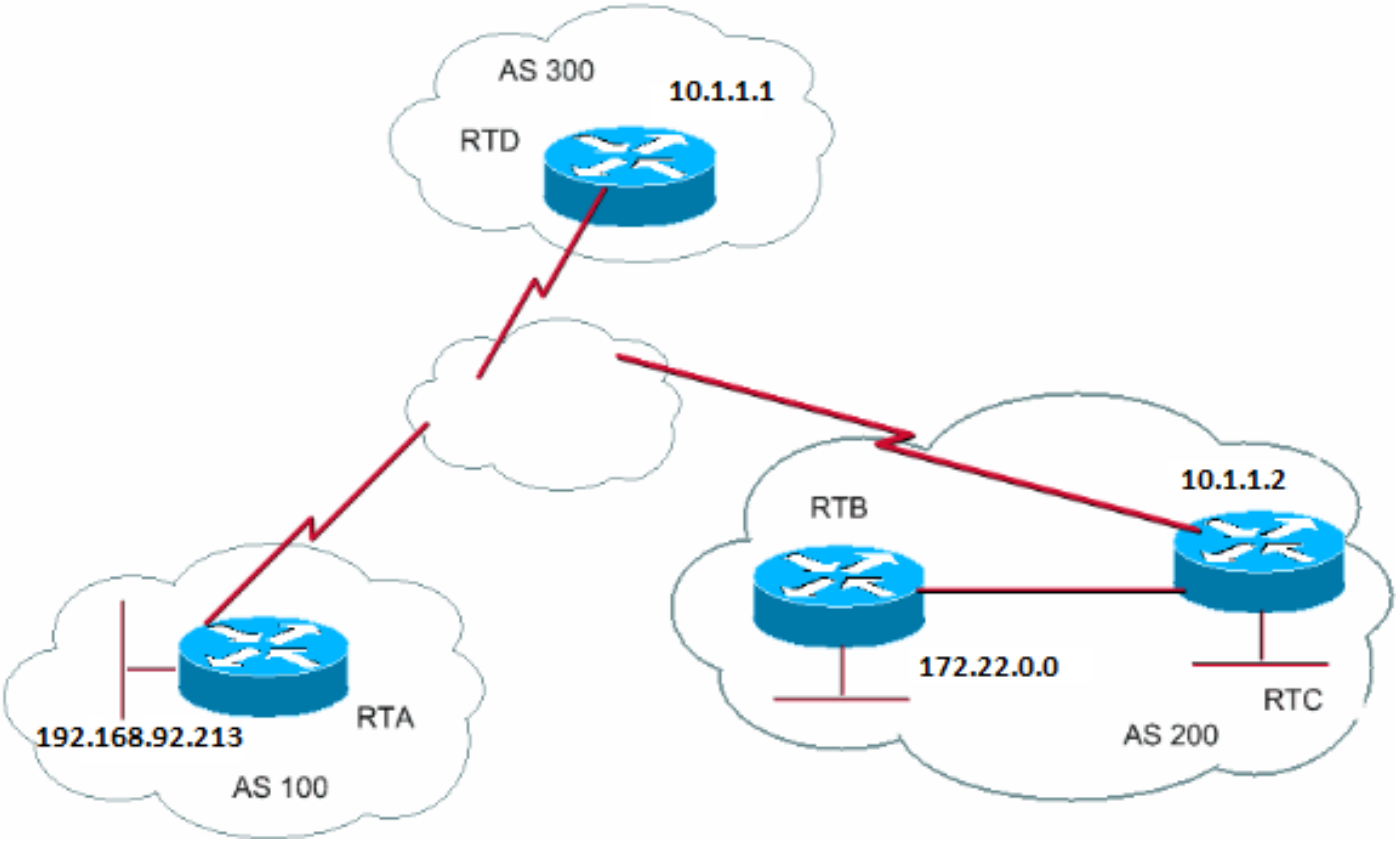


注：スタティックルートでは一致するエントリがルーティングテーブルに挿入されるため、ルータが192.168.213.0を生成するにはスタティックルートが必要です。

再配布

network コマンドは、BGP経由でネットワークをアドバタイズする方法の1つです。他にも、BGP に IGP を再配布するという方法があります。IGP には、IGRP、Open Shortest Path First（OSPF）プロトコル、RIP、Enhanced Interior Gateway Routing Protocol（EIGRP）などのプロトコルがあります。この再配布は内部ルートをすべて BGP にダンプするので危険なように思えますが、ルートの一部は BGP 経由で学習されている可能性があり、これらのルートは再度送出する必要はありません。フィルタリングを行う場合は、アドバタイズするインターネット専用ルートに必ず送信し、すべてのルートに送信しないように注意してください。次に例を示します。

RTA は 192.168.92.213 をアナウンスし、RTC は 172.22.0.0 をアナウンスします。RTC の設定を見てみましょう。



networkコマンドを発行すると、次のようになります。

```
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.
```

代わりに再配布を使用する場合は次のとおりです。

```
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.
```

この再配布により、AS から 192.168.92.213 が発信されます。あなたは192.168.92.213の発信元ではなく、AS100が発信元です。そのため、ASによるネットワークからのソースの流出を防ぐには、フィルタを使用する必要があります。正しい設定は次のとおりです。

```
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
```

access-listコマンドを使用して、AS200から発信されるネットワークを制御します。

BGP への OSPF の再配布は、他の IGP の再配布と若干異なります。redistribute ospf 1router bgp の下にある単純な問題が機能しません。internal、external、**nssa-external** などの特定のキーワードは、それぞれのルートを再配布するために必要です。詳細については、『BGPへのOSPFルートの再配布について』を参照してください。

スタティック ルートと再配布

ネットワークまたはサブネットの発信に、常にスタティック ルートを使用することもできます。他の方法との唯一の違いは、BGP がこれらのルートに不完全または不明な送信元があると見なすことです。次の例を使用すると、「再配布」セクションの例と同じ結果が得られます。

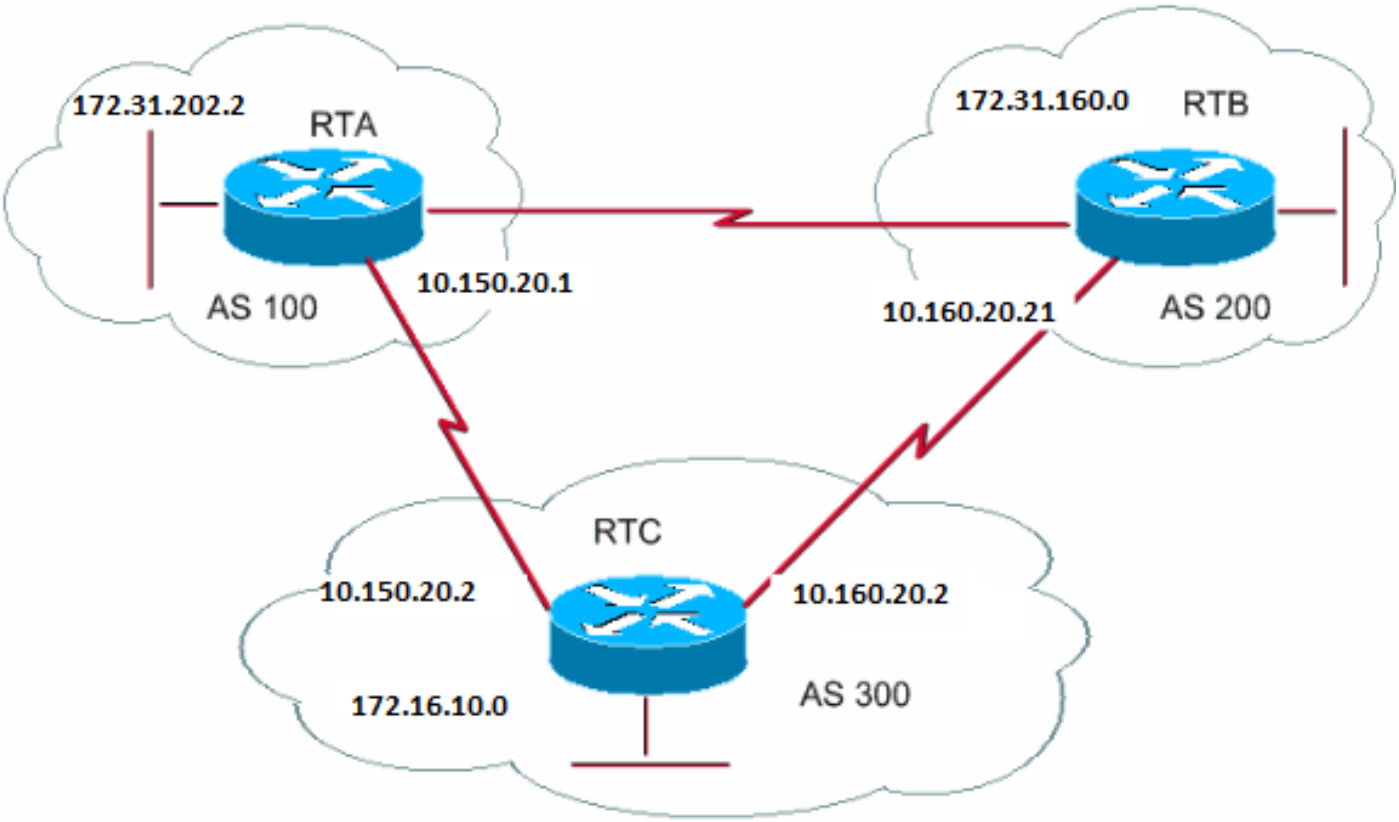
```
RTC#
router eigrp 10
 network 172.22.0.0
 redistribute bgp 200
 default-metric 1000 100 250 100 1500

router bgp 200
 neighbor 10.1.1.1 remote-as 300
 redistribute static

ip route 172.22.0.0 255.255.255.0 null0
```

null0 インターフェイスは、パケットの無視を意味します。そのため、パケットを受け取り、存在する172.22.0.0よりも具体的な一致がある場合、ルータはパケットを特定の一致に送信します。それ以外の場合、ルータはパケットを無視します。スーパーネットをアドバタイズするには、この方法が最適です。

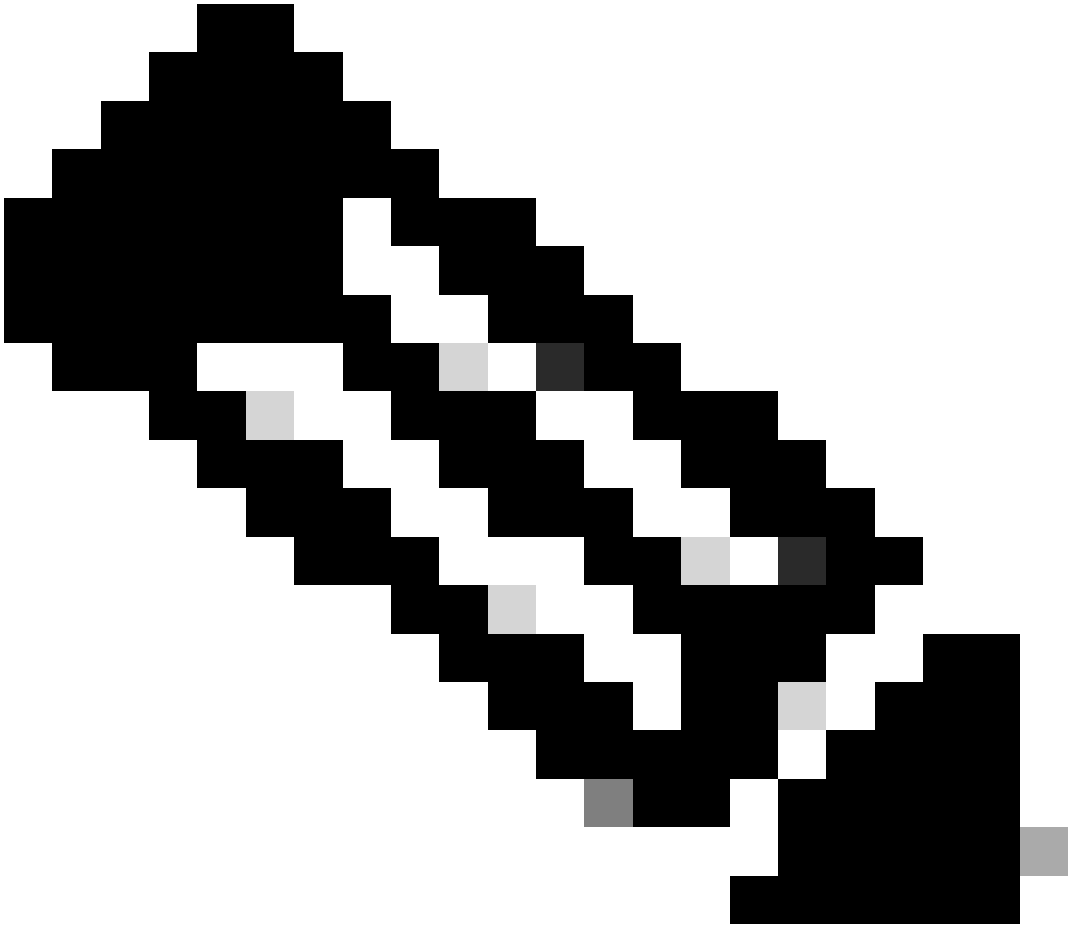
このドキュメントでは、AS からルートを発信するために使用できるさまざまな方法について説明しています。これらのルートは、BGP が内部または外部のネイバーを介して学習した他の BGP ルートとは別に生成されることに注意してください。BGP は 1 つのピアから学習した情報を他のピアへ伝えます。network コマンド、再配布、またはスタティックによって生成されたルートでは、これらのネットワークの起点(origin)がASであることが示される点が異なります。

再配布では、常に BGP が IGP にインジェクトされます。

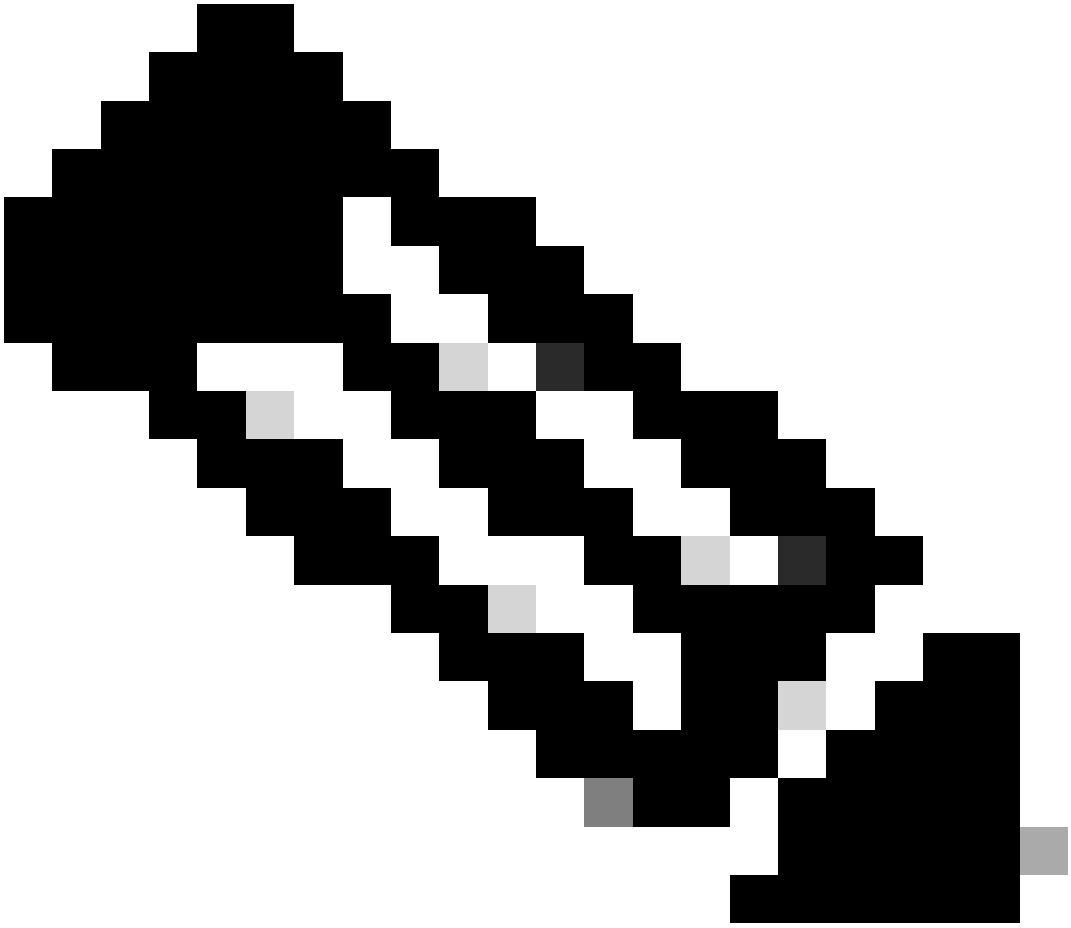
ランダム データの例は次のとおりです。




```
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:RTCに、AS100およびAS200から着信するネットワークを伝播するだけでなく、これらのネットワークを生成させる場合を除き、RTCにネットワーク172.31.202.2またはネットワーク172.31.160.0を設定する必要はありません。やはり異なる点は、networkコマンドが、これらの同じネットワークに対して、AS300もこれらのルートの送信元であることを示す追加のアドバタイズメントを付加することです。



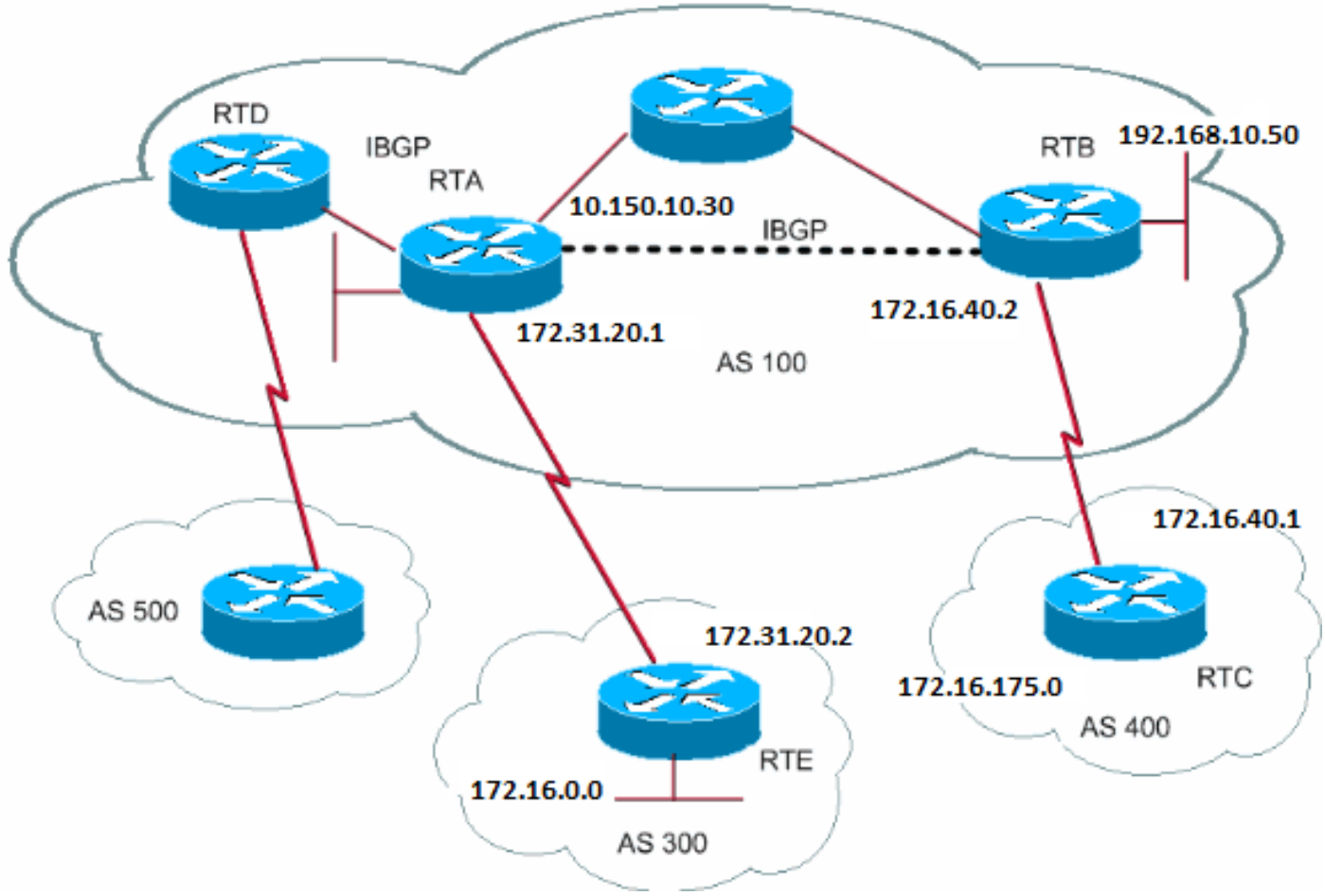
注:BGPは自身のASから発信されたアップデートを受け入れないことに注意してください。この拒否によって、ループフリーなドメイン間トポロジが実現します。

たとえば、上記の例の AS200 に AS100 への直接 BGP 接続があるとします。RTAはルート172.31.202.2を生成し、そのルートをAS300に送信します。次に、RTC はこのルートを AS200 に渡し、送信元を AS100 のまま保持します。RTBは172.31.202.2をAS100に渡しますが、送信元はAS100のままです。RTA はアップデートが自身の AS から発信されていることを検知して、このアップデートを無視します。

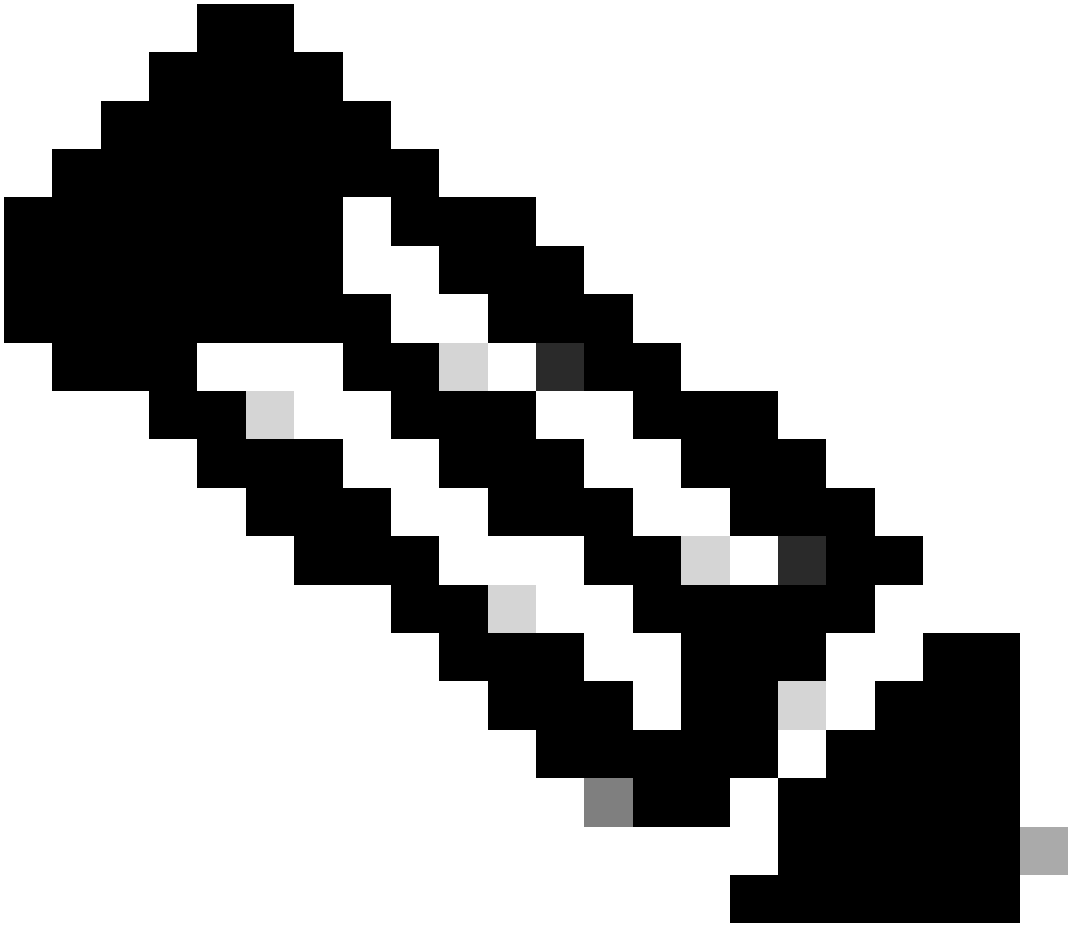
iBGP

ASを他のASへの中継システムとして機能させる場合は、iBGPを使用します。eBGPを介して学習し、IGPに再配布してから別のASに再配布する場合も、同じことができます。しかし、iBGPは、AS内で情報を交換するためのより柔軟で効率的な方法を提供し

ます。たとえば、iBGP にはローカル プリファレンスを使用して AS からの最良の出力点を制御する方法が用意されています。ローカルプリファレンスについての詳細は、「ローカルプリファレンスアトリビュート」セクションを参照してください。




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

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

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



注：BGPスピーカが自身のAS内の他のBGPスピーカ(iBGP)からアップデートを受信した場合、アップデートを受信したBGPスピーカは、自身のAS内の他のBGPスピーカにその情報を再配布しないことに注意してください。アップデートを受信した BGP スピーカは、自身の AS 外にある他の BGP スピーカにこの情報を再配布します。したがって、AS 内の iBGP スピーカ間でフル メッシュを維持する必要があります。

RTAとRTBはiBGPを実行しています。また、RTA と RTD も iBGP を実行しています。RTB から RTA に送信された BGP アップデートは、AS 外にある RTE に送信されます。このアップデートは、AS 内にある RTD には送信されません。このため、アップデートのフローが中断されないように RTB と RTD の間で iBGP ピアリングを行う必要があります。

BGP 決定アルゴリズム

さまざまな自律システムから複数の宛先に関するアップデートを受信した BGP は、特定の宛先に到達するためのパスを選択する

必要があります。BGP は特定の宛先に到達するパスを 1 つだけ選択します。

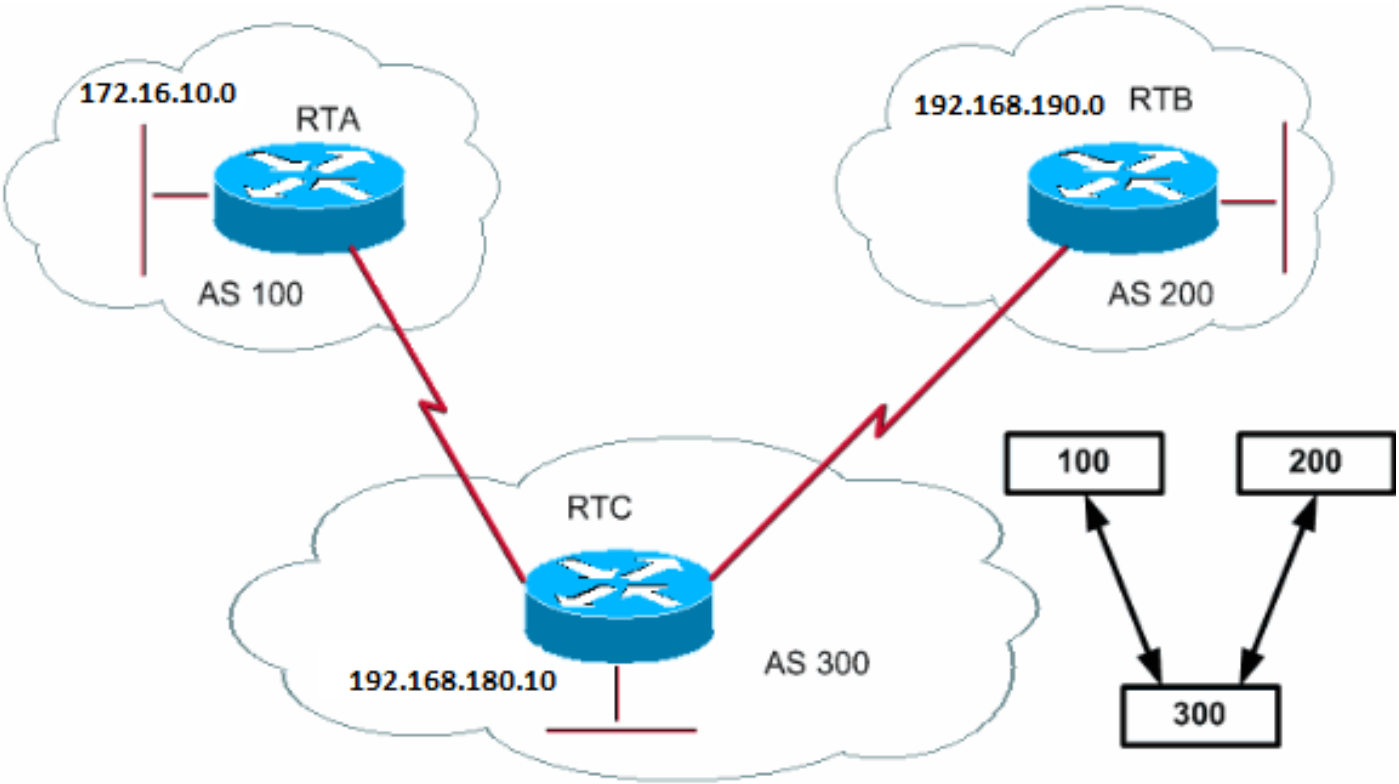
BGPはこの決定を、ネクストホップ、管理上の重み、ローカルプリファレンス、ルートの送信元、パスの長さ、送信元コード、メトリック、その他のアトリビュートなど、さまざまなattributesに基づいて行います。

BGP は常にベスト パスをネイバーに伝達します。詳細は、『BGPでベストパスを選択するアルゴリズム』を参照してください。

次のセクションでは、これらのアトリビュートとその使用法について説明します。

BGP ケース スタディ 2

AS_PATH 属性



ルート アップデートが AS を通過するたびに、AS 番号がそのアップデートに付加されます。AS_PATH 属性は、宛先に到達するために実際にルートが通過した AS 番号のリストです。AS_SET は、通過したすべての AS の順序付けられた数学的集合 {} です。AS_SETの例については、このドキュメントの「CIDR例2(as-set)」セクションを参照してください。

このセクションの例では、RTBはAS200のネットワーク192.168.190.0をアドバタイズします。そのルートが AS300 を通過すると、RTC はネットワークに自身の AS 番号を付加します。192.168.190.0がRTAに到達すると、ネットワークには2つのAS番号（最初に200、次に300）が接続されます。RTA から 192.168.190.0 に到達するパスは（300、200）になります。

同じプロセスが 172.16.10.0 と 192.168.180.10 にも当てはまります。RTBはパス(300、100)を使用する必要があります。RTBは172.16.10.0に到達するために、AS300を通過した後、AS100を通過します。RTC の場合、192.168.190.0 に到達するにはパス（200）、172.16.10.0 に到達するにはパス（100）を通過する必要があります。

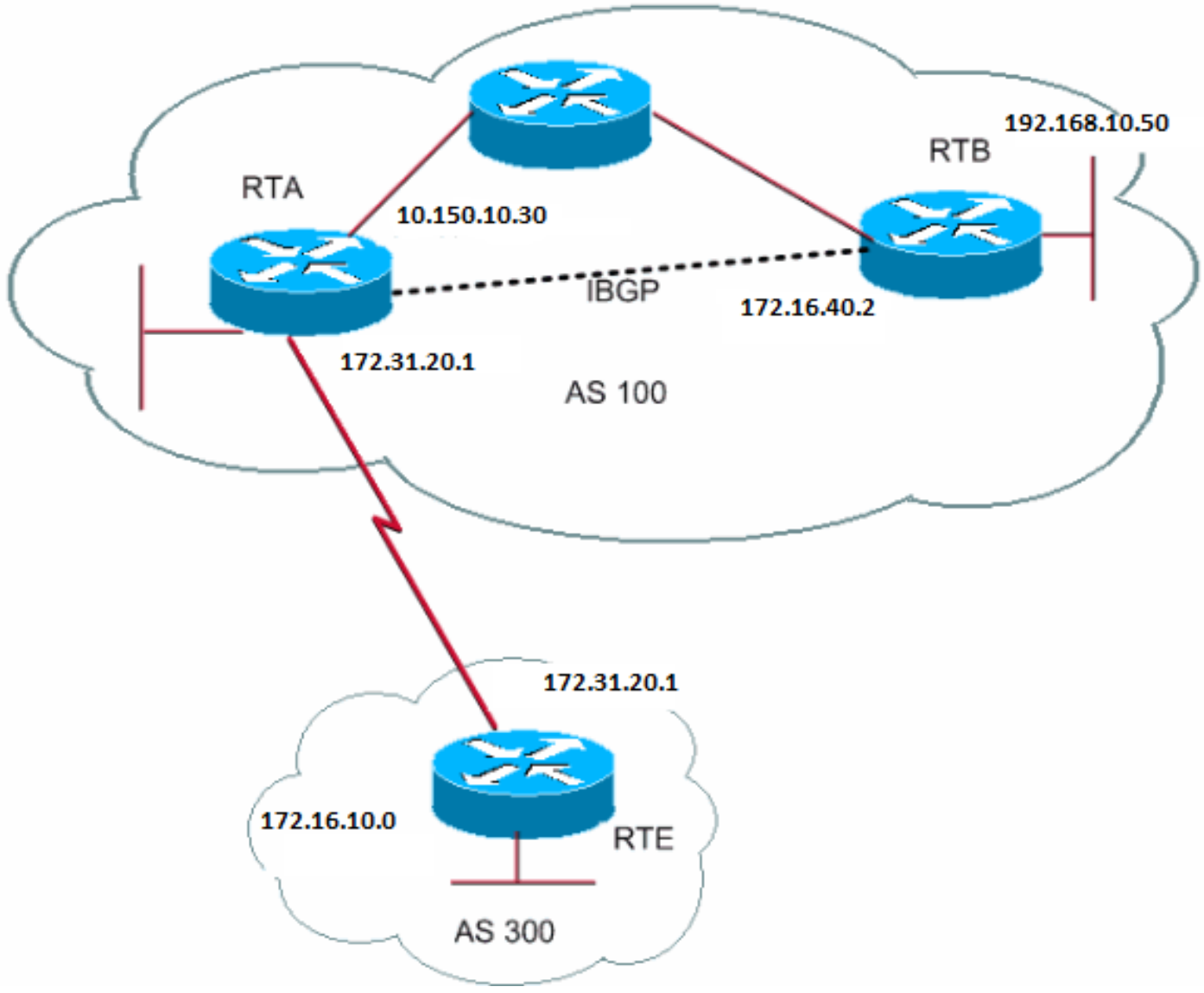
送信元属性

送信元はパス情報の送信元を定義する必須属性です。送信元属性の値は次の 3 つです。

- IGP：Network Layer Reachability Information（NLRI; ネットワーク レイヤ到着可能性情報）が発信元 AS の内部にあることを示します。これは通常、 **bgp network**コマンドを発行した場合に発生します。BGPテーブル内のaniinはIGPを示しています。

- EGP：NLRI は外部ゲートウェイ プロトコル（EGP）を介して学習されています。BGPテーブル内のAneinはEGPを示しています。

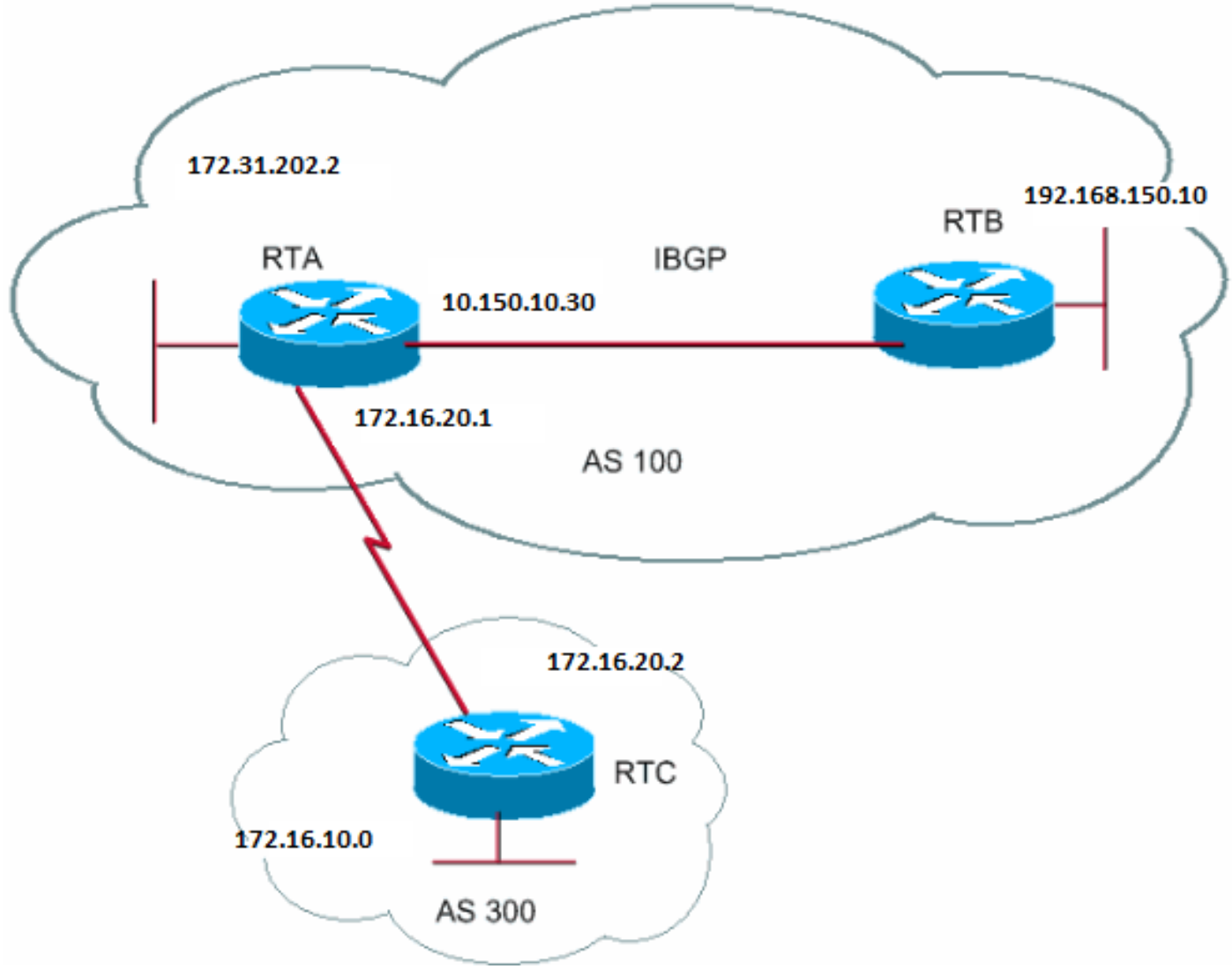
- INCOMPLETE：NLRI が不明であるか、他の手段で学習されています。INCOMPLETE は通常、他のルーティング プロトコルから BGP にルートが再配布され、ルートの送信元が不完全である場合に発生します。BGP テーブルの？は、INCOMPLETE を示しています。



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

ip route 192.168.190.0 255.255.0.0 null0

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

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

RTA は 300 i を経由して 172.16.10.0 に到達します。「300 i」は、次の AS パスが 300 で、ルートの送信元が IGP であることを意味します。また、RTA は i 経由で 192.168.10.150 に到達します。この「i」は、エントリが同じ AS 内に存在し、送信元が IGP であることを意味します。RTE は 100 i を経由して 172.31.202.2 に到達します。「100 i」は、次の AS が 100 で、送信元が IGP であることを意味します。また、RTE は 100 ? 経由で 192.168.190.0 に到達します。「100 ?」は、次の AS が 100 であり、起点 (origin) が INCOMPLETE でスタティック ルートから発信されていることを意味します。

BGP ネクスト ホップ属性



BGP ネクスト ホップ属性

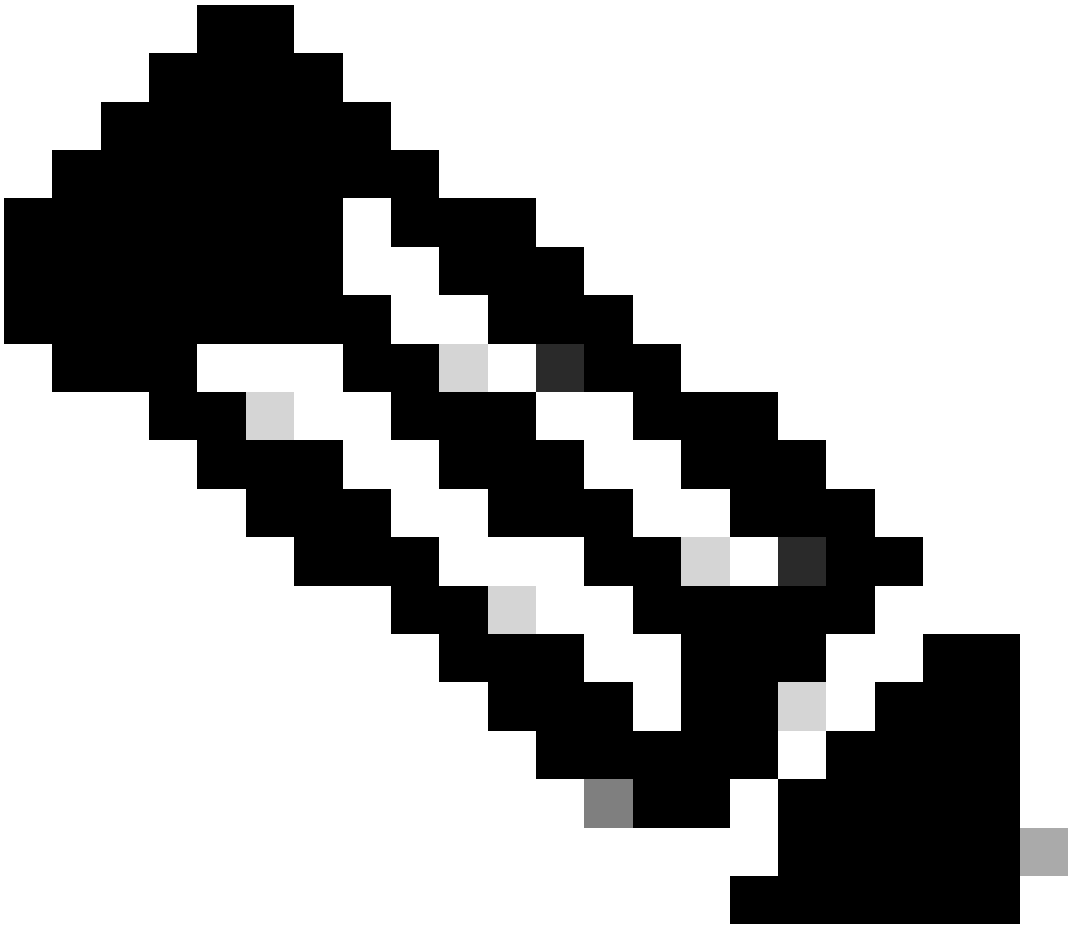
BGP ネクスト ホップ属性は、特定の宛先に到達するために使用されるネクスト ホップ IP アドレスです。

eBGPの場合、ネクストホップは常に、neighbor コマンドで指定されたネイバーのIPアドレスです。この項の例では、RTC はネクスト ホップ 172.31.20.2 を使用して RTA に 172.16.10.0 をアドバタイズします。RTA はネクスト ホップ 172.31.20.1 を使用して RTC に 172.31.202.2 をアドバタイズします。iBGPの場合、プロトコルはeBGPがアドバタイズするネクストホップをiBGPに伝達する必要があることを示します。このルールに従い、RTA はネクスト ホップ 172.31.20.2 を使用して iBGP ピアの RTB に 172.16.10.0 をアドバタイズします。RTBが172.16.10.0に到達するためのネクストホップは172.31.20.2であり、10.150.10.30ではありません。

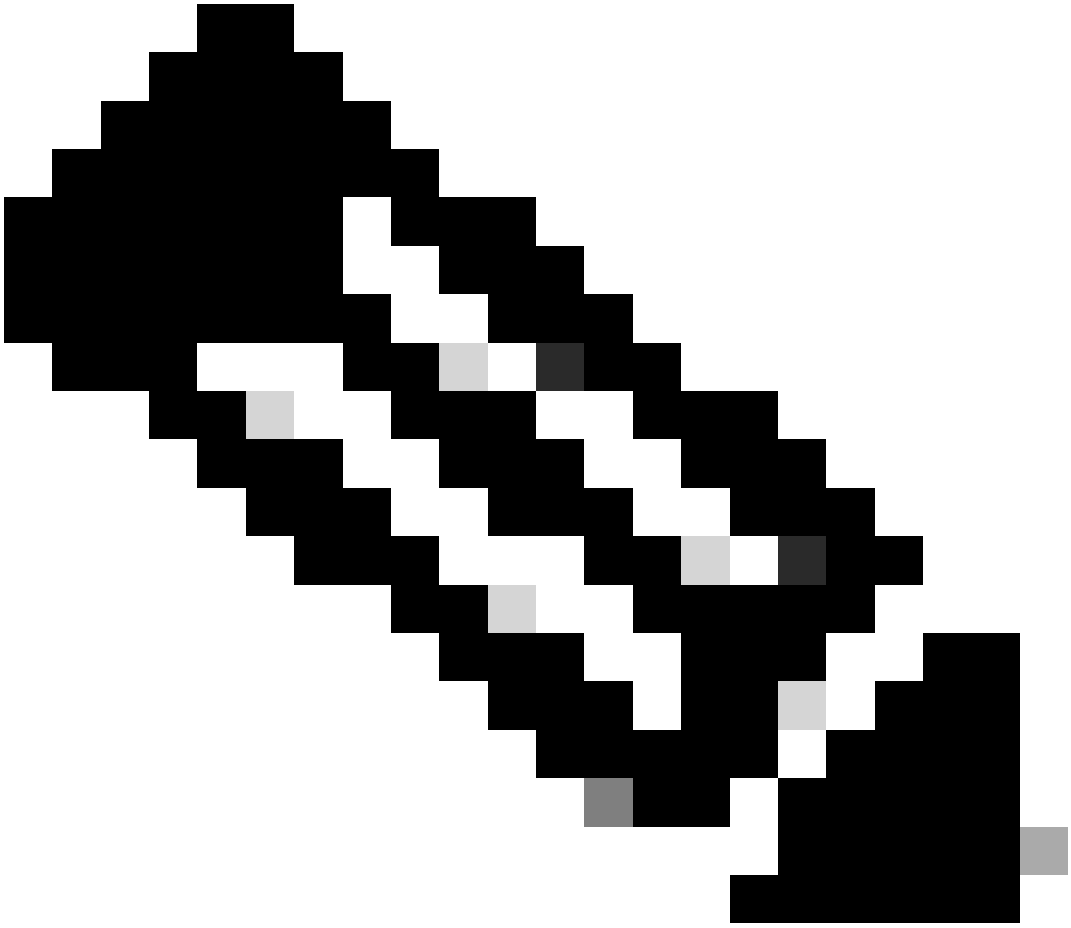
RTB が IGP 経由で 172.31.20.2 に到達できることを確認します。到達できない場合は、ネクスト ホップ アドレスがアクセス不能であるため、RTB は 172.16.10.0 宛てのパケットをドロップします。たとえば RTB で iGRP が実行されている場合は、RTA のネットワーク 172.16.10.0 でも iGRP を実行できます。RTC へのリンクで iGRP をパッシブにして、BGP のみが交換されるようにする必要があります。

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

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

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



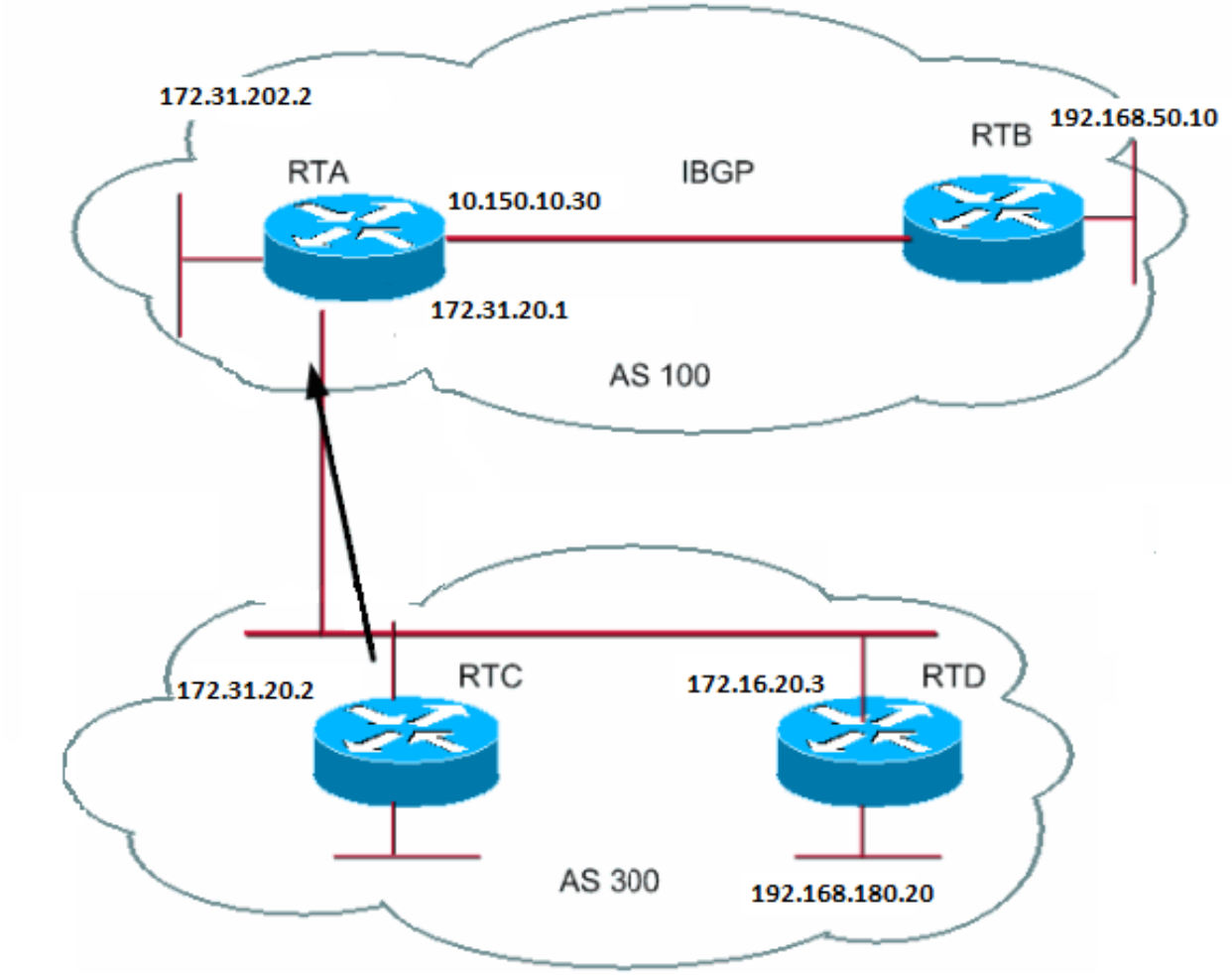
注:RTCは、ネクストホップ172.31.20.2を使用してRTAに172.16.10.0をアドバタイズします。



注:RTAは、ネクストホップ172.31.20.2を使用してRTBに172.16.10.0をアドバタイズします。eBGP のネクスト ホップは iBGP で伝達されます。

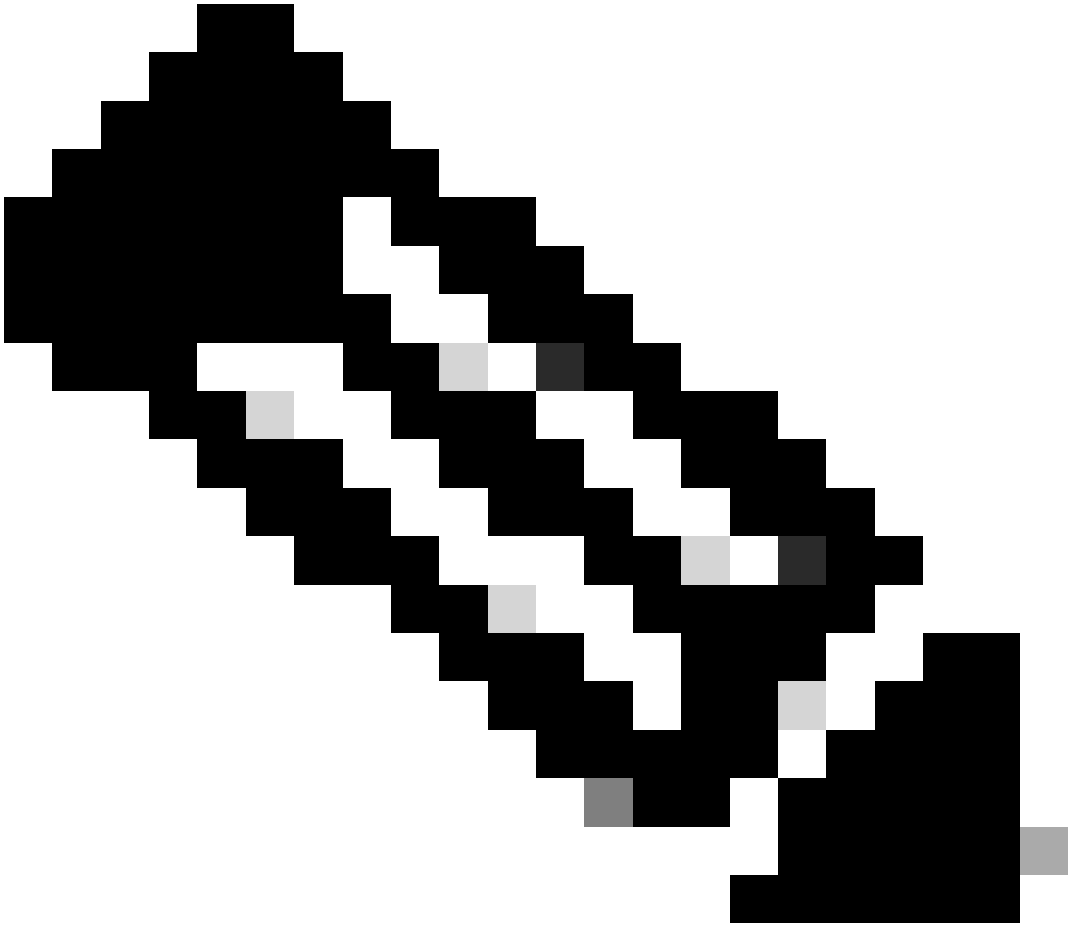
マルチアクセスおよび非ブロードキャストマルチアクセス(NBMA)ネットワークを扱う場合は、特に注意が必要です。詳細は、「BGPネクストホップ (マルチアクセスネットワーク) 」と「BGPネクストホップ(NBMA)」を参照してください。

BGP ネクスト ホップ (マルチアクセス ネットワーク)



この例は、イーサネットなどのマルチアクセス ネットワークでネクスト ホップがどのように動作するかを示しています。

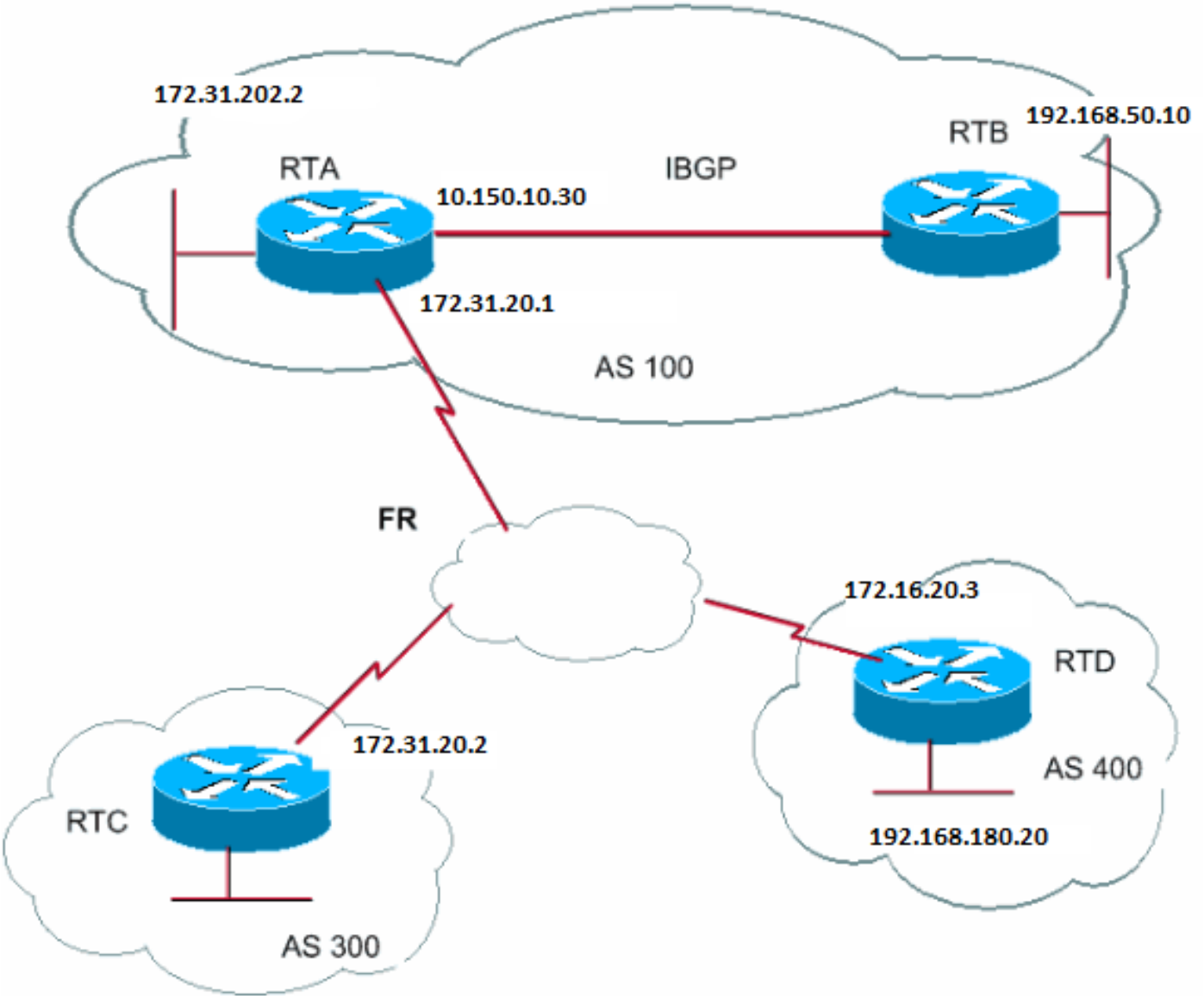
AS300 の RTC と RTD が OSPF を実行していると仮定します。RTC は RTA との間で BGP を実行しています。RTC は、172.16.20.3 経由でネットワーク 192.168.180.20 に到達できます。RTC が 192.168.180.20 に関する BGP アップデートを RTA に送信する際には、ネクスト ホップとして 172.16.20.3 が使用されます。RTC は自身の IP アドレス 172.31.20.2 を使用しません。RTC がこのアドレスを使用する理由は、RTA、RTC、RTD 間のネットワークがマルチアクセス ネットワークであるためです。RTA が 192.168.180.20 に到達するには、ネクスト ホップとして RTD を使用したほうが RTC 経由で余分にホップするよりも合理的です。



注:RTCはネクストホップ172.16.20.3を使用してRTAに192.168.180.20をアドバタイズします。

RTA、RTC、および RTD への共有メディアがマルチアクセスではなく NBMA である場合は、さらに複雑になります。

BGP ネクスト ホップ (NBMA)



この図の共有メディアはクラウドで示されています。共有メディアがフレーム リレーまたは NBMA クラウドであれば、イーサネット経由で接続している場合とまったく同じ動作になります。RTC はネクスト ホップ 172.16.20.3 を使用して RTA に 192.168.180.20 をアドバタイズします。

問題は、RTA には RTD への直接相手先固定接続（PVC）がなく、ネクスト ホップに到達できないことです。この場合、ルーティングは失敗します。

この状況に対処するには、next-hop-selfコマンドを使用します。

next-hop-self コマンド

「BGPネクストホップ(NBMA)」の例に示したネクストホップの状況では、next-hop-self コマンドを使用できます。構文は次のとおりです。

```
<#root>
```

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

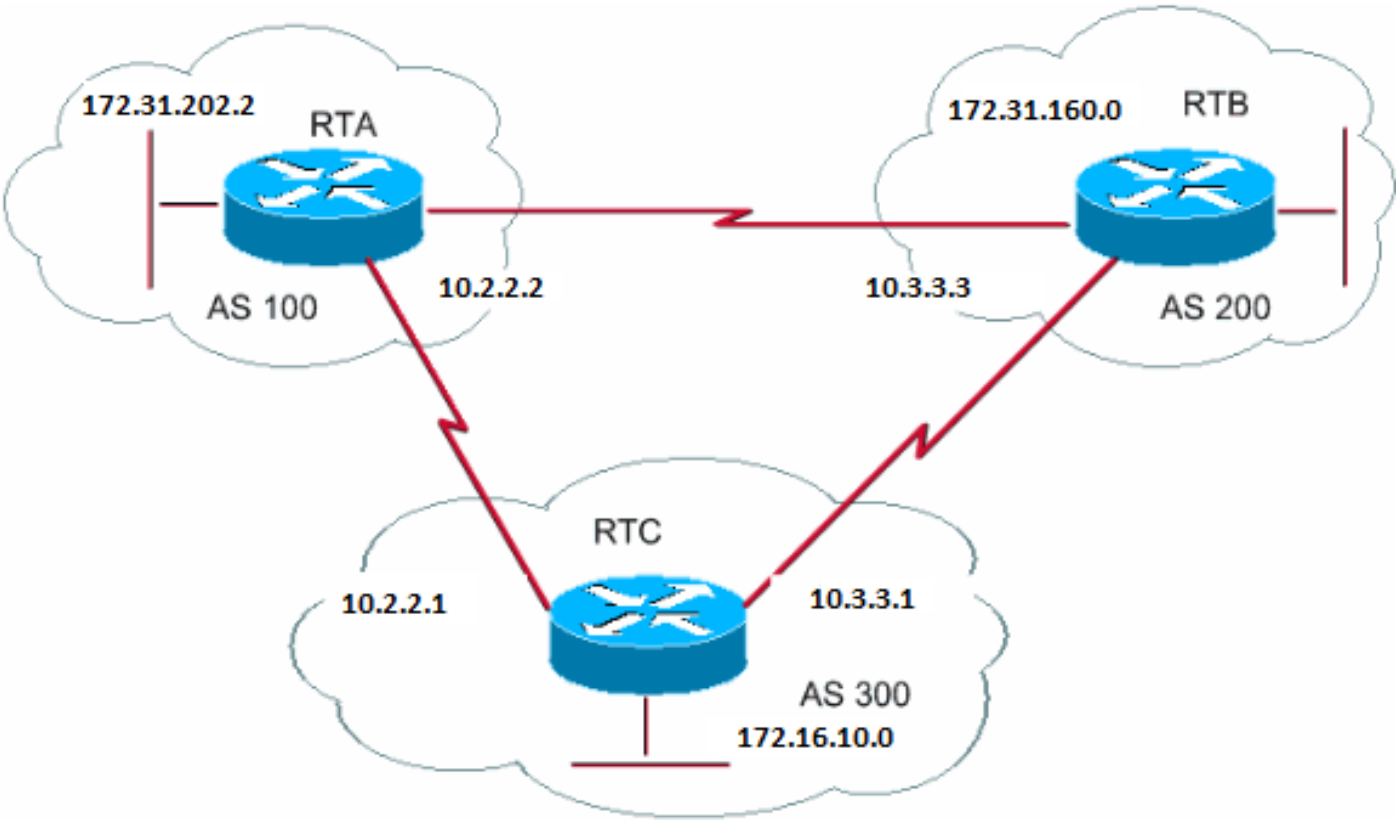
next-hop-selfコマンドを使用すると、特定のIPアドレスがBGPのネクストホップとして使用されます。

「BGP ネクスト ホップ (NBMA) 」の例では、次の設定を使用することで問題を解決できます。

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

RTC はネクスト ホップ 172.31.20.2 を使用して 192.168.180.20 をアドバタイズします。

BGP バックドア



前の図では、RTAとRTCはeBGPを実行しています。RTB と RTC は eBGP を実行しています。RTA と RTB はいずれかの

IGP（RIP、IGRP、またはその他のプロトコル）を実行しています。定義上、eBGP アップデートの距離は IGP の距離より小さい 20 です。デフォルトの距離は次のとおりです。

- RIP：120
- IGRP：100
- EIGRP：90
- OSPF：110

RTAは、次の2つのルーティングプロトコルを介して172.31.160.0に関するアップデートを受信します。

- 距離が 20 の eBGP
- 距離が 20 より大きい IGP

デフォルトでは、BGP の距離は次のとおりです。

- 外部距離：20
- 内部距離：200

- ローカル距離：200

ただし、distance コマンドを使用してデフォルトの距離を変更できます。

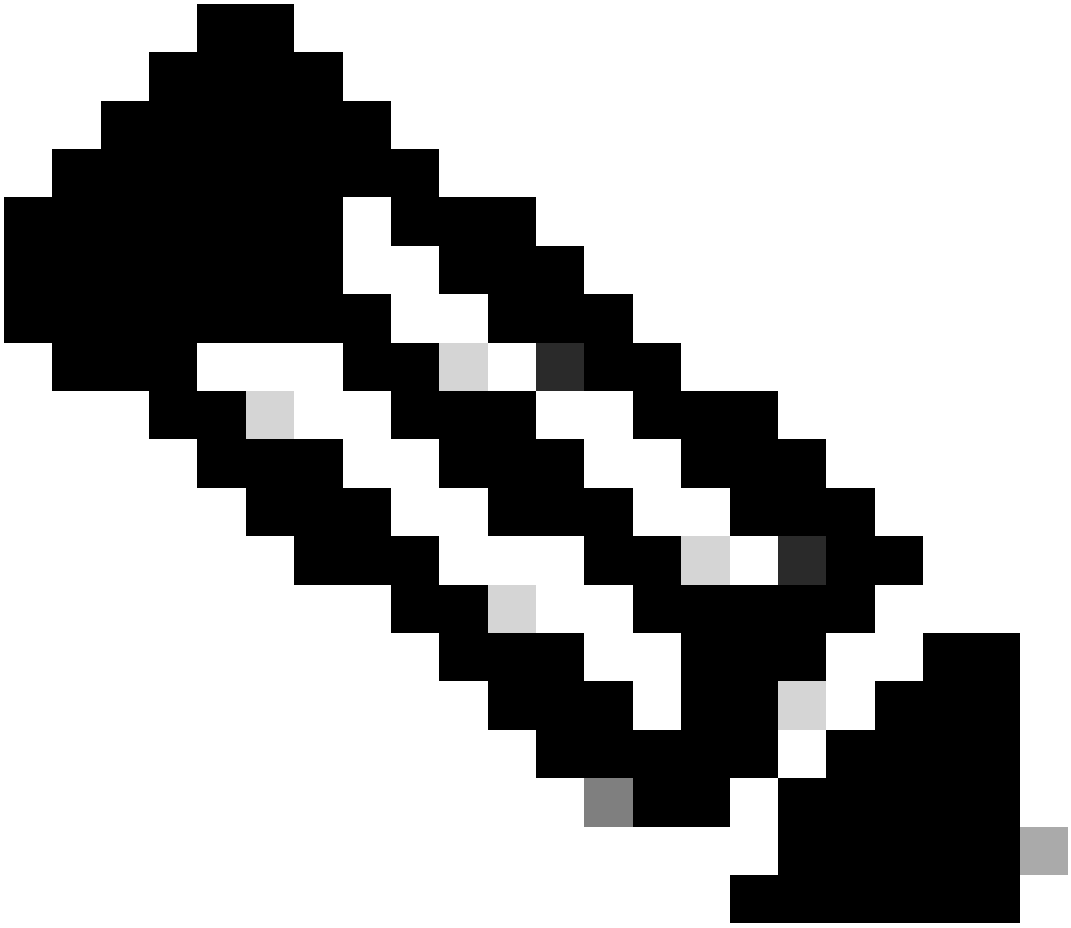
<#root>

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

RTA は、距離がより短い RTC 経由の eBGP を選択します。

RTAがRTB(IGP)を介して172.31.160.0について学習するようにするには、次の2つのオプションがあります。

- eBGP の外部距離または IGP の距離を変更する。



注：この変更は推奨されません。

- BGP バックドアを使用する。

BGP バックドアを使用すると、IGP ルートが優先ルートになります。

networkaddressbackdoorコマンドを発行します。

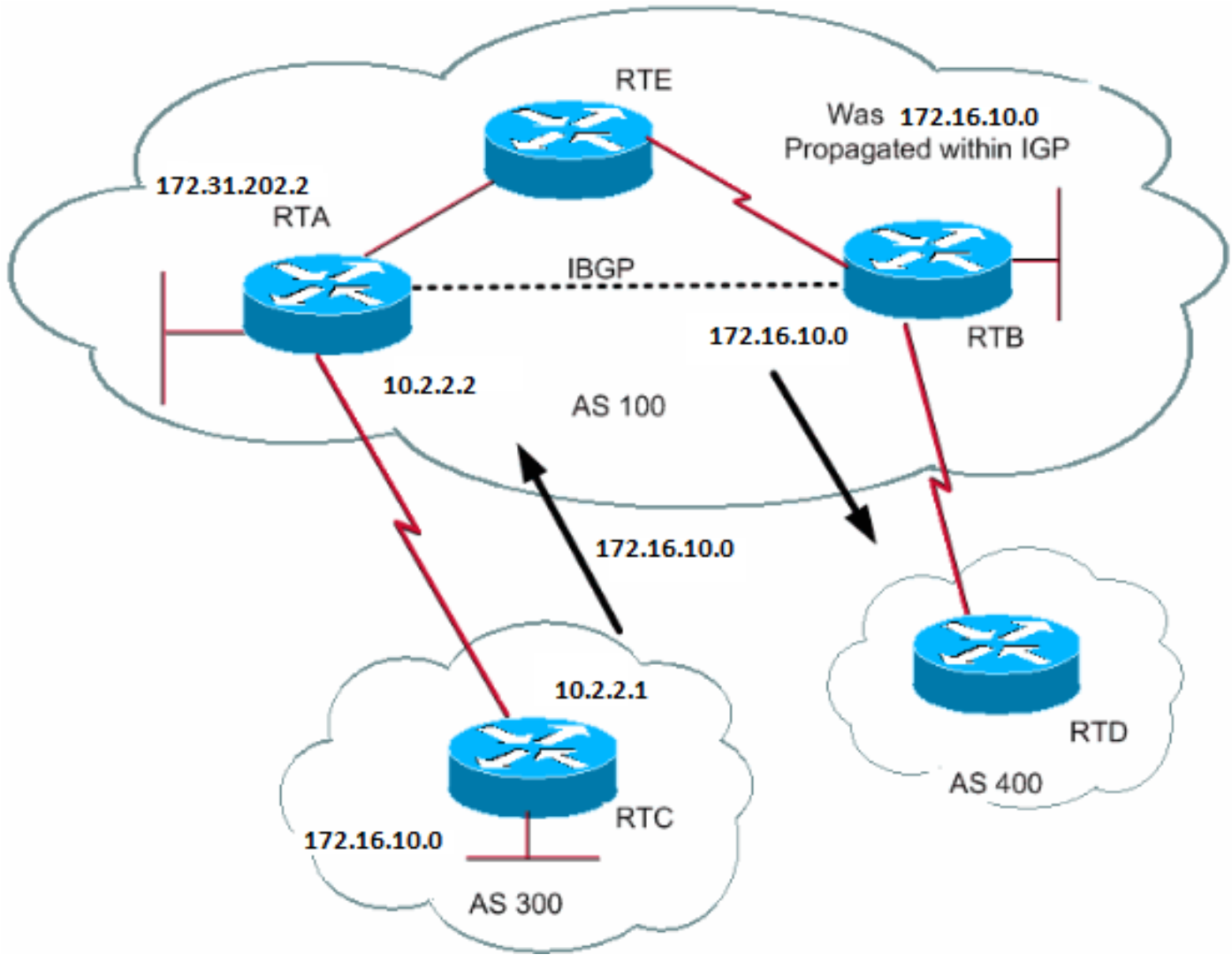
設定するネットワークは、IGP を使用して到達するネットワークです。BGP アップデートでこのネットワークがアドバタイズされない点を除き、BGP では、このネットワークをローカルに割り当てられたネットワークと同様に扱います。

```
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
```

ネットワーク172.31.160.0はローカルエントリとして扱われますが、通常のネットワークエントリとしてはアドバタイズされません。

RTA は、距離が 90 の EIGRP 経由で RTB から 172.31.160.0 を学習します。また、距離が 20 の eBGP 経由で RTC からアドレスを学習します。通常は eBGP が優先されますが、network backdoor コマンドを使用しているため、EIGRP が優先されます。

同期



同期について説明する前に、次のシナリオについて考えてみましょう。AS300のRTCが172.16.10.0に関するアップデートを送信します。RTA と RTB は iBGP を実行しているため、RTB はアップデートを受信し、ネクスト ホップ 10.2.2.1 経由で 172.16.10.0 に到達できます。ネクスト ホップは iBGP 経由で伝達されることに注意してください。RTB はネクスト ホップに到達するために、RTE にトラフィックを送信する必要があります。

RTA が IGP にネットワーク 172.16.10.0 をまだ再配布していないと仮定します。この時点で、RTE では 172.16.10.0 の存在すら認識されていません。

RTBが172.16.10.0に到達できることをAS400にアドバタイズし始めると、RTDからRTBへの宛先172.16.10.0のトラフィックが流れ込み、RTEでドロップされます。

同期では、ASがトラフィックを別のASから第3のASに渡す場合、AS内のすべてのルータがIGPを介してルートを学習するまでは、BGPはルートをアドバタイズしてはならないことを規定しています。BGP は、IGP によって AS 内にルートが伝達されるまで待機します。その後、BGP は外部ピアにルートをアドバタイズします。

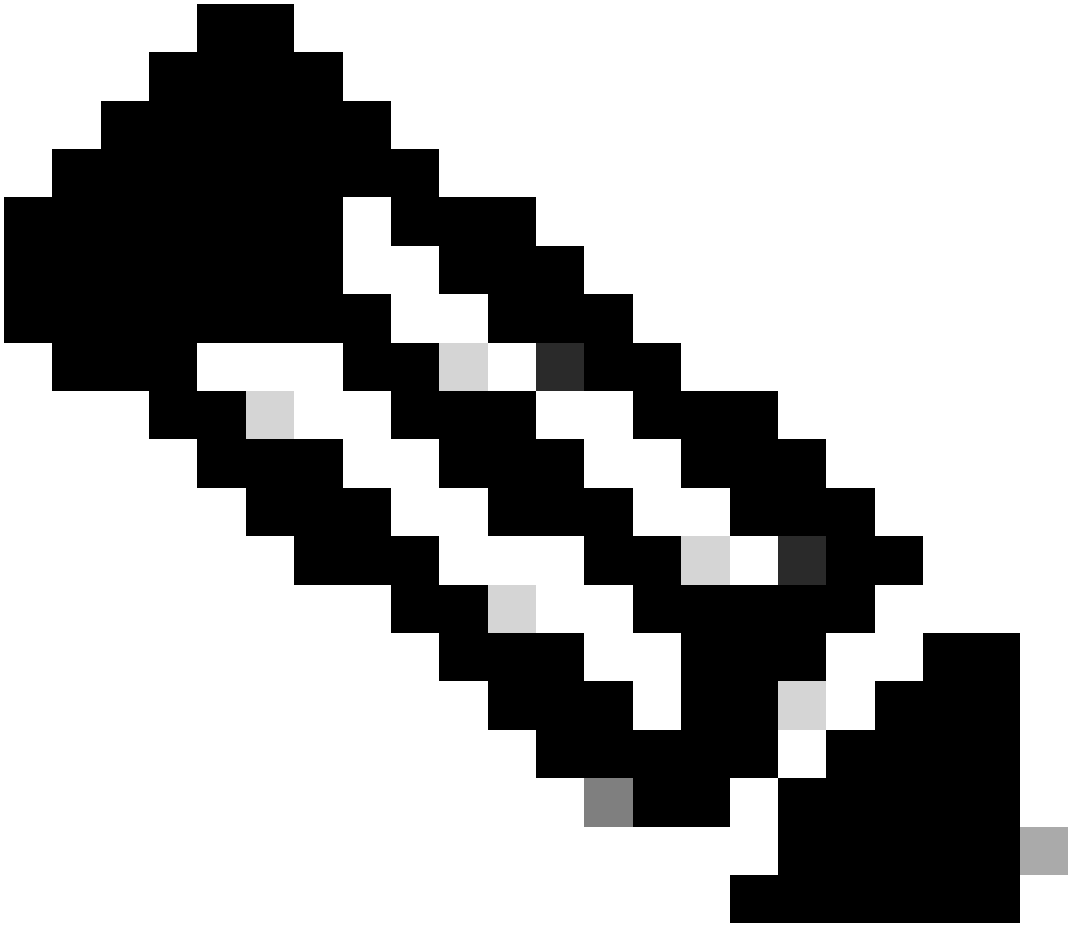
この項の例では、RTB は IGP 経由で 172.16.10.0 に関する情報が伝達されるまで待機します。その後、RTB は RTD へのアップデートの送信を開始します。172.16.10.0 を指すスタティック ルートを RTB に追加すると、RTB に IGP による情報の伝達が完了したと認識させることができます。この場合は、他のルータが 172.16.10.0 に到達できることを確認してください。

同期の無効化

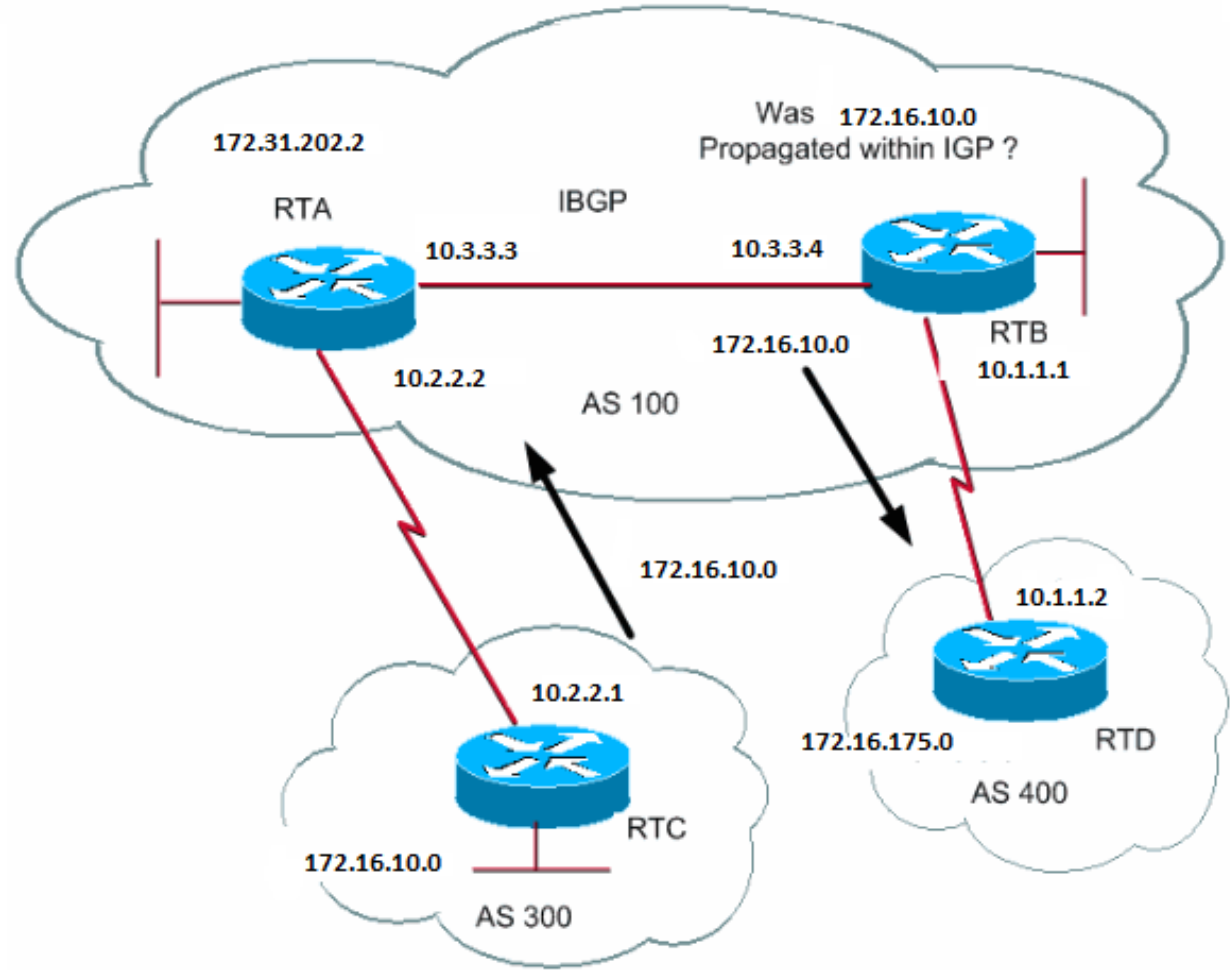
場合によっては、同期が必要ないことがあります。別の AS からのトラフィックが自 AS を通過しない場合は、同期を無効にすることができます。また、AS 内のすべてのルータで BGP を実行している場合も同期を無効にできます。この機能を無効にすると、IGP で伝達されるルートが減り、BGP の収束時間が短縮されます。

同期の無効化は自動的には行われません。AS 内のすべてのルータが BGP を実行している場合、IGP が実行されていなくてもルータはこれを認識できません。ルータは特定のルートに関する IGP アップデートを無限に待ち続け、そのルートが外部ピアに送信されることはありません。この場合は、ルーティングが正常に動作するように、手動で同期を無効にする必要があります。

```
router bgp 100
 no synchronization
```

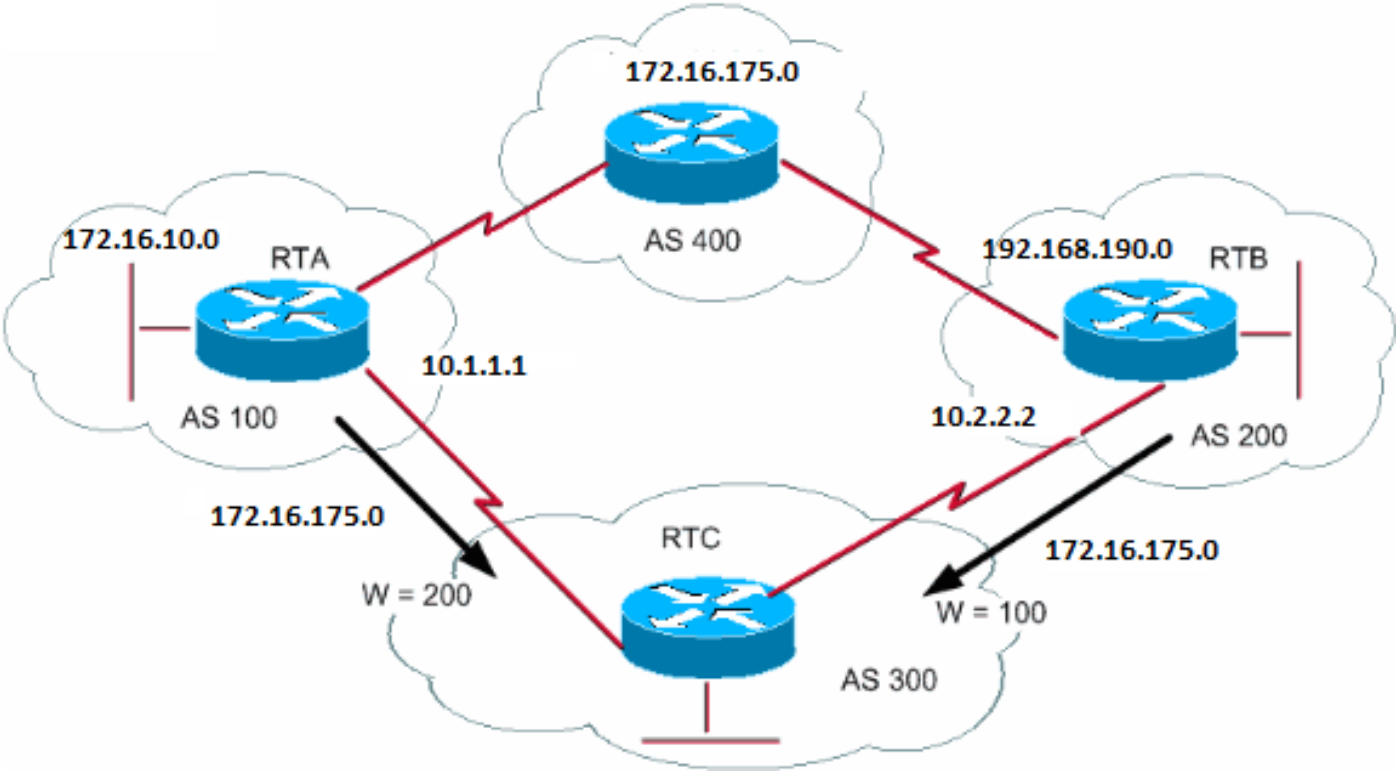


注：必ずclear ip bgp addressコマンドを発行してセッションをリセットしてください。




```
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
```

重み属性



重み属性はシスコ定義の属性です。この属性は重みを使用してベスト パスを選択します。重みはルータにローカルに割り当てられます。この値は特定のルータに対してのみ意味を持ち、値が伝達されたり、ルート アップデートで伝送されたりすることはありません。重みは 0 ~ 65,535 の範囲の数値です。ルータが送信元となるパスにはデフォルトで 32,768 の重みが割り当てられ、他のパスには 0 の重みが割り当てられます。

同じ宛先へのルートが複数存在する場合は、重み値の高いルートが優先されます。この項の例を見てみましょう。RTAはAS4からネットワーク172.16.0.0について学習しました。RTA は RTC にアップデートを伝達します。RTBはAS4からネットワーク172.16.0.0についても学習しています。RTB は RTC にアップデートを伝達します。RTCには172.16.0.0に到達する方法が2つあり、どちらに到達するかを決定する必要があります。RTC で RTA からのアップデートの重みが RTB からのアップデートの重みよりも大きくなるように設定すれば、RTC は 172.16.0.0 に到達するためのネクスト ホップとして RTA を使用することになります。このような重みを設定するには、いくつかの方法があります。

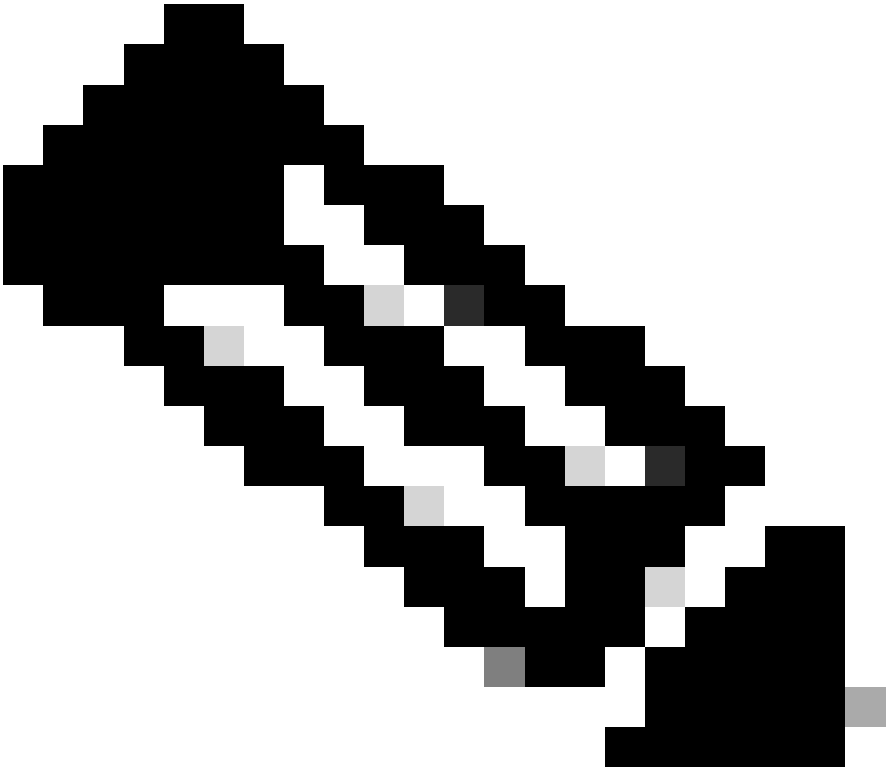
- neighbor コマンドを使用する。

  neighbor {ip-address|peer-group} weight <weight>コマンド

- AS_PATH アクセス リストを使用する。

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

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



注：シナリオによっては、一部のソフトウェアバージョンでは使用できないコマンドがほとんどないことがあります。

- ルート マップを使用する。

```
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.
```

より大きい重み値を持つ RTA がネクスト ホップとして優先されます。

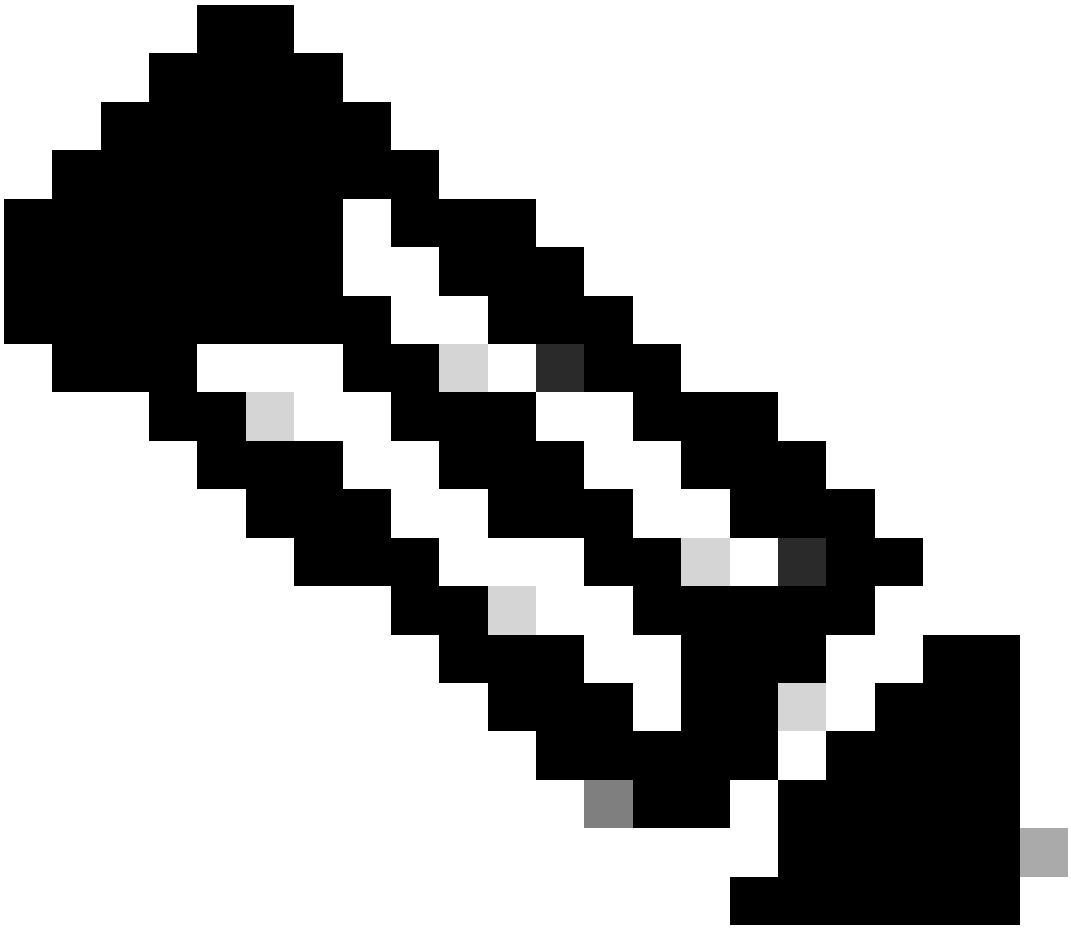
IP AS_PATH とフィルタ リストを使用した場合も同じ結果が得られます。

```
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$
...
```

また、ルート マップを使用しても同じ結果を得られます。

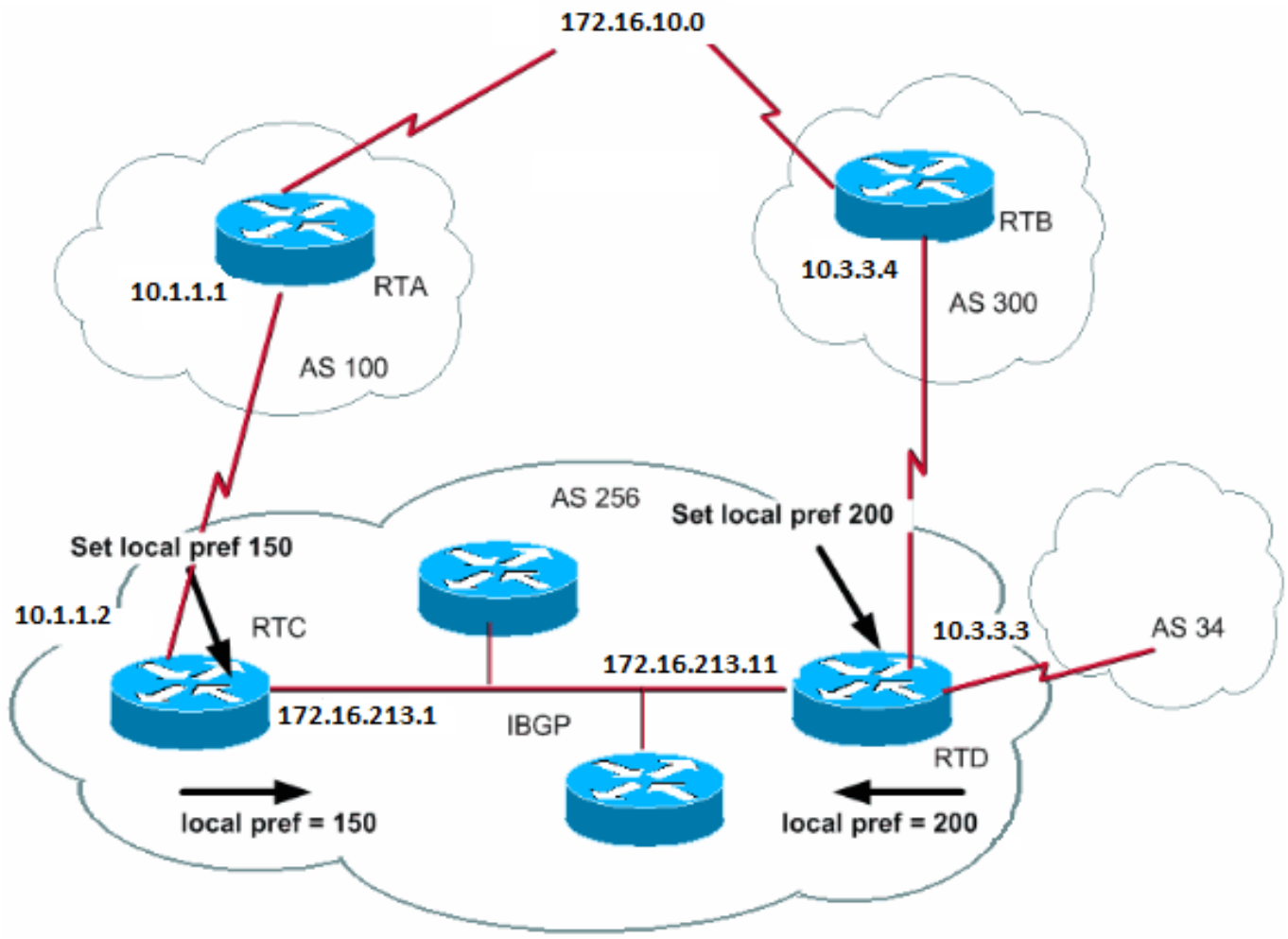
```
RTC#
router bgp 300
 neighbor 10.1.1.1 remote-as 100
 neighbor 10.1.1.1 route-map setweightin in
 neighbor 10.2.2.2 remote-as 200
 neighbor 10.2.2.2 route-map setweightin in
...
```

```
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:IGPパスをバックアップとして使用するMPLS VPN BGPパスを優先するように重みを変更できます。

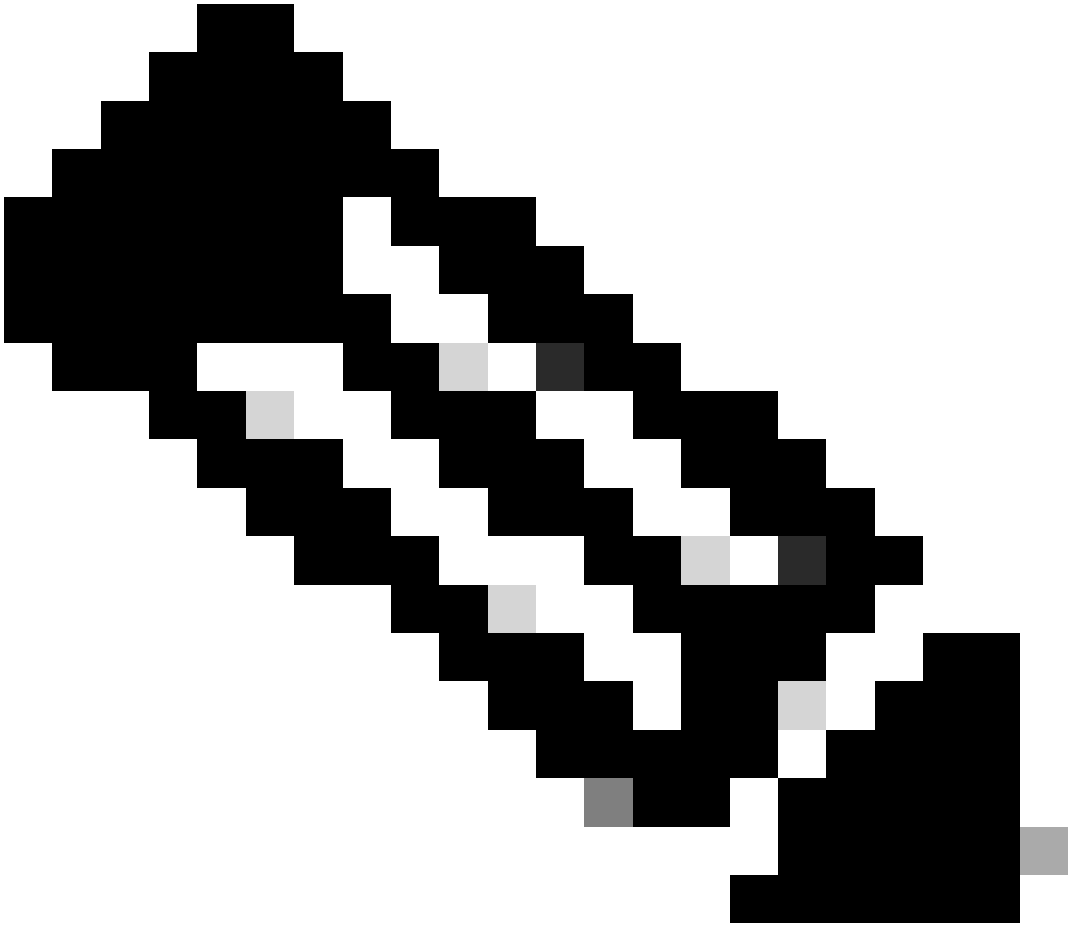
ローカル プリファレンス属性



ローカル プリファレンスは AS に対する指標で、その AS から特定のネットワークに到達する際にどのパスが優先されるかを示します。ローカル プリファレンス値の高いパスが優先されます。ローカル プリファレンスのデフォルト値は 100 です。

ローカル ルータにのみ関連する重み属性とは異なり、ローカル プリファレンスは、同じ AS 内のルータ間で交換される属性です。

ローカル プリファレンスを設定するには、bgp default local-preference value コマンドを発行します。この項の例で示すように、ルート マップを使用してローカル プリファレンスを設定することもできます。



注：変更を考慮するには、ソフトリセットを実行する（つまり、ルータでbgpプロセスをクリアする）必要があります。BGPプロセスをクリアするには、clear ip bgp [soft][in/out]コマンドを使用します。softはソフトリセットを示し、セッションを切断しません。また、[in/out]は着信または発信設定を指定します。in/out を指定しないと、インバウンドとアウトバウンドの両方のセッションがリセットされます。

---

bgp default local-preference コマンドは、ルータから同じ AS 内のピアに送信されるアップデートのローカル プリファレンスを設定します。このセクションの図では、AS256は組織の2つの異なるサイドから172.16.10.0に関するアップデートを受信します。ローカル プリファレンスによって、AS256 からそのネットワークに到達するためのルートを決定できます。優先される出力点が RTD であると仮定します。次の設定では、AS300 から到達するアップデートのローカル プリファレンスが 200、AS100 から到達するアップデートのローカル プリファレンスが 150 に設定されます。

```
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
```

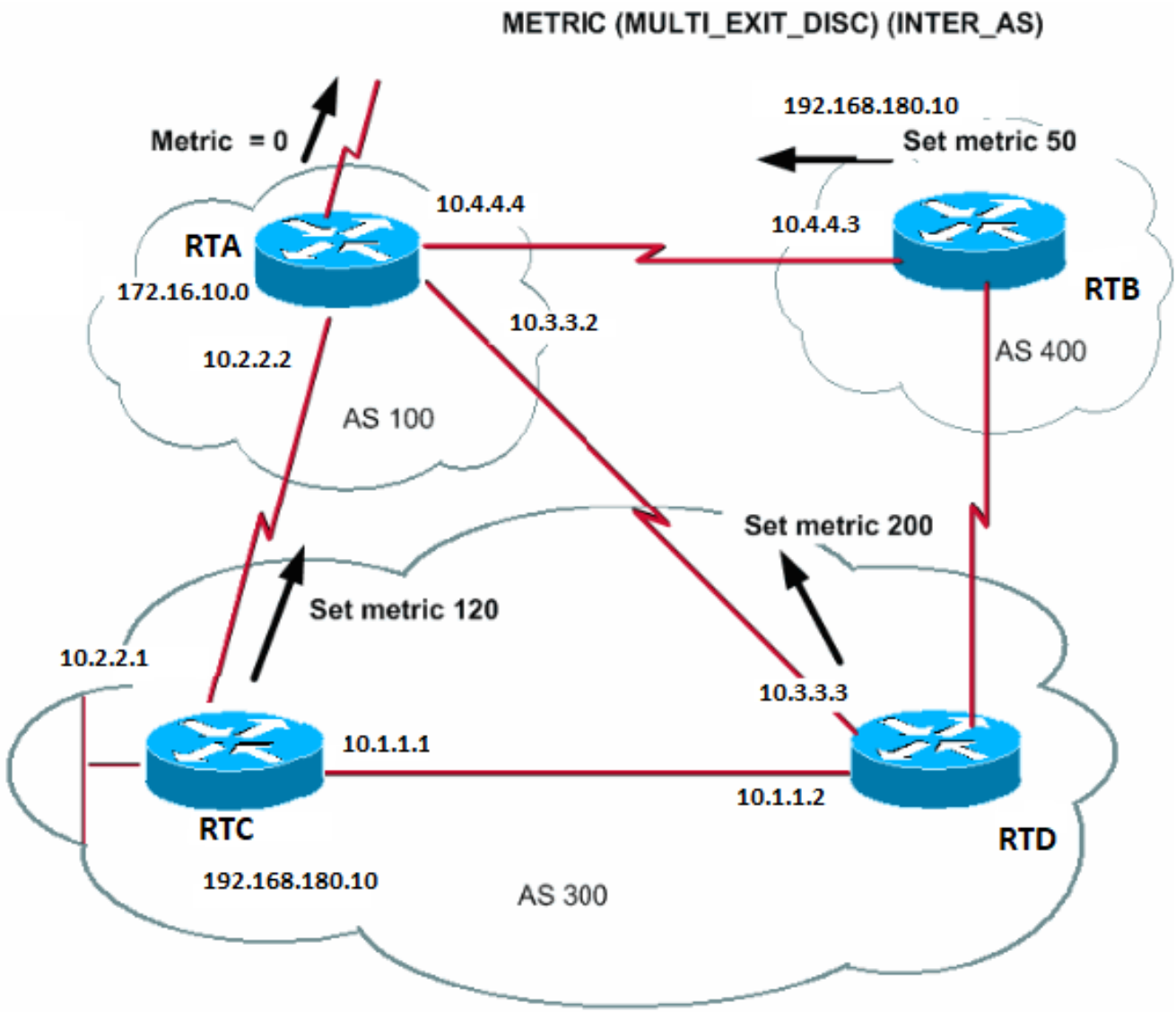
この設定では、RTC はすべてのアップデートのローカル プリファレンスを 150 に設定します。同様に、RTD はすべてのアップデートのローカル プリファレンスを 200 に設定します。AS256 内ではローカル プリファレンスの交換が行われます。したがって、RTCとRTDの両方は、ネットワーク172.16.10.0がAS100ではなくAS300からアップデートを受信する場合に、ローカルプリファレンスがより高いことを認識しています。そのネットワークを宛先とする AS256 内のトラフィックはすべて、RTD を出力点として送信されます。

ルート マップを使用すると、より柔軟な設定を行えます。この項の例では、RTD が受信するすべてのアップデートには、RTD への到達時にローカル プリファレンス 200 がタグ付けされます。AS34 から到達するアップデートにもローカル プリファレンス 200 がタグ付けされますが、このタグは不要である場合があります。その場合は、ルート マップを使用して、特定のローカル プリファレンスをタグ付けする必要がある特定のアップデートを指定できます。ランダム データの例は次のとおりです。

```
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
```

この設定により、AS300 から到達するすべてのアップデートにはローカル プリファレンス 200 がタグ付けされ、その他のアップデート (AS34 から到達するアップデートなど) には 150 の値がタグ付けされます。

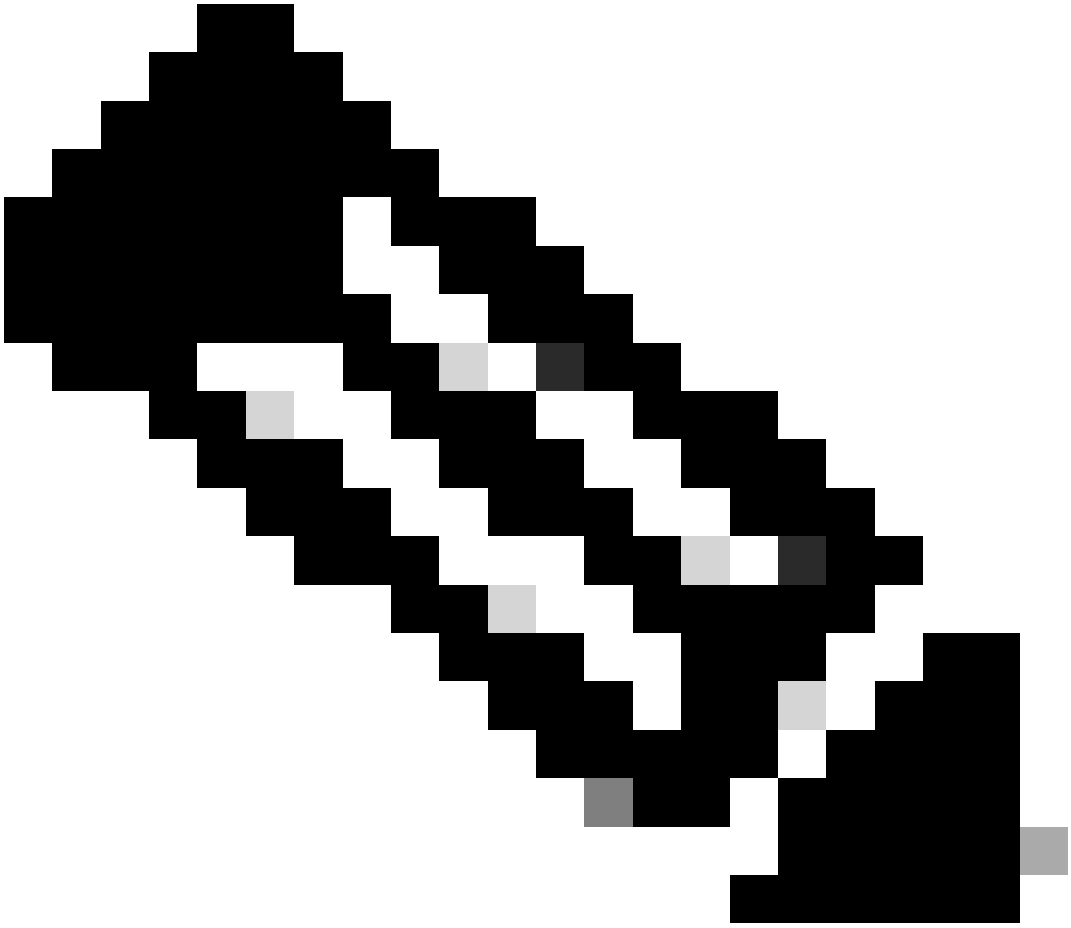
メトリック属性



メトリック属性は、MULTI_EXIT_DISCRIMINATOR、MED（BGP4）、または INTER_AS（BGP3）とも呼ばれます。この属性は、外部ネイバーにとって AS への優先パスに関するヒントになります。別の AS へのエントリ ポイントが複数ある場合、この属性を使用して、特定のルートに到達する方法に関してその AS に動的に影響を与えることができます。より小さいメトリック値が優先されます。

ローカル プリファレンスとは異なり、メトリックは AS 間で交換されます。ただし、AS に伝達されたメトリックが、さらに別の AS に伝達されることはありません。特定のメトリックが設定されたアップデートが AS に到達すると、AS 内ではそのメトリックを使用してルートが決定されます。同じアップデートが第 3 の AS に渡される場合は、メトリックが 0 に戻ります。上記の図はメトリックの設定を示しています。メトリックのデフォルト値は 0 です。

ルータは他の指示を受け取らない限り、同じ AS 内のネイバーから伝達されたパスのメトリックを比較します。ルータが別の AS のネイバーから伝達されたメトリックを比較できるようにするには、ルータで bgp always-compare-med という特別な設定コマンドを発行する必要があります。



注:Multi-Exit Discriminator(MED)ベースのパス選択に影響を与える可能性があるBGP設定コマンドは2つあります。bgp deterministic-med コマンドと bgp always-compare-med コマンドです。bgp deterministic-med コマンドを発行すると、同じ AS 内の別のピアがアドバタイズしたルートの選択時に MED 変数が比較されるようになります。bgp always-compare-med コマンドを発行すると、別の AS のネイバーから伝達されたパスの MED が比較されるようになります。bgp always-compare-med コマンドは、複数のサービス プロバイダーまたは企業が MED の設定方法に関して統一されたポリシーに合意している場合に有用です。これらのコマンドが BGP パス選択に与える影響については、『bgp deterministic-med コマンドと bgp always-compare-med コマンドの相違点』を参照してください。

このセクションの図では、AS100は3つの異なるルータ（RTC、RTD、およびRTB）を介してネットワーク192.168.180.10に関する情報を取得します。RTC と RTD は AS300 に属し、RTB は AS400 に属しています。

次の例では、bgp bestpath as-path ignore コマンドによって RTA での AS-Path 比較が無視されます。また、BGP がルート比較用の次の属性（この例ではメトリック、つまり MED）を処理するように設定されています。このコマンドを省略すると、BGPは最短の

AS-Pathを持つルータRTCからのルート192.168.180.10をインストールできます。

RTC からのメトリックを 120、RTD からのメトリックを 200、RTB からのメトリックを 50 に設定していると仮定します。デフォルトでは、ルータは同じ AS 内のネイバーから到達するメトリックを比較します。したがって、RTA は RTC から到達するメトリックと RTD から到達するメトリックのみを比較できます。120 は 200 より小さいので、RTA は最適なネクスト ホップとして RTC を選択します。RTC と RTB は別々の AS に属しているため、RTA は RTB からメトリック 50 のアップデートを受け取っても、そのメトリックを 120 と比較することはできません。RTA は他のいくつかの属性に基づいて選択を行う必要があります。

RTA にメトリックを比較させるには、RTA で bgp always-compare-med コマンドを発行する必要があります。次の設定でこのプロセスを示します。

```
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
```

これらの設定により、RTA は他のすべての属性が同じであるという事実を考慮して、ネクスト ホップとして RTC を選択します。メトリック比較に RTB を含めるには、RTA を次のように設定する必要があります。

```
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
```

この場合、RTA はネットワーク 192.168.180.10 に到達するための最適なネクスト ホップとして RTB を選択します。

default-metricnumberコマンドを発行する場合は、BGPにルートを再配布する際にもメトリックを設定できます。

この項の例で、RTB が AS100 にスタティック経由でネットワークをインジェクトするとします。設定は以下のとおりです。

```
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.
```

コミュニティ属性

コミュニティ属性は推移的なオプション属性で、値の範囲は 0 ~ 4,294,967,200 です。コミュニティアトリビュートを使用すると、宛先を特定のコミュニティにグループ化し、そのコミュニティに一致するルーティング決定を適用できます。ルーティング決定には、承認、優先、再配布などがあります。

コミュニティ属性を設定するには、ルート マップを使用します。ルート マップの set コマンドの構文は次のとおりです。

<#root>

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

このコマンドで使用する事前定義された既知のコミュニティには、次のものがあります。

- **no-export：eBGP ピアにアドバタイズしません**。このルートは AS 内に保持されます。

- **no-advertise：内部および外部のどのピアにもこのルートをアドバタイズしません**。

- **internet：このルートをインターネット コミュニティにアドバタイズします**。すべてのルータがこのコミュニティに属します。

- **local-as：コンフェデレーション シナリオで、ローカル AS の外部にパケットが送信されることを防ぐために使用します**。

コミュニティを設定するルート マップの例を 2 つ紹介します。

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

または

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

additive キーワードを設定しない場合、既存の古いコミュニティはすべて 200 に置き換えられます。additive キーワードを使用すると、コミュニティに 200 が追加されます。コミュニティ属性を設定しても、デフォルトではネイバーにこの属性は送信されません。ネイバーに属性を送信するには、次のコマンドを使用する必要があります。

```
<#root>
```

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

ランダム データの例は次のとおりです。

```
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
```

Cisco IOS ソフトウェア リリース 12.0 以降では、コミュニティを 10 進数、16 進数、AA:NN の 3 種類の形式で設定できます。デフォルトでは、Cisco IOS ソフトウェアは従来の 10 進形式を使用します。AA:NNの形式で設定と表示を行うには、**ip bgp-community new**-global configuration formatコマンドを発行します。AA:NN の前半部分は AS 番号を表し、後半部分は 2 バイトの番号を表します。

ランダム データの例は次のとおりです。

グローバル設定で**ip bgp-community new-format** コマンドを使用しない場合は、**show ip bgp 10.6.0.0** コマンドを発行すると、10進形式でコミュニティ属性値が表示されます。この例では、コミュニティ属性値は6553620と表示されています。

<#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

次に、このルータで ip bgp-community new-format コマンドをグローバルに発行します。

<#root>

Router#

configure terminal

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

ip bgp-community new-format

Router(config)#

exit

ip bgp-community new-formatglobal設定コマンドを使用すると、コミュニティ値はAA:NN形式で表示されます。この例では、**show ip bgp 10.6.0.0**コマンドの出力で、この値が100:20と表示されています。

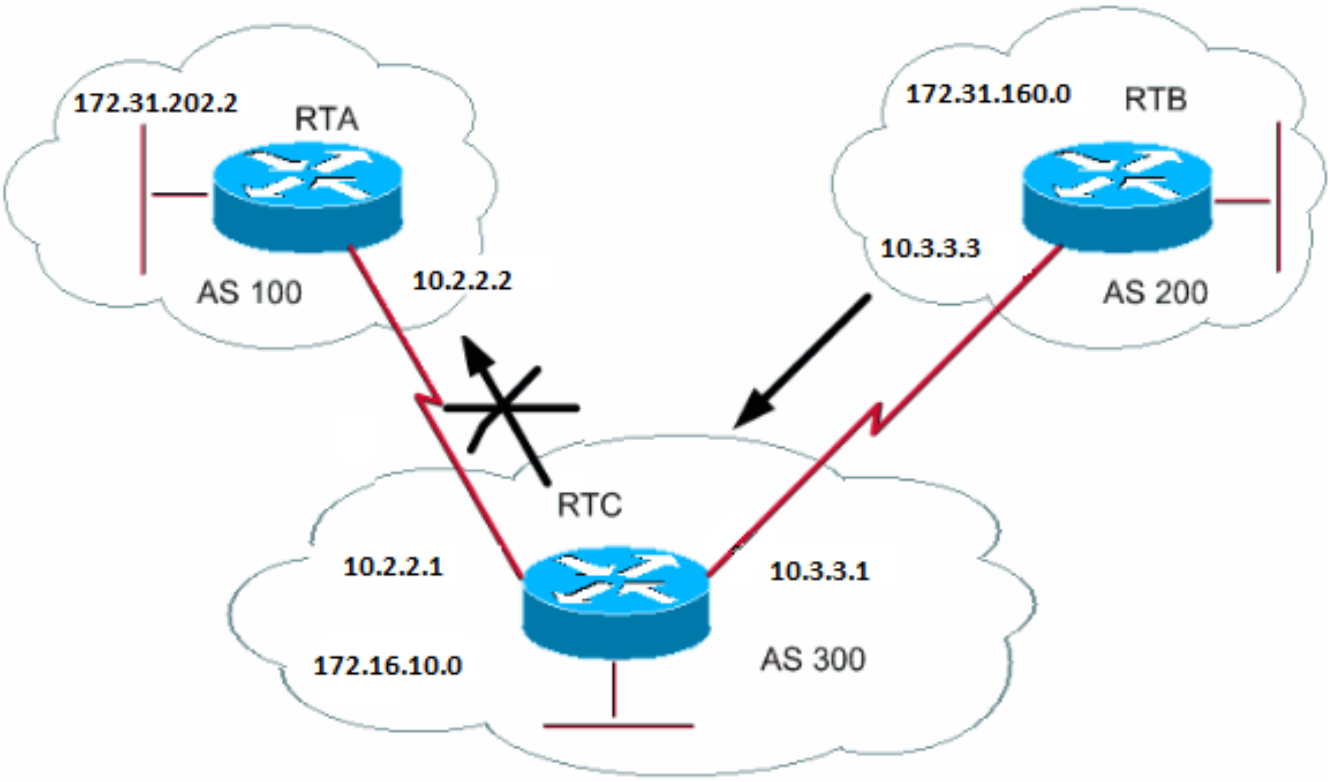
```
    <#root>

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 ケース スタディ 3

BGPフィルタ

さまざまなフィルタ方法を使用することで、BGP アップデートの送受信を制御できます。ルート情報、パス情報、またはコミュニティに基づいて BGP アップデートをフィルタリングでき、どの方法でも同じ結果を得ることができます。使用する方法は、個別のネットワーク設定に応じて決定します。

ルートフィルタ



ルータが学習またはアドバタイズするルーティング情報を制限するには、特定のネイバーとの間で送受信されるルーティング アップデートを使用して BGP をフィルタリングします。アクセス リストを定義して、ネイバーとの間で送受信するアップデートに適用します。ルータ設定モードで次のコマンドを発行します。

<#root>

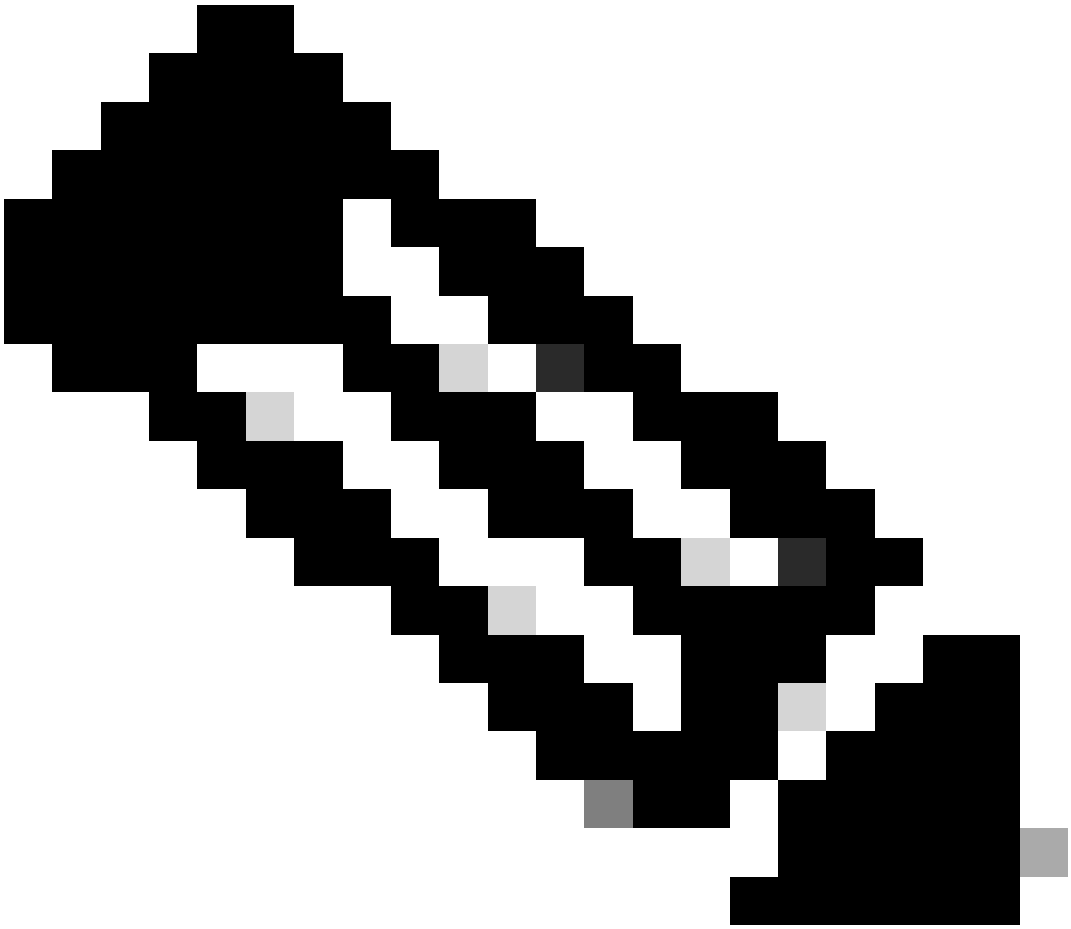
neighbor {ip-address | peer-group-name} distribute-list access-list-number {in | out}

この例では、RTB がネットワーク 172.31.160.0 を生成して、RTC にアップデートを送信します。RTC が AS100 にアップデートを伝達しないようにするには、該当するアップデートをフィルタリングするアクセス リストを定義して、RTA との通信時にアクセス リストを適用する必要があります。

```
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.
```

競合が発生する可能性のあるスーパーネットを扱う場合、アクセス リストの使用は若干複雑になります。

上記の例の RTB に 160.10.x.x という複数のサブネットがあると仮定します。ここでは、アップデートをフィルタリングし、192.168.160.0/8 のみがアドバタイズされるようにします。



注:/8表記は、IPアドレスの左端から始まる8ビットのサブネットマスクを使用することを意味します。このアドレスは192.168.160.0 255.0.0.0 に相当します。

access-list 1 permit 192.168.160.0 0.255.255.25 5

コマンドは、192.168.160.0/8、192.168.160.0/9などを許可します。アップデートを 192.168.160.0/8 のみに制限するには、次の形式の拡張アクセス リストを使用する必要があります。

<#root>

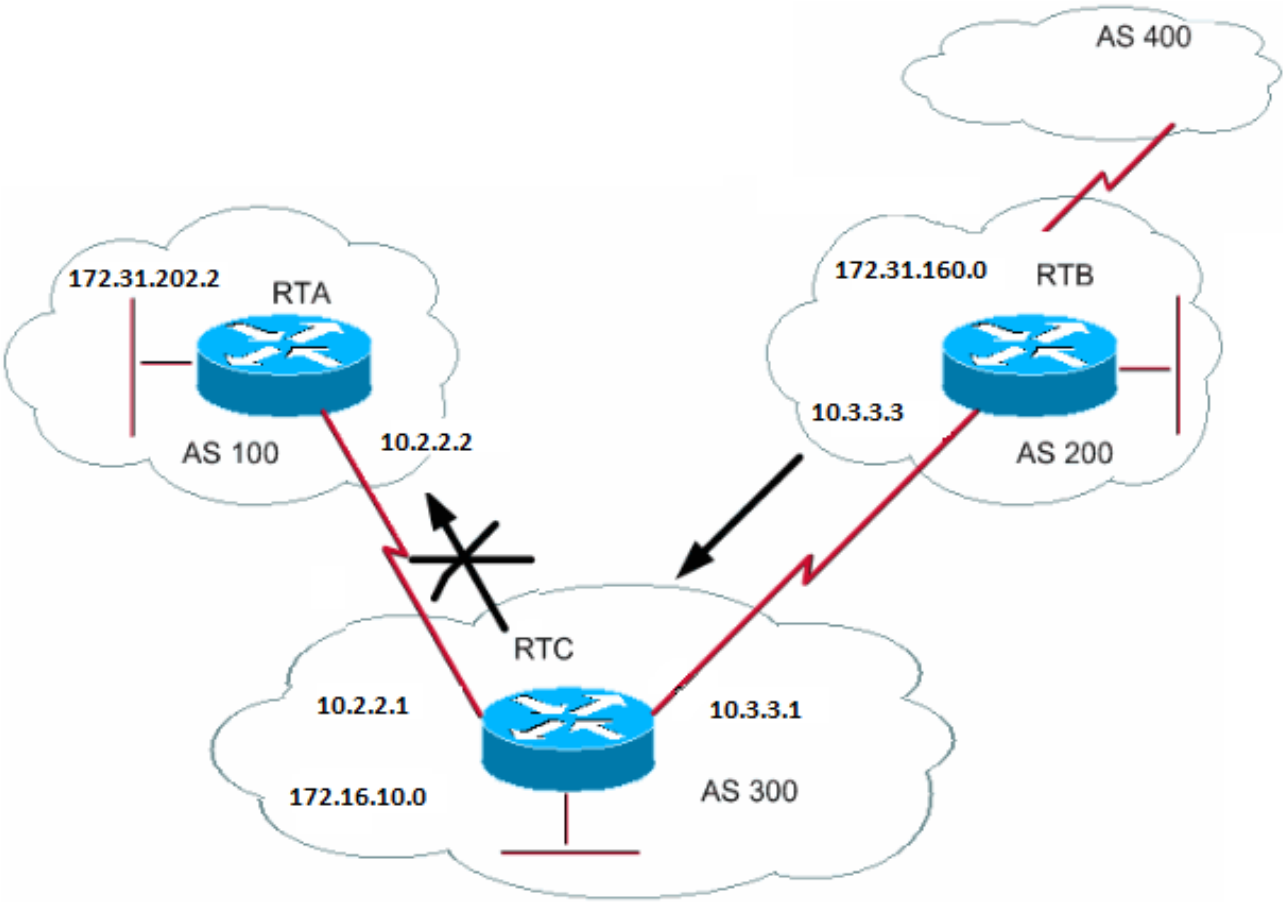
access-list 101 permit ip 192.168.160.0 0.255.255.255 255.0.0.0 0.0.0.0.

このリストは 192.168.160.0/8 のみを許可します。

BGPピアからのネットワークをフィルタリングする設定例については、『BGPピアからの1つ以上のネットワークのブロック』を参照してください。この方式では、プレフィックスリストをフィルタリングする機能だけでなく、標準および拡張アクセスコントロールリスト(ACL)で**distribute-list**コマンドを使用します。

パスフィルタ

また、パスをフィルタすることもできます。



BGP AS パス情報を使用して、着信アップデートと発信アップデートの両方にアクセス リストを指定できます。このセクションの図では、172.31.160.0に関するアップデートをブロックして、AS100に到達しないようにできます。アップデートをブロックするには、AS200 から発信されたアップデートの AS100 への送信を禁止するアクセス リストを RTC で定義します。次のコマンドを発行します。

<#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}

次の例では、RTC から RTA への 172.31.160.0 に関するアップデートの送信が停止されます。

```
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 .*
```

この例のaccess-list 1コマンドは、200で始まり200で終わるパス情報を含むすべてのアップデートを拒否するように指定しています。このコマンドの ^200$ は「正規表現」です。^ は「で始まる」を意味し、$ は「で終わる」を意味します。RTBは172.31.160.0に関するアップデートを、200で始まり200で終わるパス情報とともに送信するため、アップデートはアクセスリストに一致します。そのため、アクセス リストによってこれらのアップデートは拒否されます。

.*も正規表現です。.は「任意の文字」、*は「その文字の繰り返し」を意味します。つまり、.* はあらゆるパス情報を表します。これは、他のすべてのアップデートの送信を許可するために必要です。

^200$の代わりに^200を使用するとどうなるでしょうか。上記の図のように AS400 が存在する場合、AS400 が発信するアップデートのパス情報は (200、400) という形式になります。このパス情報は最初が 200 で最後が 400 です。これらのアップデートは、パス情報が200から始まるため、アクセスリスト^200に一致します。アクセス リストにより、RTA へのこれらのアップデートの送信が禁止されます。これは要件ではありません。

正しい正規表現が実装されているかどうかを確認するには、**show ip bgp regexpregular-expression** コマンドを発行します。このコマンドは、正規表現の設定に一致するすべてのパスを表示します。

AS 正規表現

この項では正規表現の作成について説明します。

正規表現は、入力ストリングとのマッチングを行うためのパターンです。正規表現の作成では、入力が一致する必要がある文字列を指定します。BGP の場合は、入力が一致する必要があるパス情報で構成された文字列を指定します。

「**パスフィルタ**」セクションの例では、文字列^200$を指定しています。アップデートに含まれるパス情報を文字列と一致させて決定する必要があります。

正規表現は次の要素で構成されます。

- **範囲**

  範囲は、左角カッコと右角カッコで囲まれた文字列です（例：[abcd]）。

- **アトム**

  アトムは単一の文字です。次に例を示します。

  .

  - .は、任意の1文字に一致します。

  ^

  - ^は入力文字列の先頭に一致します。

  $

- $ は入力文字列の末尾に一致します。

\

- \は文字に一致します。

-

- _は、カンマ(,)、左波カッコ({)、右波カッコ(})、入力文字列の先頭、入力文字列の末尾、またはスペースに一致します。

•

ピース

ピースは、次の記号の1つで、アトムの後に続きます。

*

- * は 0 個以上のアトムのシーケンスに一致します。

+

- + は 1 個以上のアトムのシーケンスに一致します。

?

- ? には、アトムまたはヌルの文字列が一致します。

- **ブランチ**

  ブランチは 0 個以上の結合されたピースです。

正規表現の例をいくつか示します。

a*

- この表現は、文字「a」の任意の繰り返しを示します（0 回も含む）。

a+

- この表現は、文字「a」の 1 回以上の繰り返しが存在する必要があることを示します

ab?a

- この表現は、「aa」または「aba」に一致します。

_100_

- この表現は、AS100 経由であることを意味します。

_100$

- この表現は、AS100 が送信元であることを示します。

^100 .*

- この表現は、AS100 からの送信を示します。

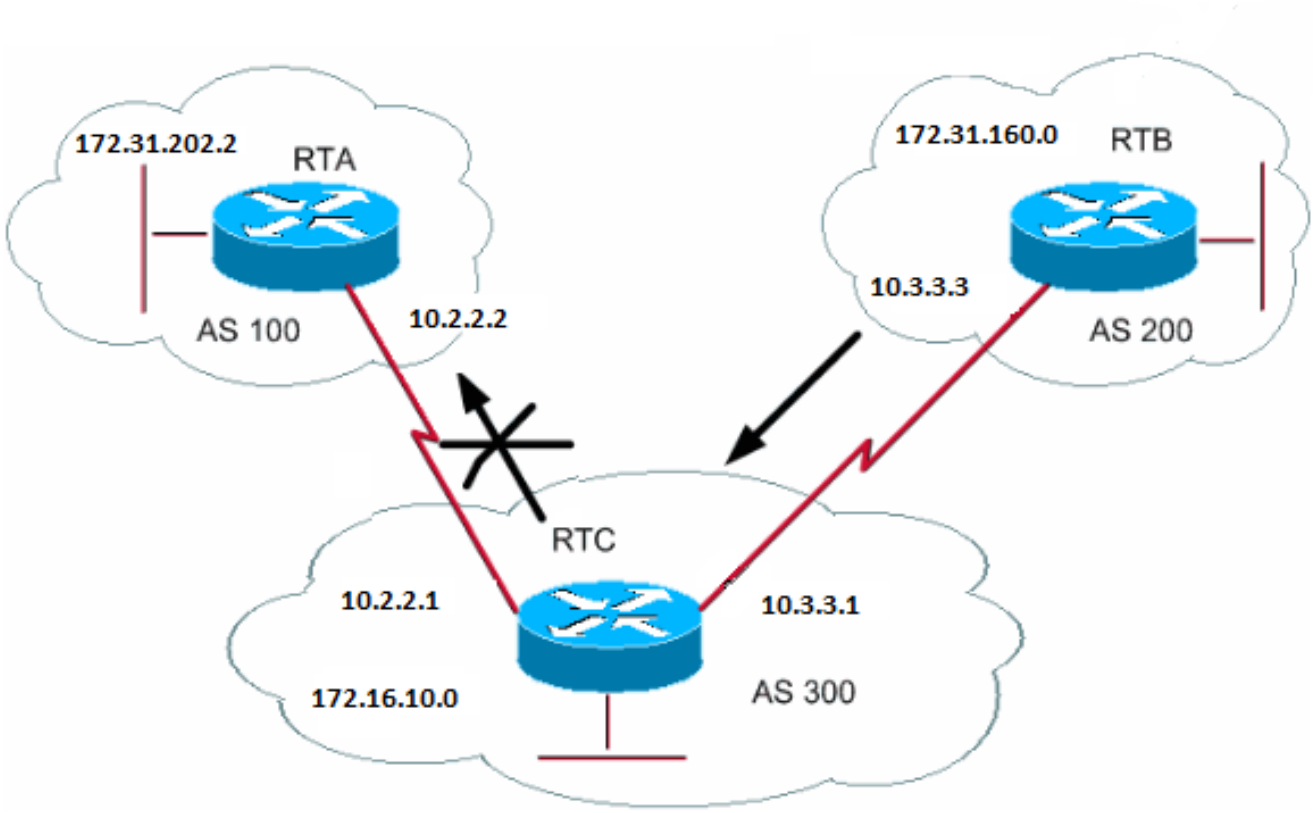
^$

- この表現は、この AS からの発信を示します。

正規表現フィルタリングの設定例については、『BGPでの正規表現の使用』を参照してください。

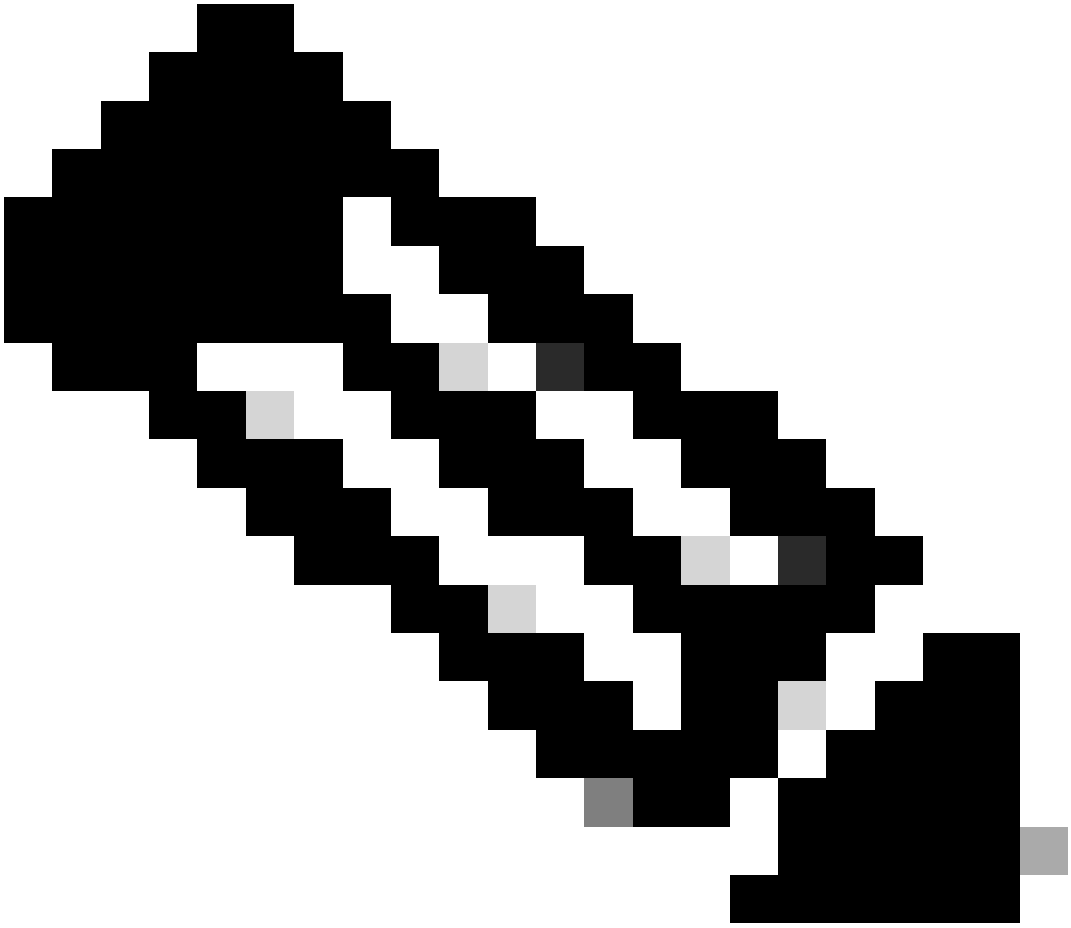
BGPコミュニティフィルタ

ここまでは、ルート フィルタリングと AS パス フィルタリングについて説明してきました。もう 1 つの方法はコミュニティ フィルタリングです。「コミュニティアトリビュート」セクションでコミュニティについて説明しているので、このセクションではコミュニティの使用例をいくつか示します。

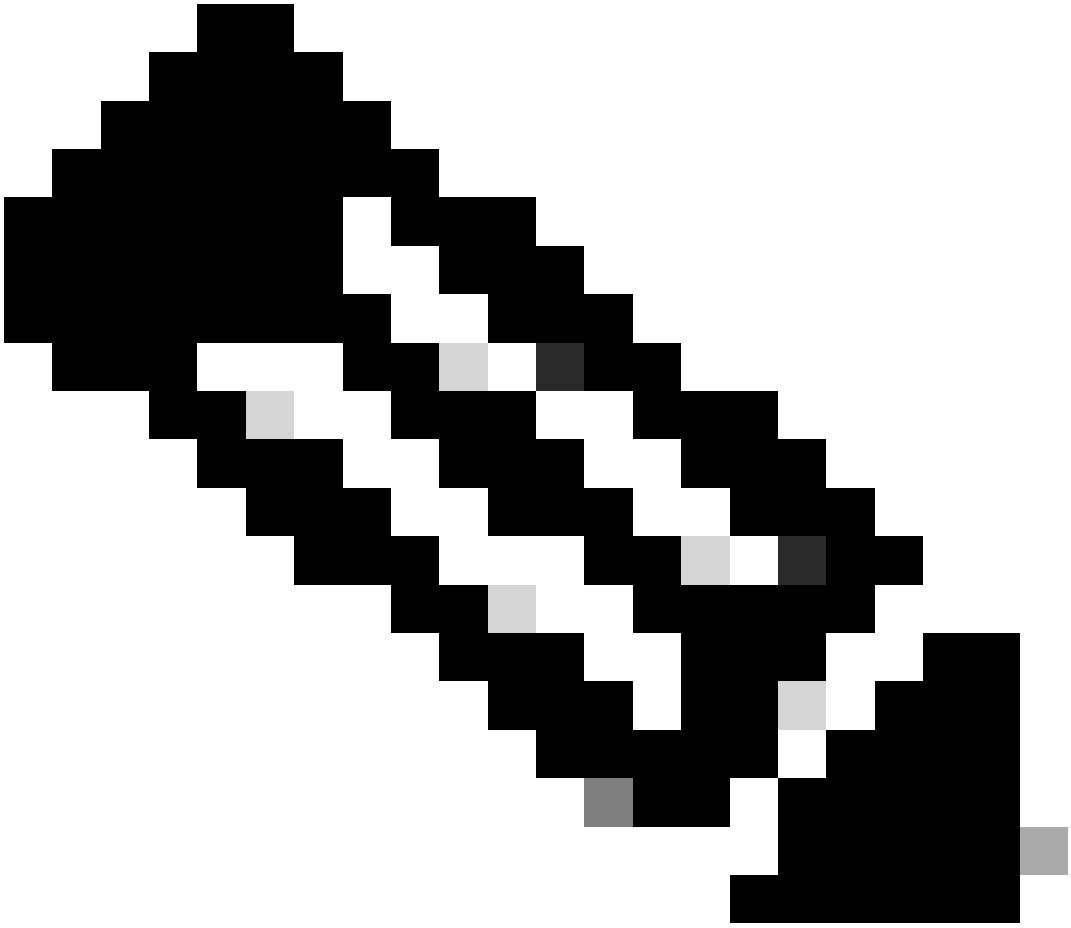


次の例では、RTB がアドバタイズした BGP ルートを RTC が外部ピアに伝達しないように、RTB によってルートにコミュニティ属性が設定されるようにします。コミュニティ属性no-exportを使用します。

```
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：この例では、コミュニティをno-exportに設定するために、route-map setcommunityコマンドを使用します。



注：このアトリビュートをRTCに送信するには、**neighbor send-community** コマンドが必要です。

RTC は属性が NO_EXPORT のアップデートを受け取った場合、外部ピアの RTA にはそのアップデートを伝達しません。

この例では、RTBはコミュニティアトリビュートをに設定しています**100 200 additive** 。この操作により、RTCに送信される前に、現在のコミュニティ値に値100 200が追加されます。

```
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
```

コミュニティ リストは、ルート マップの match 句で使用するコミュニティのグループです。コミュニティ リストを使用すると、コミュニティ番号のさまざまなリストに基づいて属性をフィルタリングまたは設定できます。

<#root>

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

たとえば、次のような match-on-community ルート マップを定義できます。

```
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.
```

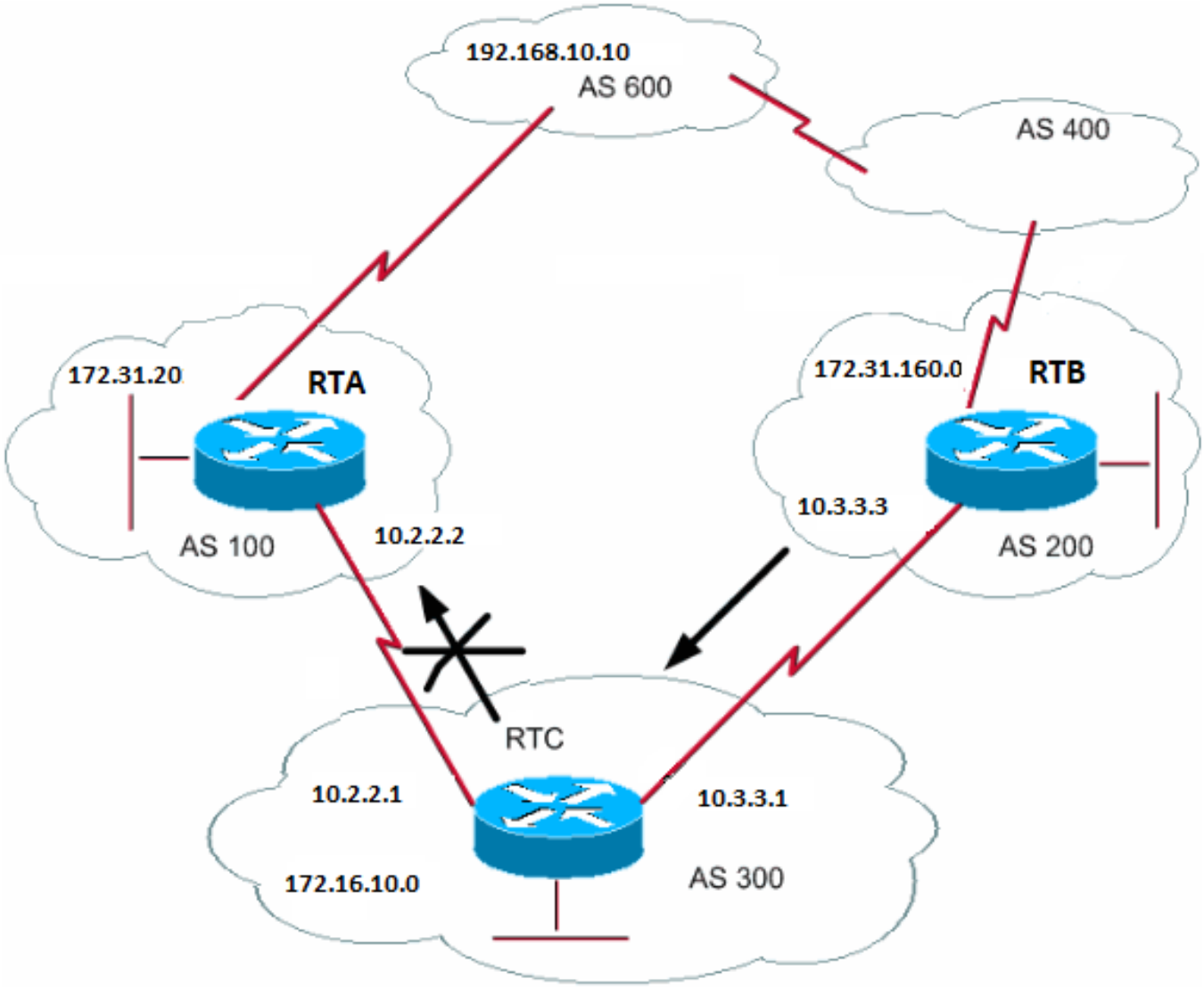
コミュニティ リストを使用して、特定のアップデートに含まれる重みやメトリックなどの特定のパラメータを、コミュニティ値に基づいてフィルタリングまたは設定できます。上記の 2 番目の例で、RTB はコミュニティ 100 200 を設定したアップデートを RTC に送信しました。RTC でこれらの値に基づいて重みが設定されるようにする場合は、以下の設定を行います。

```
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
```

この例では、コミュニティ属性に 100 を含むルートはリスト 1 に一致します。このルートの重みは 20 に設定されます。コミュニティとして 200 のみを持つルートはリスト 2 に一致し、重みは 20 になります。exact キーワードは、コミュニティが 200 のみで構成され、他の値が含まれていないことを示しています。最後のコミュニティ リストは、他のアップデートがドロップされないようにするために設定されています。デフォルトでは、一致しないアップデートはすべてドロップされることに注意してください。すべてのルートがインターネット コミュニティに属するため、internet キーワードはすべてのルートを示します。

詳細は、『BGPコミュニティ値によるアップストリームプロバイダーネットワークの設定と制御』を参照してください。

BGP ネイバーとルート マップ



neighbor コマンドをルート マップと併用して、着信および発信アップデートのパラメータをフィルタリングまたは設定できます。

IP アドレスに基づいて照合される場合、neighbor ステートメントに関連付けられたルート マップは、着信アップデートには適用されません。

<#root>

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

上記の図の RTC に、AS200 に対してローカルなネットワークに関する情報のみを AS200 から学習させるとします。さらに、承認されたルートの重みが 20 に設定されるようにします。この場合は、neighbor と as-path アクセス リストを組み合わせて使用します。

```
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$
```

AS200 から発信されるアップデートには、200 で始まって 200 で終わるパス情報が含まれています。これらのアップデートは許可され、その他のアップデートはすべてドロップされます。

次のように仮定します。

- AS200 から発信されたアップデートを承認して、重みを 20 に設定する。
- AS400 から発信されたアップデートをドロップする。
- その他のアップデートは重みを 10 に設定する。

```
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 .*
```

このステートメントにより、AS200 に対してローカルなアップデートの重みが 20 に設定されます。また、この文は、AS400の背後から到達するアップデートのウェイトを10に設定し、AS400から到達するアップデートをドロップします。

set as-path prepend コマンドの使用

場合によっては、BGP 決定プロセスを操作するためにパス情報の操作が必要になります。この場合は、ルート マップとともに次のコマンドを使用します。

<#root>

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

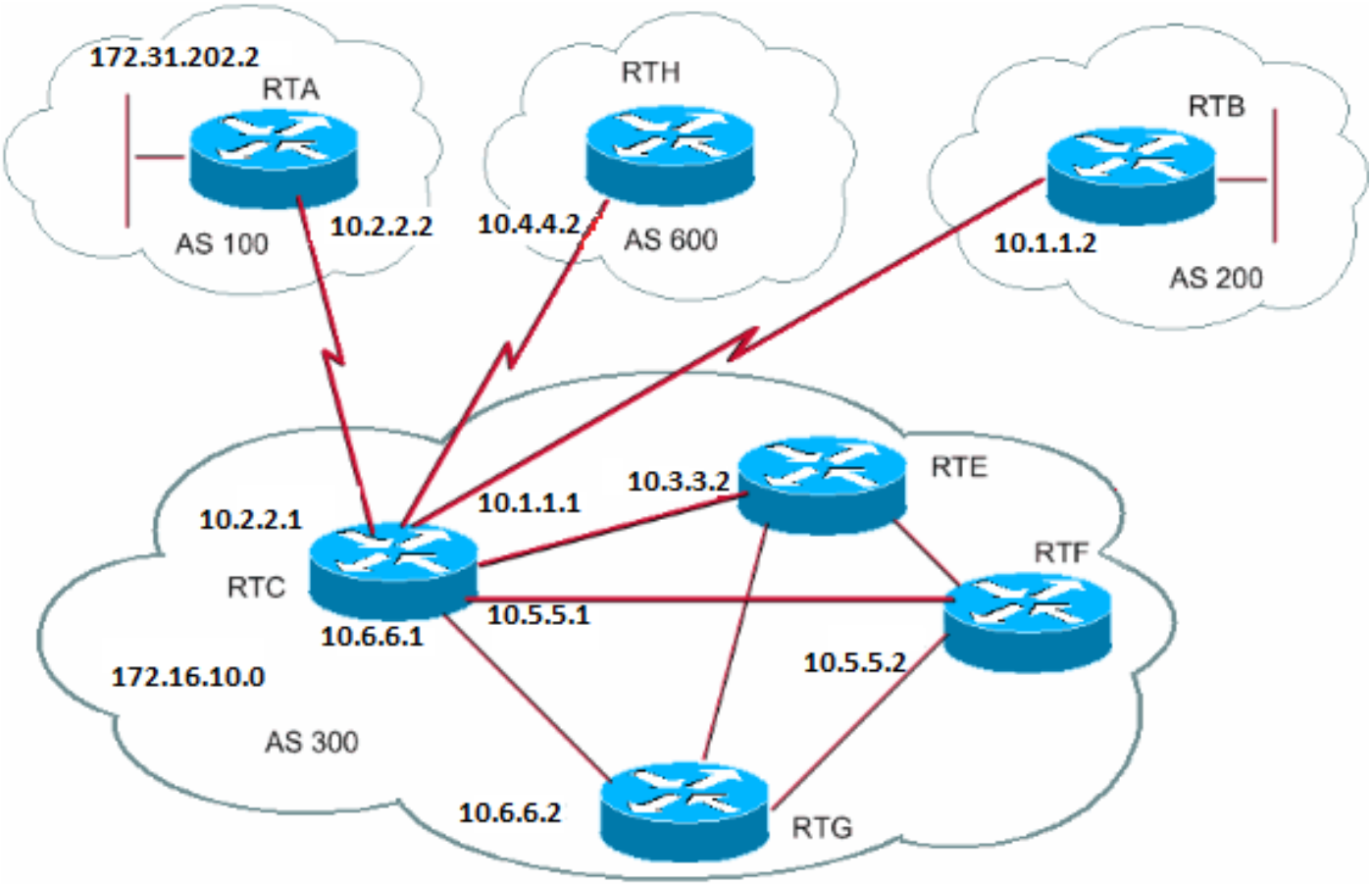
「BGPネイバーとルートマップ」セクションの図で、RTCが自身のネットワーク172.16.10.0を2つの異なるAS (AS100とAS200) にアドバタイズしていると仮定します。情報がAS600に伝搬されると、AS600内のルータには、2つの異なるルートを経由した172.16.10.0に関するネットワーク到達可能性情報が含まれます。1 つは AS100 を経由するパス (100、300) のルート、もう 1 つは、AS400 を経由するパス (400、200、300) のルートです。他の属性がすべて同じであれば、AS600 は最短パスである AS100 経由のルートを選択します。

AS300 は AS100 経由ですべてのトラフィックを取得します。AS300 側からこの決定を操作する場合は、AS100 経由のパスが AS400 を通過するパスよりも長いように見せることができます。これを行うには、AS100にアドバタイズされる現在のパス情報の先頭にAS番号を付加します。次のように自身の AS 番号を繰り返して追加する方法が一般的です。

```
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
```

この設定により、AS600は172.16.10.0に関するアップデートをAS100経由で(100、300、300、300)のパス情報とともに受信します。これは、AS600 が AS400 から受信するパス情報 (400、200、300) よりも長くなっています。

BGP ピア グループ



BGP ピア グループとは、同じアップデート ポリシーを使用する BGP ネイバーのグループのことです。通常、アップデート ポリシーはルート マップ、配布リスト、およびフィルタ リストによって設定されます。別個の隣接ルータごとに同じポリシーを定義するのではなく、代わりにピア グループ名を定義して、これらのポリシーをそのピア グループに割り当てます。

ピア グループのメンバーはピア グループのすべての設定オプションを継承します。発信アップデートに影響しないオプションの場合は、これらのオプションを上書きするようにメンバーを設定することもできます。上書きできるのは、着信に設定されたオプションのみです。

ピア グループを定義するには、次のコマンドを発行します。

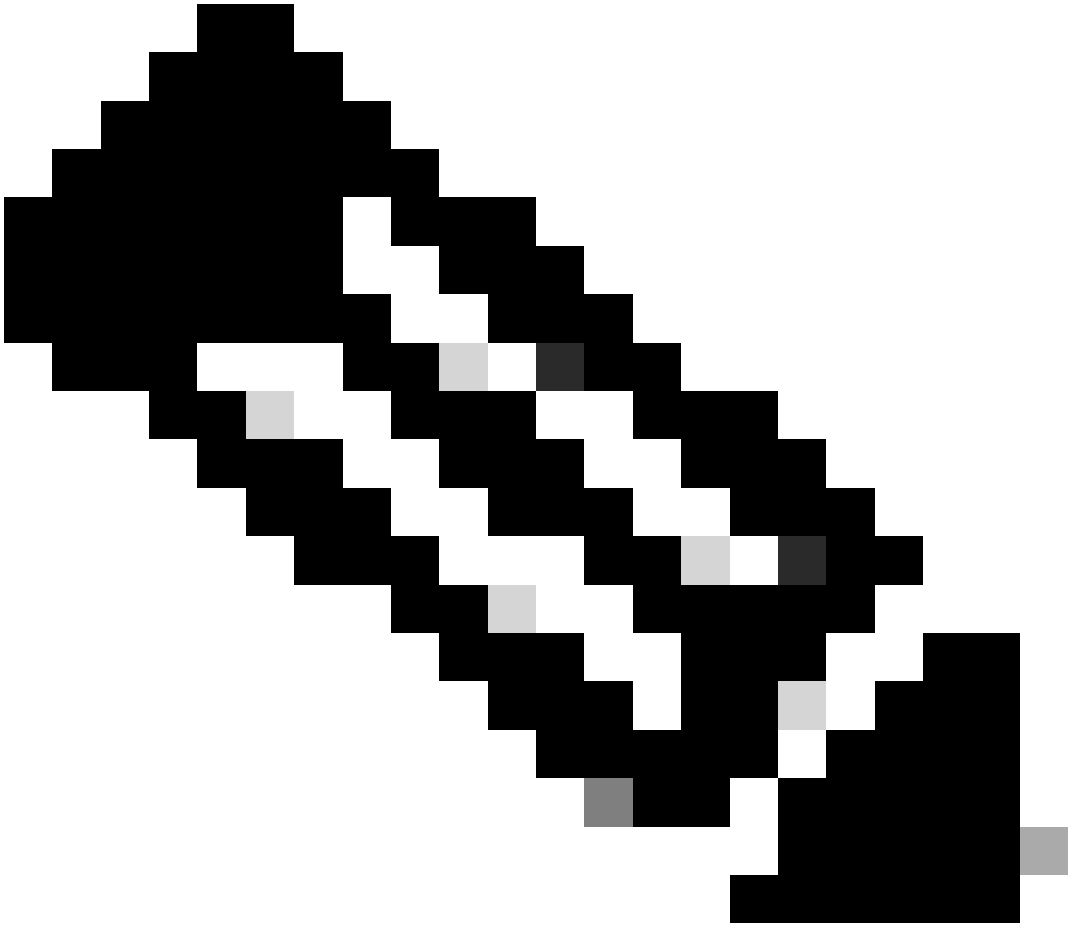
<#root>

neighbor peer-group-name peer-group

次の例では、ピア グループを内部および外部 BGP ネイバーに適用します。

```
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
```

この設定によって、internalmap という名前のピア グループが定義されます。 この設定では、このグループに対していくつかのポリシー(メトリックを5に設定するルートマップ**SETMETRIC**、2つの異なるフィルタリスト1および2など)を定義しています。この設定では、ピア グループをすべての内部ネイバー (RTE、RTF、および RTG) に適用しています。さらに、ネイバー RTE には別のフィルタ リスト 3 を定義しています。このフィルタ リストによって、ピア グループ内でフィルタ リスト 2 が上書きされます。

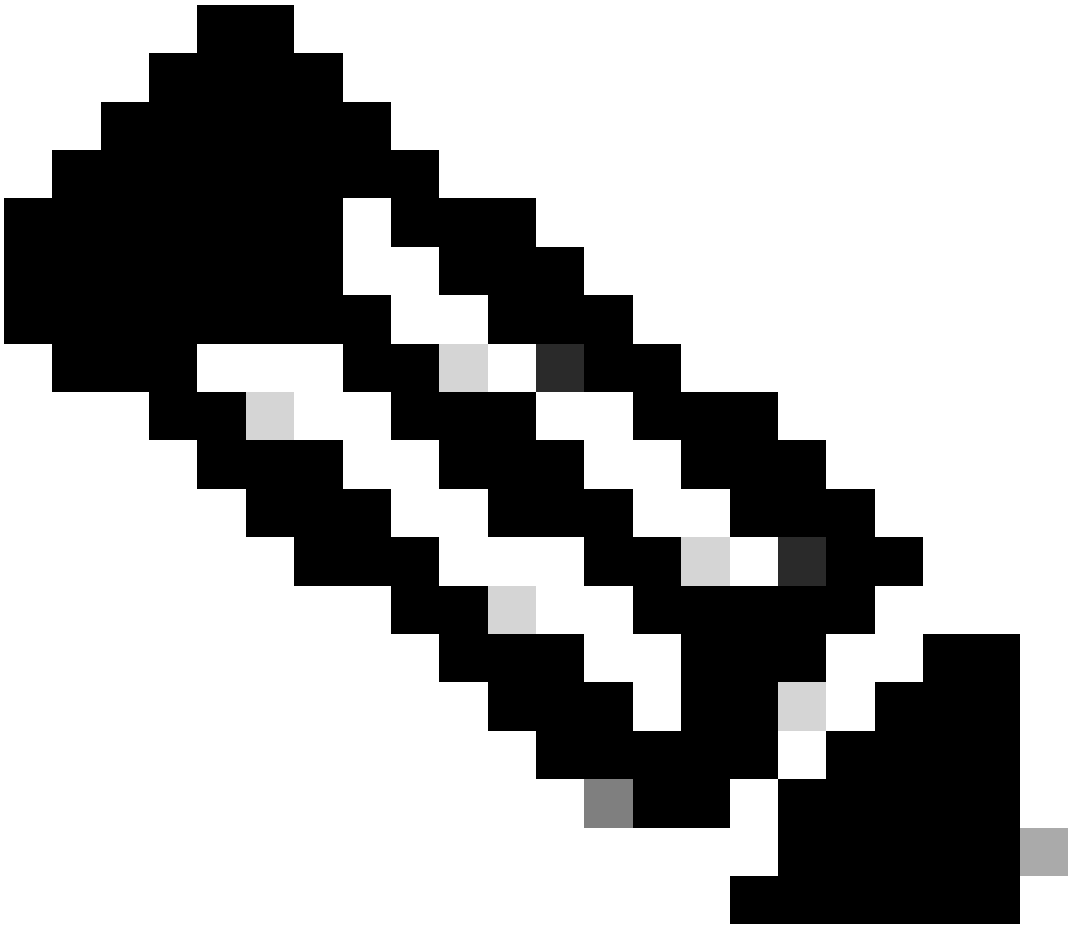


注：上書きできるのは、着信アップデートに影響を与えるオプションだけです。

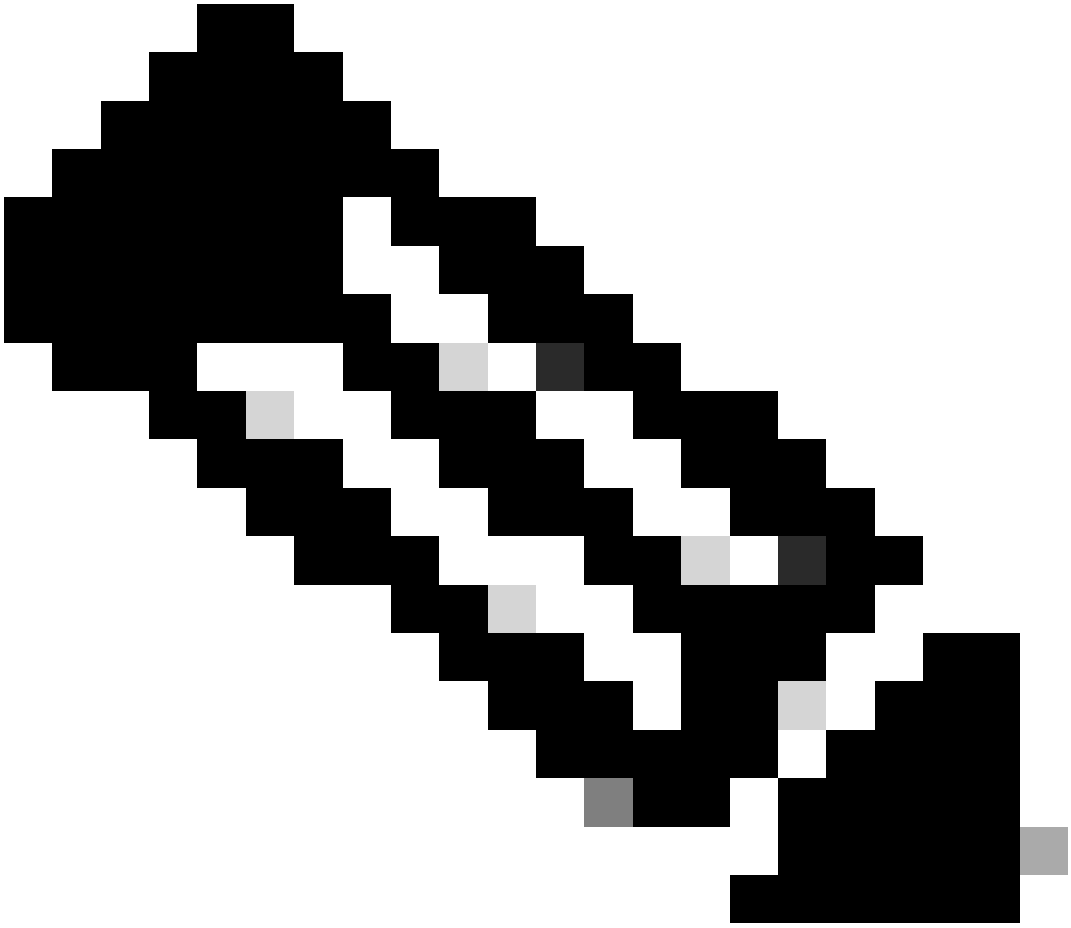
次に、外部ネイバーでのピア グループの使用方法を説明します。上記の同じ図で、RTC にピア グループ externalmap を設定し、そのピア グループを外部ネイバーに適用します。

```
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
```

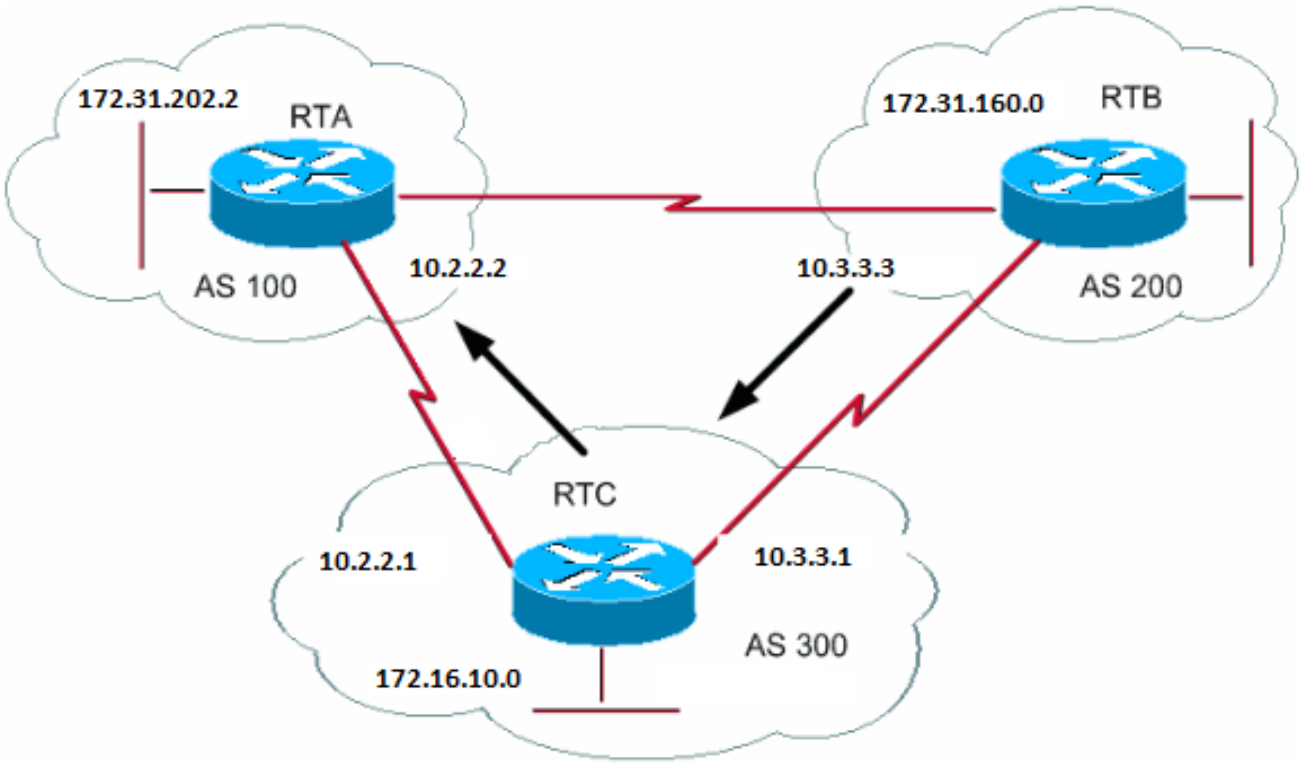


注：これらの設定では、ピアグループの外部でremote-as文を定義しています。これは、別の外部ASを複数定義する必要があるためです。また、フィルタ リスト 3 を割り当てることでネイバー 10.1.1.2 の着信アップデートを上書きします。ピア グループの詳細については、『BGP ピア グループ』を参照してください。



注:Cisco IOSソフトウェアリリース12.0(24)Sでは、BGPダイナミックアップデートピアグループ機能が導入されています。この機能は、それ以降の Cisco IOS ソフトウェア リリースでも使用できます。この機能では、同じ発信ポリシーを共有するネイバーのアップデート グループを動的に計算し、最適化する新しいアルゴリズムが使用されます。これらのネイバーは同じアップデート メッセージを共有できます。Cisco IOS ソフトウェアの以前のリリースでは、BGP アップデート メッセージのグループはピア グループ設定に基づいていました。この方法でアップデートをグループ化することで、発信ポリシーと特定のセッション設定が制限されていました。BGP ダイナミック アップデート ピア グループ機能は、ピア グループ設定からアップデート グループの複製を切り離します。これにより、コンバージェンス時間が短縮され、ネイバー設定の柔軟性が向上します。詳細については、『BGP ダイナミック アップデート ピアグループ』を参照してください。

CIDR と集約アドレス



BGP3 に対する BGP4 の主な拡張機能の 1 つは、クラスレス ドメイン間ルーティング (CIDR) です。CIDR またはスーパーネット化は IP アドレスの新しい処理方法です。CIDR では、クラス A、B、C などのクラスという概念はありません。たとえば、ネットワーク 192.168.213.0 は、かつては不正なクラス C ネットワークでしたが、現在は正当なスーパーネット 192.168.213.0/16 です。16は、IPアドレスの左端から数えたサブネットマスクのビット数を表します。これは 192.168.213.0 255.255.0.0 と同様です。

ルーティング テーブルのサイズを最小限に抑えるには、集約を使用します。集約とは、複数の異なるルートを 1 つのルートとしてアドバタイズできるように、それぞれのルートの特性を 1 つにまとめるプロセスです。この例では、RTB はネットワーク 172.31.160.0 を生成しています。このルートのスーパーネット 192.168.160.0 を RTA に伝達するように RTC を設定します。

```
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 は RTA に集約アドレス 192.168.160.0 を伝達します。

集約コマンド

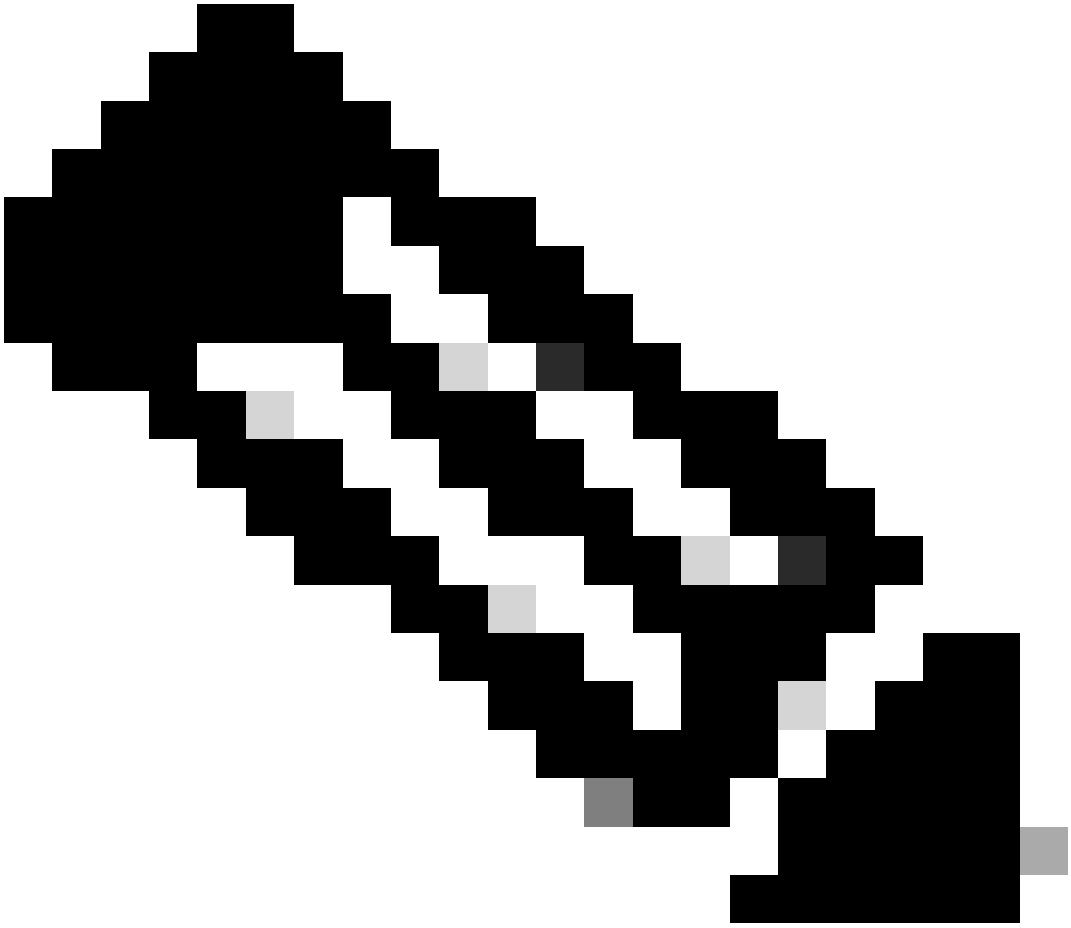
集約コマンドにはさまざまな種類があります。目的の集約動作を得るためには、それぞれのコマンドがどのように機能するかを理解する必要があります。

最初のコマンドは、「CIDRと集約アドレス」セクションの例で使用したものです。

<#root>

aggregate-address address-mask

このコマンドは、プレフィックス ルートおよびすべてのより具体的なルートをアドバタイズします。**aggregate-address 192.168.160.0**コマンドは、追加のネットワーク192.168.160.0を伝達しますが、これによってRTAへの172.31.160.0の伝達がブロックされることはありません。この結果、RTA にはネットワーク 192.168.160.0 と 172.31.160.0 が伝達されます。つまり、プレフィックス ルートとより具体的なルートの両方がアドバタイズされることになります。



注:BGPルーティングテーブルに特定のアドレスに関するより詳細なルートが存在しない場合、そのアドレスを集約することはできません。

たとえば、RTBのBGPテーブルに192.168.160.0のより詳細な工ントリがない場合、RTBは192.168.160.0の集約を生成できません。BGP テーブルにより具体的なルートをインジェクトすることは可能です。次の方法でルートをインジェクトできます。

-

  他の AS からの着信アップデート

- BGP への IGP またはスタティックの再配布

- network コマンド（例：network 172.31.160.0）

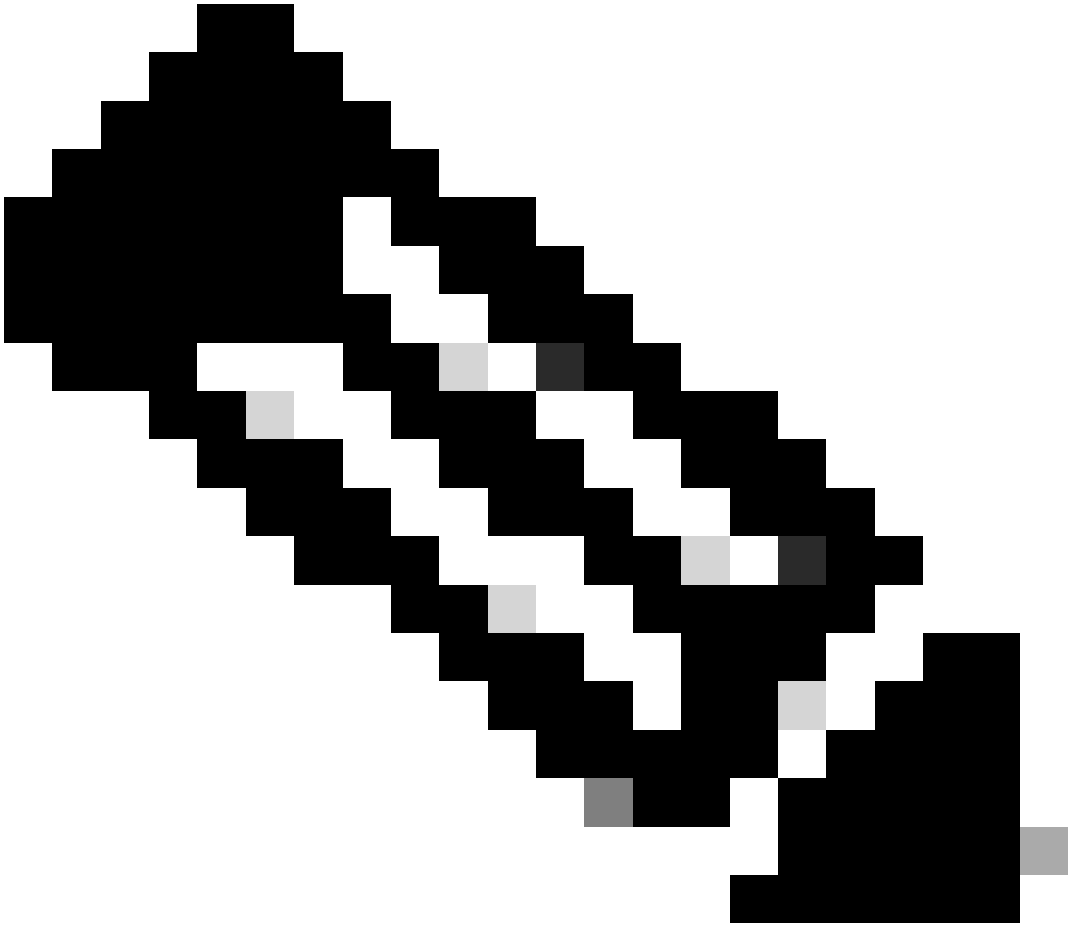
RTCがネットワーク192.168.160.0のみを伝搬し、より詳細なルートを伝搬しないようにするには、次のコマンドを発行します。

<#root>

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

このコマンドはプレフィックスのみをアドバタイズし、より具体的なルートをすべて抑制します。

aggregate 192.168.160.0 255.0.0.0 summary-onlyコマンドは、ネットワーク192.168.160.0を伝搬し、より詳細なルート172.31.160.0を抑制します。



注：network文によってBGPに注入されたネットワークを集約する場合、ネットワークエントリは常にBGPアップデートに注入されます。aggregate summary-only コマンドを使用する場合も同じようにインジェクトされます。「CIDR 例 1」の項で、この場合の例について説明します。

<#root>

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

このコマンドはプレフィックス ルートとより具体的なルートをアドバタイズしますが、ルーティング アップデートのパス情報に as-set 情報が含まれることになります。

<#root>

aggregate 192.168.0.0 255.0.0.0 as-set

「CIDR例2(as-set)」セクションは、このコマンドについて説明しています。

集約時により具体的なルートが抑制されるようにする場合は、ルート マップを定義して集約に適用します。この方法を使用すると、より具体的なルートを選択して抑制できます。

<#root>

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

このコマンドはプレフィックス ルートとより具体的なルートをアドバタイズしますが、アドバタイズメントはルート マップに基づいて抑制されます。「CIDR と集約アドレス」の項の図を基に、192.168.160.0 を集約し、より具体的なルート 192.168.160.20 を抑制して 172.31.160.0 の伝達を許可すると仮定します。この場合は次のルート マップを使用します。

```
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
```

suppress-map を定義すると、アクセス リストで許可されるパケットのアップデートが抑制されます。

次に、このルート マップを aggregate ステートメントに適用します。

```
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
```

次のような形式もあります。

<#root>

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

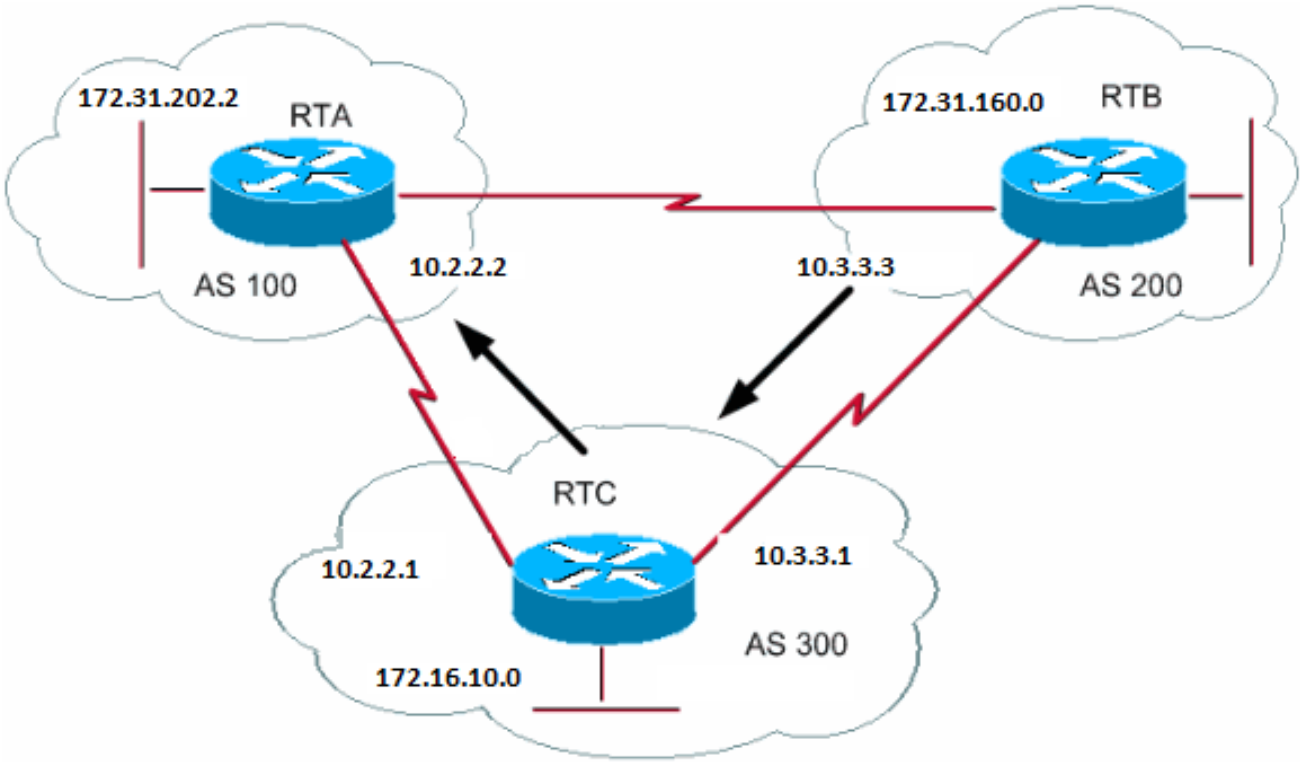
このコマンドを使用すると、集約の送信時にメトリックなどの属性を設定することができます。集約の送信元を IGP に設定するには、次のルート マップを aggregate attribute-map コマンドに適用します。

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

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

詳細は、『BGPでの経路集約について』を参照してください。

CIDR 例 1



要求：RTBがプレフィックス192.168.160.0をアドバタイズし、より詳細なルートをすべて抑制できるようにします。この要求に関する問題は、ネットワーク172.31.160.0がAS200にとってローカルである、つまり、AS200が172.31.160.0の生成元である点です。aggregate summary-only コマンドを使用しても、RTB が 172.31.160.0 のエントリを生成せずに 192.168.160.0 のプレフィックスを生成できるようにはなりません。RTB は 172.31.160.0 の発信元であるため、両方のネットワークを生成します。この問題を解決する方法は 2 つあります。

1 つは、スタティック ルートを使用して BGP に再配布する方法です。これにより、RTB は不完全（？）な送信元とともに集約をアドバタイズします。

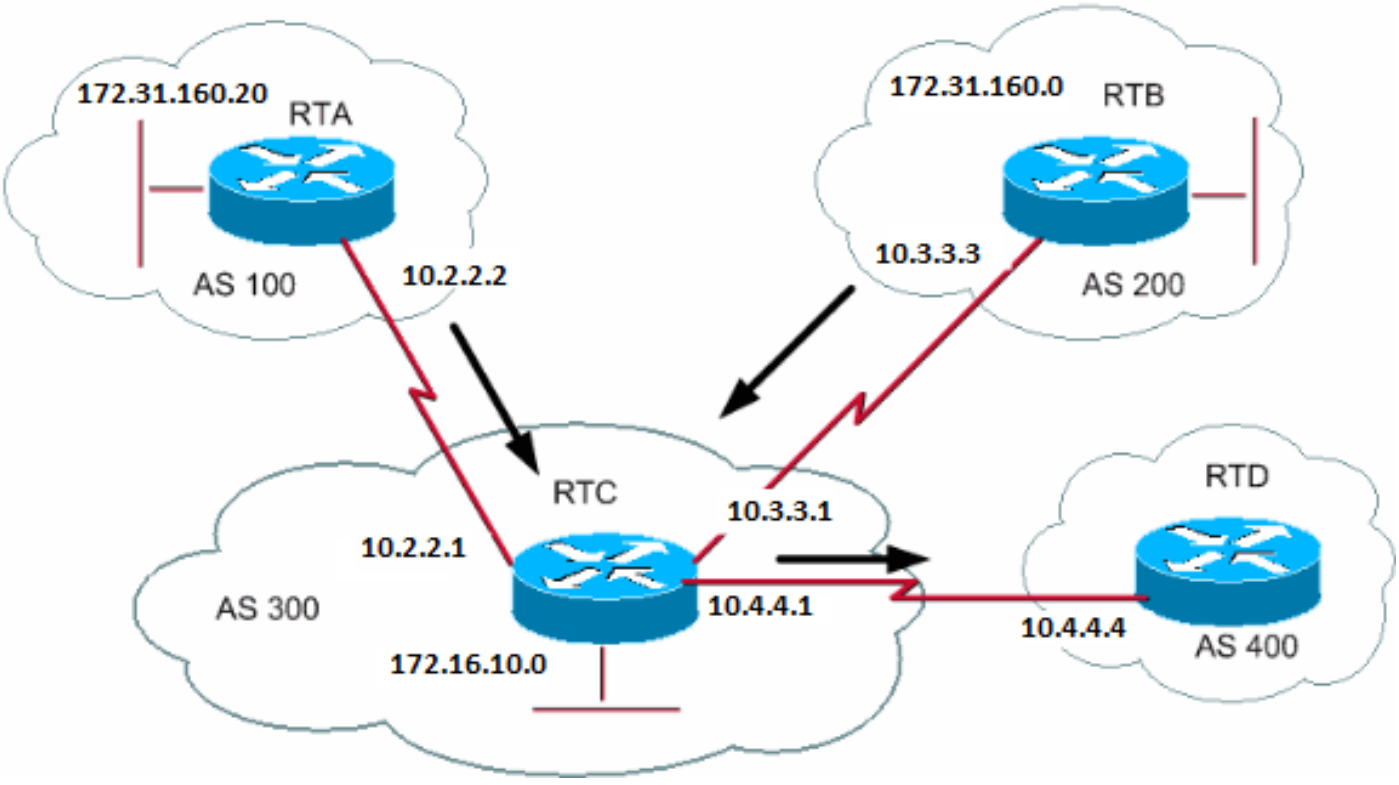
```
RTB#
router bgp 200
 neighbor 10.3.3.1 remote-as 300
 redistribute static

!--- 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
```

第2の解決策では、スタティックルートに加えて、**network**コマンドのエントリを追加します。このエントリによってアップデートの送信元が IGP に設定されますが、それ以外は同じ効果が得られます。

```
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 例 2 (as-set)

パス情報のサイズを縮小するには、集約で as-set ステートメントを使用します。as-set を使用すると、集約された複数のパスに AS 番号が複数回登場しても、AS 番号は一度しかリストされません。情報の集約によってパス属性に関する情報が失われるような場合は、aggregate as-set コマンドを使用します。次の例では、RTC は RTA から 192.168.160.20 に関するアップデート、RTB から 172.31.160.0 に関するアップデートを取得します。RTC がネットワーク 192.168.160.0/8 を集約して、RTD にそのネットワークを送信すると仮定します。RTD ではそのルートの送信元がわかりません。aggregate as-set ステートメントを追加すると、RTC は集合 {} 形式でパス情報を生成します。この集合には、パスの順番に関係なくすべてのパス情報が含まれます。




```
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
```

ケース 1 :

RTC には as-set ステートメントが設定されていません。RTCは、AS300から発信されたルートであるかのように、パス情報(300)を含むアップデート192.168.160.0/8をRTDに送信します。

```
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.
```

ケース 2 :

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

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

次の2つの項目、BGPコンフェデレーションおよびルートリフレクタは、AS内のiBGPピアリングの増大をさらに制御する必要があるInternet Service Provider（ISP；インターネットサービスプロバイダー）のためのものです。

BGP コンフェデレーション

BGP コンフェデレーションの実装により、AS 内部の iBGP メッシュが減少します。なぜなら、1 つの AS を複数の AS に分割し、グループ全体を単一のコンフェデレーションに割り当てるからです。AS はそれぞれ単独でフル メッシュ構造の iBGP を確立し、コンフェデレーション内の他の AS に接続しています。これらの AS は、コンフェデレーション内の AS と eBGP ピアで接続されていますが、iBGP を使用する場合と同様にルーティングを交換します。これにより、コンフェデレーションにネクスト ホップ、メトリック、ローカル プリファレンス情報が保持されます。外部からは、コンフェデレーションが単一の AS のように見えます。

BGP コンフェデレーションを設定するには、次のコマンドを発行します。

```
<#root>
```

bgp confederation identifier <autonomous-system>

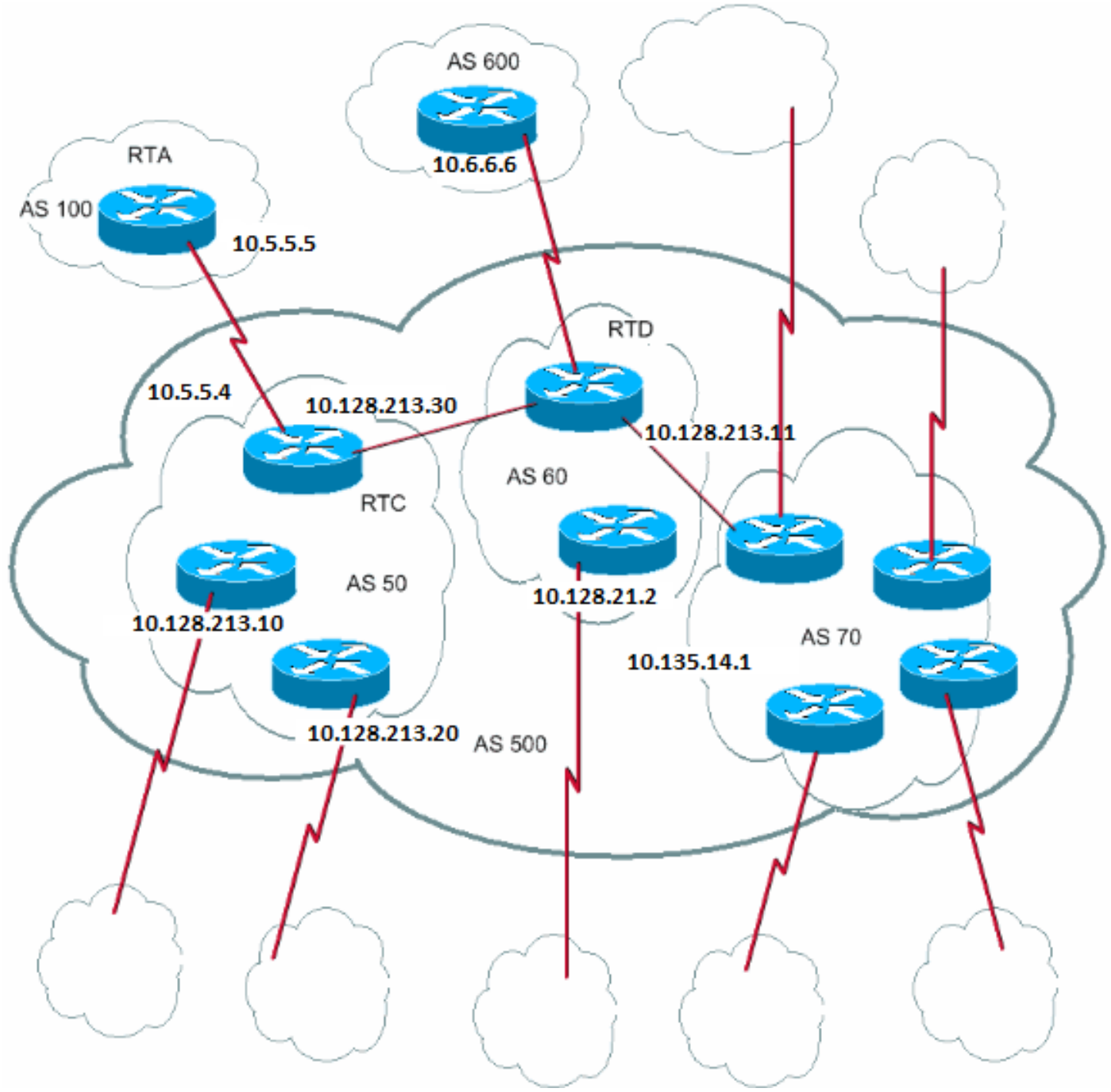
コンフェデレーション ID は、コンフェデレーション グループの AS 番号です。

次のコマンドを発行すると、コンフェデレーション内で複数の AS 間のピアリングが実行されます。

```
<#root>
```

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

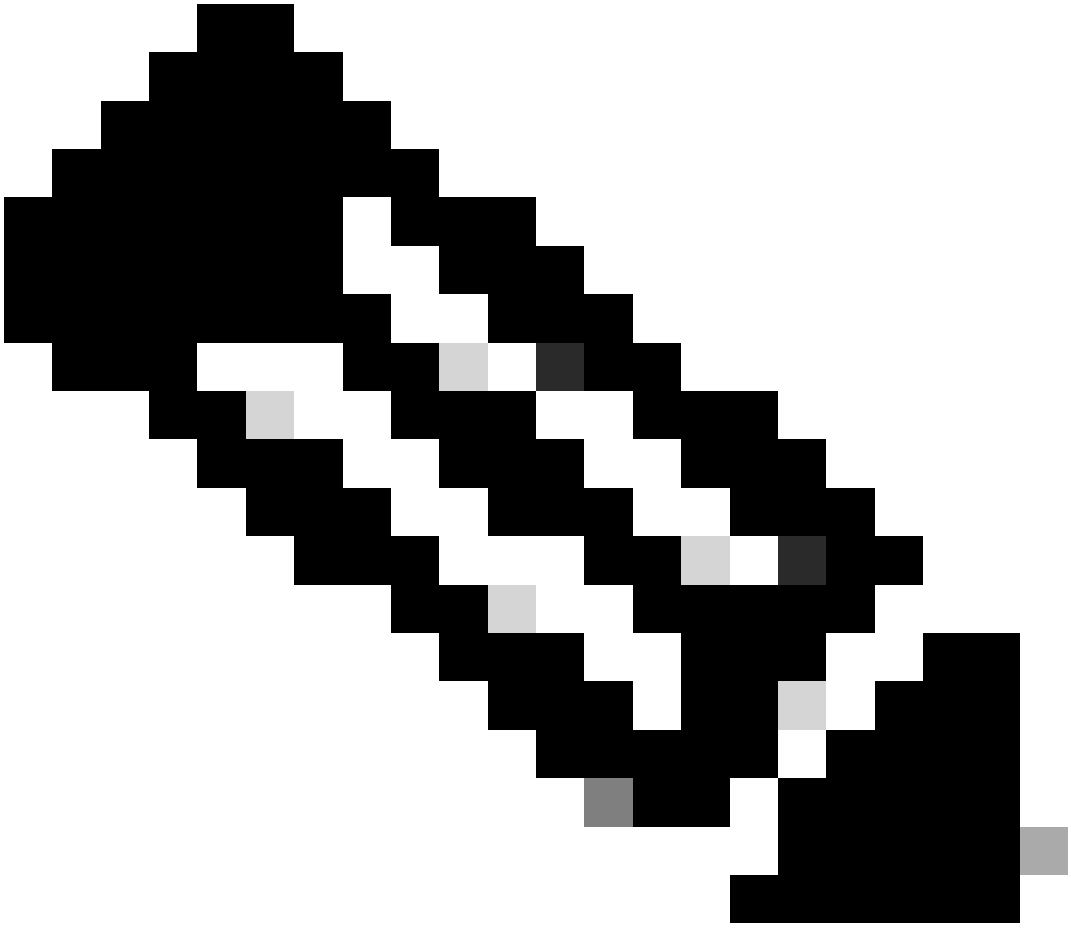
次にコンフェデレーションの例を示します。



9 台の BGP スピーカで構成された AS500 があると仮定します。BGP 以外のスピーカも別に存在しますが、ここでは他の AS への eBGP 接続を持つ BGP スピーカのみを取り上げます。AS500 内でフル メッシュ構造の iBGP を確立するには、ルータごとに 9 つのピア接続が必要になります。iBGP ピアが 8 つと、外部 AS への eBGP ピアが 1 つです。

コンフェデレーションを使用すると、AS500 を複数の AS (AS50、AS60、および AS70) に分割できます。AS のコンフェデレーション ID として 500 を割り当てます。外部からは 1 つの AS (AS500) のみが認識されます。AS50、AS60、およびAS70のそれぞれに対してiBGPピアのフルメッシュを定義し、**bgp confederation peers**コマンドを使用してコンフェデレーションピアのリストを定義します。

RTC、RTD、および RTA ルータの設定例は次のとおりです。



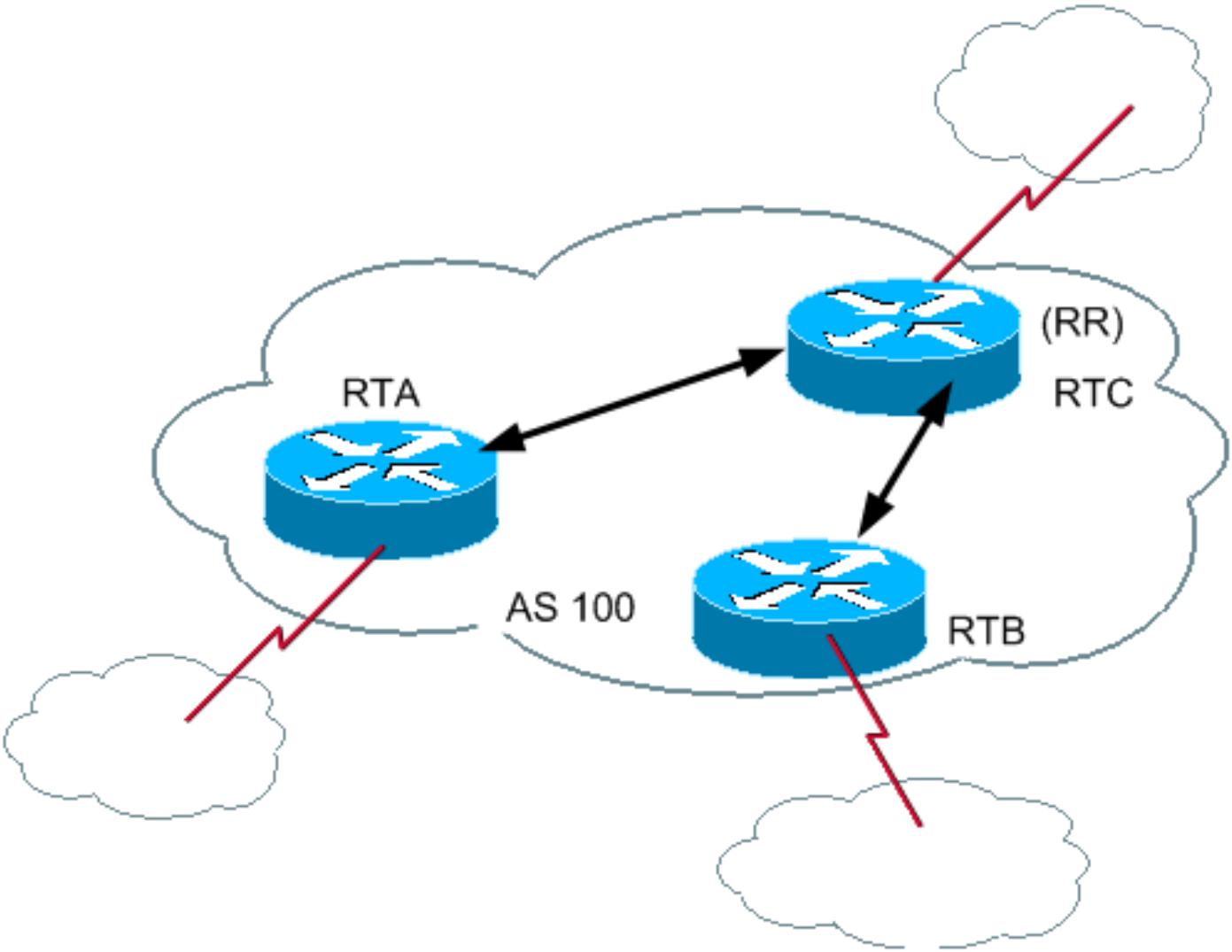
注:RTAはAS50、AS60、またはAS70を認識していません。RTA は AS500 のみを認識しています。

```
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)
```

ルート リフレクタ

AS 内での iBGP ピアリングの増大に対処するもう 1 つの方法は、ルート リフレクタ (RR) です。「iBGP」セクションで説明したように、BGPスピーカは、別のiBGPスピーカ経由で学習したルートを第3のiBGPスピーカにはアドバタイズしません。この制約を少し緩めて、ルータが iBGP で学習したルートを他の iBGP にアドバタイズ (リフレクト) できるように追加の制御を行うことができます。このルート リフレクションにより、AS 内の iBGP ピアの数は減少します。



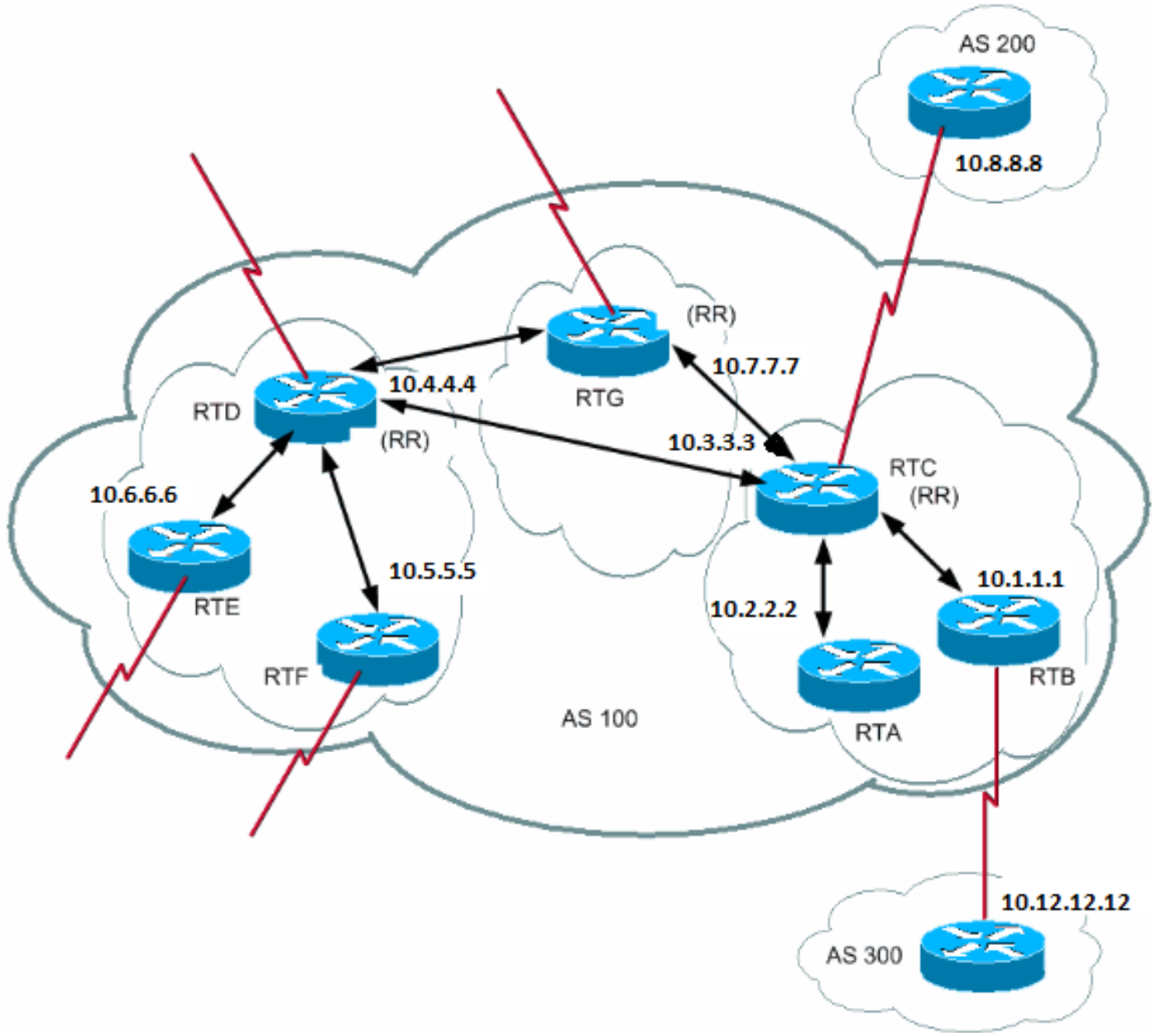
通常は、AS100 内の RTA、RTB、および RTC 間でフル メッシュ構造の iBGP を維持する必要があります。RR の概念を利用すれば、RTC を RR として選択できます。これにより、RTC は RTA および RTB との部分 iBGP ピアリングを保持します。RTC が RTA および RTB から到達するアップデートの RR であるため、RTA と RTB の間のピアリングは必要ありません。

<#root>

neighbor <ip address> route-reflector-client

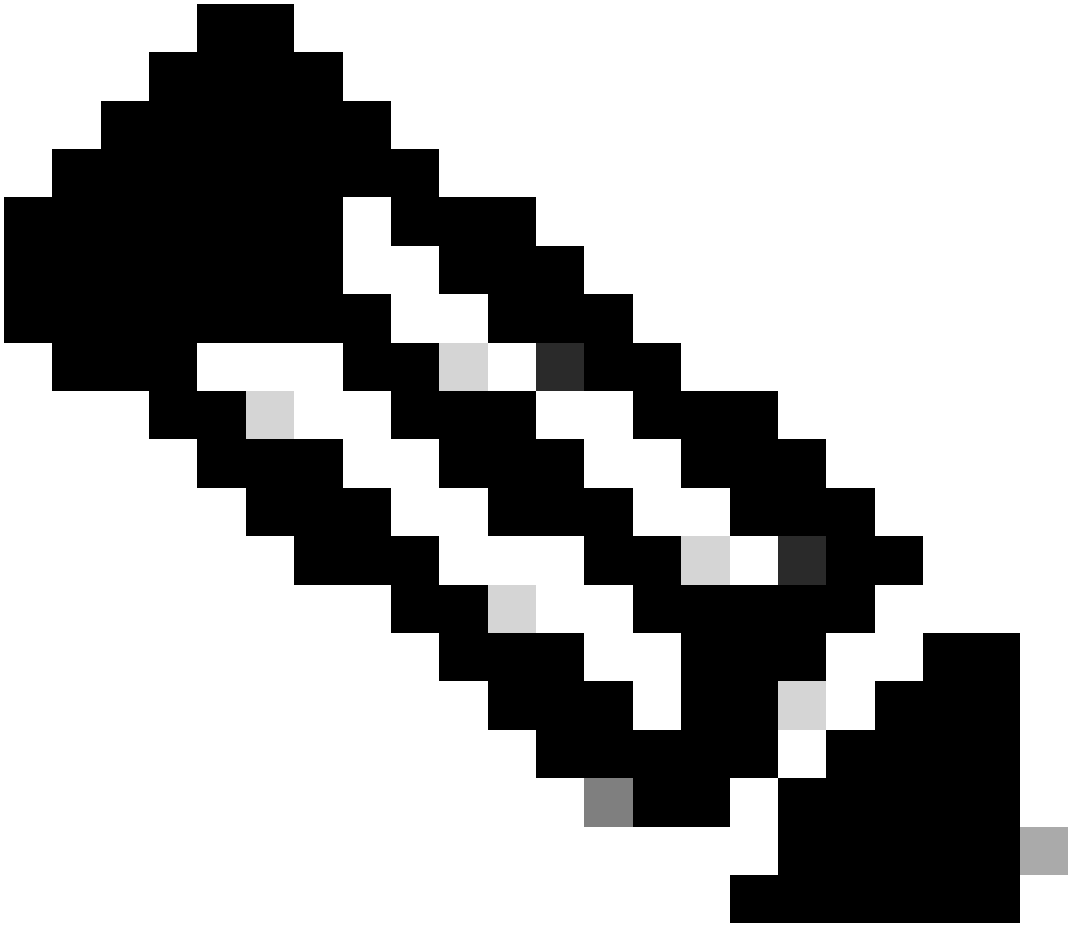
このコマンドが設定されたルータは RR になり、コマンドで指定されたネイバーはその RR のクライアントになります。この例では、RTC の設定で neighbor route-reflector-client コマンドを使用し、RTA と RTB の IP アドレスを指定します。RR とクライアントの組み合わせを「クラスタ」と呼びます。この例では、AS100 内で RTA、RTB、および RTC が 1 つの RR を含むクラスタを形成します。

クライアントではないRRの他のiBGPピアは非クライアントです。



1 つの AS に複数の RR を設定できます。この場合、各 RR は他の RR を他のすべての iBGP スピーカと同様に扱います。他の RR は同じクラスタ (クライアント グループ) に属する場合も、他のクラスタ グループに属する場合もあります。簡単な設定では、AS を複数のクラスタに分割できます。各 RR で、フル メッシュ型トポロジの非クライアント ピアとして他の RR を設定します。クライアントは、クライアントクラスタ外のiBGPスピーカとピア関係を確立できません。

上記の図では、RTA、RTB、およびRTCが1つのクラスタを形成しています。RTC は RR です。RTC にとっては、RTA と RTB がクライアントで、その他はすべて非クライアントです。neighbor route-reflector-client コマンドによって RR のクライアントが指定されることに注意してください。同様に、RTD は RTE および RTF クライアントの RR です。RTG は 3 つ目のクラスタの RR です。



注:RTD、RTC、およびRTGはフルメッシュで接続されますが、クラスタ内のルータはフルメッシュ接続されません。

RR はルートを受信すると、次のリストのように転送します。ただし、この動作はピア タイプによって異なります。

- 非クライアント ピアからのルート：クラスタ内のすべてのクライアントにリフレクトする。
- クライアント ピアからのルート：すべての非クライアント ピアおよびクライアント ピアにリフレクトする。

- eBGP ピアからのルート：すべてのクライアント ピアと非クライアント ピアにアップデートを送信する。

RTC、RTD、および RTB ルータの相対的な BGP 設定を次に示します。

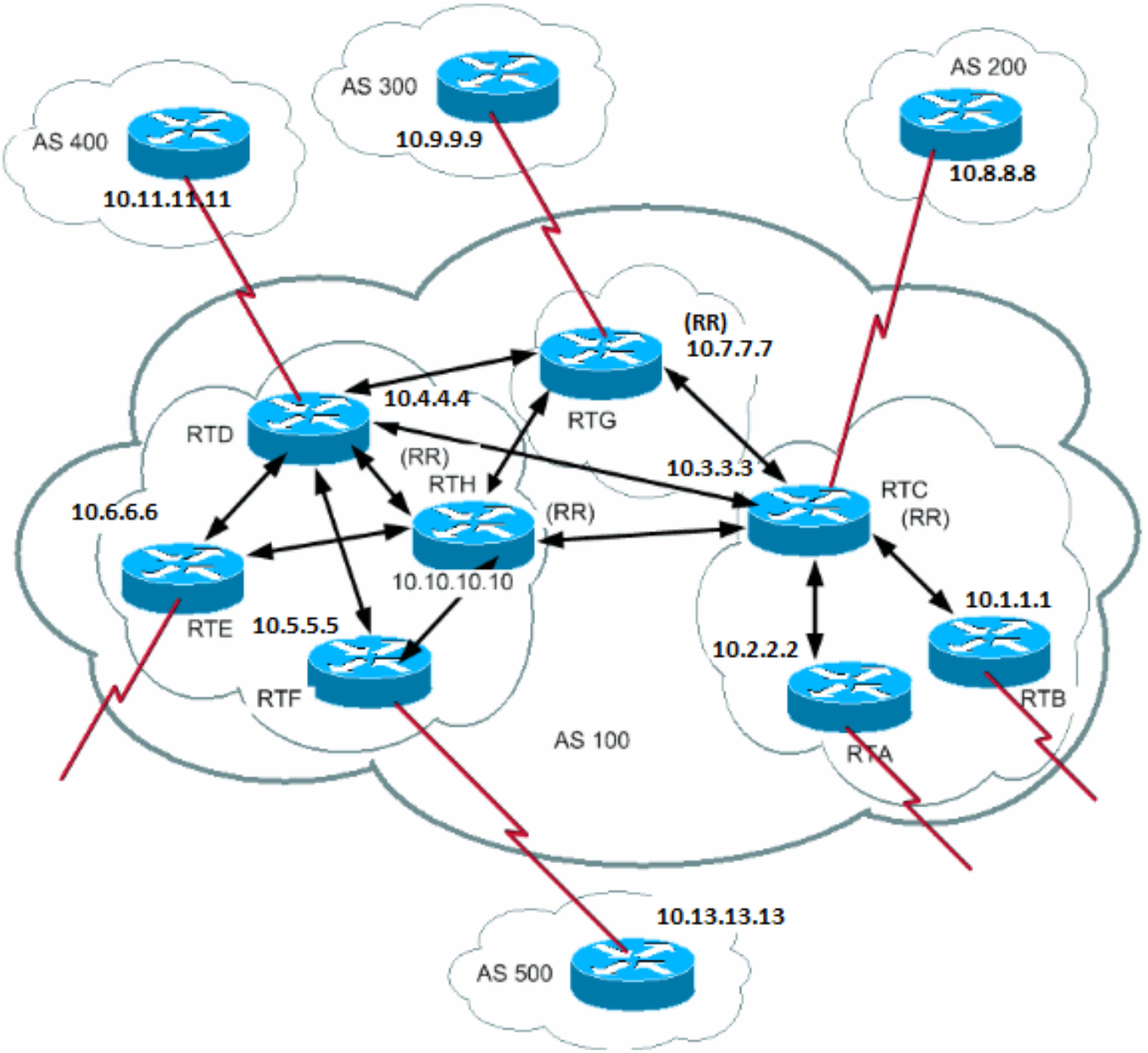
```
RTC#
router bgp 100
 neighbor 10.2.2.2 remote-as 100
 neighbor 10.2.2.2 route-reflector-client
 neighbor 10.1.1.1 remote-as 100
 neighbor 10.1.1.1 route-reflector-client
 neighbor 10.7.7.7 remote-as 100
 neighbor 10.4.4.4 remote-as 100
 neighbor 10.8.8.8 remote-as 200

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
```

iBGP で学習されたルートがリフレクトされるため、ルーティング情報のループが発生する可能性があります。RR スキームにはこのループを回避するための方法がいくつか用意されています。

- **originator-id：これは非推移的なオプションの BGP 属性で、長さは 4 バイトです**。この属性は RR によって作成され、ローカル AS 内のルート発信元のルータ ID（RID）を伝送します。設定が適切でないためにルーティング情報が発信元に戻された場合、その情報は無視されます。

- **cluster-list**：クラスタリストについては、「クラスタ内の複数のRR」を参照してください。

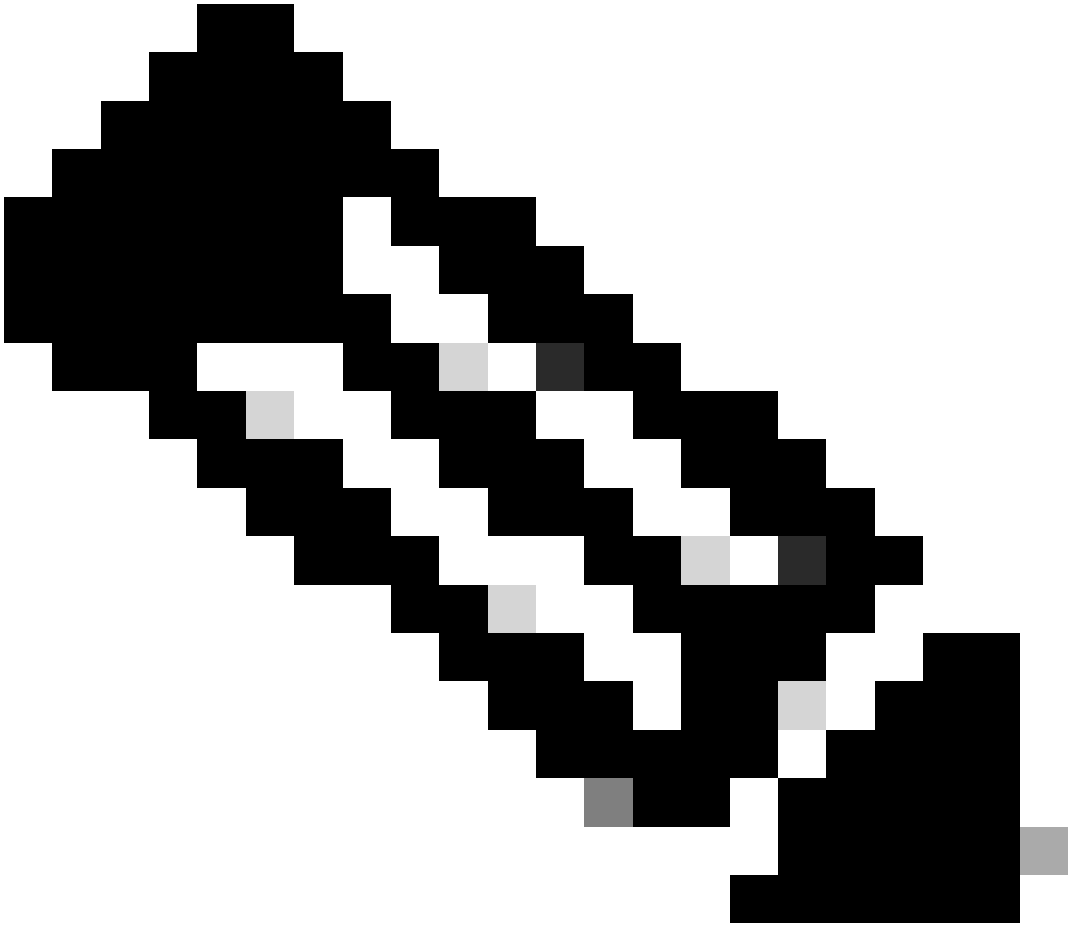
クラスタ内の複数の RR



通常、クライアントのクラスタには、RR が 1 つ存在します。この場合は、RR のルータ ID によってクラスタが識別されます。冗長性を向上してシングル ポイント障害を回避するために、1 つのクラスタに複数の RR を設定できます。この場合は、RR が同じクラスタ内の RR からのアップデートを認識できるように、同じクラスタ内のすべての RR に 4 バイトのクラスタ ID を設定する必要があります。

クラスタ リストは、ルートが通過したクラスタ ID のシーケンスです。RR は RR クライアントからのルートをクラスタ外の非クライアントにリフレクトする際に、クラスタ リストにローカル クラスタ ID を付加します。このアップデートにクラスタ リストがない場合は、RR によって作成されます。RR はこの属性を使用して、設定が適切でないためにルーティング情報が同じクラスタにループバックされていないかどうかを特定できます。クラスタ リストにローカル クラスタ ID が見つかった場合、そのアドバタイズメントは無視されます。

上記の図では、RTD、RTE、RTF、および RTH が 1 つのクラスタに属しています。RTD と RTH はどちらも同じクラスタの RR です。

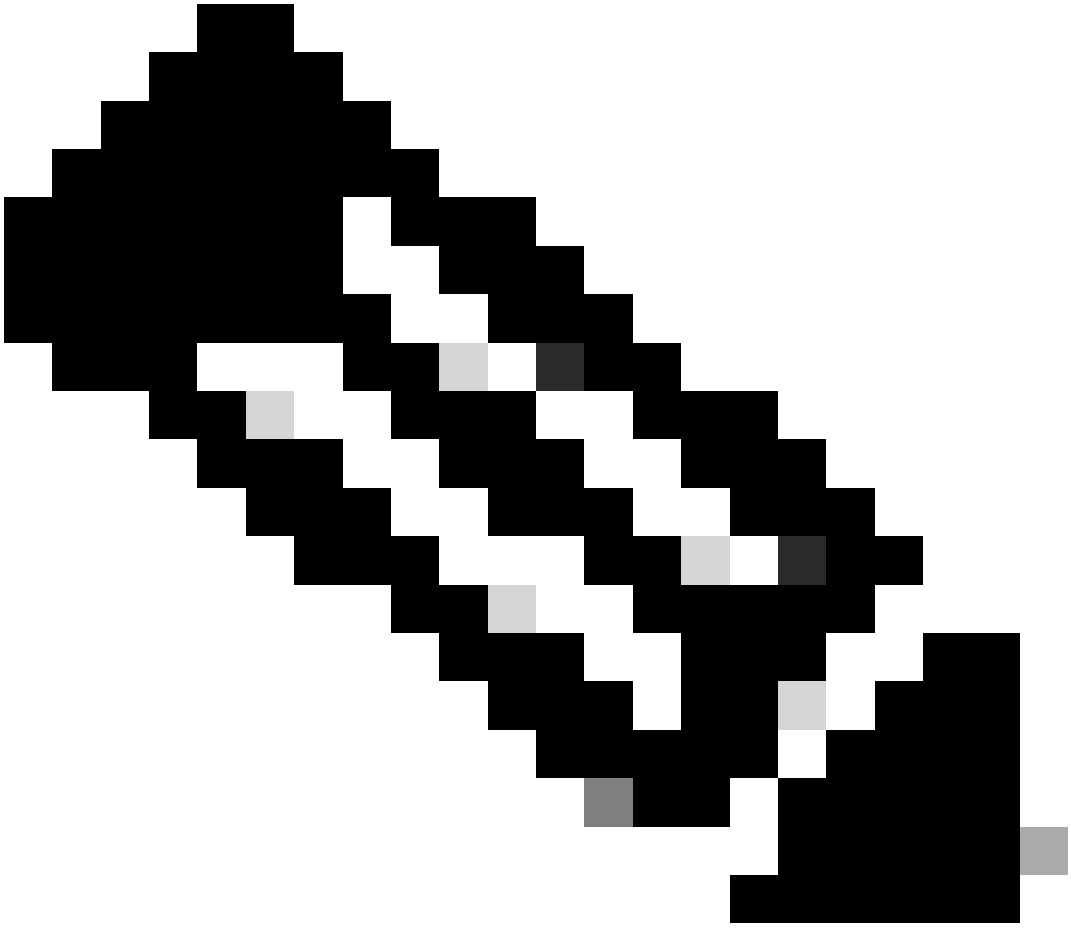


注：RTHがすべてのRRとフルメッシュのピアリングを構成しているので、冗長性が確保されています。RTD がダウンした場合は、RTH が RTD の役割を引き継ぎます。

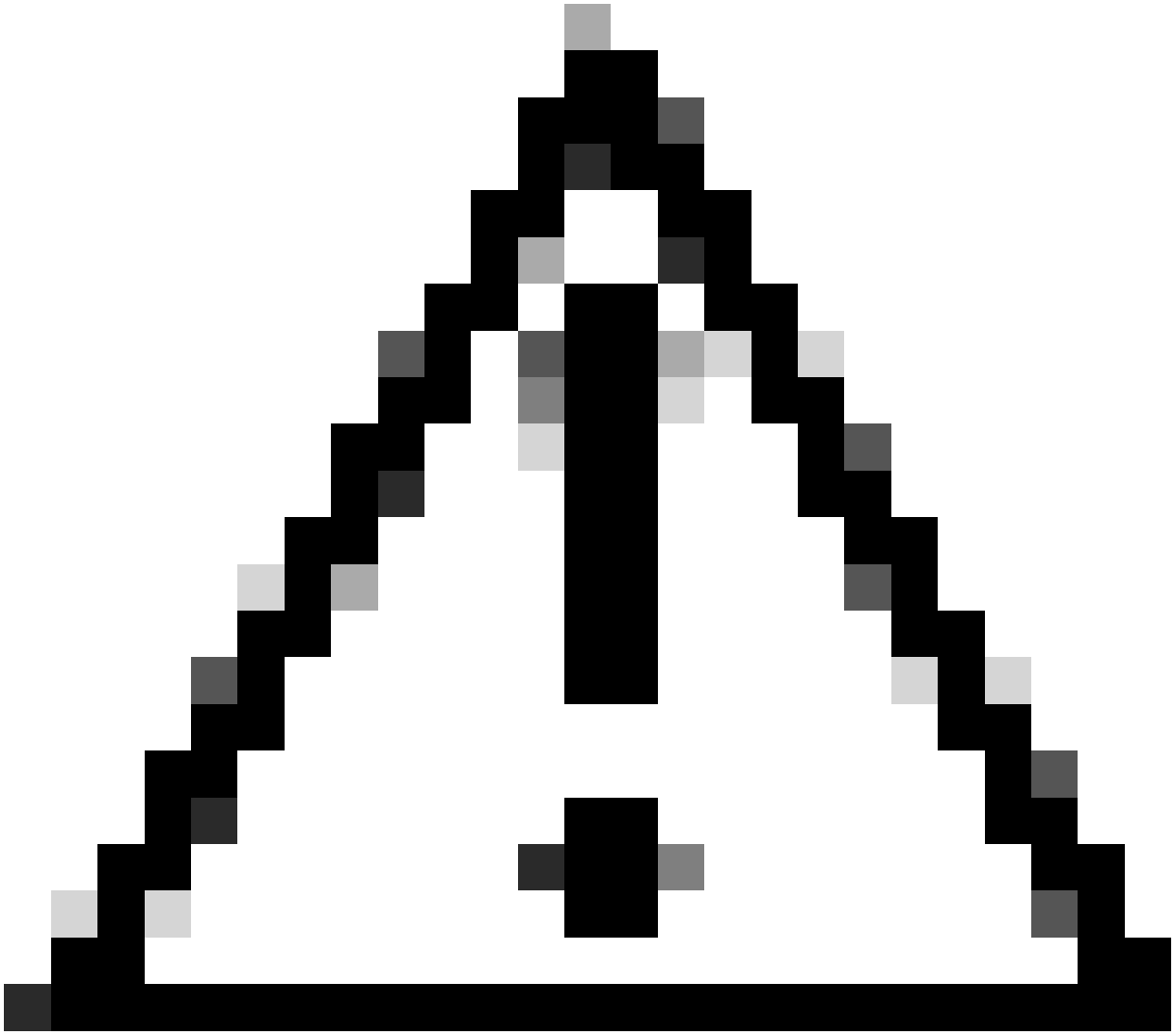
RTH、RTD、RTF、および RTC の設定は次のとおりです。

```
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：RTCのクラスタにはRRが1つしか存在しないため、RTCにbgp cluster-idコマンドを設定する必要はありません。



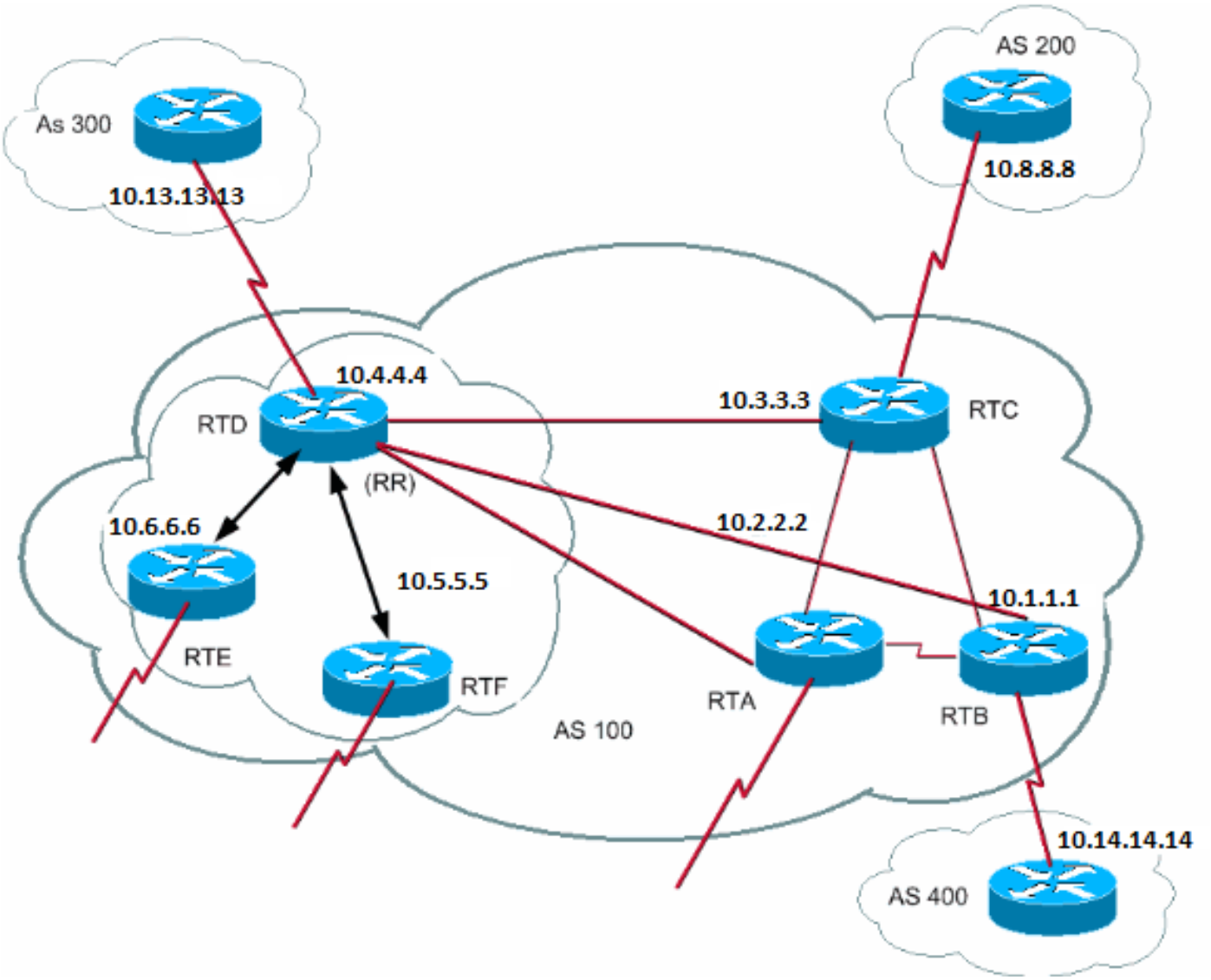
注意：この設定では、ピアグループを使用していません。クラスタ内のクライアントが互いの間に直接 iBGP ピアを確立せずに、RR 経由でアップデートを交換する場合は、ピア グループを使用しないでください。ピア グループを設定すると、RR でのルートの送信元に対する取り消しがクラスタ内のすべてのクライアントに送信される可能性があります。この送信によって問題が発生することがあります。

RR では、ルータのサブコマンドである **bgp client-to-client reflection がデフォルトでイネーブルになっています。** RR で BGP クライアント間のリフレクションをオフにして、クライアント間で冗長 BGP ピアリングを行えば、安全にピア グループを使用できます。詳細については、『ピア グループの制限』を参照してください。

RR と従来型 BGP スピーカ

AS には RR の概念に対応していない BGP スピーカが含まれる場合があります。このドキュメントでは、これらのルータを従来型

BGP スピーカと呼びます。RR スキームを使用すれば、そのような従来型 BGP スピーカの共存が可能です。これらのルータは、クライアント グループに属することも、非クライアント グループに属することもできます。これらのルータが存在すると、現在の iBGP モデルから RR モデルに簡単かつ段階的に移行できます。クラスタを作成するには、まず 1 台のルータを RR として設定し、他の RR および RR クライアントを通常の iBGP ピアにします。その後は、段階的にクラスタを追加作成できます。



この図では、RTD、RTE、および RTF がルート リフレクションの概念に対応しています。RTC、RTA、およびRTBは従来型ルータです。これらのルータを RR として設定することはできません。これらのルータと RTD の間で通常の iBGP メッシュを形成できます。その後、アップグレードする準備ができたら、RTC を RTA および RTB クライアントの RR にすることができます。クライアントはルート リフレクション スキームに対応している必要はなく、アップグレードが必要なのは RR だけです。

RTD と RTC の設定は次のとおりです。

```
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
```

RTC をアップグレードして RR にする準備ができたら、iBGP フル メッシュを削除し、RTA と RTB を RTC のクライアントにします。

ルーティング情報のループの回避

このドキュメントでは、これまでに、潜在的な情報ループを回避するために使用できる2つのアトリビュート、**originator-id**および**cluster-list**について説明しました。

ループを制御するもう1つの方法は、発信ルートマップの**set**句で追加の制限を適用することです。発信ルート マップの set 句は、iBGP ピアにリフレクトされたルートには影響しません。

また、ネイバーごとの設定オプションである**next-hop-self**で追加の制限を適用することもできます。リフレクトされたルートのネクストホップは変更できないため、RRで**next-hop-selfo**を使用した場合、この句の影響を受けるのはeBGPで学習したルートのネクストホップのみです。

ルート フラップ ダンプニング

Cisco IOS ソフトウェア リリース 11.0 でルート ダンプニングが導入されました。ルート ダンプニングとは、ルート フラッピングに起因する不安定な状態の発生を最小限に抑えるメカニズムです。また、ルート ダンプニングはネットワーク上の変動も軽減します。正常に動作していないルートを特定するために、基準を定義します。フラップが発生したルートには、フラップごとに 1000 のペナルティが割り当てられます。累積ペナルティが事前に定義された抑止限界に達すると、すぐにルートのアドバタイズメントが抑止されます。ペナルティは、事前設定された半減期に基づいて指数関数的に減少します。ペナルティが事前に定義された再使用制限の下で減少すると、ルートのアドバタイズメントは抑制されなくなります。

ルート ダンプニングは、iBGP 経由で学習された AS 外部のルートには適用されません。これにより、ルート ダンプニングでは AS 外部のルートに対して iBGP ピアがより高いペナルティを持つことが回避されます。

ペナルティは 5 秒単位で減少します。ルートは、10秒の精度で抑制されません。ルータは、ペナルティが「reuse limit」の半分よりも小さくなるまで、ダンプニング情報を保持します。半分未満になった時点で、ルータはこの情報を消去します。

最初は、ダンプニングがデフォルトでオフになっています。必要に応じて、この機能を将来的にデフォルトで有効にできます。次のコマンドを使用してルート ダンプニングを制御します。

- **bgp dampening：ダンプニングをオンにする**。

- **no bgp dampening：ダンプニングをオフにする。**

- **bgp dampeninghalf-life-time**：半減期を変更する。

一度にすべてのパラメータを設定する場合は、次のコマンドを使用します。

- **bgp dampeninghalf-life-timereuseuppressmaximum-suppress-time**

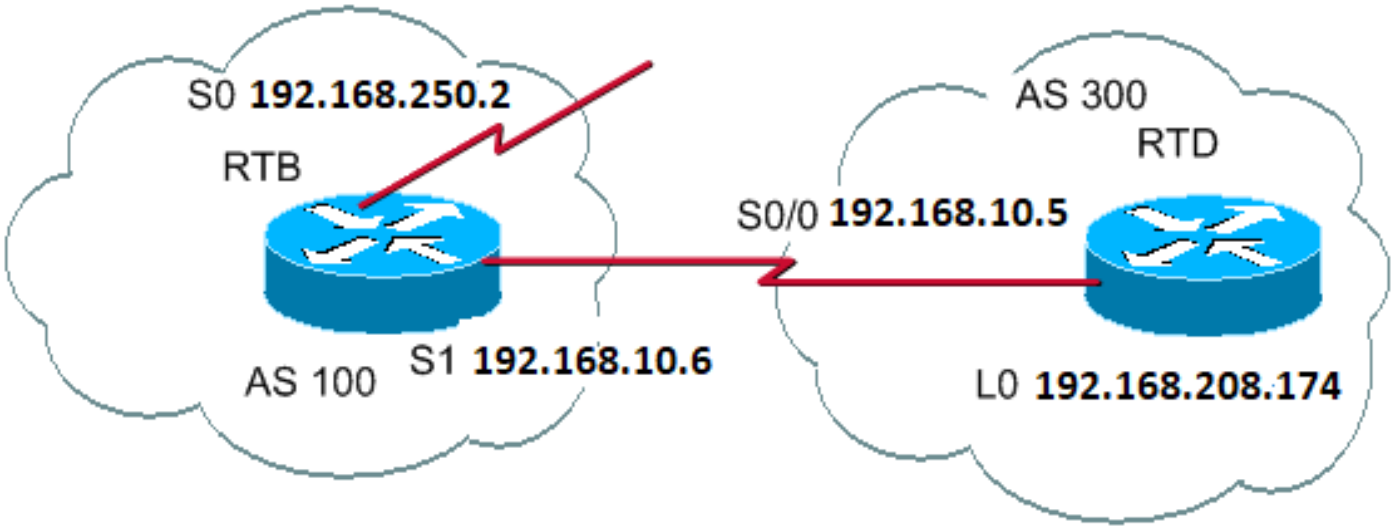
構文の詳細は次のとおりです。

- **half-life-time**：指定可能な範囲は1 ~ 45分。現在のデフォルトは15分。

- **reuse-value**：指定可能な範囲は1 ~ 20,000。デフォルトは750。

- **suppress-value**：指定可能な範囲は1 ~ 20,000。デフォルトは2000。

- **max-suppress-time**：ルートの抑制の最大期間。指定可能な範囲は 1 ~ 255 分。デフォルトは half-life-time の 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
```

RTB では、デフォルト パラメータを使用してルート ダンプニングが設定されています。RTD への eBGP リンクが安定している場合、RTB の BGP テーブルは次のように表示されます。

<#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
```

ルート フラップをシミュレートするには、RTD で clear ip bgp 192.168.10.6 コマンドを発行します。RTB の BGP テーブルは次のように表示されます。

<#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のBGPエントリはahistorystateです。この状態は、ルートへのベスト パスはないが、ルート フラッピングに関する情報はまだ保持されていることを意味します。

<#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
```

ルートはフラッピングに対するペナルティを受け取っていますが、ペナルティは「suppress limit」の下にあります。デフォルトは 2000 です。ルートの抑制はまだ実行されていません。ルート フラップがさらに数回発生すると、次のように表示されます。

```
    <#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
```

ルートはダンプニングまたは抑制されています。ルートは、ペナルティが「再使用値」に達すると再使用されます。この例の場合、再使用値はデフォルトの 750 です。ペナルティが再使用限度の半分未満になると、ダンプニング情報は消去されます。この例では、ペナルティが 375 (750/2=375) になると消去されます。次のコマンドは、フラップ統計情報を表示およびクリアする場合に使用します。

- **show ip bgp flap-statistics：すべてのパスのフラップ統計情報を表示する**。
- **show ip bgp flap-statistics regexpregular-expression**：正規表現に一致するすべてのパスのフラップ統計情報を表示する。
- **show ip bgp flap-statistics filter-listlist**：フィルタを通過するすべてのパスのフラップ統計情報を表示する。
- **show ip bgp flap-statisticsA.B.C.D m.m.m.m**：単一エントリのフラップ統計情報を表示する。
- **show ip bgp flap-statisticsA.B.C.D m.m.m.mlonger-prefix**：より具体的なエントリのフラップ統計情報を表示する。
- **show ip bgp neighbor [dampened-routes] | [flap-statistics]：ネイバーからのすべてのパスのフラップ統計情報を表示する**。
- **clear ip bgp flap-statistics：すべてのルートのフラップ統計情報をクリアする**。
-

clear ip bgp flap-statistics regexpregular-expression：正規表現に一致するすべてのパスのフラップ統計情報をクリアする。

- **clear ip bgp flap-statistics filter-listlist**：フィルタを通過するすべてのパスのフラップ統計情報をクリアする。

- **clear ip bgp flap-statisticsA.B.C.D m.m.m.m**：単一エントリのフラップ統計情報をクリアする。

- **clear ip bgpA.B.C.Dflap-statistics**：ネイバーからのすべてのパスのフラップ統計情報をクリアする。

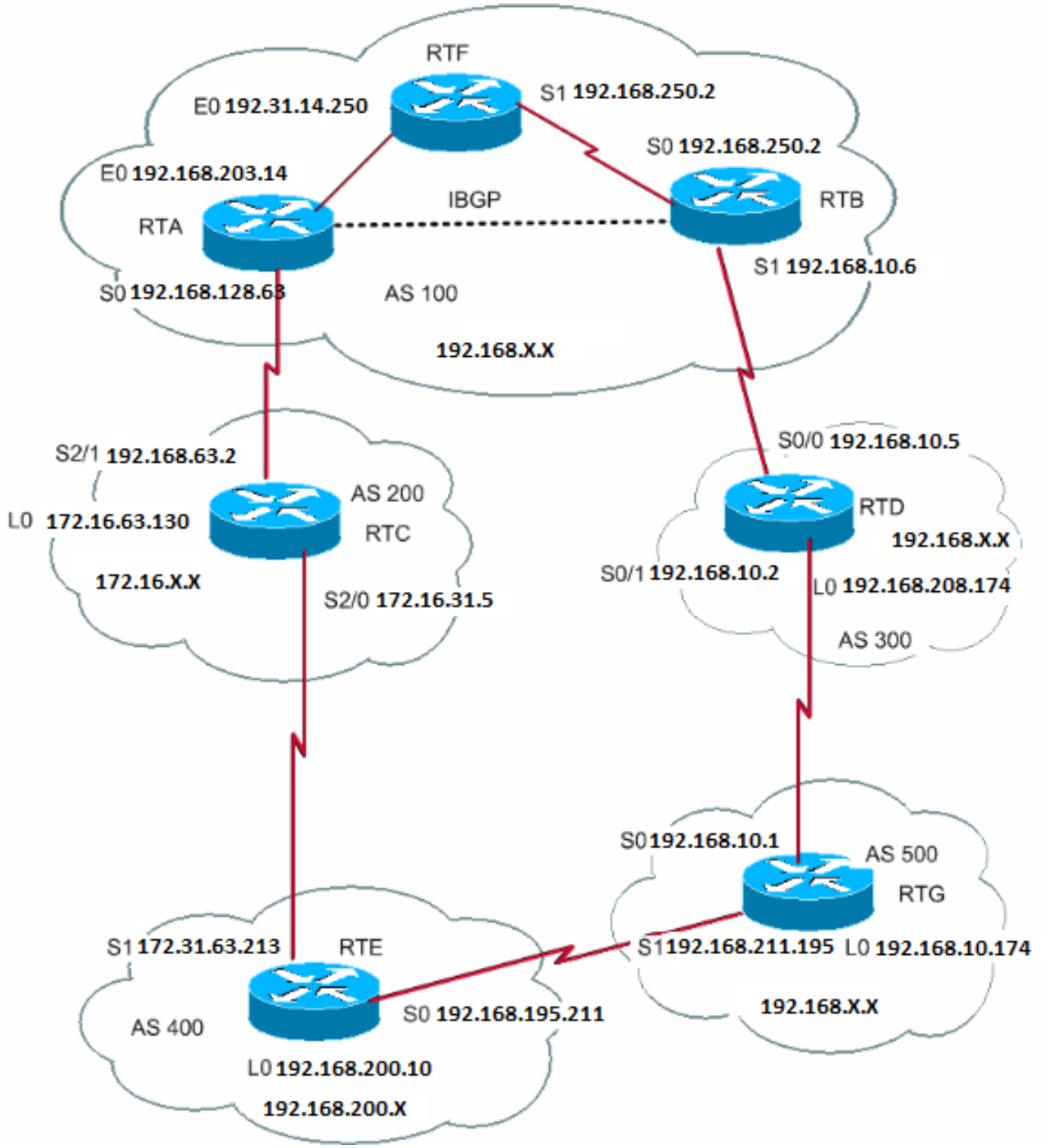
BGP によるパスの選択方法

BGP 属性と用語について十分に理解できたところで、『BGP でベスト パスを選択するアルゴリズム』を参照してください。

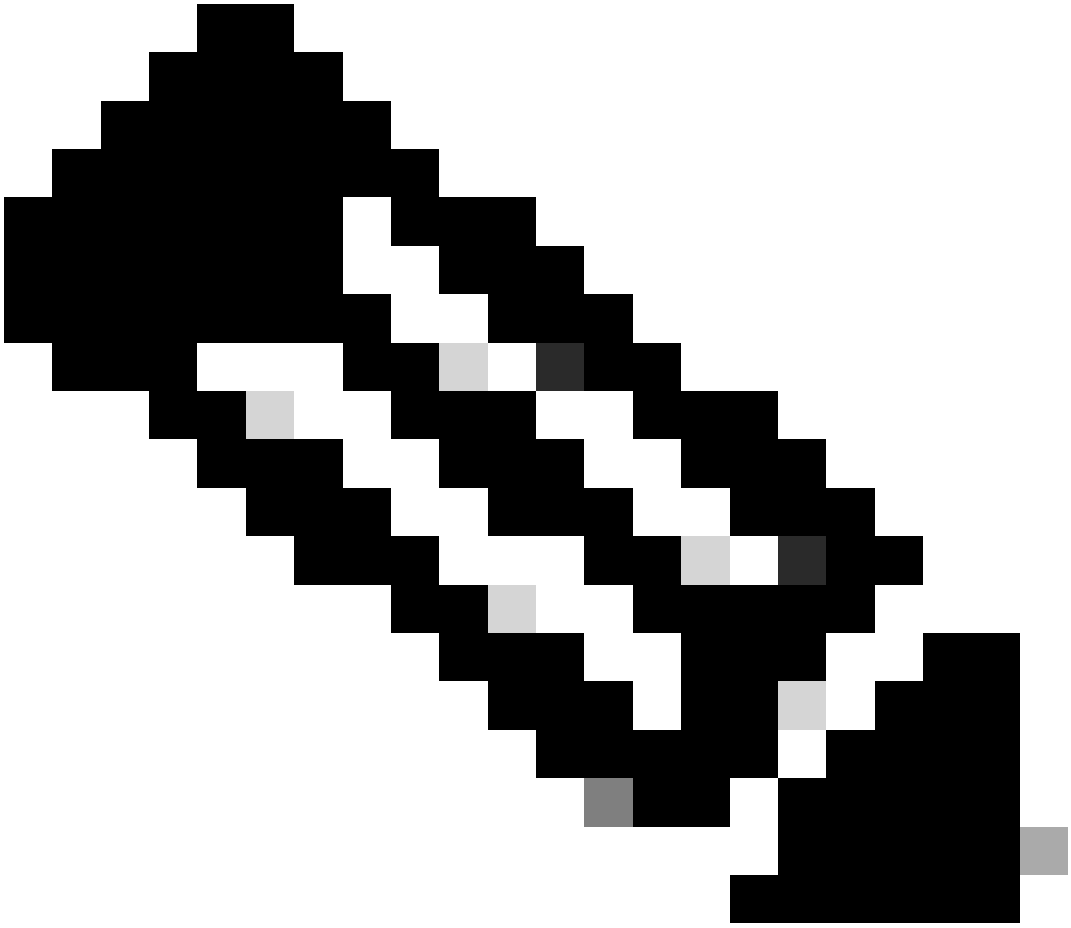
BGP ケース スタディ 5

実際の設計例

この項の設計例では、シスコ ルータに実際に表示される設定テーブルとルーティング テーブルを示します。



ここでは、この設定を段階的に構築する方法と、途中で発生し得る問題を示します。AS が eBGP 経由で 2 つの ISP に接続している場合は、ルートを適切に制御するために、AS 内で iBGP を実行してください。この例では、AS100 内の RTA と RTB の間で iBGP を実行し、IGP として OSPF を実行します。2 つの ISP (AS200 と AS300) に接続すると仮定した場合、すべてのルータの最初の設定は次のようになります。



注：これらの設定は、最終的な設定ではありません。

```
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
```

ネットワークをアドバタイズするには、常にnetwork コマンドを使用するか、BGPにスタティックエントリを再配布してください。これは、BGP に IGP を再配布する方法よりも推奨されます。この例では、network コマンドを使用してBGPにネットワークを注入しています。

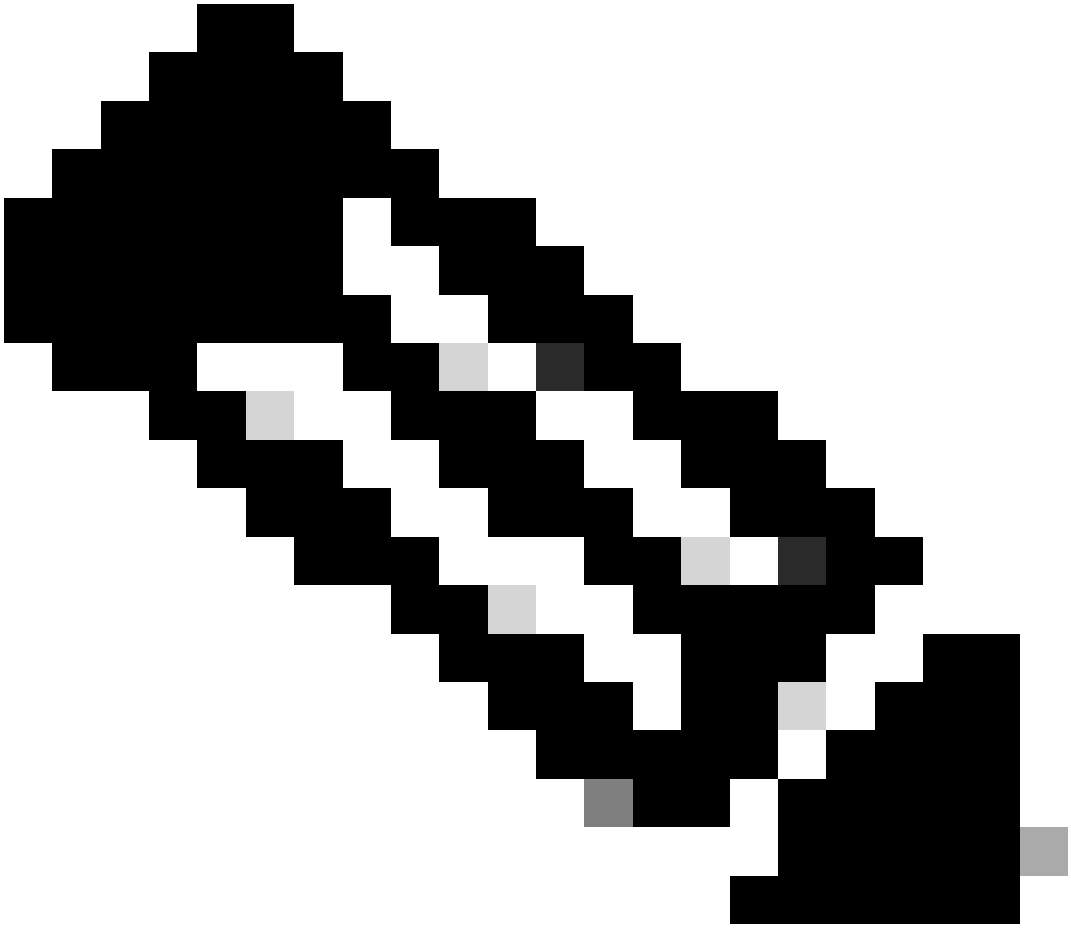
ここでは、RTB と RTD との間にリンクが存在しない場合と同様に、RTB シャットダウンの s1 インターフェイスから始めます。RTB の BGP テーブルは次のとおりです。

```
<#root>

RTB#

show ip bgp BGP

table version is 4, local router ID is 192.168.250.2 Status
codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
     Network          Next Hop          Metric LocPrf Weight Path
*i172.31.10.0      172.31.63.250           0    100      0 200 i
*i192.168.10.0     172.31.63.250                100      0 200 400 500
300 i
*i192.168.211.10     172.31.63.250                100      0 200 400 500 i
*i192.168.10.10     172.31.63.250                100      0 200 400 i
*>i192.168.203.13    192.168.203.250          0    100      0 i
*>i192.168.250.14    192.168.203.250          0    100      0 i
*>192.168.250.15     0.0.0.0                  0         32768 i
```

このテーブルでは、次の表記が使用されます。

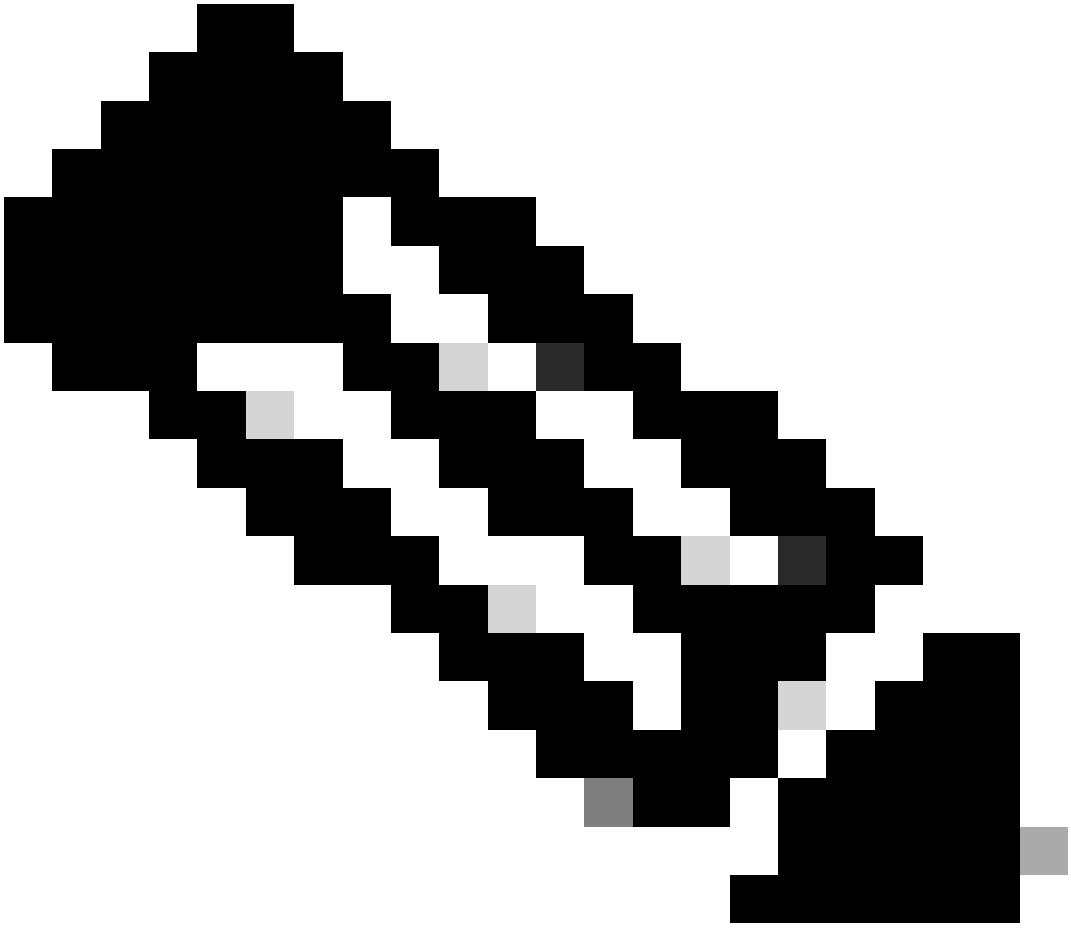
- 最初のアニメーション：エントリがiBGPピアを介して学習されたことを示します。
- 末尾のアニメーション：パス情報の起点(origin)がIGPであることを示します。
- Pathinformation：この情報は直感的に理解できます。たとえば、ネットワーク 172.31.10.0 はネクスト ホップが

172.31.63.250 のパス 200 を介して学習されることがわかります。



注：ローカルで生成されたエントリ（192.168.250.15など）のネクストホップは0.0.0.0です。

- > 記号：BGP が最適なルートを選択したことを示します。BGP は、『BGP でベスト パスを選択するアルゴリズム』ドキュメントで説明されている決定手順を使用します。BGP は宛先に到達するためのベスト パスを 1 つ選択し、そのパスを IP ルーティング テーブルにインストールして他の BGP ピアにパスをアドバタイズします。



注:Next Hop属性に注目してください。RTB は、iBGP に伝達された eBGP ネクスト ホップ 172.31.63.250 を介して 172.31.10.0 に関する情報を取得します。

IP ルーティング テーブルを見てみましょう。

<#root>

RTB#

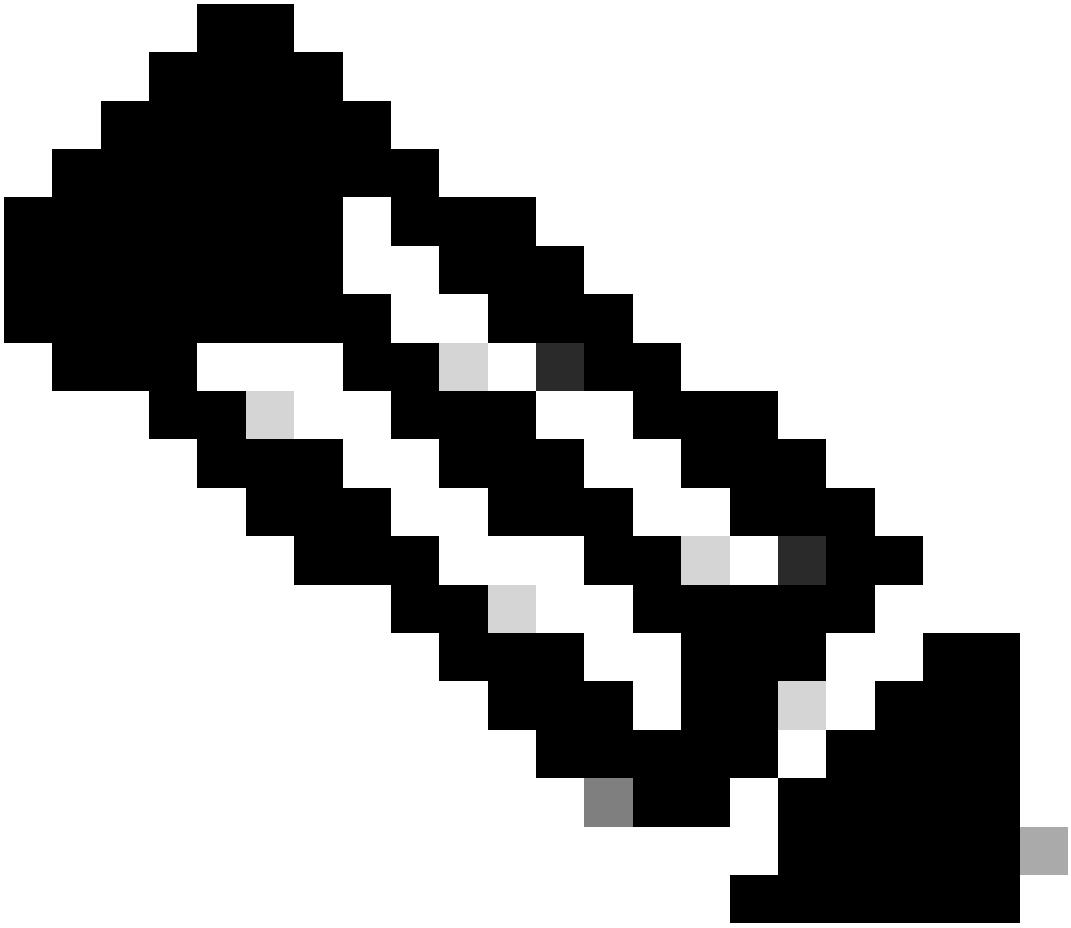
show ip route

```
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
```

一見したところ、BGP エントリはいずれもルーティング テーブルに到達していません。ここには 2 つの問題があります。

最初の問題は、これらのエントリのネクスト ホップ 172.31.63.250 が到達不能であるということです。IGP (OSPF) 経由でネクスト ホップに到達する方法がないため、RTB は OSPF 経由で 192.168.213.63 について学習していません。OSPFをRTAのs0インターフェイスで実行してパッシブにすると、RTBはネクストホップ172.31.63.250への到達方法を認識するようになります。この場合の RTA の設定は次のとおりです。

```
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:RTAとRTBの間でbgp nexthop selfコマンドを発行すると、ネクストホップを変更できます。

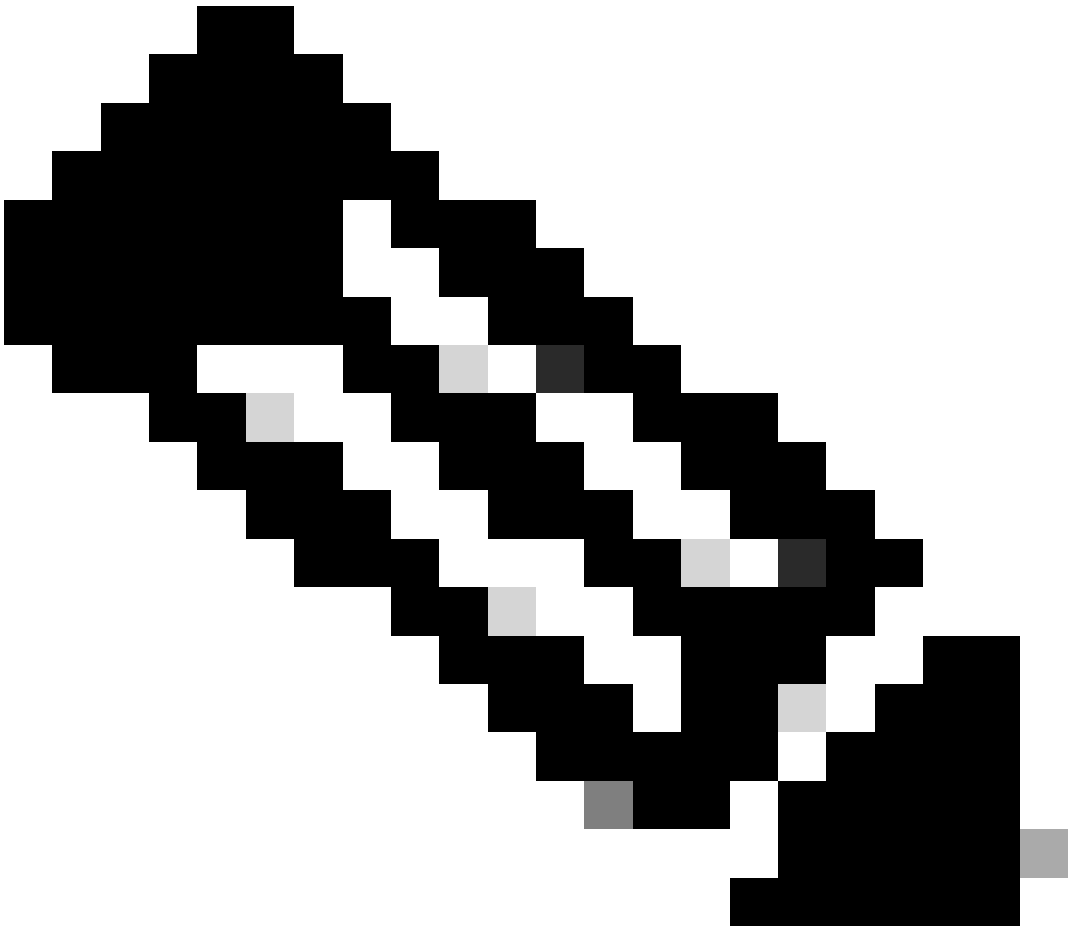
RTB の新しい BGP テーブルは次のように表示されます。

<#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:BGPがネクストホップに到達できることを意味する>が、すべてのエントリに表示されています。

ルーティング テーブルを見てみましょう。

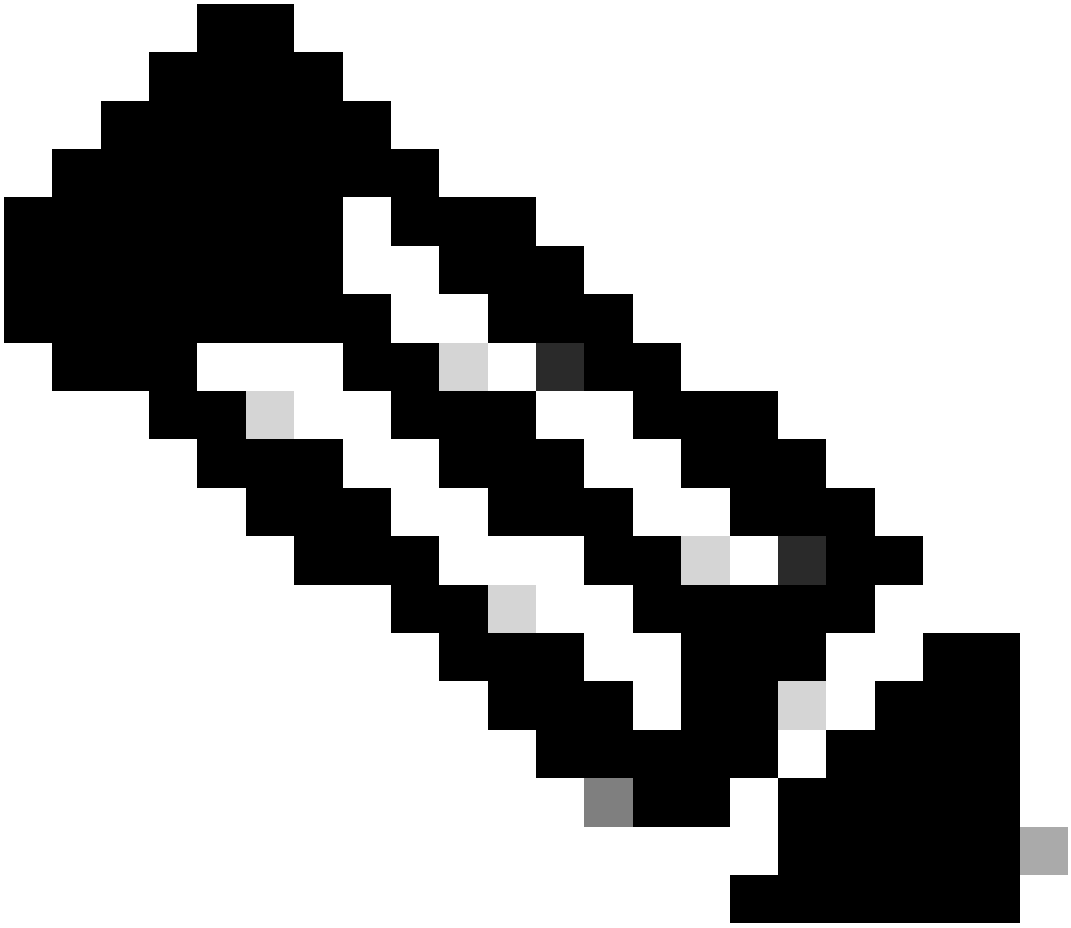
<#root>

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
```

2 つ目の問題は、ルーティング テーブルに BGP エントリがまだ表示されないことです。唯一変化したのは、192.168.213.63 が OSPF 経由で到達可能になっている点です。これは同期の問題です。IGP と同期されていないため、BGP はこれらのエントリをルーティング テーブルに挿入することも、BGP アップデートで送信することもありません。



注:BGPをOSPFにまだ再配布していないため、RTFはネットワーク192.168.10.0および192.168.211.10を認識していません。

このシナリオでは、同期をオフにするとエントリがルーティング テーブルに表示されますが、接続は切断されたままです。

RTB で同期をオフにした場合は、次のように表示されます。

<#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
```

ルーティング テーブルは問題ないように見えますが、これらのネットワークに到達する方法がありません。以下のように、中央の RTF はネットワークへの到達方法を認識していません。

```
    <#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
```

この状況で同期をオフにしても、問題はそのまま残ります。また、同期は後で他の問題を扱う際に必要です。RTA で OSPF に BGP をメトリック 2000 で再配布します。

```
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
```

ルーティング テーブルは次のように表示されます。

```
    <#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
```

OSPF は iBGP より距離が短いため、BGP エントリは表示されなくなりました。OSPF の距離が 110 であるのに対し、iBGP の距離は 200 です。

RTA が 192.168.250.15 をアドバタイズできるように、RTA で同期をオフにします。この操作が必要な理由は、RTA がマスクの違いにより OSPF と同期されないからです。RTB が 192.168.203.13 をアドバタイズできるように、RTB の同期をオフのままにします。RTB でこの操作が必要な理由も上記と同様です。

次に、RTB の s1 インターフェイスを表示してルートを確認します。また、RTB のシリアル 1 で OSPF をイネーブルにして、パッシブに設定します。この手順により、RTA は IGP 経由でネクスト ホップ 192.168.10.5 に関する情報を取得できるようになります。この手順を行わないと、ネクスト ホップ 192.168.10.5 に到達するために eBGP 経由で別のルートを通ることになるため、ルーティング ループが発生します。RTA と RTB の新しい設定を次に示します。

```
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 テーブルは次のように表示されます。

```
    <#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
```

2 つの異なる ISP (AS200 と AS300) と通信するネットワークを設計するには、複数の方法があります。1 つは、プライマリ ISP とバックアップ ISP を設定する方法です。いずれかの ISP からの部分ルートと、両方の ISP へのデフォルト ルートを学習できます。この例では、AS200 から部分ルートを受信し、AS300 からはローカル ルートのみを受信します。RTA と RTB の両方が OSPF へのデフォルト ルートを生成しますが、メトリックがより小さい RTB が優先されます。このように、2 つの ISP 間で発信トラフィックのバランスを調整できます。

RTA から発信されたトラフィックが RTB 経由で戻る場合は、非対称が発生している可能性があります。この状況は、2 つの ISP との通信時に同じ IP アドレス プール (同じメジャー ネット) を使用している場合に発生することがあります。集約により、外部からは AS 全体が 1 つのエンティティとして認識される場合があります。ネットワークへのエントリ ポイントは RTA 経由の場合も RTB 経由の場合もあります。インターネットへのポイントが複数存在するにもかかわらず、すべての着信トラフィックがシングル ポイント経由で AS に到達していることがあります。この例では、2 つの ISP との通信に 2 つのメジャー ネットを使用しています。

非対称の原因としてもう 1 つ考えられるのは、アドバタイズされた AS に到達するパスの長さが異なることです。特定の宛先には、いずれかのサービス プロバイダーがもう一方よりも近いはずです。例では、AS400 からのトラフィックは、よりパスが短い RTA 経由で常にネットワークに到達します。この決定を操作することもできます。この場合は、set as-path prepend コマンドを使用してアップデートにパス番号を付加すると、パスをより長く見せることができます。ただし、ローカル プリファレンス、メトリック、重みなどの属性が使用されている場合は、AS400 によって出力点が AS200 に設定されている可能性があります。この場合、対処法はありません。

すべてのルータの最終的な設定は次のようになります。

```
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
```

RTA では、AS200 から到達するルートのローカル プリファレンスが 200 に設定されています。また、ネットワーク 172.31.200.200 はデフォルト候補として選択されています。ip default-network コマンドを使用すると、デフォルトを選択できます。

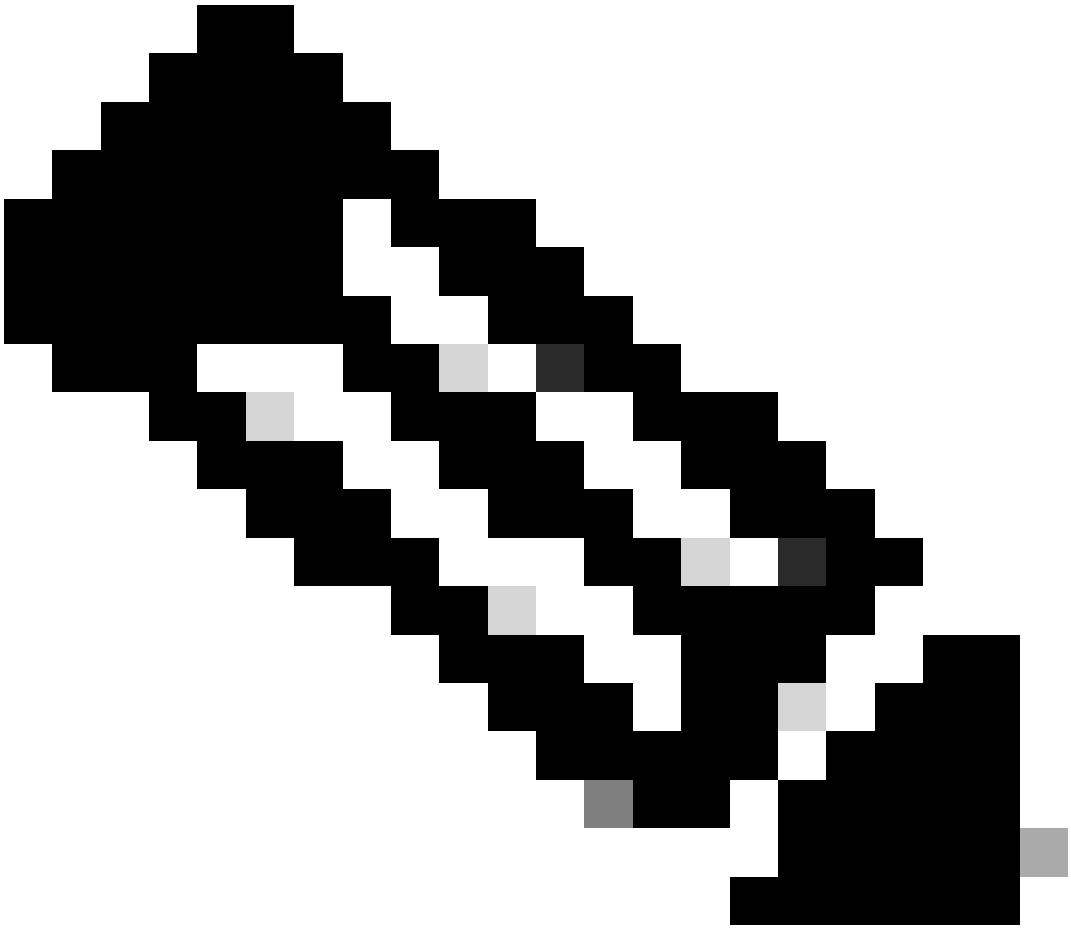
さらにこの例では、**default-information originate コマンドを OSPF に使用して、OSPF ドメイン内にデフォルト ルートをインジェクトしています。**また、Intermediate System-to-Intermediate System プロトコル（IS-IS プロトコル）と BGP にもこのコマンドを使用しています。RIP については、設定を追加しなくても 0.0.0.0 が RIP に自動的に再配布されます。IGRP および EIGRP では、BGP

が IGRP と EIGRP に再配布された後で、デフォルト情報が IGP ドメインにインジェクトされます。IGRP と EIGRP によって、0.0.0.0 へのスタティック ルートを IGP ドメインに再配布することもできます。

```
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
```

RTB

では、AS300 から到達するアップデートのローカル プリファレンスが 300 に設定されています。この値は RTA から到達する iBGP アップデートのローカル プリファレンス値よりも大きいため、AS100 は AS300 のローカル ルートとして RTB を選択します。RTB のその他のルート（存在する場合）は、内部でローカル プリファレンス 100 で送信されます。この値は RTA から到達するローカル プリファレンス 200 よりも小さいため、RTAが優先されます。



注：アドバタイズしているのは、AS300ローカルルートだけです。^300$ に一致しないパス情報はすべてドロップされます。ローカル ルートと、ISP のカスタマーであるネイバー ルートをアドバタイズする必要がある場合は、^300_[0-9]* を使用してください。

AS300 のローカル ルートを示す正規表現の出力は次のとおりです。

<#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

RTCでは、172.31.10.0/16を集約し、AS100に注入する特定のルートを指定します。ISP によってこのタスクの実行が拒否される場合は、AS100 の着信側でフィルタリングを行う必要があります。

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
```

コミュニティフィルタリングの使用方法のデモは、RTGで行います。RTDに対する192.168.211.0アップデートに no-export コミュニティを追加します。これにより、RTD はそのルートを RTB に エクスポートしません。ただしこの場合、RTB はいずれにしてもこれらのルートを受け入れません。

```
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 は 172.31.200.200/16 を集約します。RTA、RTF、および RTB の最終的な BGP テーブルとルーティング テーブルは次のとおりです。

```
<#root>

RTA#
```

show ip bgp

```
BGP table version is 21, local router ID is 192.168.203.250
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 172.31.10.0      172.31.63.250           0    200      0 200 i
*>i192.168.10.0     192.168.10.5            0    300      0 300 i
*> 172.31.200.200/16   172.31.63.250               200      0 200 400 i
*> 192.168.203.13     0.0.0.0                  0         32768 i
*> 192.168.250.14     0.0.0.0                  0         32768 i
*>i192.168.250.15     192.168.250.2             0    100      0 i

RTA#
```

show ip route

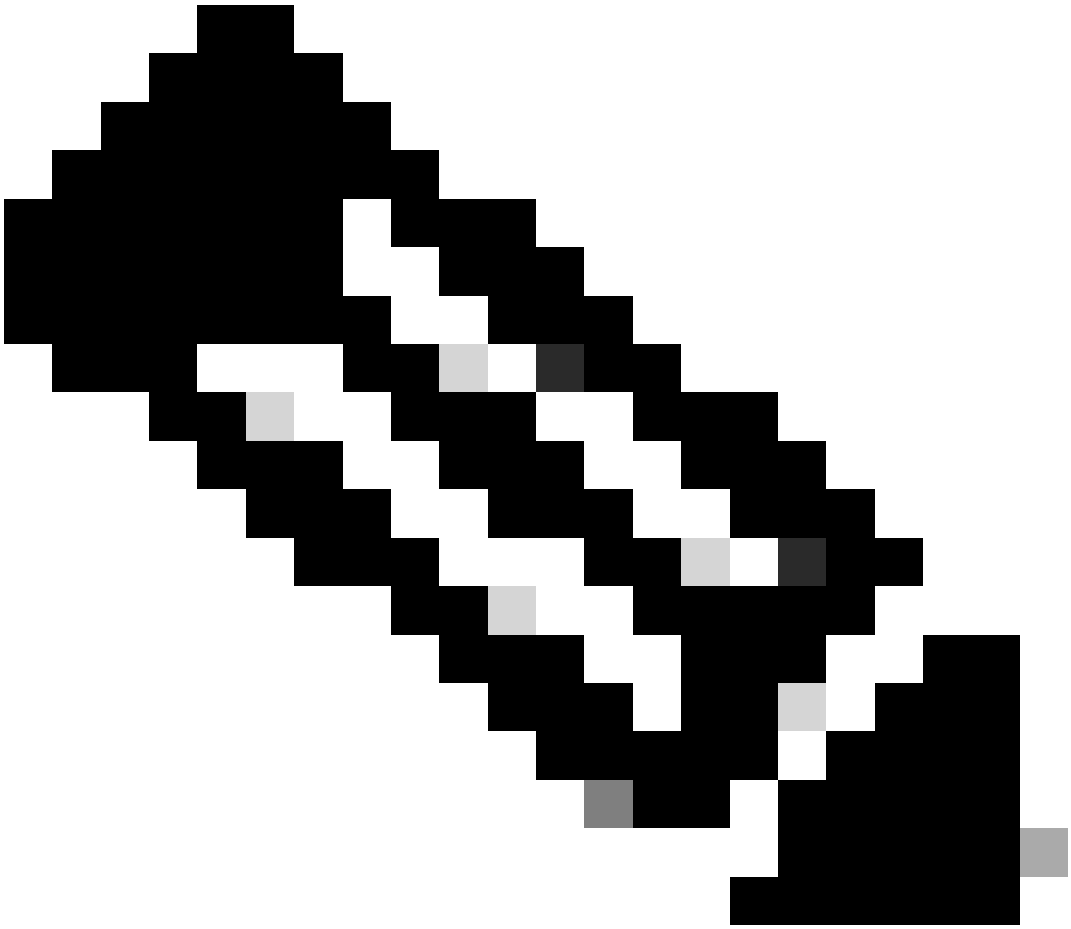
```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
```

```
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * -
candidate default

Gateway of last resort is 172.31.63.250 to network 172.31.200.200

     192.168.10.0 is variably subnetted, 2 subnets, 2 masks
O E2    192.168.10.0 255.255.255.0
           [110/1000] via 172.31.14.250, 00:41:25, Ethernet0
O       192.168.10.4 255.255.255.252
           [110/138] via 172.31.14.250, 00:41:25, Ethernet0
C    192.168.203.13 is directly connected, Loopback0
     192.168.250.15 is variably subnetted, 3 subnets, 3 masks
O       172.16.15.2500 255.255.255.255
           [110/75] via 172.31.14.250, 00:41:25, Ethernet0
O       192.168.250.15 255.255.255.252
           [110/74] via 172.31.14.250, 00:41:25, Ethernet0
B       192.168.250.15 255.255.255.0 [200/0] via 192.168.250.2, 00:41:25
C    192.168.250.14 is directly connected, Ethernet0
     172.31.10.0 is variably subnetted, 2 subnets, 2 masks
B       172.31.10.0 255.255.0.0 [20/0] via 172.31.63.250, 00:41:26
C       192.168.213.63 255.255.255.252 is directly connected, Serial0
O*E2 0.0.0.0/0 [110/1000] via 172.31.14.250, Ethernet0/0
B*   172.31.200.200 255.255.0.0 [20/0] via 172.31.63.250, 00:02:38

RTF#
```

show ip route

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

Gateway of last resort is 192.168.250.2 to network 0.0.0.0

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

```
O       192.168.213.63 255.255.255.252
           [110/74] via 192.168.203.14, 01:12:11, Ethernet0
O E2 172.31.200.200 255.255.0.0 [110/2000] via 192.168.203.14, 00:03:47, Ethernet0
O*E2 0.0.0.0 0.0.0.0 [110/1000] via 192.168.250.2, 00:03:33, Serial1
```



注:RTFのルーティングテーブルでは、AS300に対してローカルなネットワーク（192.168.10.0など）に到達する方法は、RTBを経由していることがわかります。その他の既知のネットワーク（172.31.200.200 など）に到達するルートは、RTA を経由します。ラスト リゾートのゲートウェイは RTB に設定されています。RTB と RTD の間の接続に問題が発生すると、RTA がアドバタイズしたメトリック 2000 のデフォルトが使用されます。

<#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
```

関連情報

- **BGP：FAQ**
- **PIX ファイアウォールを経由する BGP の設定例**
- **HSRP を使用してマルチホーム BGP ネットワークで冗長性を実現する方法**
- **Cat6000 MSFC上での単一ルータモードの冗長性とBGPの設定**
- **最適ルーティングの実現と BGP メモリ消費の削減**
- **一般的なBGP問題のトラブルシューティング**
- **BGPスキャナまたはルータプロセスが原因で発生するCPU高使用率のトラブルシューティング**
- **シングルホームおよびマルチホーム環境における、BGPを使用したロードシェアリングについて**
- **BGP に関するサポート ページ**
- **シスコテクニカルサポートとダウンロード**

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。