BGP ケース スタディ

目次

はじめに

このドキュメントでは、5 つの Border Gateway Protocol (BGP) ケース スタディを紹介します。

前提条件

要件

このドキュメントに関しては個別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。

表記法

ドキュメント表記の詳細は、『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、自律システム (AS) 間にループフリーの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

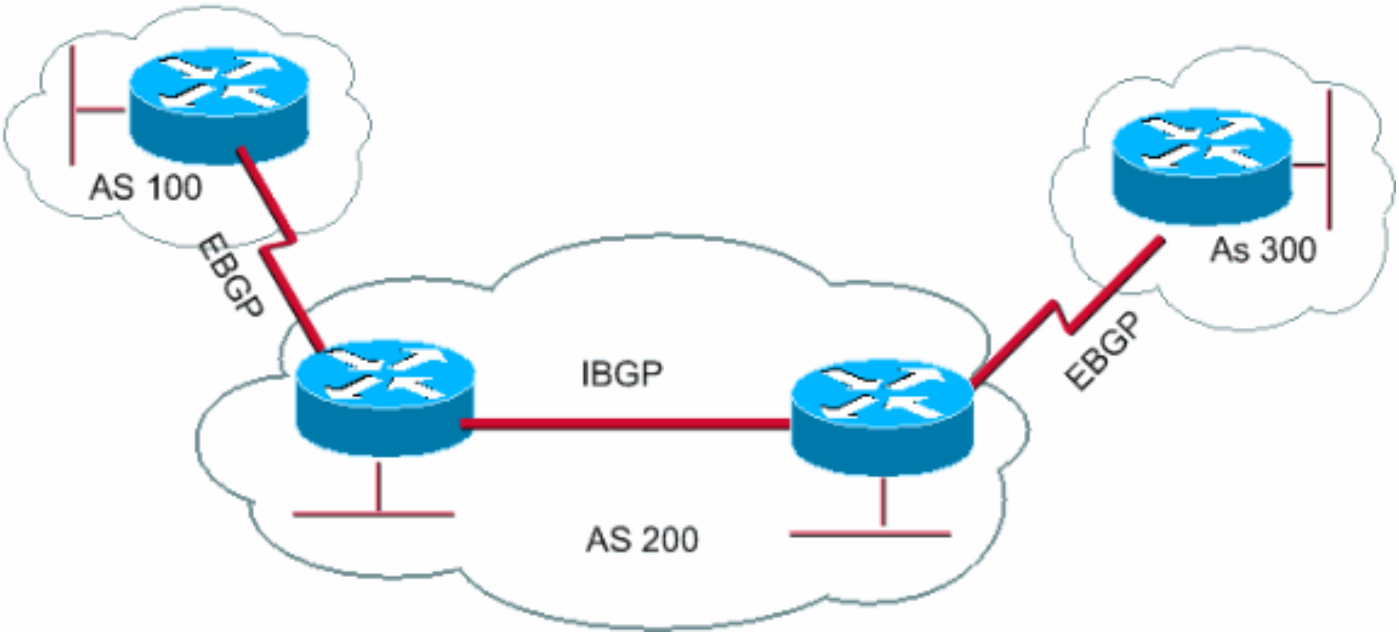
複数の BGP スピーカを含む AS は、他の AS の中継サービスとして機能できます。 この項の図に示すように、AS200 は AS100 と AS300 の中継 AS です。

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。



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。

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。

```
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。

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。 このプロセスについては、「BGP ネイバーの形成」で説明します。

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。

次の **neighbor** コマンドを発行して、TCP 接続を確立します。

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、相互に到達できることが*必要*です。 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 では、ping 発行元のルータが、**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強制されます。 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*** にはネイバー アドレスを指定します。

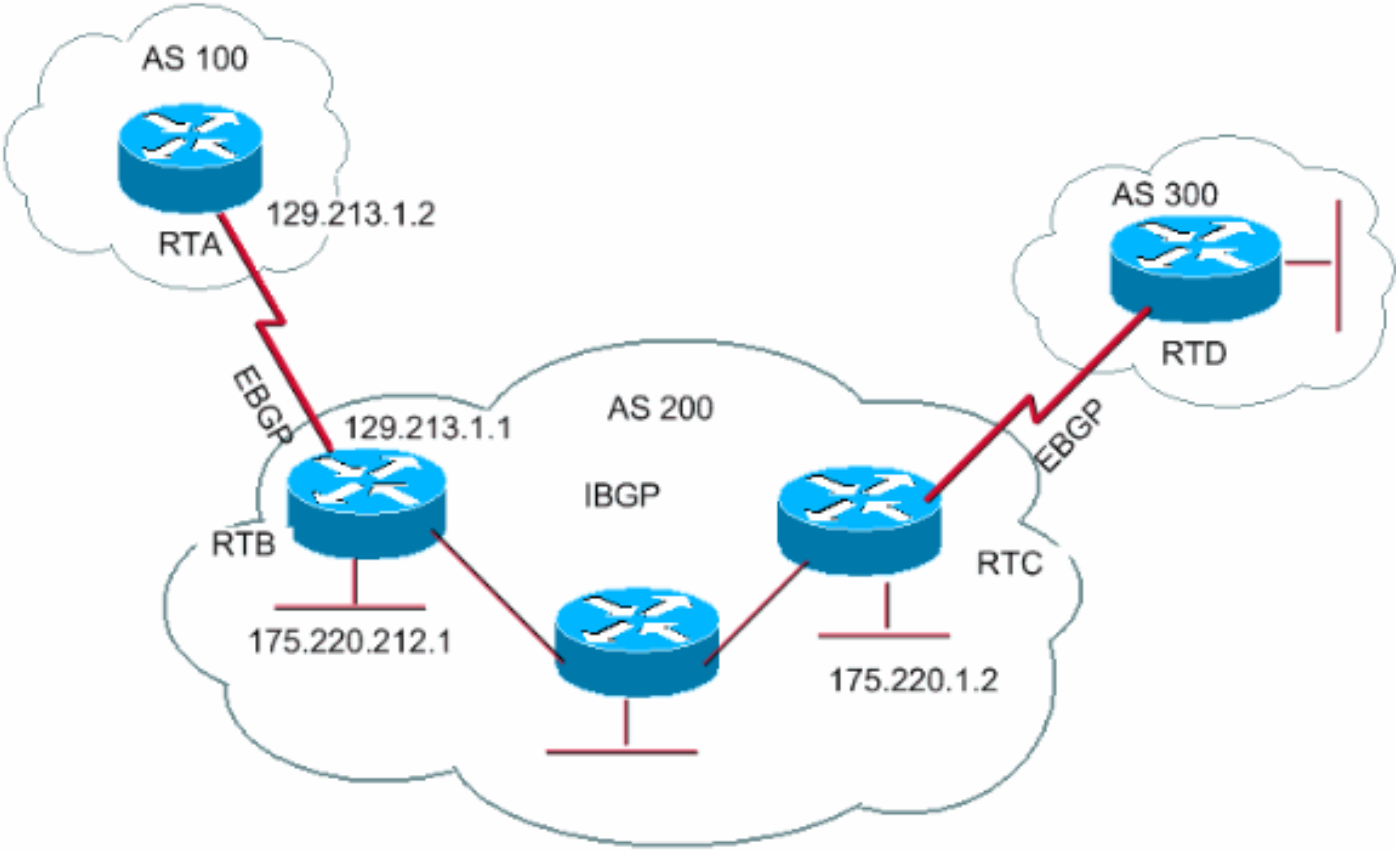
clear ip bgp *

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。

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

次に neighbor コマンド設定の例を示します。



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

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、IGP が動作していて、2 つのネイバーが相互に到達可能である必要があります。

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 以外である場合は、ピアが確立されていません。

注: 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。

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

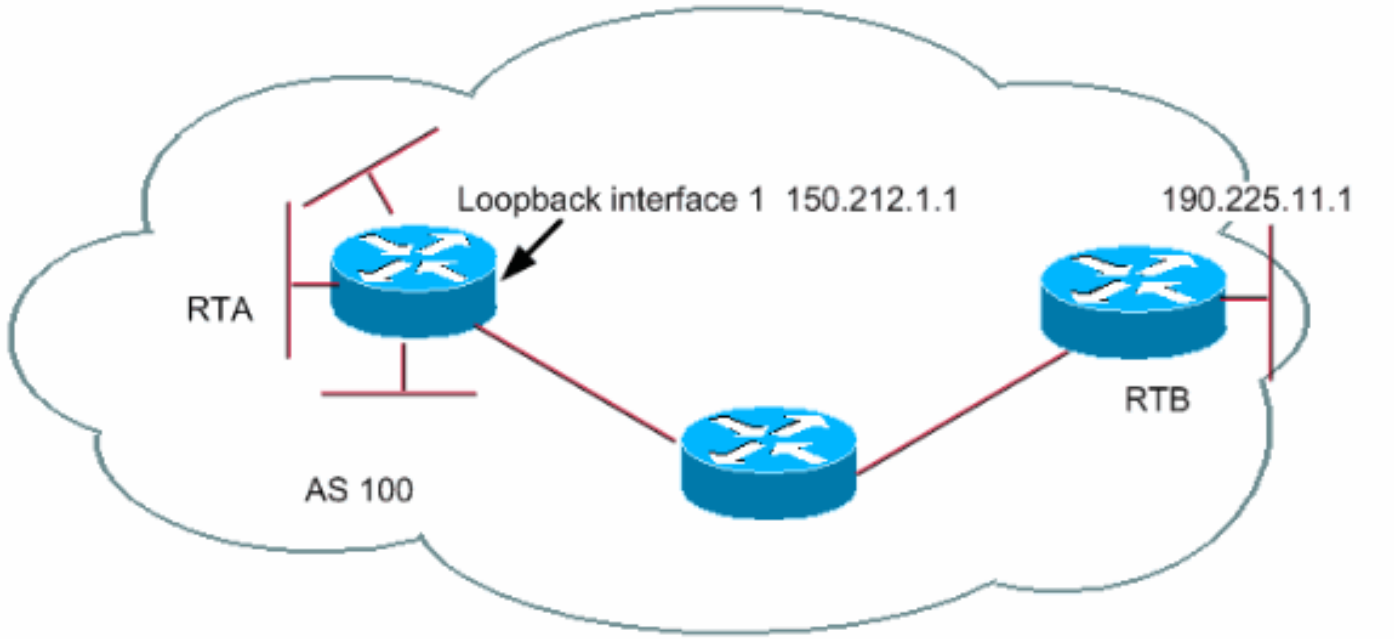
BGP とループバック インターフェイス

ネイバーの定義にループバック インターフェイスを使用する方法は、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。

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

次に、このコマンドの使用例を示します。



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

この例では、RTA と RTB は AS100 内で iBGP を実行します。 この **neighbor** コマンドの場合、RTB は RTA のループバック インターフェイス (150.212.1.1) 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 では **update-source interface-type interface-number** を追加します。その結果、コマンドは **neighbor 190.225.11.1 update-source loopback 1** になります。 このステートメントにより、BGP はネイバー 190.225.11.1 との通信時にループバック インターフェイスの IP アドレスを使用するように強制されます。

注: RTA は、RTB の物理インターフェイス IP アドレス (190.225.11.1) をネイバーとして使用しています。 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

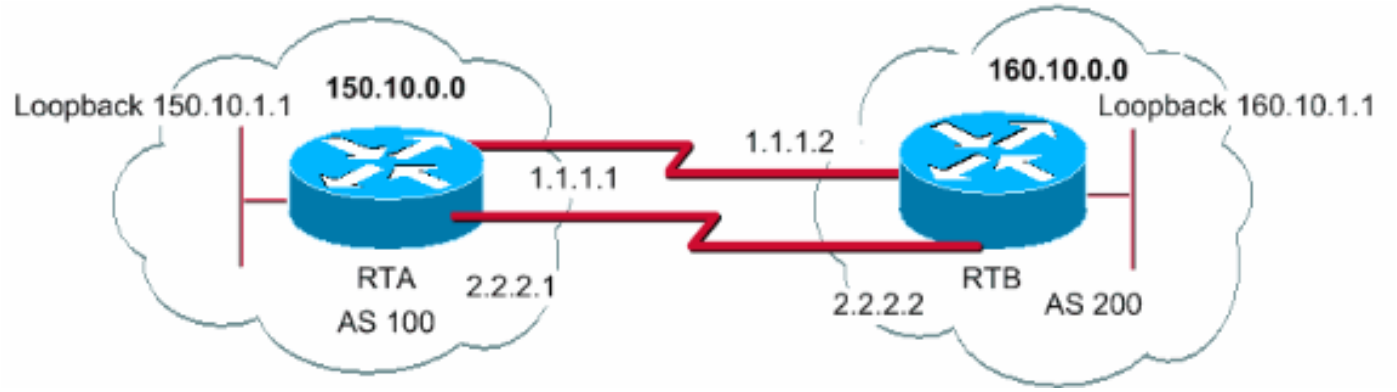
場合によっては、シスコ ルータは 2 つの外部ピアの直接接続を許可しないサード パーティ製ルータとの eBGP を実行できます。 この接続を実現するには、eBGP マルチホップを使用します。 eBGP マルチホップを使用すると、直接接続されていない 2 つの外部ピアをネイバー接続できます。 マルチホップは eBGP のみを対象としており、iBGP では使用されません。 次の例で eBGP マルチホップについて説明します。

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

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 (129.213.1.2) 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。 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 ルーティングの設定も必要です。

「eBGP マルチホップ (ロード バランシング)」の項の例で、パラレル回線上で eBGP を使用する場合に BGP でロード バランシングを実現する方法を示します。

eBGP マルチホップ (ロード バランシング)



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

これはループバック インターフェイス、update-source、および ebgp-multihop の使用例です。 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 160.10.1.1 に到達する方法は 2 つあります。 1 つは 1.1.1.2 経由のパスで、もう 1 つは 2.2.2.2 経由のパスです。 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。

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 **match** 基準を指定し、この **match** コマンドによる基準が満たされた場合の **set** アクションを set で指定します。

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 1.1.1.1 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。

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

この一致基準が満たされた場合、**permit** が指定されていれば、set アクションの指定ど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**承認も転送もされません**。

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 マップを使用した場合、IP アドレスで *match* コマンドを使用する際は、着信でフィルタリング**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。

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

match route-type

match tag

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 next-hop

set level

set local-preference

set metric

set metric-type

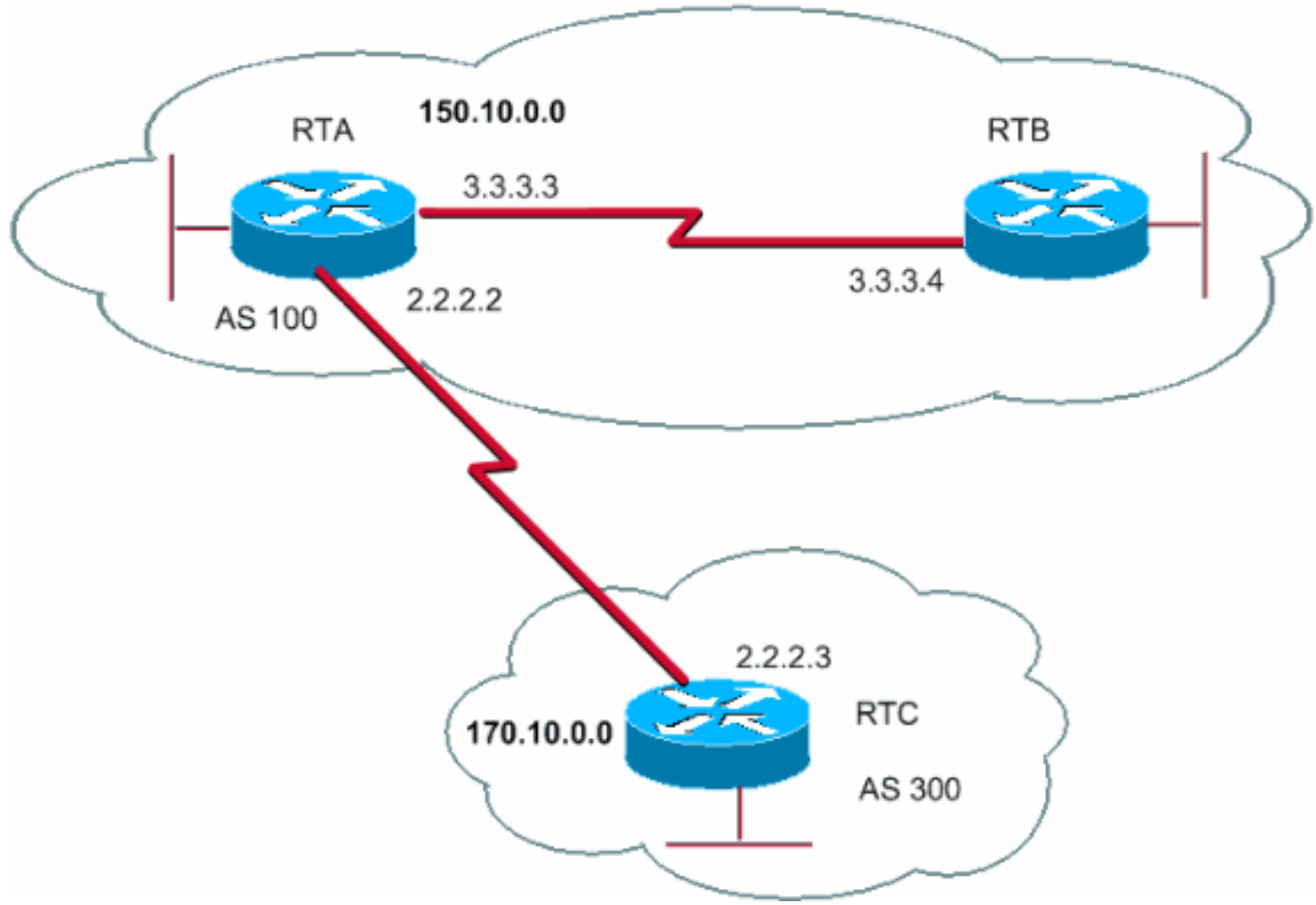
set next-hop

set origin

set tag

set weight

ルートマップの例をいくつか紹介します。



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 は、170.10.0.0 に関するルートをメトリック 2 で、他のすべてのルートをメトリック 5 で RTB に再配布するとします。この場合は次の設定を使用できます。

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

この例では、IP アドレス 170.10.0.0 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。 一致しない場合は、リスト内の次のルート マップに進みます。つまり、他のすべてのルートはメトリック 5 に設定されます。

注: どの match ステートメントにも一致しないルートは、 デフォルトでドロップされます。

例 2

例 1 の AS100 が 170.10.0.0 に関するアップデートを受け入れないようにするとします。 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。

```
match ip address 1.1.1.1
```

```
set metric 5
```

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。

```
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、 そのルータから発信すべきネットワークを 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。 設定可能な **network** コマンドのエントリ数は最大 200 です。

アドバタイズしようとするネットワークが、接続済み、スタティック、またはダイナミックに学習したネットワークとしてルータで認識されている場合は、**network** コマンドが機能します。

network コマンドの例は、次のとおりです。

```
network network-number [mask network-mask]
```

この例では、ルータ A が 192.213.0.0/16 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。

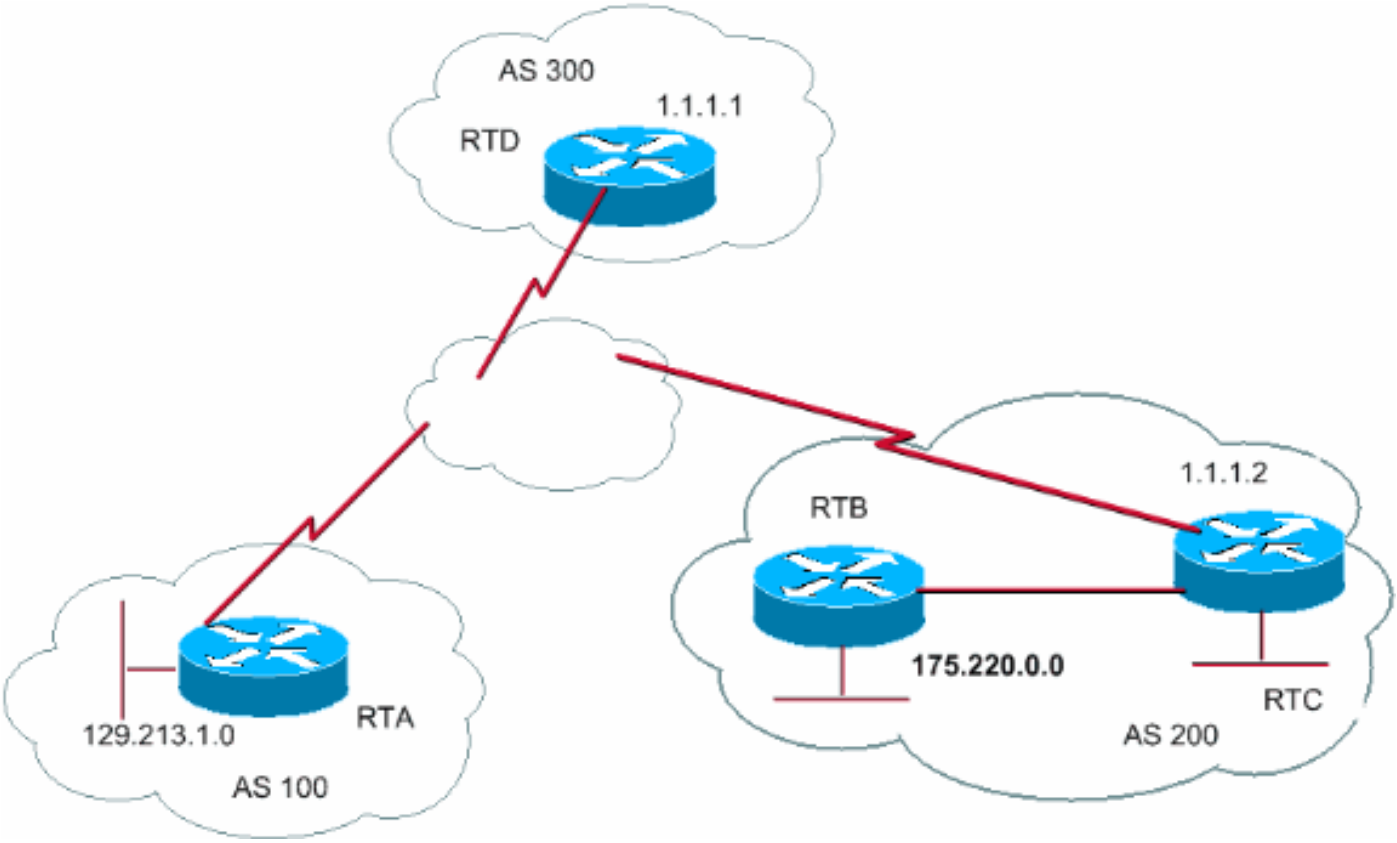
注: 一致するエントリがスタティック ルートによってルーティング テーブルに挿入されるため、ルータが 192.213.0.0 を生成するにはスタティック ルートが必要です。

再配布

network コマンドは、ネットワークを BGP 経由で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。 この再配布はすべての内部ルートを BGP にダンプするため不安が感じられることがあります。 実際、ルートの一部は BGP を介してすでに学習されている可能性があ

り、再度送信する必要はありません。 すべてのルートではなく、アドバタイズするインターネット専用のルートに送信するように、慎重にフィルタリングを適用してください。 次に例を示します。

RTA は 129.213.1.0 をアナウンスし、RTC は 175.220.0.0 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。

```
network network-number [mask network-mask]
```

代わりに再配布を使用する場合は次のとおりです。

```
network network-number [mask network-mask]
```

この再配布により、AS から 129.213.1.0 が発信されます。 ユーザは 129.213.1.0 の発信元ではなくなり、 AS100 が発信元になります。 したがって、AS によってそのネットワークが発信されないように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。

```
network network-number [mask network-mask]
```

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。 **router bgp** の下で **redistribute ospf 1** を発行するだけでは機能しません。 それぞれのルートを再配布するには、**internal**、**external**、**nssa-external** などの特定のキーワードが必要です。 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。

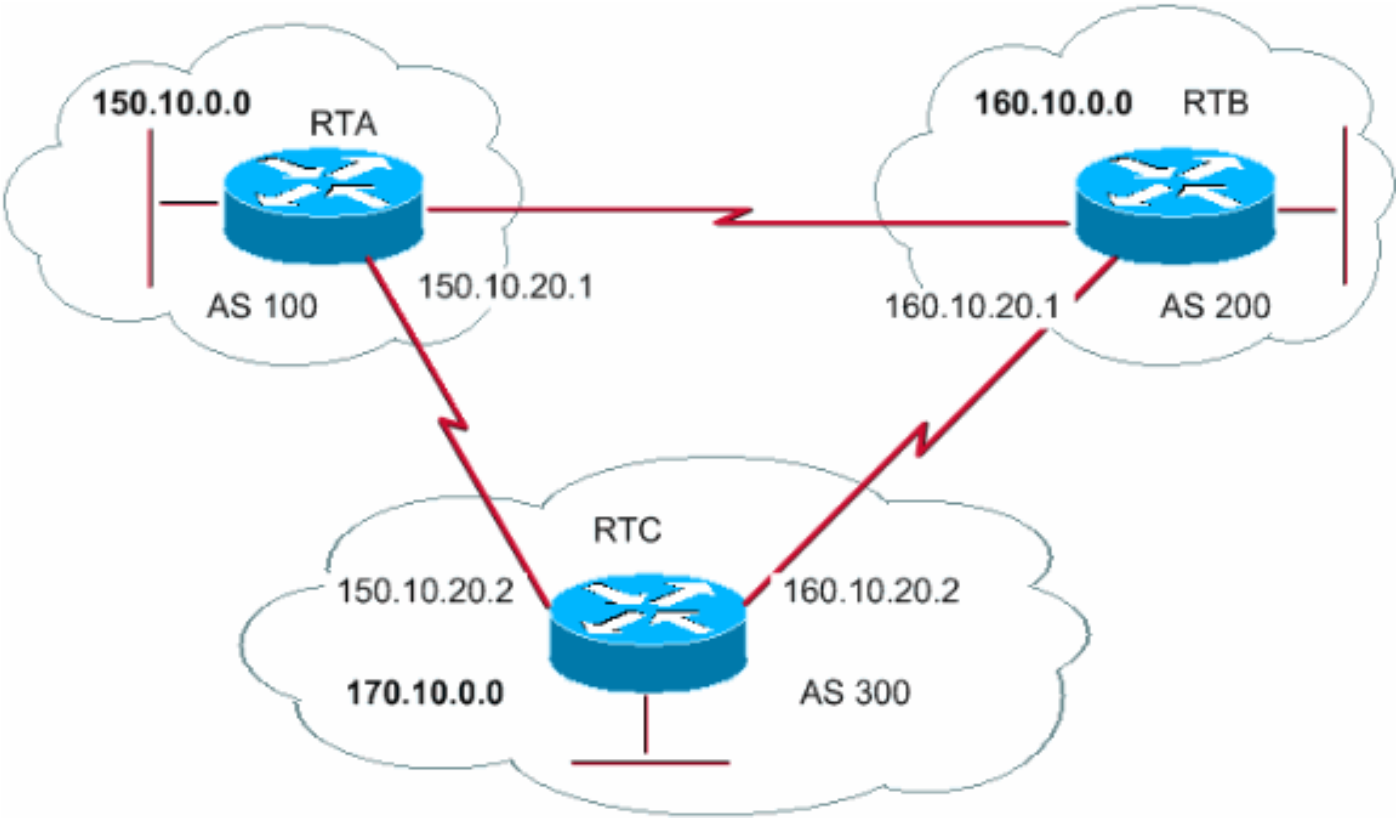
network *network-number* **[mask** *network-mask***]**

null0 インターフェイスは、パケットの無視を意味します。 パケットを受信して 175.220.0.0 よりも具体的な一致が見つかった（実際に存在する）場合、ルータはそのパケットを一致先に送信します。 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から生成されたルートでは、これらのネットワークの起点（origin）としてこの AS 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。

次に例を示します。



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

注: ネットワーク 150.10.0.0 およびネットワーク 160.10.0.0 は、AS100 と AS200 から RTC に到達するため、RTC がこれらのネットワークを伝えるだけでなく生成する場合を除いて、RTC にこれらのネットワークは必要ありません。 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 がこれらのルートの起点（origin）であることを示す追加のア

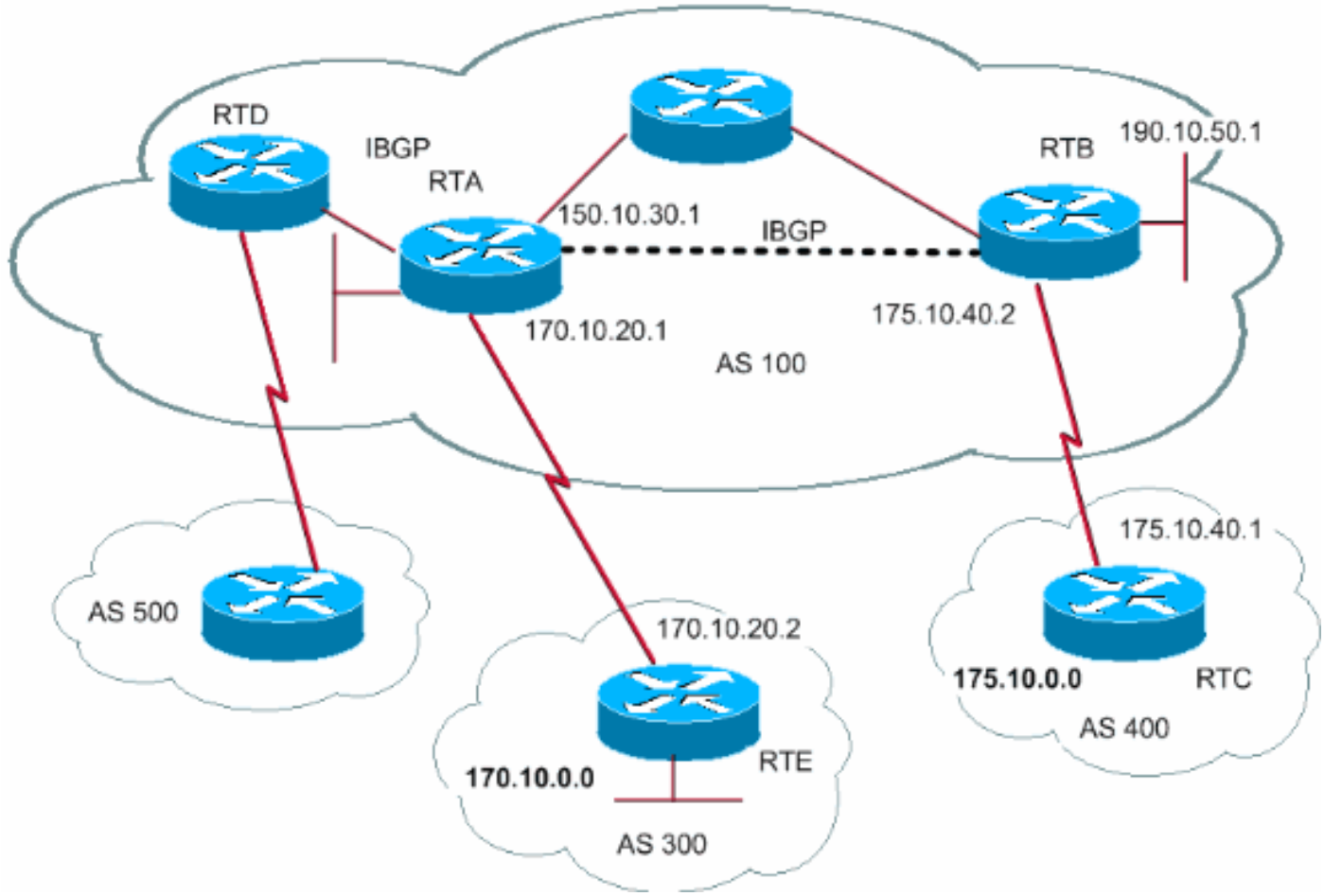
ドバタイズメントを付加することです。

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を念頭に置いてください。 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 150.10.0.0 を生成して AS300 にそのルートを送信します。 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 は送信元を AS100 にしたまま 150.10.0.0 を AS100 に渡します。 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

1 つの AS を他の AS 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 に再配布してから再び他の AS に再配布する方法でも同じ結果を得られますが、 iBGP では、より柔軟かつ効率的な方法を使用して AS 内で情報を交換できます。 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は、「ローカル プリファレンス属性」で詳しく説明します。



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

注: BGP スピーカは、自身の AS 内の他の BGP スピーカからアップデートを受信したときに（iBGP）、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。

上記の図では、RTA と RTB が 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。

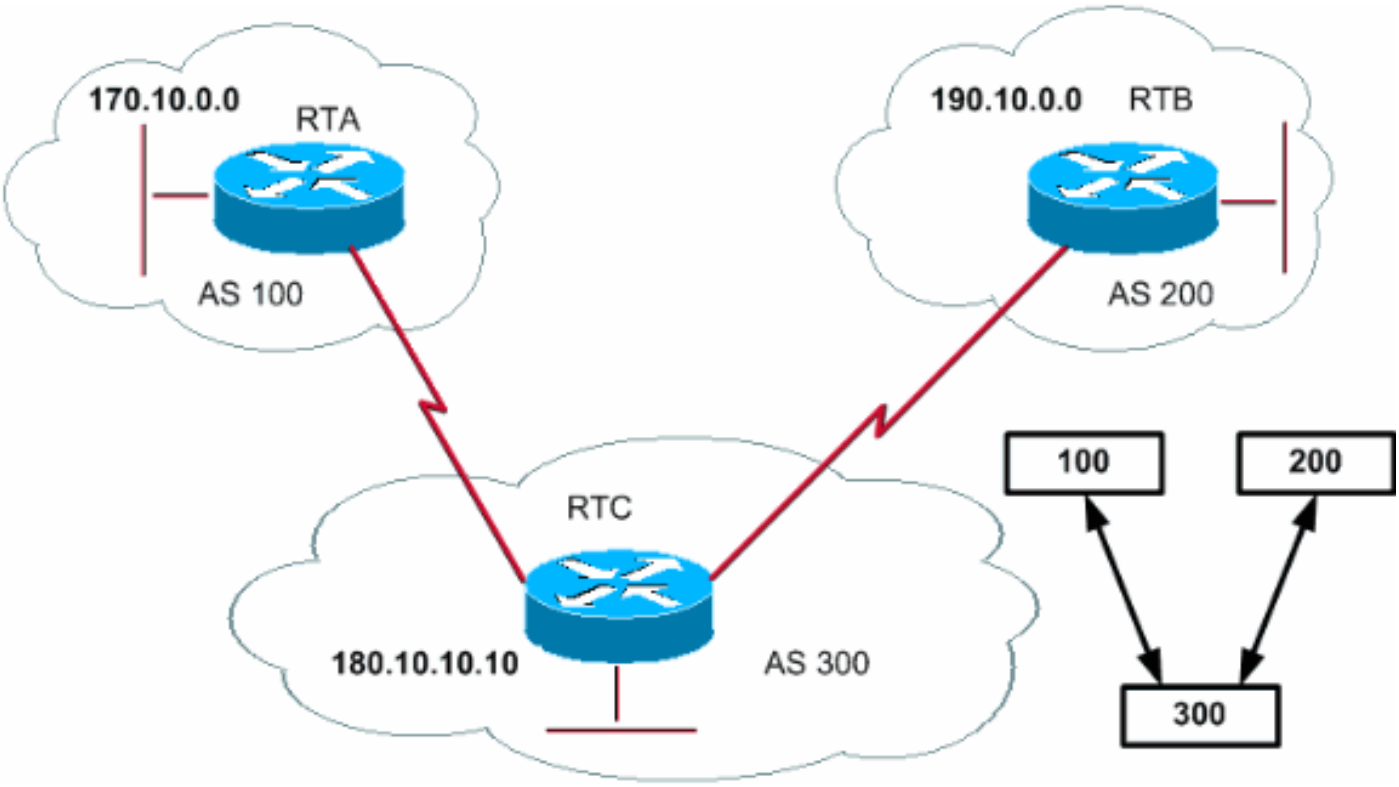
BGP がこの決定を行う際に基準となるのは、ネクスト ホップ、管理上の重み、ローカル プリファレンス、ルートの送信元、パス長、送信元コード、メトリックなどのさまざまな**属性**です。

BGP は常にベスト パスをネイバーに伝達します。 詳細については、BGP ベスト パス選択アルゴリズムを参照してください。

「BGP ケース スタディ 2」では、これらの属性とその使用方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。 このドキュメントの「CIDR 例 2 (as-set)」で AS_SET の例を示しています。

この項の例では、RTB は AS200 のネットワーク 190.10.0.0 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。 その結果、

190.10.0.0 が RTA に到達するときには、ネットワークに 2 つの AS 番号が付いています（最初が 200 で次が 300）。 RTA から 190.10.0.0 に到達するパスは（300、200）になります。

同じプロセスが 170.10.0.0 と 180.10.0.0 にも当てはまります。 RTB が 170.10.0.0 に到達するには、パス（300、100）を通る必要があります。 つまり、RTB は AS300、AS100 の順に通過します。 RTC の場合、190.10.0.0 に到達するにはパス（200）、170.10.0.0 に到達するにはパス（100）を通過する必要があります。

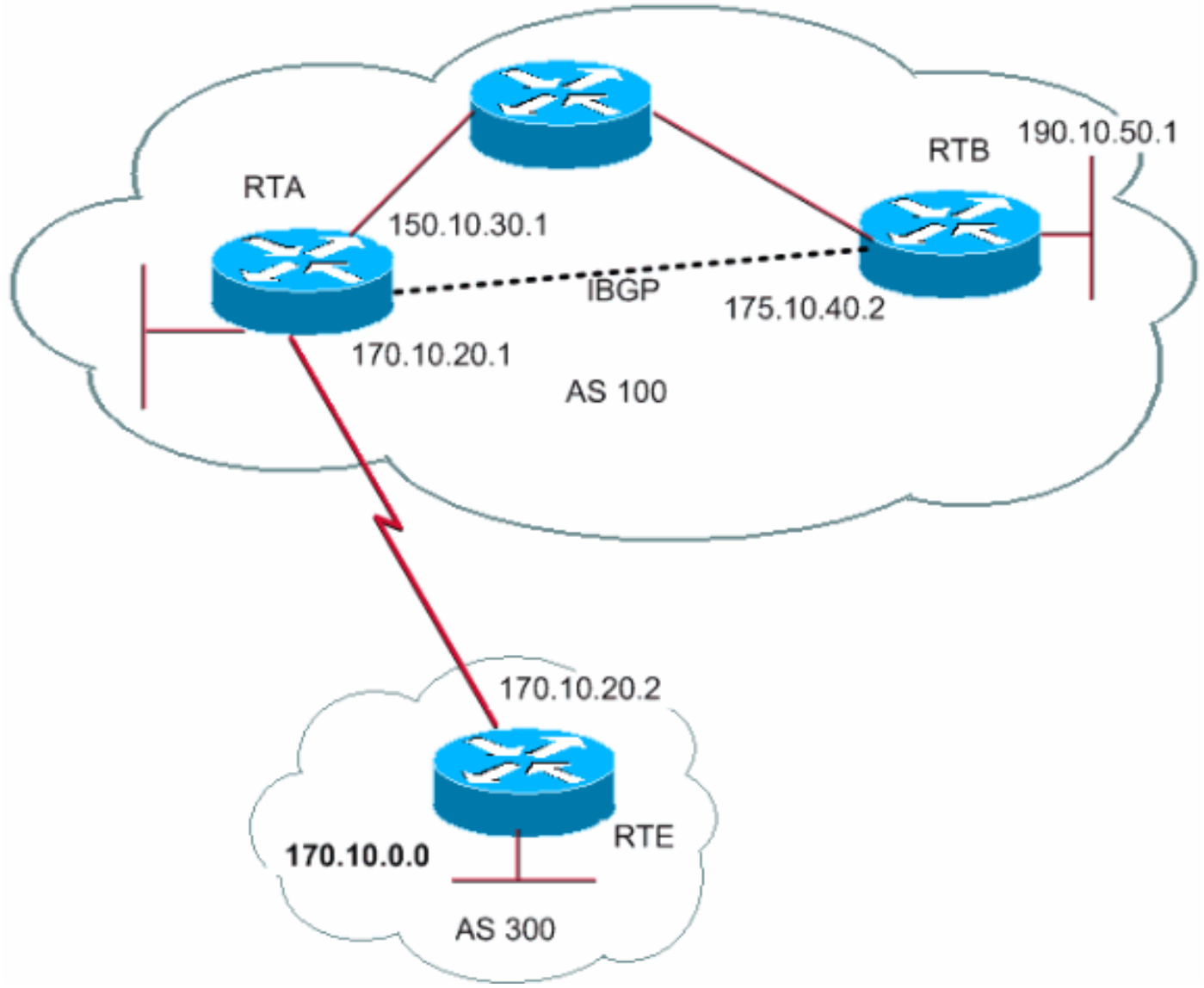
送信元属性

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： ネットワーク層到達可能性情報（NLRI）が送信元 AS の内部に存在します。 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 i 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 e は EGP を示しています。

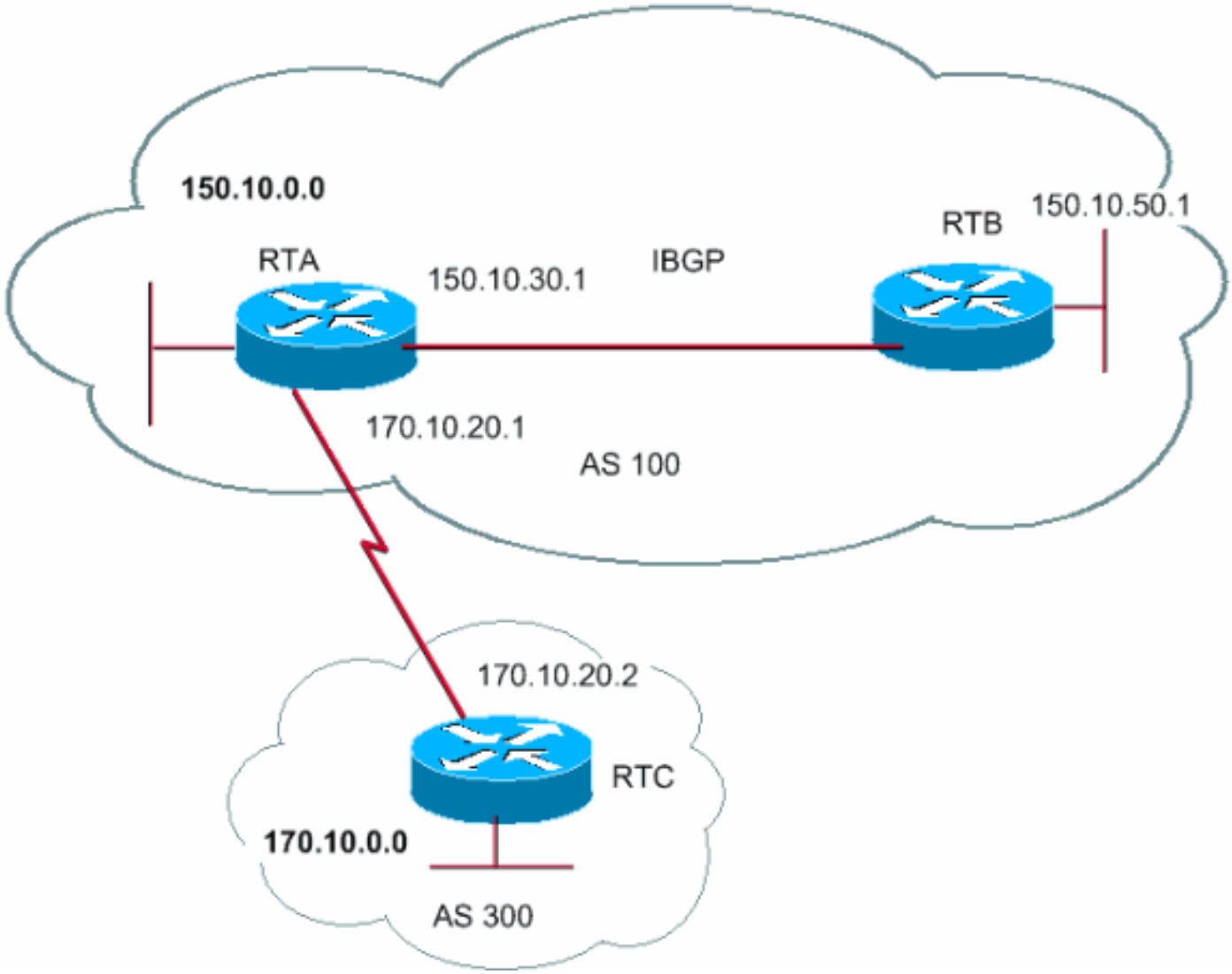
INCOMPLETE：NLRI が不明であるか、他の手段で学習されています。 INCOMPLETE は通常、他のルーティング プロトコルから BGP にルートが再配布され、ルートの送信元が不完全である場合に発生します。 BGP テーブル内の？は INCOMPLETE を示しています。



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

RTA は 300 i を経由して 170.10.0.0 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 190.10.50.0 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 150.10.0.0 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 190.10.0.0 に到達します。 「100 ?」は、 次の AS が 100 で、送信元は不完全でスタティック ルートから到達し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隣接ルータの IP アドレスです。 この項の例では、RTC はネクスト ホップ 170.10.20.2 を使用して RTA に 170.10.0.0 をアドバタイズします。 RTA はネクスト ホップ 170.10.20.1 を使用して RTC に 150.10.0.0 をアドバタイズします。 iBGP の場合は、eBGP がアドバタイズしたネクスト ホップは iBGP に伝達される必要があることがプロトコルで規定されています。 このルールに従い、RTA はネクスト ホップ 170.10.20.2 を使用して iBGP ピアの RTB に 170.10.0.0 をアドバタイズします。 したがって、RTB が 170.10.0.0 に到達するためのネクスト ホップは 150.10.30.1 ではなく 170.10.20.2 です。

RTB が IGP 経由で 170.10.20.2 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 170.10.0.0 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 170.10.0.0 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。

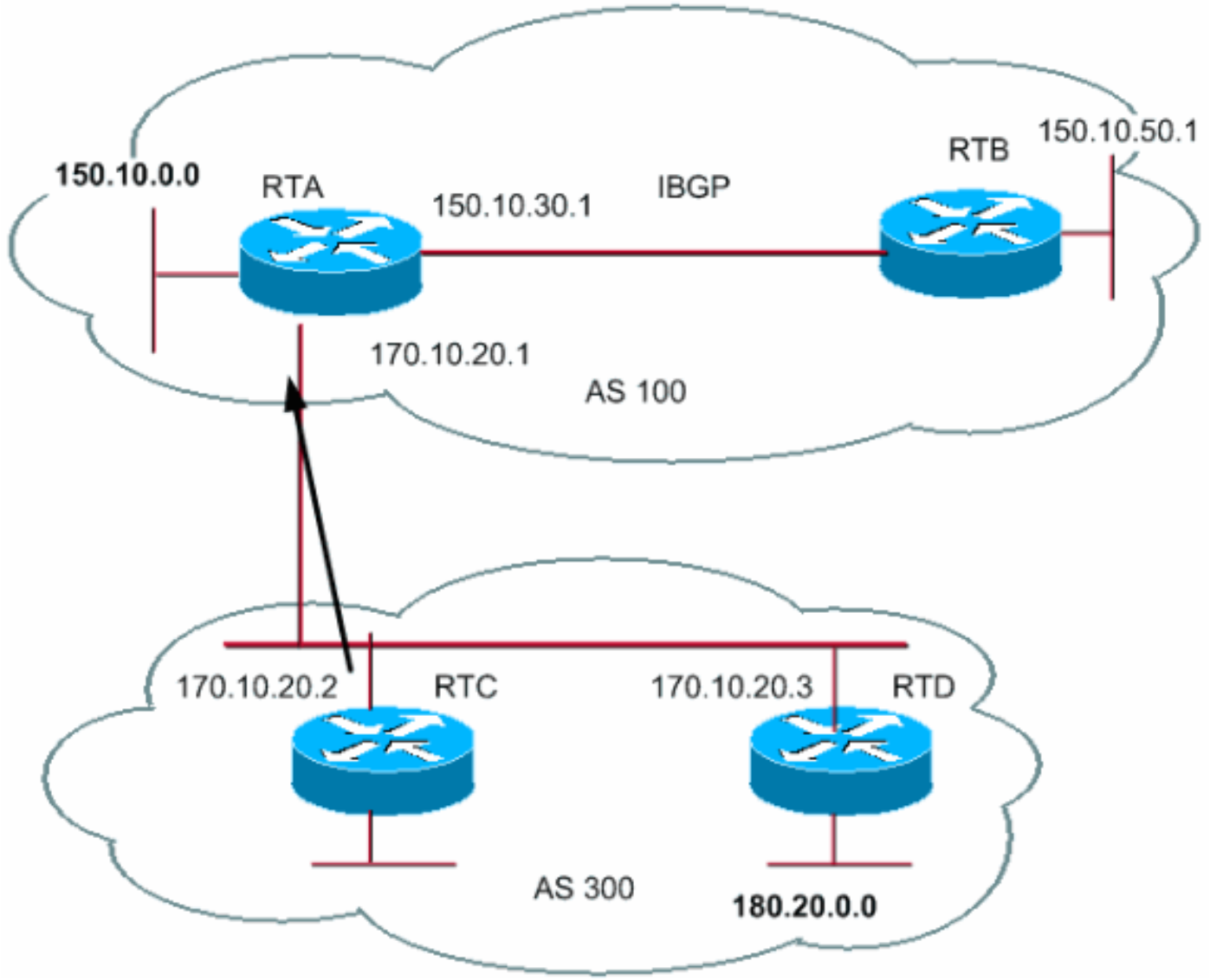
network *network-number* **[mask** *network-mask***]**

注: RTC は同じネクスト ホップ 170.10.20.2 を使用して RTA に 170.10.0.0 をアドバタイズします。

注: RTA は同じネクスト ホップ 170.10.20.2 を使用して RTB に 170.10.0.0 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 ネットワークおよびノンブロードキャスト 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。 詳細については、「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。

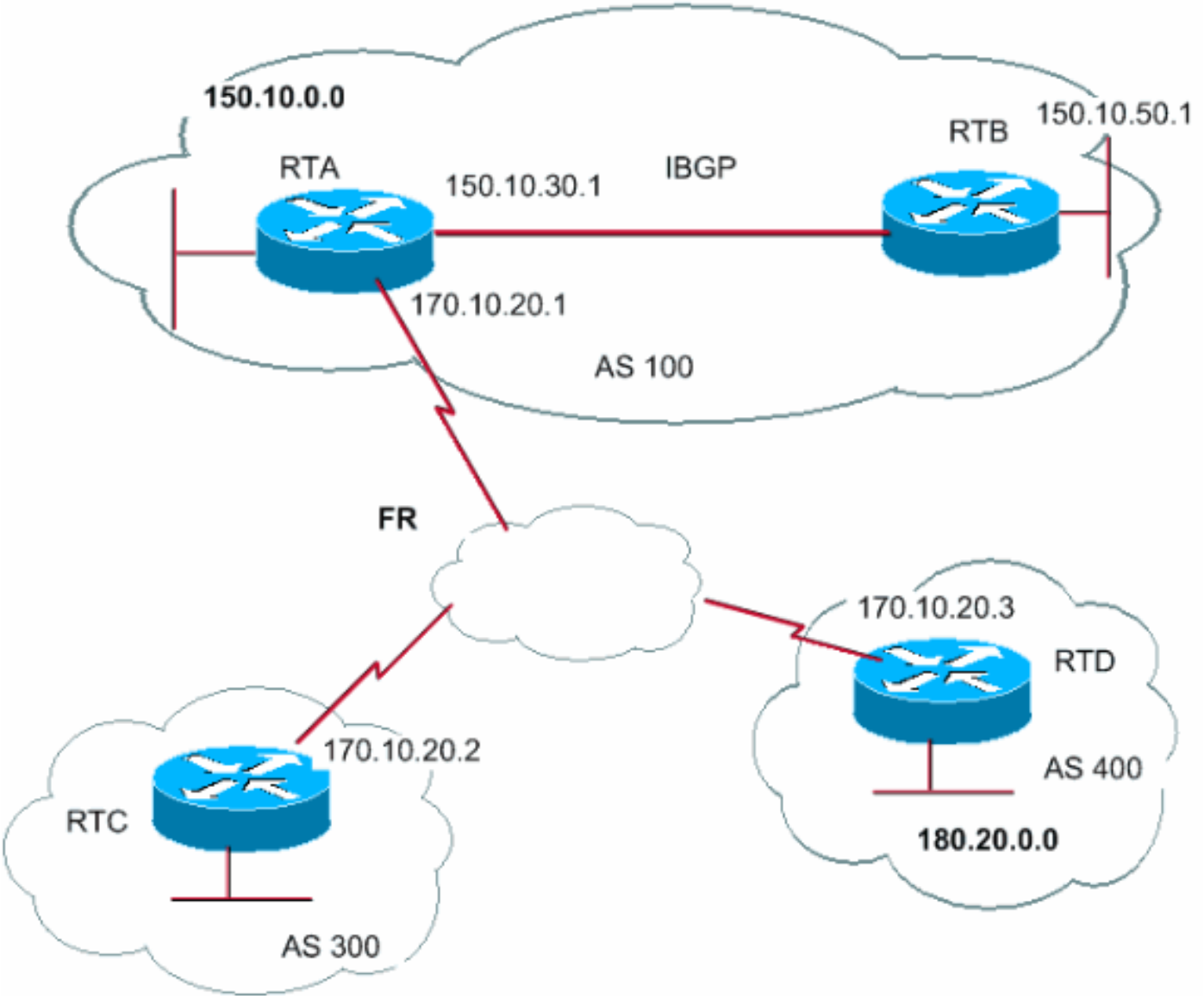
AS300 の RTC と RTD が OSPF を実行していると仮定します。 RTC は RTA との間で BGP を実行しています。 RTC は、170.10.20.3 経由でネットワーク 180.20.0.0 に到達できます。 RTC が 180.20.0.0 に関する BGP アップデートを RTA に送信する際には、ネクスト ホップとして 170.10.20.3 が使用されます。 RTC は自身の IP アドレス 170.10.20.2 を使用しません。 RTC がこのアドレスを使用する理由は、RTA、RTC、RTD 間のネットワークがマルチアクセス ネットワークであるためです。 RTA が 180.20.0.0 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 170.10.20.3 を使用して RTA に 180.20.0.0 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 170.10.20.3 を使用して RTA に 180.20.0.0 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)」の例で示したようなネクストホップの状況においては、**next-hop-self** コマンドを使用できます。 構文は次のとおりです。

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

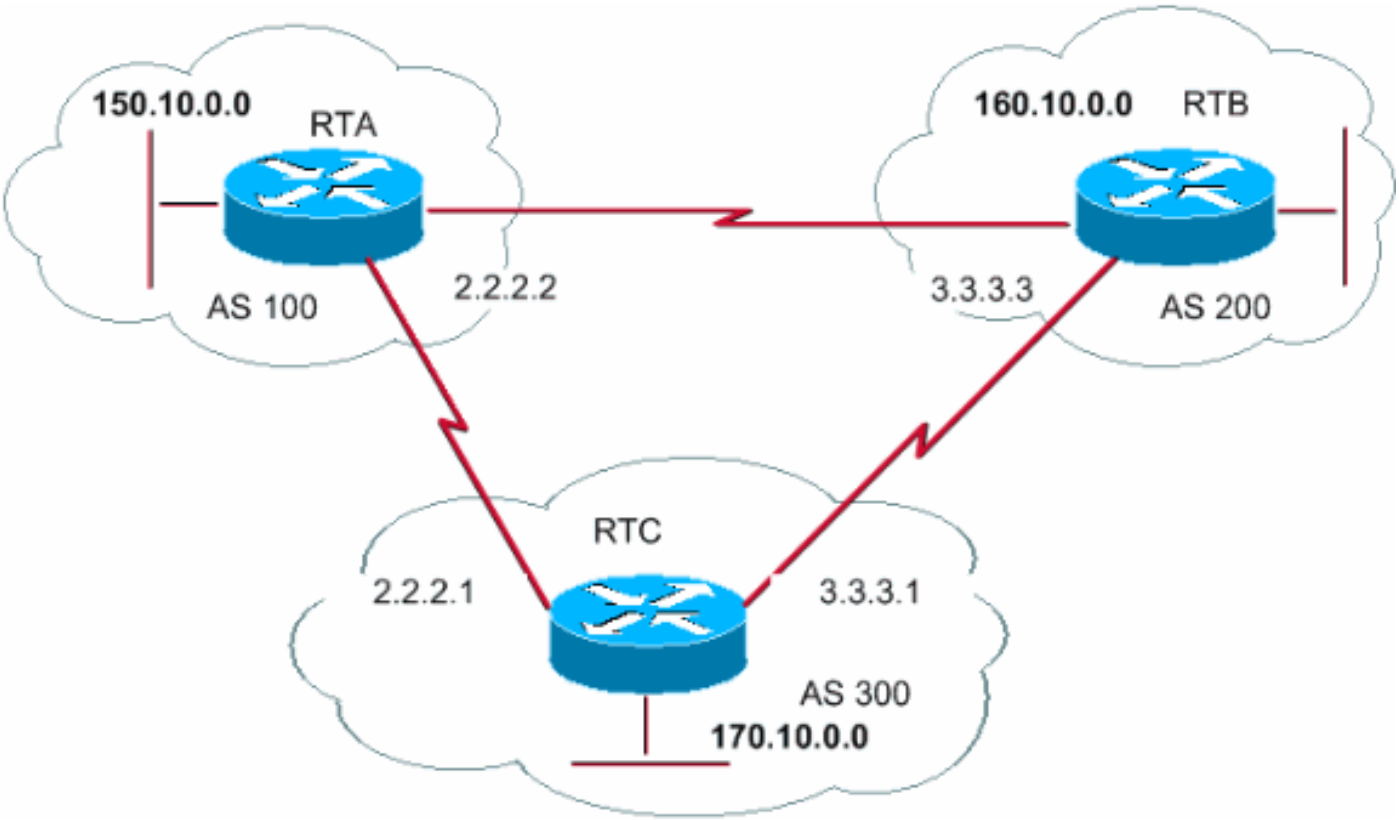
next-hop-self コマンドを使用すると、特定の IP アドレスが BGP のネクストホップとして使用さ

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。

neighbor {*ip-address* | *peer-group-name*} **next-hop-self**
RTC はネクスト ホップ 170.10.20.2 を使用して 180.20.0.0 をアドバタイズします。

BGP バックドア



この図では、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 プロトコル経由で 160.10.0.0 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

distance コマンドを使用すると、デフォルトの距離を変更できます。

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 に RTB (IGP) 経由で 160.10.0.0 について学習させるには、次の 2 つの方法があります。

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。

network address backdoor コマンドを発行します。

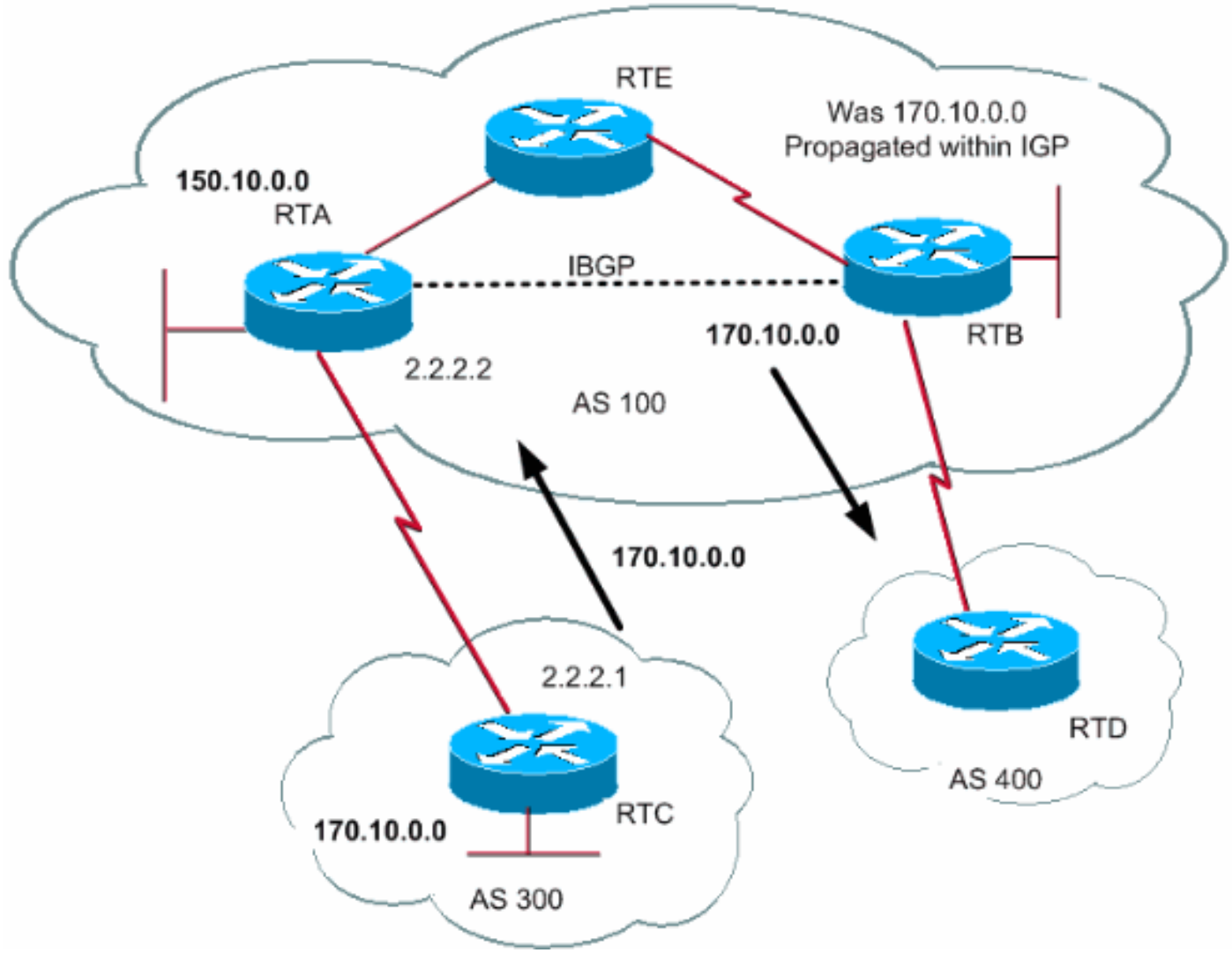
設定するネットワークは、IGP を使用して到達するネットワークです。 BGP アップデートでこのネットワークがアドバタイズされない点を除き、BGP では、このネットワークをローカルに割り当てられたネットワークと同様に扱います。

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

ネットワーク 160.10.0.0 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 エントリとしてアドバタイズされることはありません。

RTA は、距離が 90 の EIGRP 経由で RTB から 160.10.0.0 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 は 170.10.0.0 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 2.2.2.1 経由で 170.10.0.0 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 170.10.0.0 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 170.10.0.0 の存在すら認識されていません。

RTB が 170.10.0.0 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 に 170.10.0.0 宛てのトラフィックが流れますが、RTE でドロップされます。

同期の規定では、自 AS が別の AS からのトラフィックを第 3 の AS に渡す場合、自 AS 内のすべてのルータが IGP 経由でルートを学習するまでは、BGP はルートをアドバタイズできないことになっています。 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 170.10.0.0 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。 170.10.0.0 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 170.10.0.0 に到達できることを確認してください。

同期の無効化

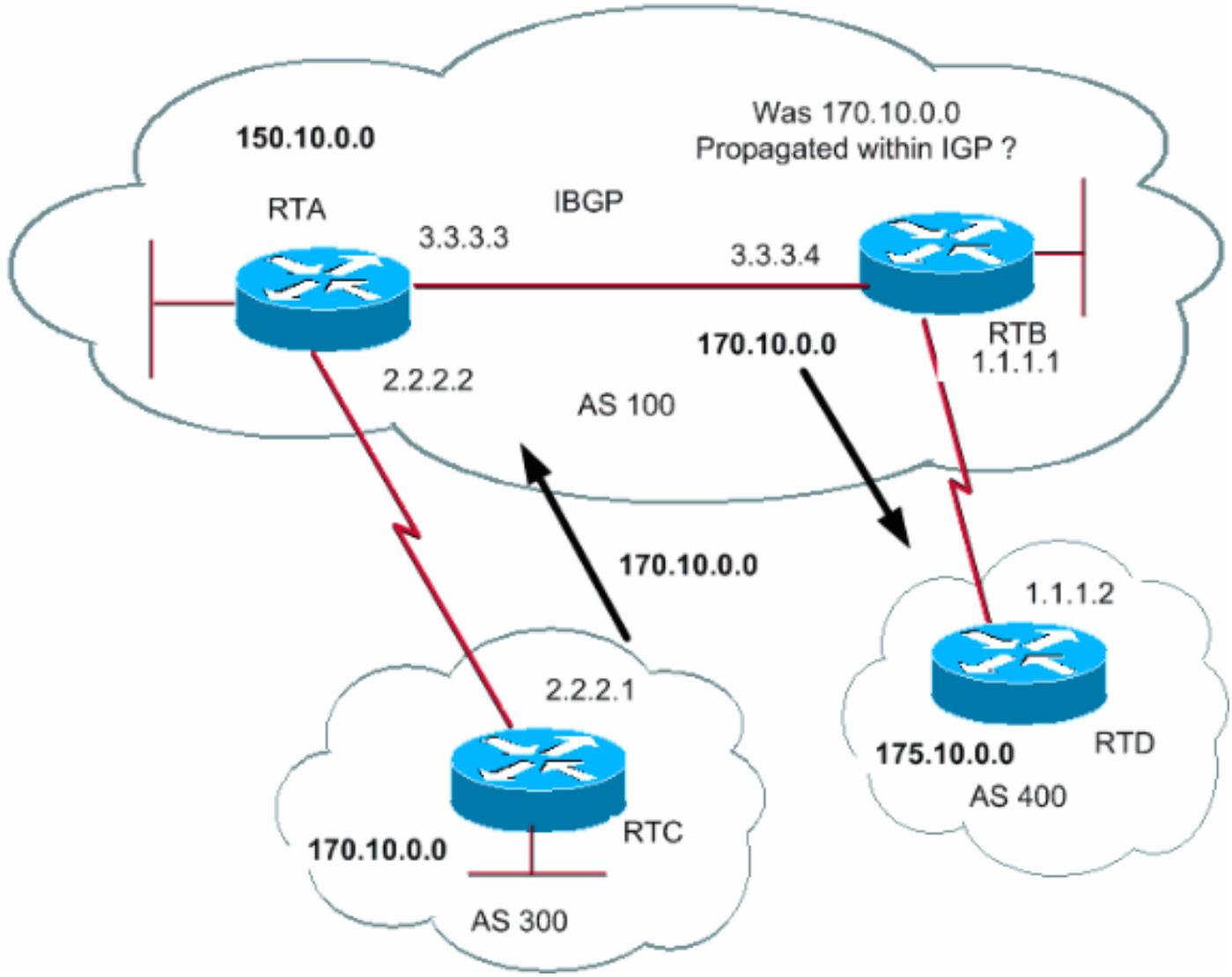
場合によっては、同期が必要ないことがあります。 別の AS からのトラフィックが自 AS を通過

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。

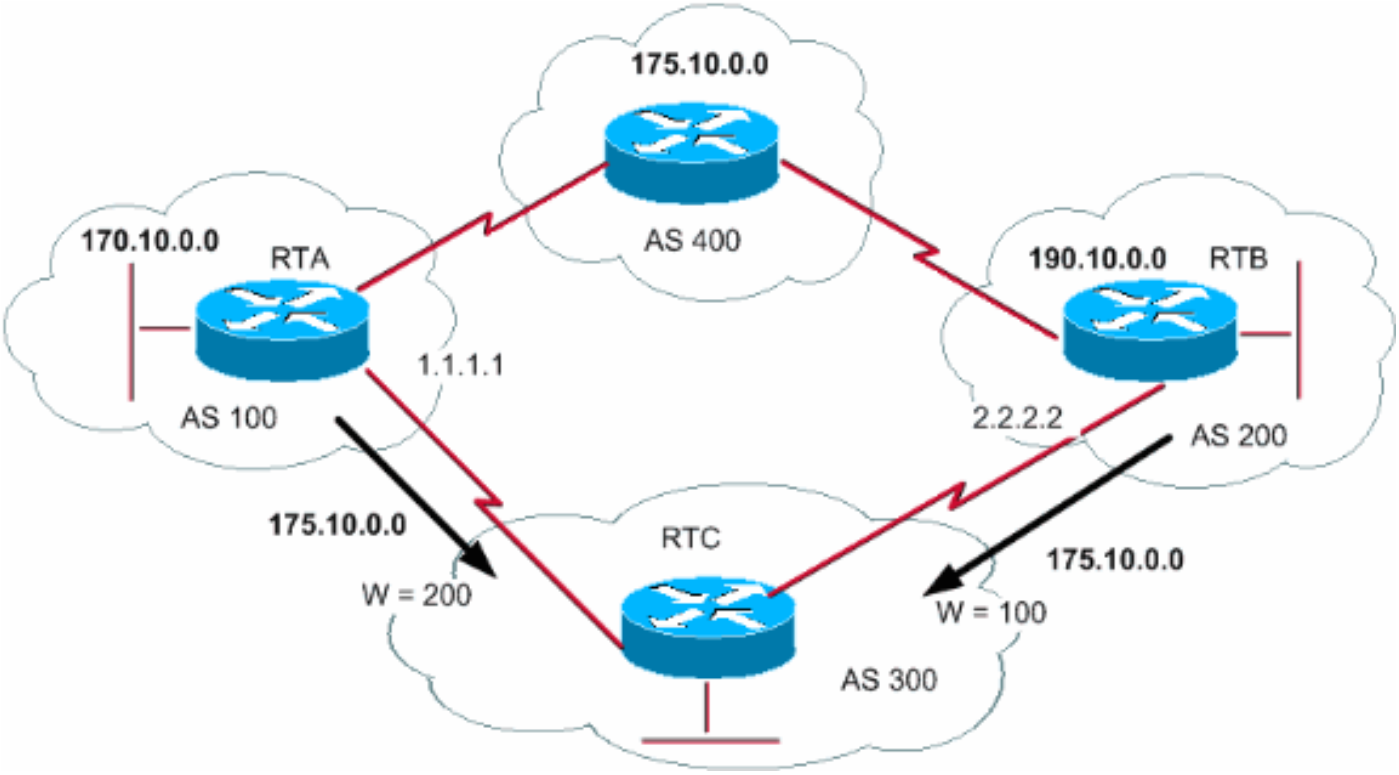
distance bgp *external-distance internal-distance local-distance*

注: 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。



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

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 175.10.0.0 について学習しています。 RTA は RTC にアップデートを伝達します。 RTB も AS4 からネットワーク 175.10.0.0 について学習しています。 RTB は RTC にアップデートを伝達します。 これで RTC には 175.10.0.0 に到達するルートが 2 つ存在することになり、どちらを使用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 175.10.0.0 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

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

ルート マップを使用する。

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

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。

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

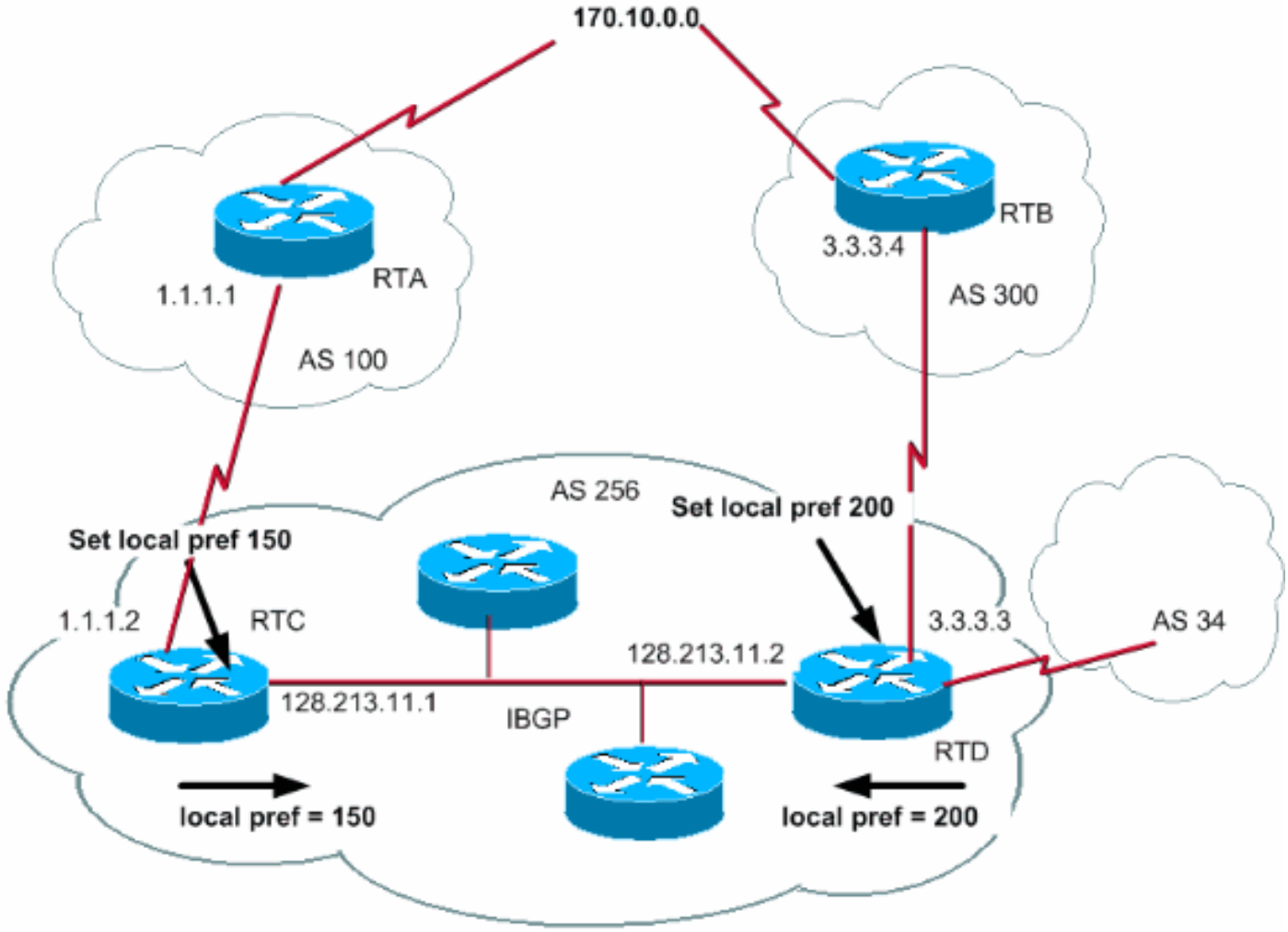
また、ルート マップを使用しても同じ結果を得られます。

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

注: バックアップとして IGP パスを指定した MPLS VPN BGP パスを優先するように重みを変更できます。

注: 詳細については、シスコ サポート コミュニティの次のドキュメントを参照してください。プライマリ条件と障害条件の両方に対する優先パスを指定し、プライマリ パスの回復後に再ルーティングするようにルータを設定する方法について説明しています。 IGP バックアップを指定した MPLS VPN BGP パスを優先する

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: 変更が反映されるように、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** はセッションを切断しないソフト リセットを示し、**[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、同じ AS 内のピアに到達するルータからのアップデートでローカル プリファレンスを設定します。 上記の図では、AS256 は組織の異なる 2 台のルータから 170.10.0.0 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。

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

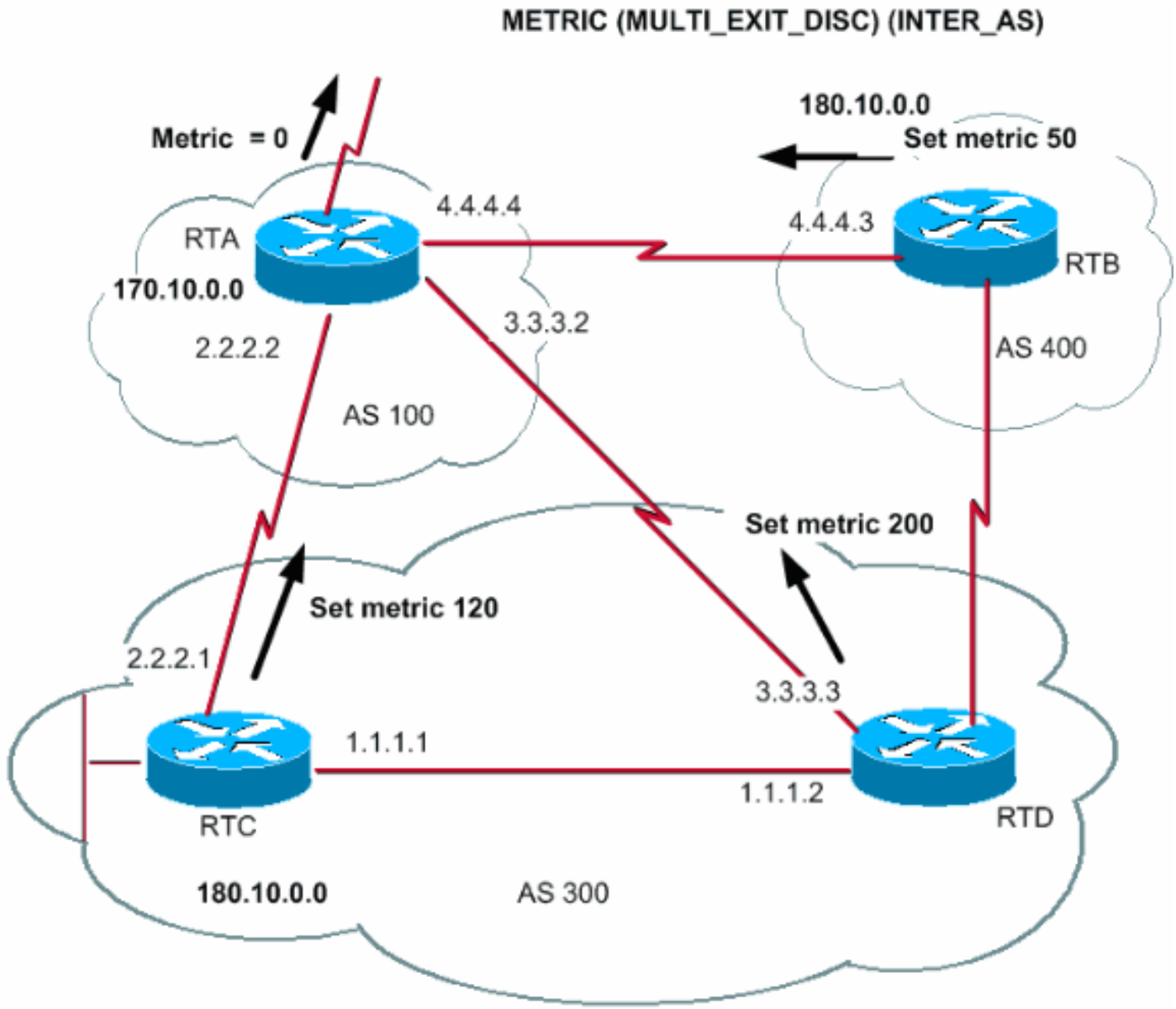
この設定では、RTC はすべてのアップデートのローカル プリファレンスを 150 に設定します。 同様に、RTD はすべてのアップデートのローカル プリファレンスを 200 に設定します。 AS256 内ではローカル プリファレンスの交換が行われます。 したがって RTC と RTD の両方が、ネットワーク 170.10.0.0 のアップデートは、AS100 ではなく AS300 から到達する場合にローカル プリファレンスがより高いことを認識します。 そのネットワークを宛先とする AS256 内のトラフィックはすべて、RTD を出力点として送信されます。

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。 次に例を示します。

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

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 識別子（MED）ベースのパス選択に影響を与えることができる BGP 設定コマンドは 2 つあります。 **それらのコマンドは、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 内の異なるピアからアドバタイズされたルートを選択するときに 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、異なる AS の隣接ルータからのパスについて、

MED が比較されるようになります。 複数のサービス プロバイダーまたは企業が MED の設定に関して統一されたポリシーに合意している場合には、**bgp always-compare-med** コマンドが便利です。 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。

上記の図では、AS100 はネットワーク 180.10.0.0 に関する情報を 3 台の異なるルータ（RTC、RTD、および RTB）から取得します。 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した場合、BGP は最短の AS-Path を持つ RTC ルータからルート 180.10.0.0 をインストールします。

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 がメトリックの比較を行うようにするには、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。

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

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。

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

この場合、RTA はネットワーク 180.10.0.0 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-metric *number* 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。

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

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。 コミュニティ属性を使用すると、宛先を特定のコミュニティにグループ化し、それぞれのコミュニティに応じた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。

```
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 内に維持する。

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：パケットがローカルの AS 外部へ送信されないようにするために、コンフェデレーション シナリオで使用。

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。

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

  または

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

additive キーワードを設定しなければ、すでに存在している古いコミュニティが 200 に置き換わります。 **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。

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

次に例を示します。

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

Cisco IOS ソフトウェア リリース 12.0 以降では、3 種類の形式（10 進数、16 進数、および 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: **ip bgp-community new-format** グローバル設定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 バイト番号を表します。

次に例を示します。

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 6.0.0.0 コマンドを発行すると、10 進形式でコミュニティ アトリビュート値が表示されます。 この例では、コミュニティ属性値が 6553620 と表示されます。

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

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。

```
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-format グローバル設定コマンドを使用すると、AA: NN 形式でコミュニティ値が表示されます。 この例では、**show ip bgp 6.0.0.0** コマンドの出力で、この値が 100:20 と表示されます。

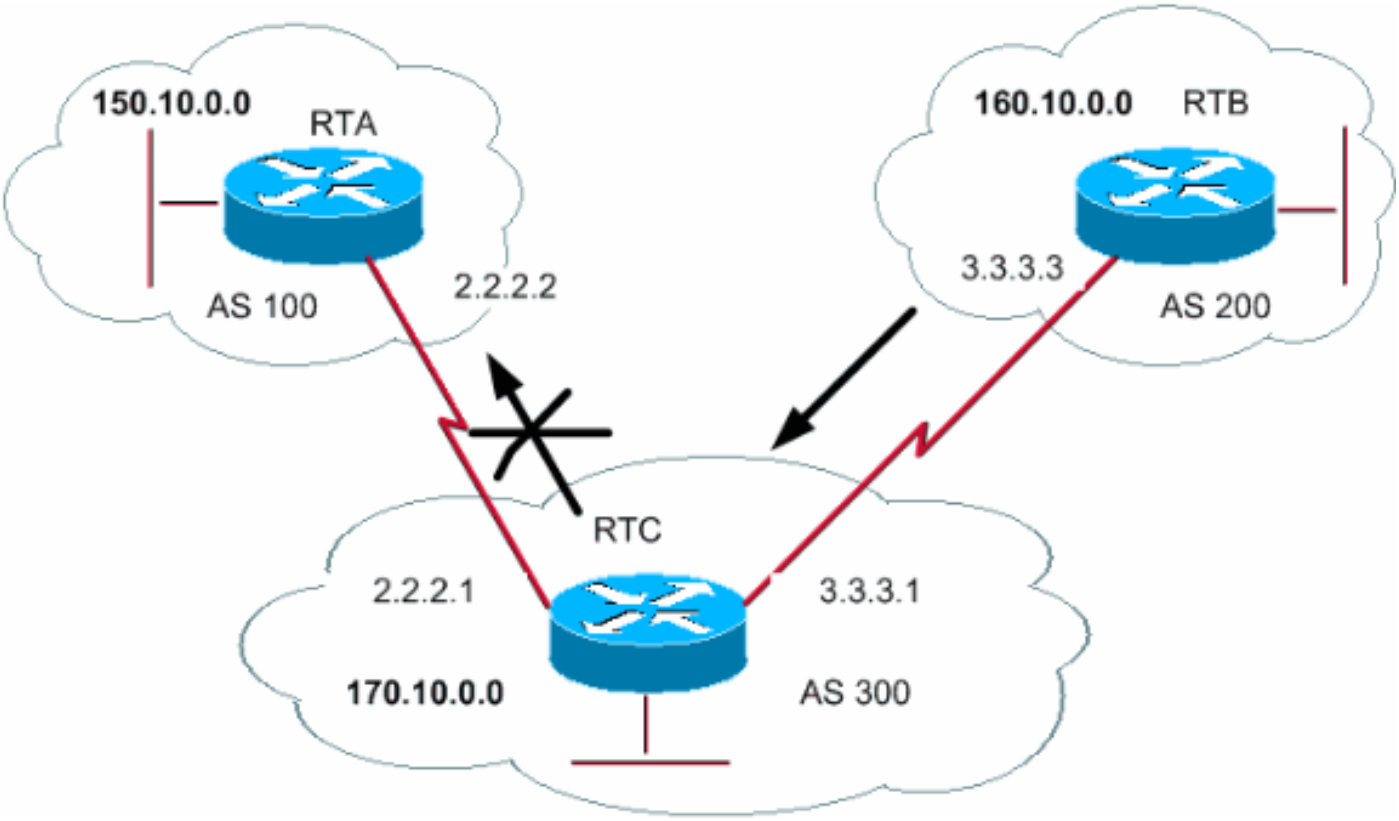
```
Router# show ip bgp 6.0.0.0
BGP routing table entry for 6.0.0.0/8, version 9
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  1
    10.10.10.1 from 10.10.10.1 (200.200.200.1)
      Origin IGP, metric 0, localpref 100, valid, external, best
      Community: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

BGP ケース スタディ 3

BGP フィルタリング

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 フィルタリング



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。

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 160.10.0.0 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。

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

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、160.0.0.0/8 のみがアドバタイズされるようにします。

注: /8 の表記は、IP アドレスの左端から 8 ビットをサブネット マスクとして使用することを意味します。 このアドレスは 160.0.0.0 255.0.0.0 に相当します。

access-list 1 permit 160.0.0.0 0.255.255.255 コマンドは、160.0.0.0/8、160.0.0.0/9 などを許可します。 アップデートを 160.0.0.0/8 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。

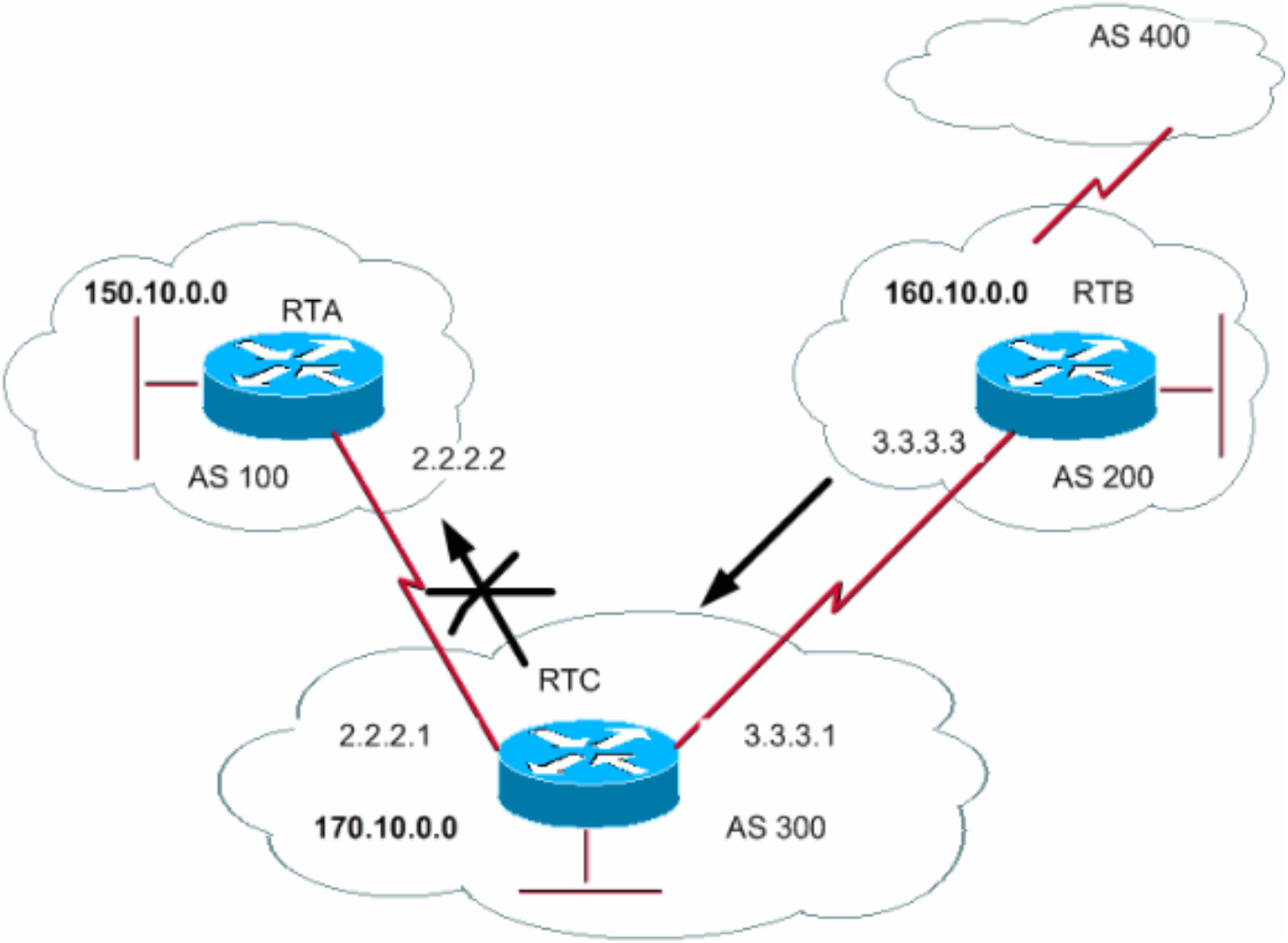
```
access-list 101 permit ip 160.0.0.0 0.255.255.255 255.0.0.0 0.0.0.0.
```

このリストは 160.0.0.0/8 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キング設定例』を参照してください。 この方式では、プレフィクス リスト フィルタリングだけでなく、標準 Access Control List (ACL; アクセス コントロール リスト) および拡張 ACL とともに **distribute-list** コマンドを使用しています。

パス フィルタリング

もう 1 つのフィルタリング タイプは、パス フィルタリングです。



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。 上記の図では、160.10.0.0 に関するアップデートをブロックして AS100 に送信されないように設定できます。 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。

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

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

次の例では、RTC から RTA への 160.10.0.0 に関するアップデートの送信が停止されます。

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

この例の access-list 1 コマンドにより、200 で始まって 200 で終わるパス情報を持つすべてのアップデートが拒否されます。 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 160.10.0.0 に関するアップデートを 200 で始まって 200 で終わるパス情報とともに送信するため、この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」を意味し、? 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チェックするには、show ip bgp regexp *regular 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 フィルタリング」の項の例では、文字列 ^200$ を指定しました。 フィルタリングを行うには、アップデートに含まれるパス情報がこの文字列に一致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。

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

. は任意の 1 文字に一致します。

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

^ は入力文字列の先頭に一致します。

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

$ は入力文字列の末尾に一致します。

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

\ は指定した文字に一致します。

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

_ 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、アトムの後に続く次のいずれかの記号です。

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

? は 0 個以上のアトムのシーケンスに一致します。

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

+ は 1 個以上のアトムのシーケンスに一致します。

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

[?] はアトムまたはヌル ストリングに一致します。

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。

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

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）。

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

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

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

この表現は、「aa」または「aba」に一致します。

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

この表現は、AS100 経由であることを意味します。

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

この表現は、AS100 が送信元であることを示します。

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

この表現は、AS100 からの送信を示します。

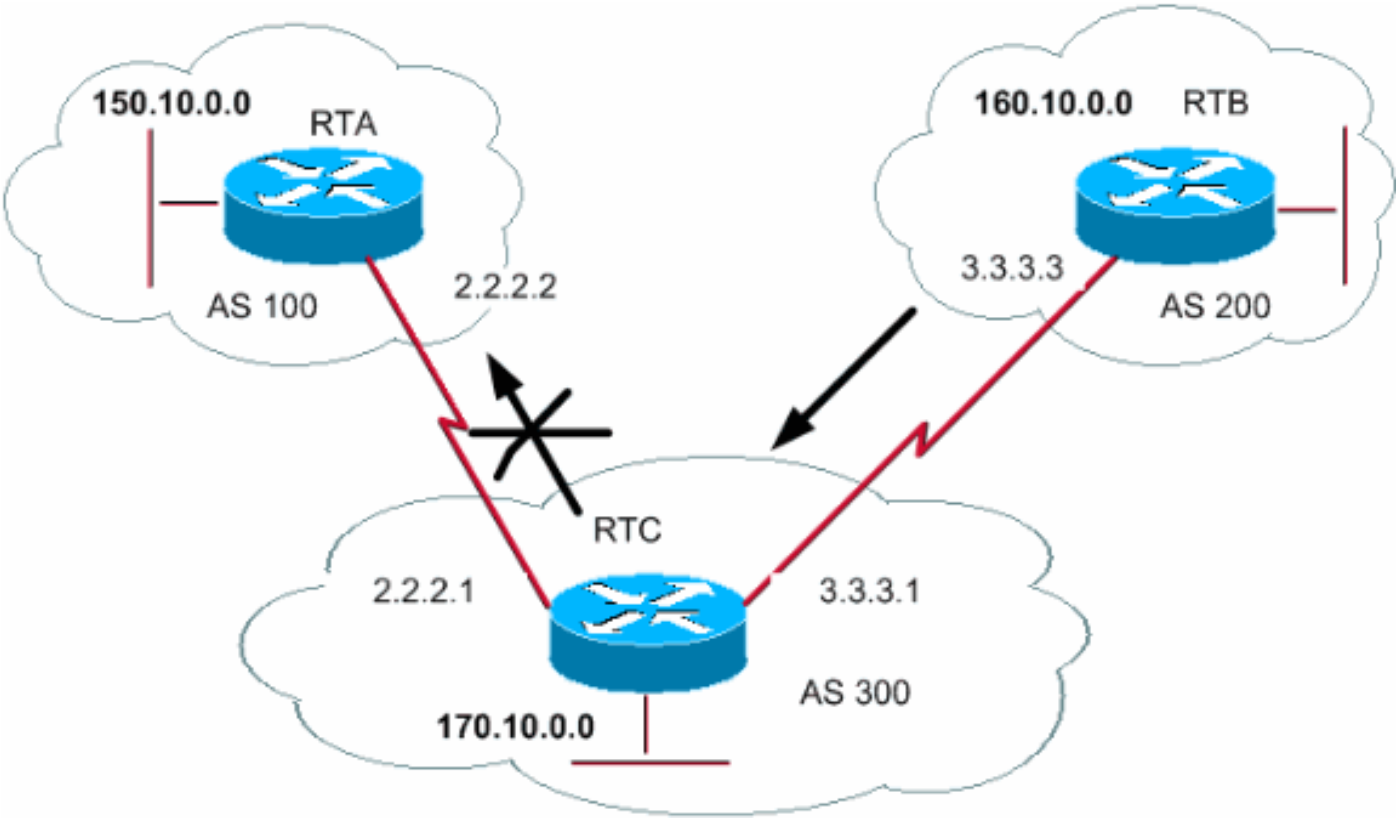
```
neighbor {ip-address | peer-group-name} filter-list access-list-number {in | out}
```

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 フィルタリング

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。 コミュニティについては「コミュニティ属性」の項で説明しているため、ここではコミュニティの使用例をいくつか紹介します。



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。 この場合は、**no-export** コミュニティ アトリビュートを使用します。

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

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** コマンドを使用しています。

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。

次の例では、RTB はコミュニティ属性を **100 200 additive** に設定します。 これにより、RTC に

送信される前に既存のすべてのコミュニティ値に値 100 200 が追加されます。

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

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。

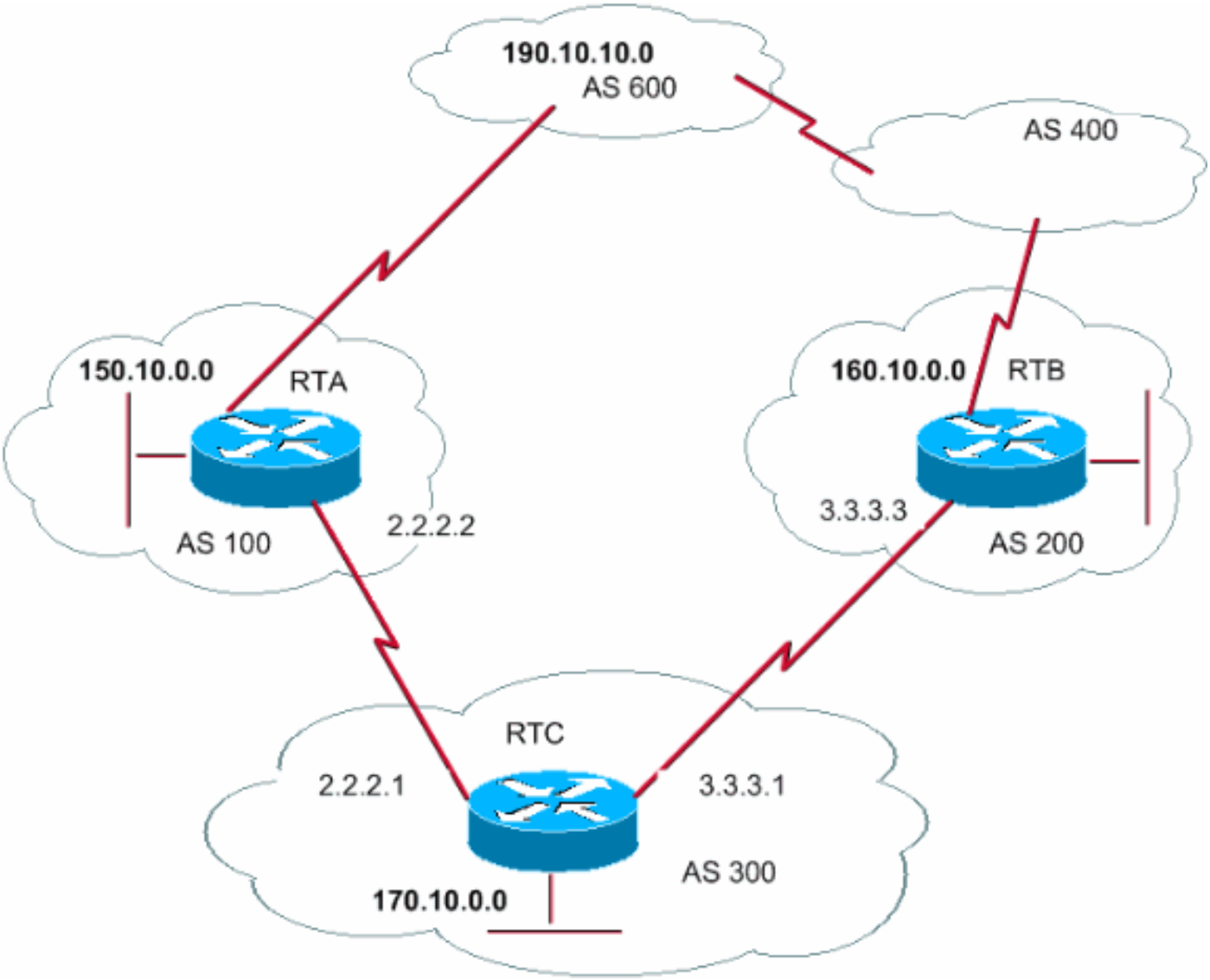
ip community-list *community-list-number* **{permit | deny}** *community-number*
たとえば、次のような **match-on-community** ルート マップを定義できます。

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

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

詳細については、『BGP コミュニティ値を使用した、アップストリーム プロバイダー ネットワークでのルーティング ポリシーの制御』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 マップと組み合せて使用すれば、送受信されるアップデートに対してフィルタリングまたはパラメータ設定を実行できます。

neighbor 文と関連付けられたルート マップは、次のように IP アドレスに基づいて照合される場合には、受信されるアップデートに影響しません。

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。

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

neighbor *ip-address* **route-map** *route-map-name*
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、AS400 の背後から到達するアップデートの重みは 10 に設定され、AS400 から到達するアップデートはドロップされます。

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。

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

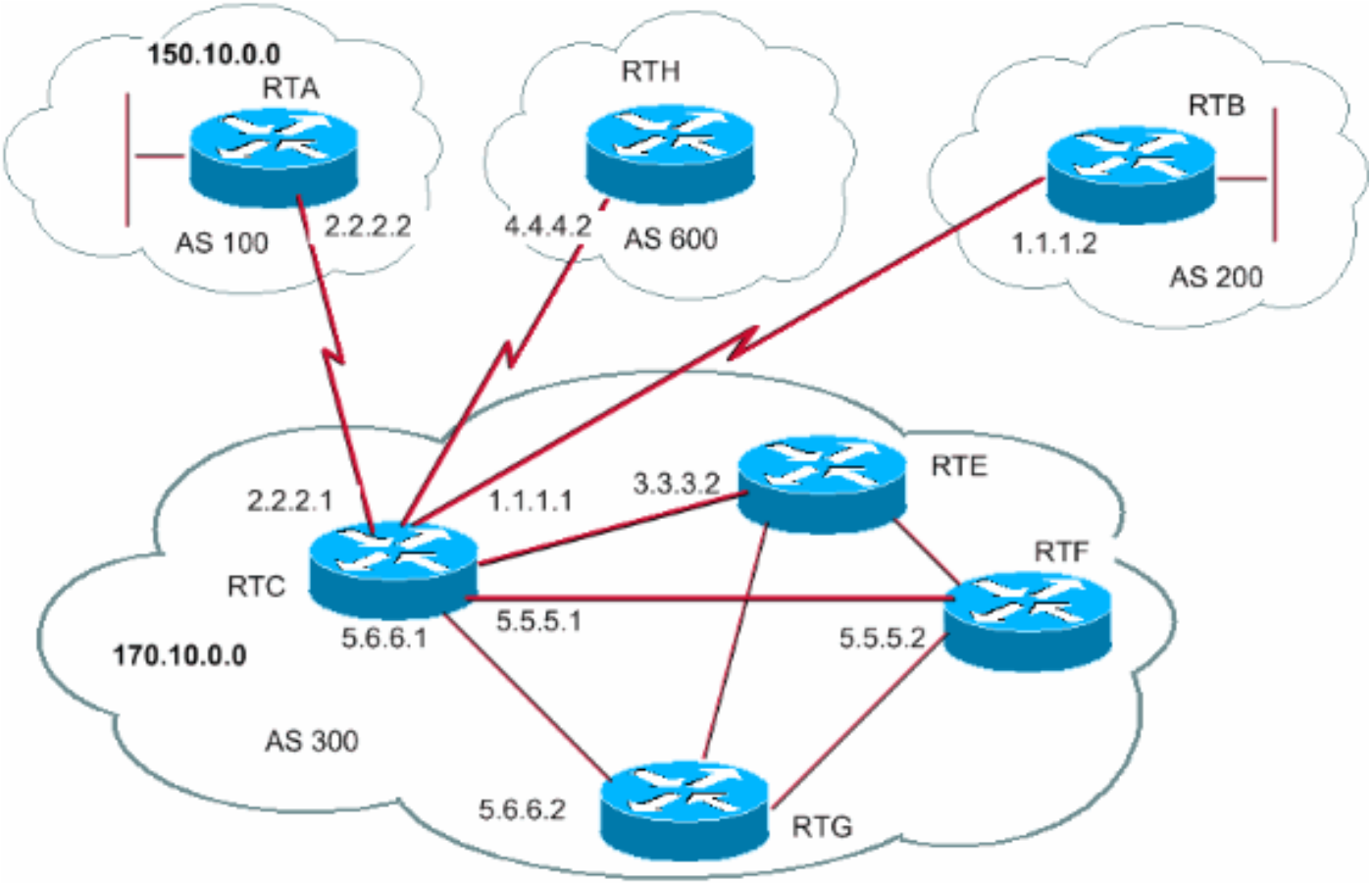
「BGP ネイバーとルート マップ」の項の図で、RTC が自身のネットワーク 170.10.0.0 を 2 つの異なる AS (AS100 と AS200) にアドバタイズすると仮定します。 情報が AS600 に伝達されると、AS600 のルータは、170.10.0.0 に関して 2 つのルートを経由するネットワーク到達可能性情報を持つことになります。 1 つは AS100 を経由するパス (100、300) のルート、もう 1 つは、AS400 を経由するパス (400、200、300) のルートです。 他の属性がすべて同じであれば、AS600 は最短パスである AS100 経由のルートを選択します。

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。 そのためには、AS100 にアドバタイズされる既存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。

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

この設定により、AS600 が AS100 経由で受信する 170.10.0.0 に関するアップデートには、(100、300、300、300) というパス情報が含まれることになります。 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。 個別のネイバーごとに同じポリシーを定義するのではなく、 ピア グループ名を定義して、そのピア グループにこれらのポリシーを割り当てます。

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。

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。

neighbor *peer-group-name* **peer-group**
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。 また、このグループに対して、メトリックを 5 に設定するルート マップ **SETMETRIC** や 2 つの異なるフィルタ リスト (1 および 2) などのポリシーが定義されます。 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作用するオプションのみです。

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。

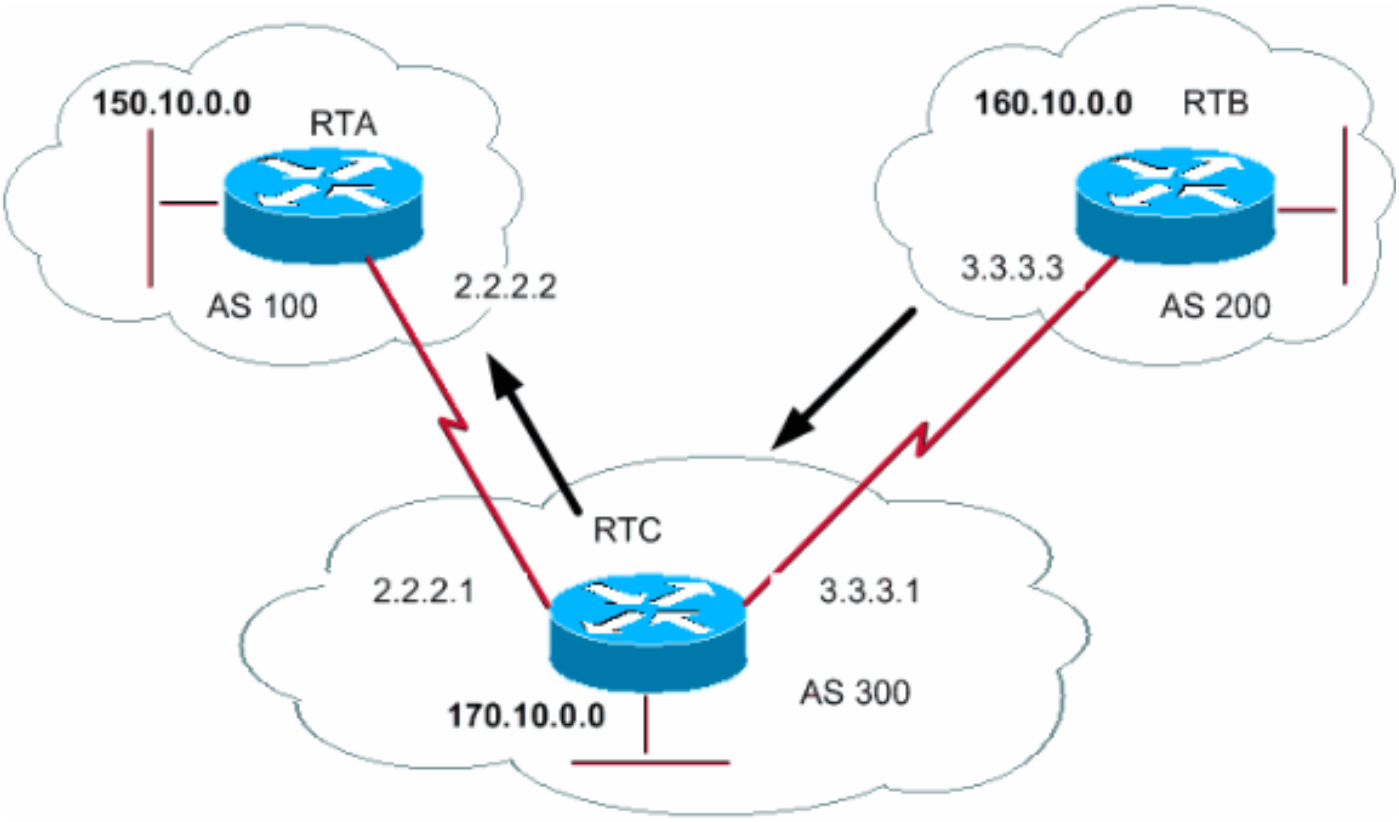
neighbor *peer-group-name* **peer-group**
注: これらの設定では、複数の外部 AS を定義する必要があるため、ピア グループの外部で **remote-as** ステートメントを定義しています。 また、フィルタ リスト 3 を割り当てることでネイバー 1.1.1.2 の着信アップデートを上書きします。

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、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ました。 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。

BGP ケース スタディ 4

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、または C といったクラスの概念がありません。たとえばネットワーク 192.213.0.0 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 192.213.0.0/16 です。 「16」は、IP アドレスの左端から数えたサブネット マスク内のビット数を表します。 これは 192.213.0.0 255.255.0.0 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 160.10.0.0 を生成しています。 このルートのスーパーネット 160.0.0.0 を RTA に伝達するように RTC を設定します。

neighbor *peer-group-name* **peer-group**
RTC は RTA に集約アドレス 160.0.0.0 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」の項の例で使用しています。

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 160.0.0.0** コマンドは、追加のネットワーク 160.0.0.0 を伝達しますが、これによって RTA への 160.10.0.0 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 160.0.0.0 と 160.10.0.0 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具体的なルートが含まれていない場合、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 160.0.0.0 のより具体的な エントリが存在しない場合は、RTB は 160.0.0.0 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（**network 160.10.0.0** など）

RTC がネットワーク 160.0.0.0 のみを伝達し、より具体的なルートを**伝達しない**ようにするには、次のコマンドを発行します。

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 160.0.0.0 255.0.0.0 summary-only コマンドは、ネットワーク 160.0.0.0 を伝達し、よ

り具体的なルート 160.10.0.0 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。 この注入は、**aggregate summary-only** コマンドを使用する場合でも発生します。 「CIDR 例 1」の項で、この場合の例について説明します。

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内に **as-set** 情報が含まれています。

aggregate 129.0.0.0 255.0.0.0 as-set

「CIDR 例 2 (as-set)」の項で、このコマンドについて説明します。

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。

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、160.0.0.0 を集約し、より具体的なルート 160.20.0.0 を抑制して 160.10.0.0 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。

aggregate-address *address-mask* **suppress-map** *map-name*
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。

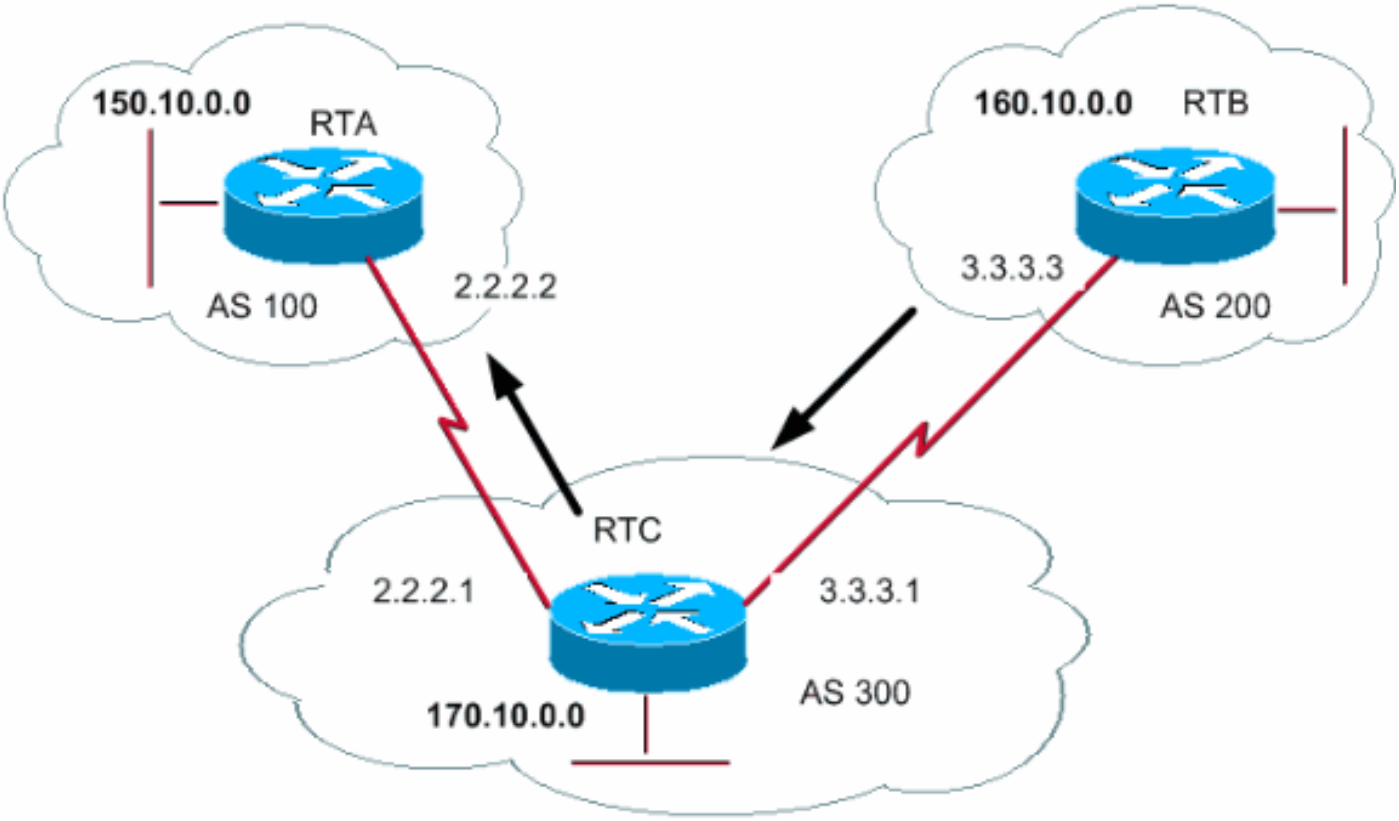
次に、このルート マップを **aggregate** 文に適用します。

aggregate-address *address-mask* **suppress-map** *map-name*
次のような形式もあります。

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。 集約の起点 (origin) を IGP に設定するには、**aggregate attribute-map** コマンドに次のルート マップを適用します。

aggregate-address *address-mask* **attribute-map** *map-name*
詳しくは、『BGP での経路集約について』を参照してください。

CIDR 例 1



Request： RTB がプレフィクス 160.0.0.0 をアドバタイズして、より具体的なルートをすべて抑制するようにします。 この要求の問題点は、ネットワーク 160.10.0.0 が AS200 に対してローカルである、つまり AS200 が 160.10.0.0 の発信元であるということです。 RTB では、160.10.0.0 のエントリを生成しなければ、160.0.0.0 のプレフィクスを生成できません。**aggregate summary-only** コマンドを使用しても結果は同じです。 RTB は 160.10.0.0 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。

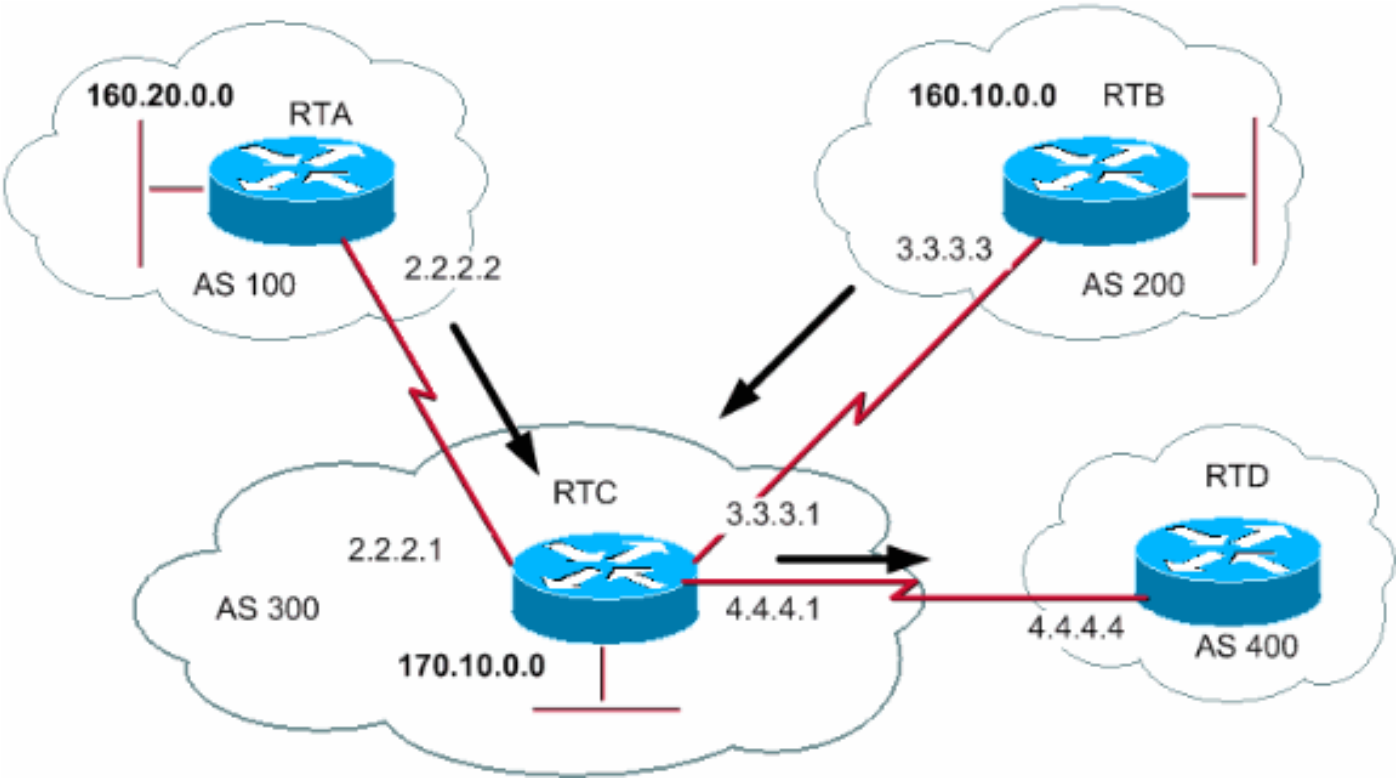
1 つは、スタティック ルートを使用して BGP に再配布する方法です。 これにより、RTB は不完全（？）な送信元とともに集約をアドバタイズします。

aggregate-address *address-mask* **attribute-map** *map-name*
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。

aggregate-address *address-mask* **attribute-map** *map-name*

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 160.20.0.0 に関するアップデート、RTB から 160.10.0.0 に関するアップデートを取得します。 RTC がネットワーク 160.0.0.0/8 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** 文を追加すると、RTC は集合 {} の形式でパス情報を生成するようになります。 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。



aggregate-address *address-mask* **attribute-map** *map-name*
ケース 1 :

RTC には **as-set** 文が設定されていません。 RTC は、AS300 から発信されたルートに見えるように、パス情報 (300) を付けてアップデート 160.0.0.0/8 を RTD に送信します。

aggregate-address *address-mask* **attribute-map** *map-name*
Case 2:

aggregate-address *address-mask* **attribute-map** *map-name*
次の 2 つの項、「BGP コンフェデレーション」と「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インターネット サービス プロバイダー (ISP) 向けです。

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。

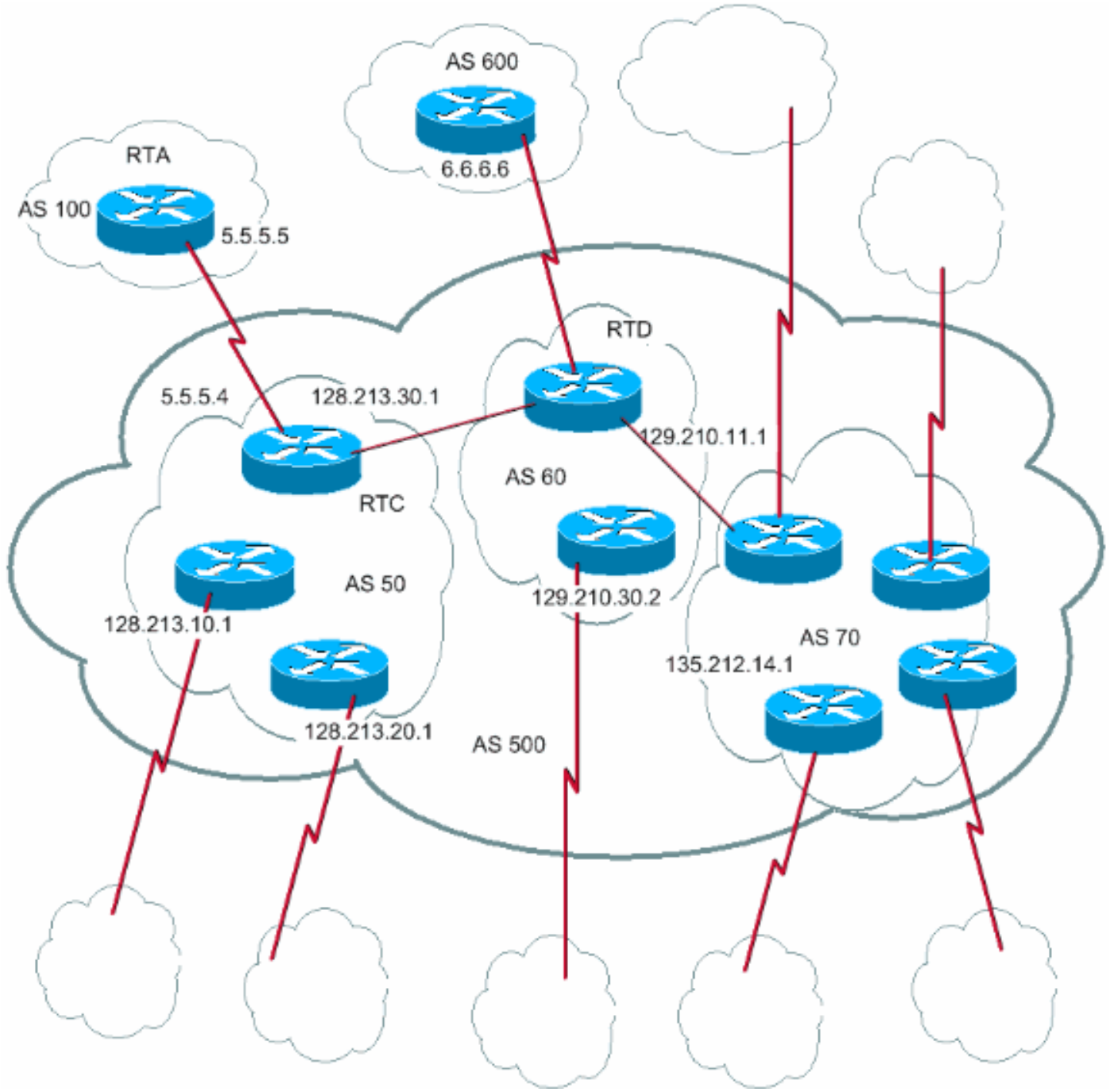
bgp confederation identifier *autonomous-system*

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。

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。 各 AS50、AS60、および AS70 に対して、フル メッシュ構造の iBGP ピアを定義し、**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。

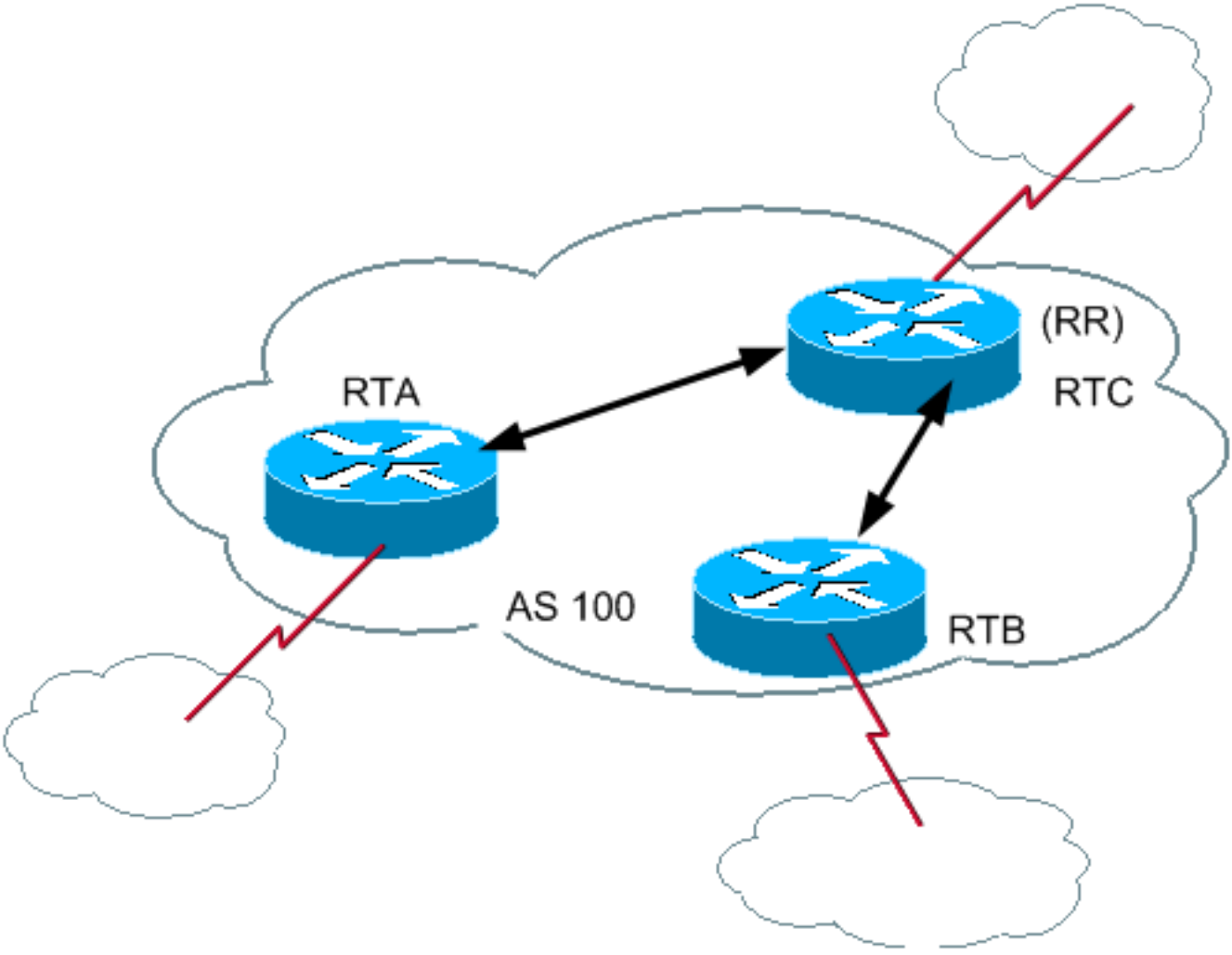
RTC、RTD、および RTA ルータの設定例は次のとおりです。

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

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

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」の項で説明したように、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。



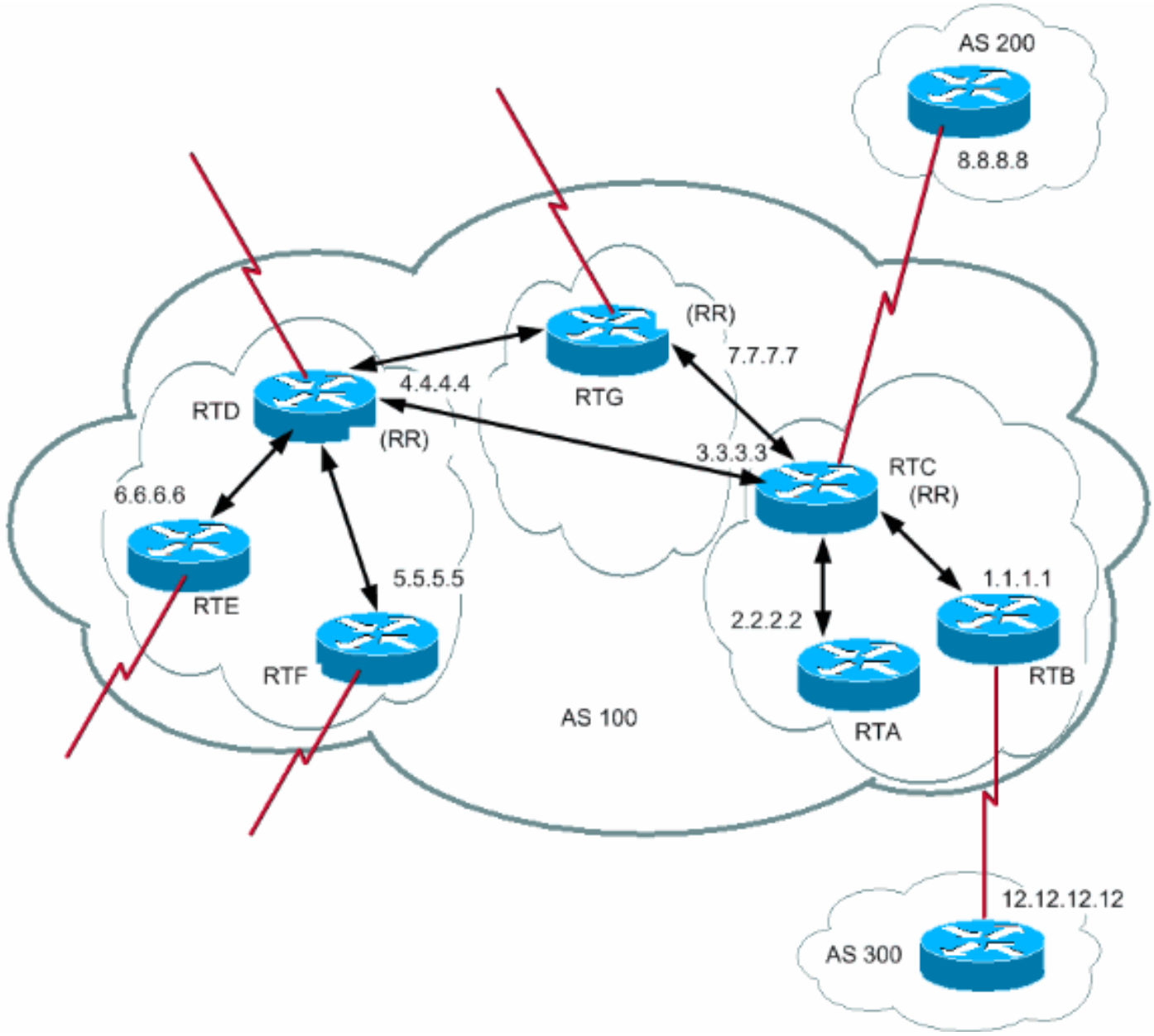
通常は、AS100 内の RTA、RTB、および RTC 間でフル メッシュ構造の iBGP を維持する必要があります。 RR の概念を利用すれば、RTC を RR として選択できます。 これにより、RTC は

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。

neighbor 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 アドレスを指定しています。 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」と呼ばれます。



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。 クライアントがクライアント クラスタ外の iBGP スピーカとピアを確立することはできま

せん。

上の図をご覧ください。 RTA、RTB、および RTC は 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 のクライアントがポイントされてい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 メッシュ化されていますが、クラスタ内のルータはフル メッシュ化されていません。 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。

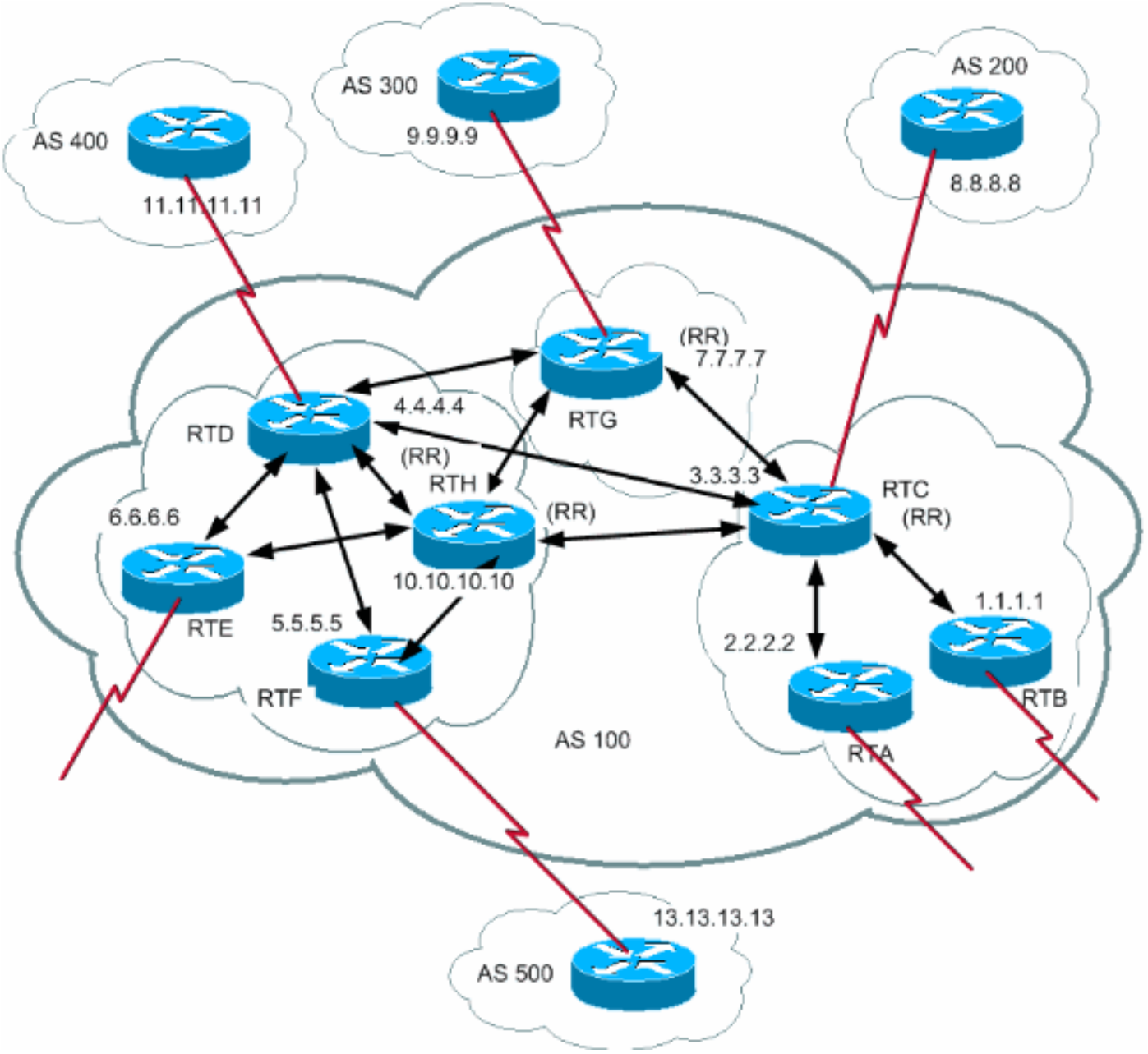
```
neighbor route-reflector-client
```

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 メッシュでピアリングしているため、冗長性が確保されます。 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。

`neighbor route-reflector-client`

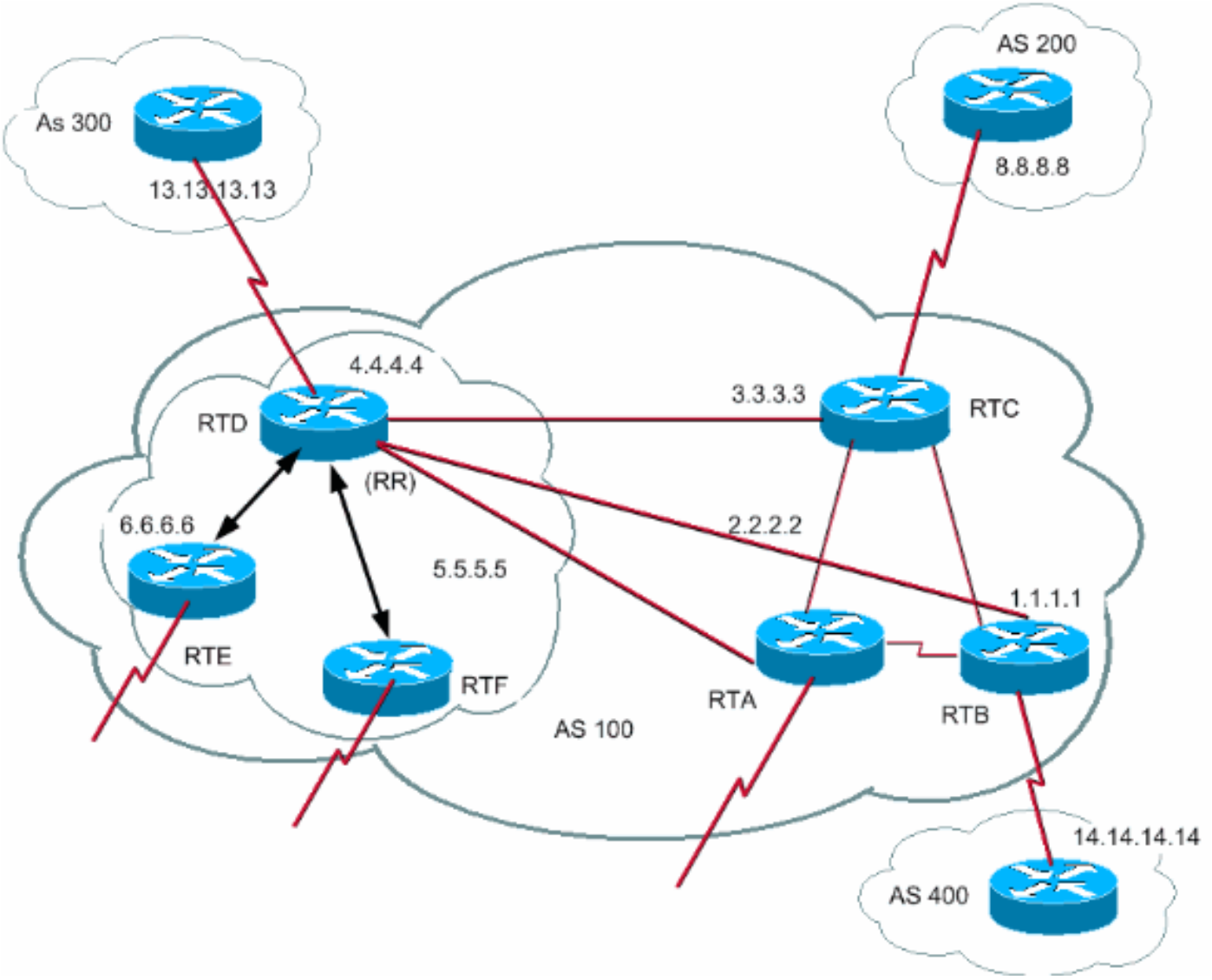
注: RTC のクラスタには RR が 1 つしか存在しないため、RTC に bgp cluster-id コマンドを設定する必要はありません。

特記事項： この設定ではピア グループを使用しません。 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」のルータです。 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 のみです。

RTD と RTC の設定は次のとおりです。

neighbor route-reflector-client

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、潜在的な情報ループを回避するために使用できる 2 つの属性 (**originator-id** と **cluster-list**) についてすでに説明しました。

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、ネイバー単位の設定オプションである **nexthop-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 **nexthop-self** 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定義された「抑制限度」に達するとすぐに、ルートのアドバタイズメントが抑制されます。 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。 ペナルティが事前定義された「再使用限度」を下回ると、ルートのアドバタイズメントの抑制が解除されます。

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。 ルートの抑制解除は 10 秒ごとに実行されます。 ルータは、ペナルティが「再使用限度」の半分未満に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。 必要があれば、この機能は将来的にデフォルトで有効にされる可能性があります。 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 dampening half-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 dampening *half-life-time reuse suppress maximum-suppress-time*

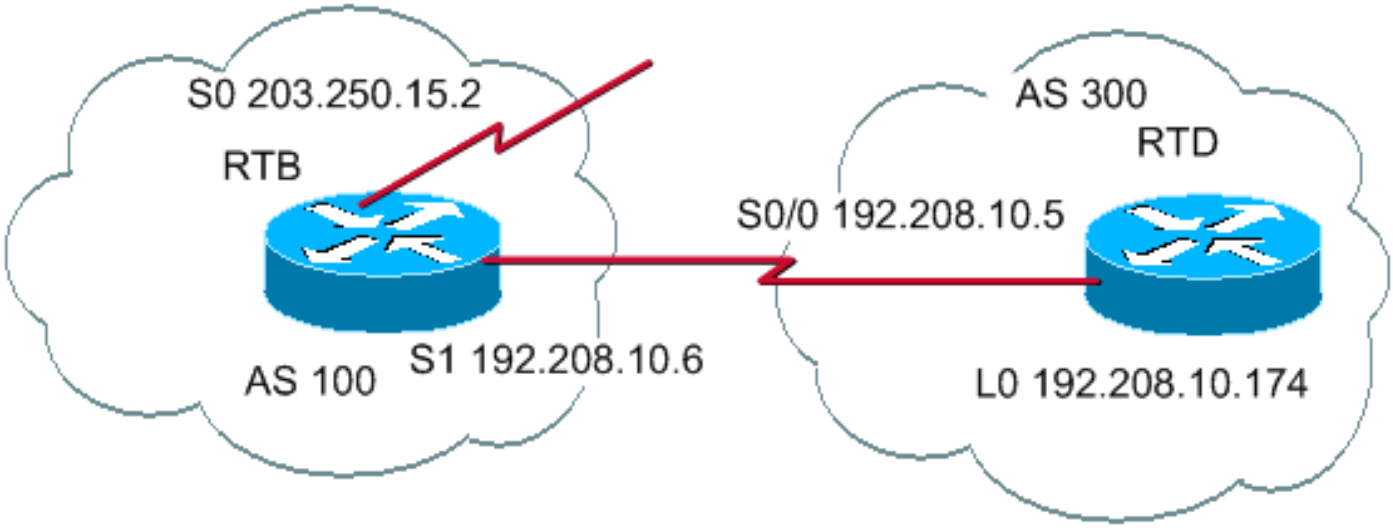
構文の詳細は次のとおりです。

half-life-time：値の範囲は 1 ~ 45 分で、現在のデフォルトは 15 分です。

reuse-value：値の範囲は 1 ~ 20,000 で、デフォルトは 750 です。

suppress-value：値の範囲は 1 ~ 20,000 で、デフォルトは 2000 です。

max-suppress-time：ルートを抑制する最長時間です。 値の範囲は 1 ~ 255 分で、デフォルトは半減期の 4 倍です。



neighbor route-reflector-client

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。

```
RTB# show ip bgp
BGP table version is 24, local router ID is 203.250.15.2 Status codes: s
suppressed, d damped, h history, * valid, > best, i - internal Origin
codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 192.208.10.0     192.208.10.5           0             0 300 i
*> 203.250.15.0     0.0.0.0                0         32768 i
```

ルート フラップをシミュレートするには、RTD で **clear ip bgp 192.208.10.6** 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。

```
RTB# show ip bgp
BGP table version is 24, local router ID is 203.250.15.2 Status codes: s
suppressed, d damped, h history, * valid, > best, i - internal Origin
codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
 h 192.208.10.0     192.208.10.5           0             0 300 i
*> 203.250.15.0     0.0.0.0                0         32768 i
```

192.208.10.0 の BGP エントリは、history 状態になっています。 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。

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

ルートにはフラッピングに対するペナルティが与えられていますが、まだ「抑制限度」を下回っています。 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。

```
RTB# show ip bgp
BGP table version is 32, local router ID is 203.250.15.2 Status codes:
s suppressed, d damped, h history, * valid, > best, i - internal Origin codes:
i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*d 192.208.10.0     192.208.10.5           0             0  300 i
*> 203.250.15.0     0.0.0.0                0       32768  i

RTB# show ip bgp 192.208.10.0
BGP routing table entry for 192.208.10.0 255.255.255.0, version 32
Paths: (1 available, no best path)
300, (suppressed due to dampening)
192.208.10.5 from 192.208.10.5 (192.208.10.174)
      Origin IGP, metric 0, valid, external
Dampinfo: penalty 2615, flapped 3 times in 0:05:18 , reuse in 0:27:00
```

ルートはダンプニング (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 regexp regular-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-list *list*：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-statistics A.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-statistics A.B.C.D m.m.m.m longer-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 regexp regular-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-list *list*：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-statistics A.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 bgp A.B.C.D 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。

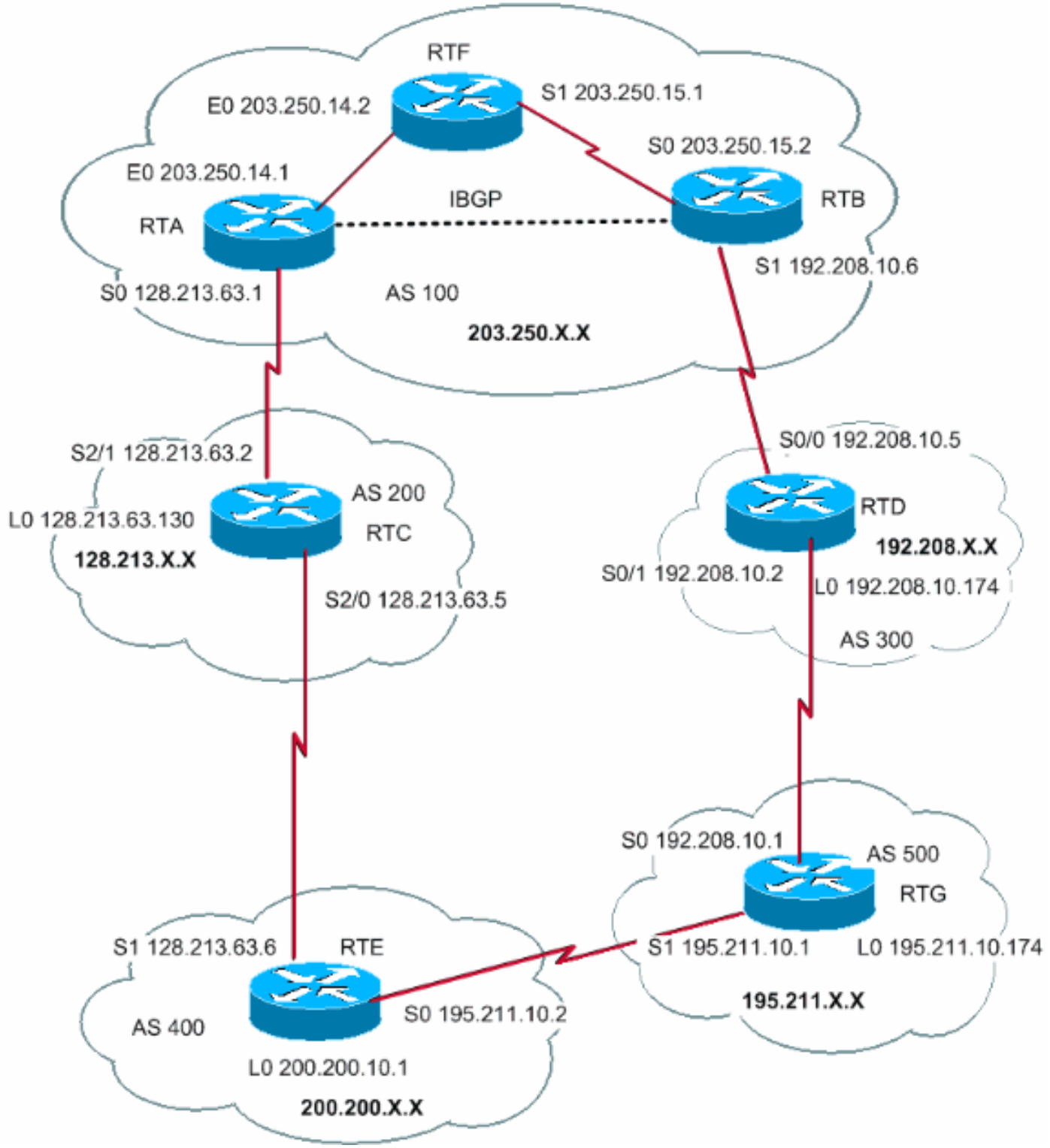
BGP によるパスの選択方法

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: これらは最終的な設定ではありません。

```
RTB# show ip bgp
BGP table version is 32, local router ID is 203.250.15.2 Status codes:
s suppressed, d damped, h history, * valid, > best, i - internal Origin codes:
i - IGP, e - EGP, ? - incomplete
```

```
   Network          Next Hop          Metric LocPrf Weight Path
*d 192.208.10.0     192.208.10.5           0             0  300 i
*> 203.250.15.0     0.0.0.0                0       32768  i

RTB# show ip bgp 192.208.10.0
BGP routing table entry for 192.208.10.0 255.255.255.0, version 32
Paths: (1 available, no best path)
300, (suppressed due to dampening)
192.208.10.5 from 192.208.10.5 (192.208.10.174)
      Origin IGP, metric 0, valid, external
Dampinfo: penalty 2615, flapped 3 times in 0:05:18 , reuse in 0:27:00
```

ネットワークをアドバタイズするには、**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。

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

このテーブルでは、次の表記が使用されます。

先頭の i：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 i：パス情報の送信元が IGP であることを示します。

Path 情報：これは直感的に理解できる情報です。 たとえば、ネットワーク 128.213.0.0 はネクスト ホップが 128.213.63.2 のパス 200 を介して学習されることがわかります。

注: 203.250.15.0 など、ローカルで生成されたエントリのネクスト ホップは 0.0.0.0 と表示されます。

>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 128.213.63.2 を介して 128.213.0.0 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。

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

Gateway of last resort is not set

     203.250.13.0 255.255.255.255 is subnetted, 1 subnets
O       203.250.13.41 [110/75] via 203.250.15.1, 02:50:45, Serial0
     203.250.15.0 255.255.255.252 is subnetted, 1 subnets
C       203.250.15.0 is directly connected, Serial0
O    203.250.14.0 [110/74] via 203.250.15.1, 02:50:46, Serial0
```

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 128.213.63.2 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 128.213.63.0 について学習していません。 RTA の s0 インターフェイスで OSPF を実行してパッシブにすると、 RTB はネクスト ホップ 128.213.63.2 に到達する方法を認識します。 この場合の RTA の設定は次のとおりです。

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

Gateway of last resort is not set

     203.250.13.0 255.255.255.255 is subnetted, 1 subnets
O       203.250.13.41 [110/75] via 203.250.15.1, 02:50:45, Serial0
     203.250.15.0 255.255.255.252 is subnetted, 1 subnets
C       203.250.15.0 is directly connected, Serial0
O    203.250.14.0 [110/74] via 203.250.15.1, 02:50:46, Serial0
```

注: RTA と RTB の間で **bgp nexthopself** 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。

```
RTB# show ip bgp
BGP table version is 10, local router ID is 203.250.15.2
Status codes: s suppressed, d damped, h history, * valid, > best,
i - internal Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*>i128.213.0.0      128.213.63.2           0    100      0 200 i
*>i192.208.10.0     128.213.63.2                100      0 200 400 500
300 i
*>i195.211.10.0     128.213.63.2                100      0 200 400 500 i
*>i200.200.10.0     128.213.63.2                100      0 200 400 i
*>i203.250.13.0     203.250.13.41          0    100      0 i
*>i203.250.14.0     203.250.13.41          0    100      0 i
*> 203.250.15.0     0.0.0.0                0         32768 i
```

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。

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

Gateway of last resort is not set

     203.250.13.0 255.255.255.255 is subnetted, 1 subnets
O       203.250.13.41 [110/75] via 203.250.15.1, 00:04:46, Serial0
     203.250.15.0 255.255.255.252 is subnetted, 1 subnets
C       203.250.15.0 is directly connected, Serial0
O    203.250.14.0 [110/74] via 203.250.15.1, 00:04:46, Serial0
     128.213.0.0 255.255.255.252 is subnetted, 1 subnets
O       128.213.63.0 [110/138] via 203.250.15.1, 00:04:47, Serial0
```

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、128.213.63.0 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ので、RTF はネットワーク 192.208.10.0 および 195.211.10.0 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。

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

Gateway of last resort is not set

B    200.200.10.0 [200/0] via 128.213.63.2, 00:01:07
B    195.211.10.0 [200/0] via 128.213.63.2, 00:01:07
B    192.208.10.0 [200/0] via 128.213.63.2, 00:01:07
     203.250.13.0 is variably subnetted, 2 subnets, 2 masks
O       203.250.13.41 255.255.255.255
           [110/75] via 203.250.15.1, 00:12:37, Serial0
B       203.250.13.0 255.255.255.0 [200/0] via 203.250.13.41, 00:01:08
     203.250.15.0 255.255.255.252 is subnetted, 1 subnets
C       203.250.15.0 is directly connected, Serial0
O    203.250.14.0 [110/74] via 203.250.15.1, 00:12:37, Serial0
     128.213.0.0 is variably subnetted, 2 subnets, 2 masks
B       128.213.0.0 255.255.0.0 [200/0] via 128.213.63.2, 00:01:08
O       128.213.63.0 255.255.255.252
           [110/138] via 203.250.15.1, 00:12:37, Serial0
```

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。

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

Gateway of last resort is not set

     203.250.13.0 255.255.255.255 is subnetted, 1 subnets
O       203.250.13.41 [110/11] via 203.250.14.1, 00:14:15, Ethernet0
     203.250.15.0 255.255.255.252 is subnetted, 1 subnets
C       203.250.15.0 is directly connected, Serial1
C    203.250.14.0 is directly connected, Ethernet0
     128.213.0.0 255.255.255.252 is subnetted, 1 subnets
O       128.213.63.0 [110/74] via 203.250.14.1, 00:14:15, Ethernet0
```

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。

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

Gateway of last resort is not set

     203.250.13.0 255.255.255.255 is subnetted, 1 subnets
O       203.250.13.41 [110/11] via 203.250.14.1, 00:14:15, Ethernet0
     203.250.15.0 255.255.255.252 is subnetted, 1 subnets
C       203.250.15.0 is directly connected, Serial1
C    203.250.14.0 is directly connected, Ethernet0
     128.213.0.0 255.255.255.252 is subnetted, 1 subnets
O       128.213.63.0 [110/74] via 203.250.14.1, 00:14:15, Ethernet0
```

ルーティング テーブルは次のように表示されます。

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

Gateway of last resort is not set

O E2 200.200.10.0 [110/2000] via 203.250.15.1, 00:00:14, Serial0
O E2 195.211.10.0 [110/2000] via 203.250.15.1, 00:00:14, Serial0
O E2 192.208.10.0 [110/2000] via 203.250.15.1, 00:00:14, Serial0
     203.250.13.0 is variably subnetted, 2 subnets, 2 masks
O       203.250.13.41 255.255.255.255
           [110/75] via 203.250.15.1, 00:00:15, Serial0
O E2    203.250.13.0 255.255.255.0
           [110/2000] via 203.250.15.1, 00:00:15, Serial0
     203.250.15.0 255.255.255.252 is subnetted, 2 subnets
C       203.250.15.8 is directly connected, Loopback1
C       203.250.15.0 is directly connected, Serial0
O    203.250.14.0 [110/74] via 203.250.15.1, 00:00:15, Serial0
```

```
     128.213.0.0 is variably subnetted, 2 subnets, 2 masks
O E2    128.213.0.0 255.255.0.0 [110/2000] via 203.250.15.1,
00:00:15,Serial0
O       128.213.63.0 255.255.255.252
           [110/138] via 203.250.15.1, 00:00:16, Serial0
```

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 203.250.15.0 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 203.250.13.0 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 192.208.10.5 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 192.208.10.5 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。

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

Gateway of last resort is not set

O E2 200.200.10.0 [110/2000] via 203.250.15.1, 00:00:14, Serial0
O E2 195.211.10.0 [110/2000] via 203.250.15.1, 00:00:14, Serial0
O E2 192.208.10.0 [110/2000] via 203.250.15.1, 00:00:14, Serial0
     203.250.13.0 is variably subnetted, 2 subnets, 2 masks
O       203.250.13.41 255.255.255.255
           [110/75] via 203.250.15.1, 00:00:15, Serial0
O E2    203.250.13.0 255.255.255.0
           [110/2000] via 203.250.15.1, 00:00:15, Serial0
     203.250.15.0 255.255.255.252 is subnetted, 2 subnets
C       203.250.15.8 is directly connected, Loopback1
C       203.250.15.0 is directly connected, Serial0
O    203.250.14.0 [110/74] via 203.250.15.1, 00:00:15, Serial0
     128.213.0.0 is variably subnetted, 2 subnets, 2 masks
O E2    128.213.0.0 255.255.0.0 [110/2000] via 203.250.15.1,
00:00:15,Serial0
O       128.213.63.0 255.255.255.252
           [110/138] via 203.250.15.1, 00:00:16, Serial0
```

BGP テーブルは次のように表示されます。

```
RTA# show ip bgp
BGP table version is 117, local router ID is 203.250.13.41
Status codes: s suppressed, d damped, h history, * valid, > best,
i -internal Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 128.213.0.0      128.213.63.2           0             0 200 i
*>i192.208.10.0     192.208.10.5           0    100      0 300 i
*>i195.211.10.0     192.208.10.5                100      0 300 500 i
*                   128.213.63.2                         0 200 400 500 i
*> 200.200.10.0     128.213.63.2                         0 200 400 i
```

```
*> 203.250.13.0     0.0.0.0               0         32768 i
*> 203.250.14.0     0.0.0.0               0         32768 i
*>i203.250.15.0     203.250.15.2          0    100      0 i

RTB# show ip bgp
BGP table version is 12, local router ID is 203.250.15.10
Status codes: s suppressed, d damped, h history, * valid, > best,
i -internal Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*>i128.213.0.0      128.213.63.2           0    100      0 200 i
*                   192.208.10.5                        0 300 500 400
200 i
*> 192.208.10.0     192.208.10.5           0             0 300 i
*> 195.211.10.0     192.208.10.5                         0 300 500 i
*>i200.200.10.0     128.213.63.2                100      0 200 400 i
*                   192.208.10.5                         0 300 500 400 i
*>i203.250.13.0     203.250.13.41          0    100      0 i
*>i203.250.14.0     203.250.13.41          0    100      0 i
*> 203.250.15.0     0.0.0.0                0         32768 i
```

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。 **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ことで、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# show ip bgp
BGP table version is 117, local router ID is 203.250.13.41
Status codes: s suppressed, d damped, h history, * valid, > best,
i -internal Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 128.213.0.0      128.213.63.2           0             0 200 i
*>i192.208.10.0     192.208.10.5           0    100      0 300 i
*>i195.211.10.0     192.208.10.5                100      0 300 500 i
*                   128.213.63.2                         0 200 400 500 i
*> 200.200.10.0     128.213.63.2                         0 200 400 i
*> 203.250.13.0     0.0.0.0                0         32768 i
```

```
*> 203.250.14.0     0.0.0.0                0         32768 i
*>i203.250.15.0     203.250.15.2           0    100      0 i

RTB# show ip bgp
BGP table version is 12, local router ID is 203.250.15.10
Status codes: s suppressed, d damped, h history, * valid, > best,
i -internal Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*>i128.213.0.0      128.213.63.2           0    100      0 200 i
*                   192.208.10.5                        0 300 500 400
200 i
*> 192.208.10.0     192.208.10.5           0             0 300 i
*> 195.211.10.0     192.208.10.5                         0 300 500 i
*>i200.200.10.0     128.213.63.2                100      0 200 400 i
*                   192.208.10.5                         0 300 500 400 i
*>i203.250.13.0     203.250.13.41          0    100      0 i
*>i203.250.14.0     203.250.13.41          0    100      0 i
*> 203.250.15.0     0.0.0.0                0         32768 i
```

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 200.200.0.0 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。

```
RTA# show ip bgp
BGP table version is 117, local router ID is 203.250.13.41
Status codes: s suppressed, d damped, h history, * valid, > best,
i -internal Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 128.213.0.0      128.213.63.2           0             0 200 i
*>i192.208.10.0     192.208.10.5           0    100      0 300 i
*>i195.211.10.0     192.208.10.5                100      0 300 500 i
*                   128.213.63.2                         0 200 400 500 i
*> 200.200.10.0     128.213.63.2                         0 200 400 i
*> 203.250.13.0     0.0.0.0                0         32768 i
*> 203.250.14.0     0.0.0.0                0         32768 i
*>i203.250.15.0     203.250.15.2           0    100      0 i

RTB# show ip bgp
BGP table version is 12, local router ID is 203.250.15.10
Status codes: s suppressed, d damped, h history, * valid, > best,
i -internal Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*>i128.213.0.0      128.213.63.2           0    100      0 200 i
*                   192.208.10.5                        0 300 500 400
200 i
*> 192.208.10.0     192.208.10.5           0             0 300 i
*> 195.211.10.0     192.208.10.5                         0 300 500 i
*>i200.200.10.0     128.213.63.2                100      0 200 400 i
*                   192.208.10.5                         0 300 500 400 i
```

```
*>i203.250.13.0     203.250.13.41          0    100      0 i
*>i203.250.14.0     203.250.13.41          0    100      0 i
*> 203.250.15.0     0.0.0.0                0         32768 i
```

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: アドバタイズされるのは AS300 のローカル ルートのみです。 ^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。

```
RTB# show ip bgp regexp ^300$
BGP table version is 14, local router ID is 203.250.15.10
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 192.208.10.0     192.208.10.5           0    300      0 300
```

```
RTC#
hostname RTC

ip subnet-zero

interface Loopback0
 ip address 128.213.63.130 255.255.255.192

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

router bgp 200
 network 128.213.0.0
 neighbor 128.213.63.1 remote-as 100
 neighbor 128.213.63.1 distribute-list 1 out
 neighbor 128.213.63.6 remote-as 400

ip classless
access-list 1 deny   195.211.0.0 0.0.255.255
access-list 1 permit any
```

RTC では、128.213.0.0/16 を集約し、AS100 にインジェクト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。

```
RTB# show ip bgp regexp ^300$
BGP table version is 14, local router ID is 203.250.15.10
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
```

```
*> 192.208.10.0     192.208.10.5           0    300      0 300

RTC#
hostname RTC

ip subnet-zero

interface Loopback0
 ip address 128.213.63.130 255.255.255.192

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

router bgp 200
 network 128.213.0.0
 neighbor 128.213.63.1 remote-as 100
 neighbor 128.213.63.1 distribute-list 1 out
 neighbor 128.213.63.6 remote-as 400

ip classless
access-list 1 deny   195.211.0.0 0.0.255.255
access-list 1 permit any
```

RTG でコミュニティ フィルタリングが使用されています。 no-export コミュニティを RTD に対する 195.211.0.0 アップデートに追加します。 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。

```
RTB# show ip bgp regexp ^300$
BGP table version is 14, local router ID is 203.250.15.10
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 192.208.10.0     192.208.10.5           0    300      0 300

RTC#
hostname RTC

ip subnet-zero

interface Loopback0
 ip address 128.213.63.130 255.255.255.192

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

router bgp 200
 network 128.213.0.0
 neighbor 128.213.63.1 remote-as 100
 neighbor 128.213.63.1 distribute-list 1 out
 neighbor 128.213.63.6 remote-as 400

ip classless
access-list 1 deny   195.211.0.0 0.0.255.255
access-list 1 permit any
```

RTE は 200.200.0.0/16 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。

```
RTA# show ip bgp
BGP table version is 21, local router ID is 203.250.13.41
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 128.213.0.0      128.213.63.2           0    200      0 200 i
*>i192.208.10.0     192.208.10.5           0    300      0 300 i
*> 200.200.0.0/16   128.213.63.2                200      0 200 400 i
*> 203.250.13.0     0.0.0.0                0         32768 i
*> 203.250.14.0     0.0.0.0                0         32768 i
*>i203.250.15.0     203.250.15.2           0    100      0 i

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

Gateway of last resort is 128.213.63.2 to network 200.200.0.0

     192.208.10.0 is variably subnetted, 2 subnets, 2 masks
O E2    192.208.10.0 255.255.255.0
           [110/1000] via 203.250.14.2, 00:41:25, Ethernet0
O       192.208.10.4 255.255.255.252
           [110/138] via 203.250.14.2, 00:41:25, Ethernet0
C    203.250.13.0 is directly connected, Loopback0
     203.250.15.0 is variably subnetted, 3 subnets, 3 masks
O       203.250.15.10 255.255.255.255
           [110/75] via 203.250.14.2, 00:41:25, Ethernet0
O       203.250.15.0 255.255.255.252
           [110/74] via 203.250.14.2, 00:41:25, Ethernet0
B       203.250.15.0 255.255.255.0 [200/0] via 203.250.15.2, 00:41:25
C    203.250.14.0 is directly connected, Ethernet0
     128.213.0.0 is variably subnetted, 2 subnets, 2 masks
B       128.213.0.0 255.255.0.0 [20/0] via 128.213.63.2, 00:41:26
C       128.213.63.0 255.255.255.252 is directly connected, Serial0
O*E2 0.0.0.0/0 [110/1000] via 203.250.14.2, Ethernet0/0
B*   200.200.0.0 255.255.0.0 [20/0] via 128.213.63.2, 00:02:38

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

Gateway of last resort is 203.250.15.2 to network 0.0.0.0

     192.208.10.0 is variably subnetted, 2 subnets, 2 masks
O E2    192.208.10.0 255.255.255.0
           [110/1000] via 203.250.15.2, 00:48:50, Serial1
O       192.208.10.4 255.255.255.252
           [110/128] via 203.250.15.2, 01:12:09, Serial1
     203.250.13.0 is variably subnetted, 2 subnets, 2 masks
O       203.250.13.41 255.255.255.255
```

```
           [110/11] via 203.250.14.1, 01:12:09, Ethernet0
O E2    203.250.13.0 255.255.255.0
           [110/2000] via 203.250.14.1, 01:12:09, Ethernet0
     203.250.15.0 is variably subnetted, 2 subnets, 2 masks
O       203.250.15.10 255.255.255.255
           [110/65] via 203.250.15.2, 01:12:09, Serial1
C       203.250.15.0 255.255.255.252 is directly connected, Serial1
C    203.250.14.0 is directly connected, Ethernet0
     128.213.0.0 is variably subnetted, 2 subnets, 2 masks
O E2    128.213.0.0 255.255.0.0
           [110/2000] via 203.250.14.1, 00:45:01, Ethernet0
O       128.213.63.0 255.255.255.252
           [110/74] via 203.250.14.1, 01:12:11, Ethernet0
O E2 200.200.0.0 255.255.0.0 [110/2000] via 203.250.14.1, 00:03:47,
Ethernet0
O*E2 0.0.0.0 0.0.0.0 [110/1000] via 203.250.15.2, 00:03:33, Serial1
```

注: RTF のルーティング テーブルには、AS300 に対してローカルなネットワーク (192.208.10.0 など) に到達するルートが、RTB 経由であることが示されています。 その他の既知のネットワーク (200.200.0.0 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。

```
RTB# show ip bgp
BGP table version is 14, local router ID is 203.250.15.10
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*>i128.213.0.0      128.213.63.2           0    200      0 200 i
*> 192.208.10.0     192.208.10.5           0    300      0 300 i
*>i200.200.0.0/16   128.213.63.2                200      0 200 400 i
*>i203.250.13.0     203.250.13.41          0    100      0 i
*>i203.250.14.0     203.250.13.41          0    100      0 i
*> 203.250.15.0     0.0.0.0                0         32768 i

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

Gateway of last resort is 192.208.10.5 to network 192.208.10.0

 *   192.208.10.0 is variably subnetted, 2 subnets, 2 masks
B*      192.208.10.0 255.255.255.0 [20/0] via 192.208.10.5, 00:50:46
C       192.208.10.4 255.255.255.252 is directly connected, Serial1
     203.250.13.0 is variably subnetted, 2 subnets, 2 masks
O       203.250.13.41 255.255.255.255
           [110/75] via 203.250.15.1, 01:20:33, Serial0
O E2    203.250.13.0 255.255.255.0
           [110/2000] via 203.250.15.1, 01:15:40, Serial0
     203.250.15.0 255.255.255.252 is subnetted, 2 subnets
C       203.250.15.8 is directly connected, Loopback1
C       203.250.15.0 is directly connected, Serial0
O    203.250.14.0 [110/74] via 203.250.15.1, 01:20:33, Serial0
     128.213.0.0 is variably subnetted, 2 subnets, 2 masks
O E2    128.213.0.0 255.255.0.0 [110/2000] via 203.250.15.1, 00:46:55, Serial0
O       128.213.63.0 255.255.255.252
```

```
          [110/138] via 203.250.15.1, 01:20:34, Serial0
O*E2 0.0.0.0/0 [110/2000] via 203.250.15.1, 00:08:33, Serial0
O E2 200.200.0.0 255.255.0.0 [110/2000] via 203.250.15.1, 00:05:42, Serial0
```

関連情報

- 「BGP： よく寄せられる質問（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 ネットワークを冗長構成にする方法
- 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
- BGP に関するトラブルシューティング
- BGP スキャナまたは BGP 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：設定例
- BGP に関するサポート ページ
- テクニカル サポートとドキュメント – Cisco Systems