BGP ケース スタディ

目次

概要

このドキュメントでは、Border Gateway Protocol（BGP; ボーダー ゲートウェイ プロトコル）に関する 5 つのケース スタディを紹介しています。

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、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 Interior Gateway Protocols（IGP; 内部ゲートウェイ プロトコル）を使用して AS 内部のルーティング情報を交換できます。 これらのルータは、Exterior Gateway Protocol を使用して AS 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 つの BGP ルータは、相互に TCP 接続を確立します。 これらのルータは、ピア ルータです。 ピア ルータは接続パラメータを開いて確認するためにメッセージを交換します。

BGP ルータはネットワーク到達可能性情報を交換します。 この情報は、主に宛先ネットワークに到着するためにルートで使用するフル 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、ループのない AS のグラフの作成に役立ちます。 このグラフは、ルーティング動作に対していくつかの制限を適用するためのルーティング ポリシーをどこに適用するかについても示します。

BGP ルーティング情報を交換するために TCP 接続が確立された 2 つのルータは、「ピア」または「隣接ルータ」と呼ばれます。 BGP ピアは最初に、BGP ルーティング テーブル全体を交換します。 この交換後、ピアは、ルーティング テーブルが変更されるたびに、差分更新を送ります。BGP は、BGP テーブルのバージョン番号を保持します。 バージョン番号は、すべての BGP ピアで同じです。 ルーティング情報の変更により 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 パケットの送信により、BGP ピア間の接続が機能していることが確認されます。 通知パケットは、エラーや特殊な状況が発生した場合に送信されます。

eBGP と iBGP

AS に複数の BGP スピーカがある場合、AS は他の AS の中継サービスとして機能することができます。 このセクションの図に示すように、AS200 は AS100 と AS300 の中継 AS です。

情報を外部 AS に送るには、ネットワークの到達可能性が保証されている必要があります。 ネットワークの到達可能性を保証するために、次のプロセスが実行されます。

AS 内のルータ間の Internal BGP (iBGP; 内部 BGP) ピアリング

AS 内で動作している IGP への BGP 情報の再配布

BGP が 2 つの異なる AS に属するルータ間で動作している場合は、Exterior BGP (eBGP; 外部 BGP) 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 を有効にして設定するには、次の手順を実行します。

BGP を使用して対話を行う 2 つのルータ (RTA と RTB) があるものと仮定します。 最初の例では、RTA と RTB は異なる AS 内に存在します。 2 番目の例では、両方のルータが同じ AS に属しています。

ルータのプロセスと、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、ルータ上で BGP を有効にします。

router bgp *autonomous-system* RTA# router bgp 100 RTB# router bgp 200
これらの文は、RTA が BGP を実行し、AS100 に属することを示します。 RTB は BGP を実行し、AS200 に属します。

BGP 隣接ルータを定義します。

BGP 隣接ルータの構成は、BGP を使用して対話を試みるルータを示します。 このプロセスについては、「BGP ネイバーの作成」セクションを参照してください。

BGP 隣接ルータの作成

2 つの BGP ルータは、相互に TCP 接続を確立することによって隣接ルータになります。 TCP 接続は、2 つのピア ルータがルーティング アップデートの交換を開始するために欠かせません。

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、ルータで稼動する BGP のバージョン、BGP ルータ ID、キープアライブ保留時間などが含まれます。 これらの値を確認して承認した後、隣接ルータの接続が確立されます。 「Established」以外の状態は、2 つのルータが隣接ルータになっておらず、BGP アップデートを交換できないことを示してます。

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*** の部分では、BGP を使用して接続するルータの AS 番号を指定します。 ***ip-address*** の部分では、eBGP の直接接続でのネクストホップ アドレスを指定します。 iBGP の場合は、***ip-address*** の部分で他のルータの IP アドレスを指定します。

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方法の 1 つは、2 つの IP アドレス間で拡張 ping を行うことです。 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、パケットが送信され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隣接ルータの接続をリセットする*必要*があります。

clear ip bgp address

注: ***address*** の部分では、隣接ルータのアドレスを指定します。

clear ip bgp *

このコマンドは、すべての隣接ルータ接続をクリアします。

デフォルトでは、BGP セッションは最初に BGP バージョン 4 を使用して開始し、必要であれば旧バージョンへのダウンワード移行をネゴシエートします。 ネゴシエーションを回避して、ルータの隣接ルータとの通信に、このバージョンが使用されるようにできます。 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。

```
RTA#
router bgp 100
neighbor 129.213.1.1 remote-as 200

RTB#
router bgp 200
neighbor 129.213.1.2 remote-as 100
neighbor 175.220.1.2 remote-as 200

RTC#
router bgp 200
neighbor 175.220.212.1 remote-as 200
```

この例では、RTA と RTB では eBGP が実行されます。 RTB と RTC では iBGP が実行されます。 リモート AS 番号は外部 AS または内部 AS をポイントし、eBGP または iBGP のどちらかを示します。 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、稼働している IGP があり、これにより 2 つの隣接ルータ間での通信が可能になっている必要があります。

このセクションでは、show ip bgp neighbors コマンドが表示する情報の例を示しています。

注: 特に BGP の状態に注意してください。 Established 以外の状態は、ピアが確立されていないことを示しています。

注: また、次の項目に注意してください。

BGP version (値 = 4)

remote router ID

この番号は、ルータの最上位の IP アドレスか最上位のループバック 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。 新しい情報が入ってくると、テーブルのバージョンの値が上がります。 バージョンが上がり続けている場合は、ルート フラップが発生していて、ルートが継続的に更新されていることを示します。

```
# 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

ループバック インターフェイスを使用して隣接ルータを定義するのは 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隣接ルータの 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します。 eBGP の場合、ピア ルータは直接接続されることが多く、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。 隣接ルータは BGP に、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 隣接ルータへの TCP 接続を開始することを伝える必要があります。 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*

次の例では、このコマンドの使用法を説明しています。

```
RTA#
     router bgp 100
     neighbor 190.225.11.1 remote-as 100
     neighbor 190.225.11.1 update-source loopback 1
RTB#
     router bgp 100
     neighbor 150.212.1.1 remote-as 100
```

この例では、RTA と RTB は AS100 の内部で 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 BGP に対し、TCP 隣接ルータ接続の送信元としてループバック IP アドレスを強制的に使用させる必要があります。 この動作を強制するために、RTA は **update-source** ***interface-type interface-number*** を追加します。その結果、このコマンドは **neighbor 190.225.11.1 update-source loopback 1** になります。 この文により、BGP が隣接ルータ 190.225.11.1 と通信する場合は、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 は隣接ルータとして、RTB の物理インターフェイス IP アドレス 190.225.11.1 を使用しています。 この IP アドレスが使用されるため、RTB では特別な設定は必要ありません。 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

状況により、Cisco 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間で 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できます。 eBGP マルチホップにより、直接接続されていない 2 つの外部ピア間の隣接ルータ接続が可能になります。 マルチホップは、eBGP でだけ使用され、iBGP では使用されません。 次の例は、eBGP マルチホップについて説明しています。

```
RTA#
router bgp 100
neighbor 180.225.11.1 remote-as 300
neighbor 180.225.11.1 ebgp-multihop
RTB#
router bgp 300
neighbor 129.213.1.2 remote-as 100
```

RTA は、直接接続されていない外部隣接ルータを示します。 RTA は neighbor ebgp-multihop コマンドの使用を示す必要があります。 一方、RTB は、直接接続されている隣接ルータ (129.213.1.2) を示しています。 このような理由でダイレクト接続は、RTB **neighbor ebgp-multihop コマンドを**必要としません。 また、接続されていない隣接ルータを互いに到達可能にするように、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。

「eBGP 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 150.10.1.1 255.255.255.0
router bgp 100
neighbor 160.10.1.1 remote-as 200
neighbor 160.10.1.1 ebgp-multihop
neighbor 160.10.1.1 update-source loopback 0
network 150.10.0.0

ip route 160.10.0.0 255.255.0.0 1.1.1.2
ip route 160.10.0.0 255.255.0.0 2.2.2.2
RTB#
int loopback 0
```

```
ip address 160.10.1.1 255.255.255.0
router bgp 200
neighbor 150.10.1.1 remote-as 100
neighbor 150.10.1.1 update-source loopback 0
neighbor 150.10.1.1 ebgp-multihop
network 160.10.0.0


ip route 150.10.0.0 255.255.0.0 1.1.1.1
ip route 150.10.0.0 255.255.0.0 2.2.2.1
```

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** の使用方法を示しています。 この例は、パラレル シリアル回線上で 2 つの eBGP スピーカ間のロード バランシングを実現するための回避策です。 通常の状況では、BGP はパケットを送信する回線の一方を選択し、ロード バランシングは発生しません。 ループバック インターフェイスを導入することにより、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。 ここではスタティック 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 には、ネクストホップ 160.10.1.1 に到達するために、2 つの選択肢があります。 1 つのパスは 1.1.1.2 経由、もう 1 つのパスは 2.2.2.2 経由です。 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 マップが頻繁に使用されます。 BGP のコンテキストでは、ルート マップは、ルーティング情報を制御および修正するための方式です。 ルーティング情報の制御および修正は、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の定義によって発生します。 また、ルーティング情報の制御は、BGP との間のルート注入時にも発生する場合があります。 ルート マップの形式は次のようになります。

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

map tag は、route map に付ける単なる名前です。 同じルート マップまたは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、同じ名前ですでに設定されているルート マップのリストの中で、新しいルート マップのポジションを示します。

次の例では、MYMAP というルート 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。

route-map MYMAP permit 10 (最初の条件セットがここに入ります。)

route-map MYMAP permit 20 (2 番目の条件セットがここに入ります。)

ルート マップ MYMAP を着信ルートまたは発信ルートに適用する場合、最初の条件セットはインスタンス 10 経由で適用されます。 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

に一致するエントリがある場合、そのアップデートのメトリックは 5 に設定されます。 次に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*
ここで一致基準が満たされて、**許可**された場合は、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ることになります。

一致基準が満たされて、**拒否**された場合、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ることになります。

一致基準が満たされず、**許可または拒否**された場合は、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。 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を終了するまで、次のインスタンスのチェックは続きます。 一致することなくリストを終了した場合、このルートは**許可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、この制限がありません。

match の関連コマンドには、次のものがあります。

- 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 の関連コマンドには、次のものがあります。

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

次に、ルート マップの例をいくつか示します。

例 1

RTA および RTB が Routing Information Protocol (RIP) を実行していて、RTA および RTC が BGP を実行していると仮定します。 RTA は BGP 経由でアップデートを取得し、RIP にアップデートを再配布します。 この設定を使用できることをことを RTA 2 のメトリックの 170.10.0.0 についての RTB ルーティングおよび 5.のメトリックの他のすべてのルーティングにこの場合再配布したいと思う仮定して下さい:

```
RTA#
router rip
network 3.0.0.0
```

```
network 2.0.0.0
network 150.10.0.0
passive-interface Serial0
redistribute bgp 100 route-map SETMETRIC

router bgp 100
neighbor 2.2.2.3 remote-as 300
network 150.10.0.0

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

route-map SETMETRIC permit 20
set metric 5

access-list 1 permit 170.10.0.0 0.0.255.255
```

この例では、IP アドレス 170.10.0.0 に一致したルートは、メトリック 2 になります。 その後、ルート マップ リストから抜けます。 一致するエントリがない場合は、ルート マップ リストの検索が続けられ、その他すべてがメトリック 5 に設定されます。

注:「match 文のいずれとも一致しないルートはどうなるか」という疑問を持つことは常に重要です。 そのようなルートは、デフォルトでは廃棄されます。

例 2

例 1 で、170.10.0.0 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合する場合、ルート マップを着信で適用することはできません。 そのため、RTC では発信ルート マップを使用する必要があります。

```
RTC#

router bgp 300
network 170.10.0.0
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-map STOPUPDATES out

route-map STOPUPDATES permit 10
match ip address 1

access-list 1 deny 170.10.0.0 0.0.255.255
access-list 1 permit 0.0.0.0 255.255.255.255
```

ここまでに BGP の開始方法および隣接ルータの定義方法について説明したので、次にネットワーク情報の交換を開始する方法について説明します。

BGP を使用してネットワーク情報を送信するには、さまざまな方法があります。 次のセクションでは、これらの方式について 1 つずつ説明します。

- ネットワーク コマンド
- 再配布

スタティック ルートおよび再配布

ネットワーク 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 を使用するこれまでの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で BGP を実行するためのものではありません。 代わりに、BGP に対して、BGP がこのルータから発信するのがどのネットワークかを指定します。 BGP バージョン 4（BGP4）ではサブネット化およびスーパーネット化を処理できるので、このコマンドではマスク部分が使用されます。 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。

```
RTA#
router bgp 1
network 192.213.0.0 mask 255.255.0.0
ip route 192.213.0.0 255.255.0.0 null 0
```
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 つのオクテットまたは最初の 16 ビットをアドバタイズしていることを示しています。

注: 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。これは、スタティック ルートにより一致エントリがルーティング テーブルに挿入されるためです。

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。 この他に、IGP を BGP に再配布するという方法もあります。 IGP には、IGRP、Open Shortest Path First（OSPF）プロトコル、RIP、Enhanced Interior Gateway Routing Protocol（EIGRP）、その他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。 慎重なフィルタリングを行って、アドバタイズするインターネット専用のルートに送信し、すべてのルートを送らないよう注意してください。 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。

```
RTC#
router eigrp 10
network 175.220.0.0
redistribute bgp 200
default-metric 1000 100 250 100 1500
```

```
router bgp 200
neighbor 1.1.1.1 remote-as 300
network 175.220.0.0 mask 255.255.0.0
!--- This limits the networks that your AS originates to 175.220.0.0.
```

再配布を代わりに使用すると、次のようになります。

```
RTC#
router eigrp 10
network 175.220.0.0
redistribute bgp 200
default-metric 1000 100 250 100 1500

router bgp 200
neighbor 1.1.1.1 remote-as 300
redistribute eigrp 10
!--- EIGRP injects 129.213.1.0 again into BGP.
```

この再配布により、AS から 129.213.1.0 が発信されます。 ここでユーザは 129.213.1.0 の発信元ではなくなり、 AS100 が発信元になります。 そのため、AS によってネットワークからソースが送出されないように、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。

```
RTC#
router eigrp 10
network 175.220.0.0
redistribute bgp 200
default-metric 1000 100 250 100 1500

router bgp 200
neighbor 1.1.1.1 remote-as 300
neighbor 1.1.1.1 distribute-list 1 out
redistribute eigrp 10

access-list 1 permit 175.220.0.0 0.0.255.255
```

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 での再配布とは若干異なります。 **redistribute ospf 1** を **router bgp** の下で発行した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 ルートおよび再配布

常にスタティック ルートを使用して、ネットワークまたはサブネットを発信できます。 唯一の違いは、BGP ではこれらのルートが、起点 (origin) が不完全または不明であると見なされることです。 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 175.220.0.0
redistribute bgp 200
default-metric 1000 100 250 100 1500

router bgp 200
neighbor 1.1.1.1 remote-as 300
```

```
redistribute static
...
ip route 175.220.0.0 255.255.255.0 null0
....
```
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、パケットを受信した際に 175.220.0.0 よりも明確な一致がある場合（これは実際に存在します）、ルータはパケットをその一致先に送信します。 それ以外の場合、ルータではパケットが無視されます。 この方式は、スーパーネットをアドバタイズする優れた方法です。

このドキュメントでは、さまざまな方式を使用して AS からルートを発信する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いずれかの隣接ルータを経由して学習した他の 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 は学習した情報をあるピアから別のピアへ伝えます。 **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。

再配布は常に、IGP に BGP を注入するための方式です。

次に例を示します。

```
RTA#
router bgp 100
neighbor 150.10.20.2 remote-as 300
network 150.10.0.0

RTB#
router bgp 200
neighbor 160.10.20.2 remote-as 300
network 160.10.0.0

RTC#
router bgp 300
neighbor 150.10.20.1 remote-as 100
neighbor 160.10.20.1 remote-as 200
network 170.10.00
```
注: RTC にネットワーク 150.10.0.0 またはネットワーク 160.10.0.0 を生成させ、これらのネットワークが AS100 と AS200 から到着したときに、これらを伝播する必要がない場合、これらのネットワークは RTC には必要ありません。 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に注意してください。 この拒否により、ループフリーなドメイン間トポロジが保証されます。

たとえば、このセクションの例にある AS200 に、AS100 への直接 BGP 接続があると仮定します。 RTA はルート 150.10.0.0 を生成し、AS300 にルートを送ります。 その後、RTC はこのルートを AS200 に渡し、起点（origin）を AS100 として保持します。 RTB は AS100 に 150.10.0.0 を渡します。起点（origin）は AS100 のままです。 RTA では、アップデートが自身の AS から発信されていることが検知され、このアップデートは無視されます。

iBGP

AS を他の AS の中継システムとして機能させるには、iBGP を使用します。 eBGP 経由で学習して、IGP へ再配布し、さらに別の AS へ再配布することによって、同じことができるというのは

本当でしょうか。 それは本当です。ただし、iBGP には、AS 内に情報を交換するための、より柔軟で効率的な方法があります。 たとえば、iBGP には、ローカル プリファレンスを使用して AS からの最適な出口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 190.10.50.1 remote-as 100
neighbor 170.10.20.2 remote-as 300
network 150.10.0.0

RTB#
router bgp 100
neighbor 150.10.30.1 remote-as 100
neighbor 175.10.40.1 remote-as 400
network 190.10.50.0

RTC#
router bgp 400
neighbor 175.10.40.2 remote-as 100
network 175.10.0.0
```

注: BGP スピーカが自身の AS (iBGP) 内の他の BGP スピーカ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、その情報を自身の AS 内の他の BGP スピーカには再配布しない点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、この情報を自身の AS 外にある別の BGP スピーカに再配布します。 このため、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。

このセクションの図では、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 へ送られる BGP アップデートは、AS の外部にある RTE へ送信されます。 これらのアップデートは、AS 内部にある RTD へは送信されません。 このため、アップデートのフローを中断しないように、RTB と RTD の間に iBGP ピアを作成する必要があります。

BGP 決定アルゴリズム

BGP がさまざまな自律システムからさまざまな宛先に関するアップデートを受信すると、プロトコルは特定の宛先に到達するために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、さまざまな**アトリビュート**を基準にして、この決定を行います。このアトリビュートには、ネクストホップ、管理上の重み付け (weight) 、ローカル プリファレンス、ルートの起点 (origin) 、パスの長さ、起点 (origin) コード、メトリック、その他があります。

BGP は常に最適なパスを隣接ルータに伝搬します。 詳細は、『BGP で最適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 2」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ています。

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、そのアップデートに AS 番号がプリペンドされます。 AS_PATH アトリビュートは、実際には、宛先に到達するため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 の順序付き数学的集合 { } 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 190.10.0.0 をアドバタイズしています。 そのルートが AS300 をたどると、RTC はネットワークの末尾に自身の AS 番号をアペンドします。 そのため、190.10.0.0 が RTA に到達すると、ネットワークには 2 つの AS 番号が付加されています (最初に 200、次に 300)。 RTA からは、190.10.0.0 に到達するためのパスは (300、200) となります。

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 は、170.10.0.0 に到達するためにパス (300、100) をとる必要があります。 つまり RTB は、AS300 をたどり、次に AS100 をたどります。 RTC は、190.10.0.0 に到達するためにパス (200) をたどり、170.10.0.0 に到達するためにパス (100) をたどる必要があります。

起点 (origin) アトリビュート

起点 (origin) は、パス情報の生成元を定義する必須アトリビュートです。 オリジン アトリビュートは次の 3 種類の値をとることができます。

IGP — ネットワーク レイヤ到着可能性情報 (NLRI) は発生現在でに内部です。 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 テーブルの i は、IGP を示しています。

EGP：NLRI が Exterior Gateway Protocol (EGP; エクステリア ゲートウェイ プロトコル) を通じて学習されたことを示します。 BGP テーブルの e は、EGP を示しています。

INCOMPLETE：NLRI が不明であるか、他の手段によって学習されたことを示します。 INCOMPLETE は通常、他のルーティング プロトコルからルートが BGP に再配布されて、ルートの起点 (origin) が不完全である場合に発生します。 BGPテーブルで不完全示します。

```
RTA#
router bgp 100
neighbor 190.10.50.1 remote-as 100
neighbor 170.10.20.2 remote-as 300
network 150.10.0.0
redistribute static

ip route 190.10.0.0 255.255.0.0 null0

RTB#
router bgp 100
neighbor 150.10.30.1 remote-as 100
network 190.10.50.0
RTE#
router bgp 300
neighbor 170.10.20.1 remote-as 100
```

```
network 170.10.0.0
```
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 であり、ルートの起点（origin）が IGP であることを意味します。 また、RTA は i を経由して 190.10.50.0 に到達します。 「i」は、そのエントリが同じ AS 内にあり、起点（origin）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。 「100i」は、次の AS が 100 であり、起点（origin）が IGP であることを意味します。 また、RTE は 100？を経由して 190.10.0.0 に到達します。 "100 か」。 、そして原点が不完全である意味し、ことスタティック ルートからであることを次の AS が 100 来ます。

BGP ネクストホップ属性

BGP ネクストホップ アトリビュートは、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。 このセクションの例では、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、プロトコルの規定により、eBGP によってアドバタイズされたネクストホップは iBGP に伝達されます。 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。 そのため、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 になります。

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 宛てのパケットを廃棄します。 たとえば、RTB で iGRP が動作している場合は、RTA のネットワーク 170.10.0.0 でも iGRP を実行できます。 RTC へのリンクでは、BGP の交換だけが行われるように、iGRP をパッシブにする必要があります。

```
RTA#
router bgp 100
neighbor 170.10.20.2 remote-as 300
neighbor 150.10.50.1 remote-as 100
network 150.10.0.0
RTB#
router bgp 100
neighbor 150.10.30.1 remote-as 100
RTC#
router bgp 300
neighbor 170.10.20.1 remote-as 100
network 170.10.0.0
```
注: 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。

注: RTA は、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 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 NonBroadcast MultiAccess（NBMA; 非ブロードキャスト マルチアクセス）ネットワークを扱う場合は、特別な注意が必要です。 詳細は、「BGP ネクストホップ（マルチアクセス ネットワーク）」セクションと「BGP ネクストホップ（NBMA）」セクション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）

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

AS300 の RTC と RTD で OSPF が動作しているとします。 RTC は、RTA との間で BGP を動作させています。 RTC は 170.10.20.3 を経由してネットワーク 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 を使用します。 RTC では、自身の IP アドレスである 170.10.20.2 を使用しません。 RTA、RTC、および RTD の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、RTC はこのアドレスを使用します。 RTA が 180.20.0.0 に到達するには、ネクストホップとして RTD を使用するほうが、RTC 経由で余分にホップするよりも理にかなっています。

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 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)

この図では、共有メディアはクラウド (雲の絵) として示されています。 共通メディアがフレームリレーまたは NBMA クラウドである場合は、イーサネット経由で接続している場合とまったく同様の動作になります。 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 から RTD への直接の Permanent Virtual Circuit (PVC; 相手先固定接続) がないために、RTA がネクストホップに到達できないことです。 この場合は、ルーティングが失敗します。

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。

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

RTC はネクストホップを 170.10.20.2 として 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 では eBGP が実行されています。 RTB と RTC では eBGP が実行されています。 RTA および RTB では特定の種類の IGP (RIP、IGRP、または他のプロトコルなど) が実行されています。 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 の場合は 120

IGRP の場合は 100

EIGRP の場合は 90

OSPF の場合は 110

RTA は、次の 2 つのルーティング プロトコルを経由して 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 の距離は次のようになります。

external-distance（外部距離）：20

internal-distance（内部距離）：200

local-distance（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、距離がより短いので、RTC 経由で eBGP を選択します。

RTA が 160.10.0.0 について RTB（IGP）から学習するようにするには、次の 2 通りの方法があります。

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れば、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。

設定されるネットワークは、IGP 経由で到達しようとするネットワークです。 BGP では、このネットワークは、BGP アップデートでアドバタイズされない点を除き、ローカルに割り当てられ

たネットワークと同様に扱われます。

```
RTA#
router eigrp 10

network 150.10.0.0

router bgp 100
neighbor 2.2.2.1 remote-as 300
network 160.10.0.0 backdoor
```

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ません。

RTA は、距離 90 の RTB から EIGRP 経由で 160.10.0.0 を学習します。 また、RTA は、距離 20 の RTC からも eBGP 経由でこのアドレスを学習します。 通常 eBGP はプリファレンスですが、**network backdoor コマンド**が理由で、EIGRP はプリファレンスです。

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す。 AS300 の RTC が 170.10.0.0 に関するアップデートを送信します。 RTA および RTB は iBGP を実行しています。したがって、RTB はアップデートを取得し、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 経由で伝送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は、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。 この時点では、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 に到達可能であることを AS400 にアドバタイズし始めると、RTD から RTB を向かう 170.10.0.0 宛てのトラフィックが流れますが、RTE で廃棄されます。

同期の規定では、AS が他の AS から第 3 の AS へトラフィックを送っている場合、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になっています。 BGP は、IGP が AS 内でルートを伝搬するまで待機します。 その後、BGP はそのルートを外部ピアへアドバタイズします。

このセクションの例では、RTB は IGP を通じて 170.10.0.0 に関する情報が伝播されるまで待機します。 その後、RTB は RTD にアップデートを送信し始めます。 RTB に 170.10.0.0 を指すスタティック ルートを追加すれば、IGP によってその情報が伝搬されたと RTB を思い込ませることができます。 そのためには、他のルータが 170.10.0.0 に到達できるようにする必要があります。

同期の無効化

場合によっては、同期が不要であることがあります。 別の AS からのトラフィックが特定の AS を通過しないようにすると、同期を無効にできます。 また、AS 内のすべてのルータで BGP が動作している場合も、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 のコンバージ速度が向上します。

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で BGP が動作していて、IGP が動作していない場合、ルータでは IGP が動作していないことを知る手段がありません。 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るため、そのルートを外部ピアに送信できません。 この場合、ルーティングを正常に動作させるためには、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 150.10.0.0
neighbor 1.1.1.2 remote-as 400
neighbor 3.3.3.3 remote-as 100
no synchronization
!--- RTB puts 170.10.0.0 in its IP routing table and advertises the network !--- to RTD, even if
RTB does not have an IGP path to 170.10.0.0. RTD# router bgp 400 neighbor 1.1.1.1 remote-as 100
network 175.10.0.0 RTA# router bgp 100 network 150.10.0.0 neighbor 3.3.3.4 remote-as 100
```

ウェイト属性

ウェイト アトリビュートは、Cisco で定義されたアトリビュートです。 このアトリビュートは、ウェイトを使用して最適なパスを選択します。 ウェイトは、ルータにローカルに割り当てられます。 この値が意味を持つのは、特定のルータだけです。 この値は、別のルータに伝搬されたり、ルート アップデートによって伝達されたりはしません。 ウェイトは 0 ~ 65,535 の数値を取ることができます。 このルータが発信元のパスのウェイトはデフォルトで 32,768 に設定されており、他のパスのウェイトは 0 に設定されています。

1 つの宛先に対して複数のルートが存在する場合は、ウェイト値の高いルートが優先されます。 このセクションの例を参照してください。 RTA は AS4 からのネットワーク 175.10.0.0 について学習しています。 RTA は、アップデートを RTC に伝搬します。 また、RTB も AS4 からのネットワーク 175.10.0.0 について学習しています。 RTB は、アップデートを RTC に伝搬します。 これで RTC は 175.10.0.0 に到達可能なルートを 2 つ持つこととなり、どちらを使用するかを決定する必要があります。 RTA から伝播されるアップデートのウェイトが RTB から伝播されるアップデートのウェイトよりも大きくなるように RTC で設定する場合は、ネクストホップとして RTA を使用して 175.10.0.0 に到達するように RTC に強制します。 このウェイト設定を実現するには、複数の方式があります。

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。

```
RTC#
router bgp 300
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 weight 200
!--- The route to 175.10.0.0 from RTA has a 200 weight. neighbor 2.2.2.2 remote-as 200
```

```
    neighbor 2.2.2.2 weight 100 !--- The route to 175.10.0.0 from RTB has a 100 weight.
```

より大きいウェイト値を持つ 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 リストを使用しても同じ結果が得られます。

```
RTC#
router bgp 300
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 filter-list 5 weight 200
neighbor 2.2.2.2 remote-as 200
neighbor 2.2.2.2 filter-list 6 weight 100
...
ip as-path access-list 5 permit ^100$
!--- This only permits path 100. ip as-path access-list 6 permit ^200$ ...
```

ルート マップを使用しても同じ結果が得られます。

```
RTC#
router bgp 300
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 route-map setweightin in
neighbor 2.2.2.2 remote-as 200
neighbor 2.2.2.2 route-map setweightin in
...
ip as-path access-list 5 permit ^100$
...

route-map setweightin permit 10
match as-path 5
set weight 200
!--- Anything that applies to access list 5, such as packets from AS100, has weight 200. route-
map setweightin permit 20 set weight 100 !--- Anything else has weight 100.
```

注: バックアップとして IGP パスの MPLS VPN BGP パスを選ぶためにウエイトを修正できます。

注: 詳細については、この Ciscoサポート コミュニティ 資料を参照して下さいルータを設定する方法をプライマリおよび障害状態の優先パスを持つためにおよびプライマリ パス リカバリで再ルーティングするために記述する: IGP バックアップの MPLS VPN BGP パスを選ぶこと

ローカル プレファレンス属性

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 ルータにだけ関連するウェイト アトリビュートとは異なり、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アトリビュートです。

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。 このセクションの例で示すように、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: 考慮事項に奪取 されるべき変更のためにソフト リセットを (すなわち、ルータの BGPプロセスをクリアして下さい) 行うことは必要です。 BGPプロセスをクリアするために、clear ip bgp

[ソフト] [イン/アウト使用して下さい]示し、[イン/アウト]受信か送信 設定をかソフトがセッションを引き裂かないでソフト リセットを規定 するところコマンド。 イン/アウトが規定 されなかった両方受信 および 送信 セッション リセットされればでなければ。

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。 このセクションの図では、AS256 は、組織の異なる 2 つのルータから 170.10.0.0 に関するアップデートを受信しています。 AS256 からそのネットワークに到達する際に選択されるルートは、ローカル プリファレンスにより決定できます。 優先される出口ポイントが RTD であると仮定します。 次の設定は、AS300 から到達するアップデートのローカル プリファレンスを 200 に設定し、AS100 から到達するアップデートのローカル プリファレンスを 150 に設定します。

```
RTC#
router bgp 256
neighbor 1.1.1.1 remote-as 100
neighbor 128.213.11.2 remote-as 256
bgp default local-preference 150

RTD#
router bgp 256
neighbor 3.3.3.4 remote-as 300
neighbor 128.213.11.1 remote-as 256
bgp default local-preference 200
```

この設定では、RTC が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。 同じ RTD が、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。 ローカル プリファレンスの交換は AS256 内で行われます。 そのため、RTC および RTD の両方は、AS100 ではなく AS300 からアップデートが到達するときに、ネットワーク 170.10.0.0 がより高いローカル プリファレンスを持っていることを認識します。 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、出口ポイントとして RTD を使用して送信されます。

ルート マップを使用すると、さらに柔軟な設定を行えます。 このセクションの例では、RTD に到達するすべてのアップデートに対して、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。 このタグは、必要ではない可能性があります。 このため、ルート マップを使用すると、どのアップデートに特定のローカル プリファレンスをタグ付けするかを指定できます。 次に例を示します。

```
RTD#
router bgp 256
neighbor 3.3.3.4 remote-as 300
neighbor 3.3.3.4 route-map setlocalin in
neighbor 128.213.11.1 remote-as 256
....
ip as-path access-list 7 permit ^300$
...

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

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

この設定では、AS300 から到達するすべてのアップデートにローカル プリファレンス 200 がタグ付けされます。 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 と

いう値がタグ付けされます。

メトリック属性

メトリック アトリビュートは、MULTI_EXIT_DISCRIMINATOR、MED（BGP4）、または INTER_AS（BGP3）とも呼ばれます。 このアトリビュートは、AS への優先パスに関する外部隣接ルータへのヒントです。 このアトリビュートは、AS への入口ポイントが複数ある場合に、特定のルートに到達するように、別の AS に対して動的に影響を与える方法を提供します。 メトリックの値の小さい方が優先されます。

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。 別の AS に伝達されたメトリックは、その AS からさらに別の AS へ伝達されることはありません。 特定のメトリックを持つアップデートが AS に到達すると、AS 内部では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 に送信される場合、メトリックは 0 に戻ります。 このセクションの図は、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、他の指示を受信しない限り、同じ AS 内の隣接ルータから伝達されたパスのメトリックを比較します。 **異なる AS から到達する隣接ルータからのメトリックをルータが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する可能性がある 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。 これらのコマンドによって BGP パス選択がどのような影響を受けるかについては、『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 つの異なるルータ経由でネットワーク 180.10.0.0 に関する情報を取得します。 RTC、RTD、RTB がこの 3 つのルータです。 RTC と RTD は AS300 に属し、RTB は AS400 に属します。

この例では、コマンド bgp bestpath as-path ignore による RTA の AS-PATH 比較は無視されます。 ルート比較のための次のアトリビュートに落ちるために BGP を強制することを設定します（この場合メトリックか MED）。 コマンドが省略される場合、BGP はルータ RTC からそれに最も短い AS-PATH があるようにルート 180.10.0.0 をインストールします。

RTC から到達するメトリックを 120、RTD から到達するメトリックを 200、RTB から到達するメトリックを 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 内の隣接ルータから到達するメトリックを比較します。 したがって、RTA では、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比較だけが行えます。 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。 RTA が RTB からのアップデートをメトリック 50 で取得した場合、RTC と RTB は異なる AS にあるため、RTA はメトリックを 120 と比較できません。 RTA は、別のアトリビュート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。 次の設定は、このプロセスを示しています。

```
RTA#
   router bgp 100
```

```
   neighbor 2.2.2.1 remote-as 300
   neighbor 3.3.3.3 remote-as 300
   neighbor 4.4.4.3 remote-as 400
   bgp bestpath as-path ignore
   ....

RTC#
   router bgp 300
   neighbor 2.2.2.2 remote-as 100
   neighbor 2.2.2.2 route-map setmetricout out
   neighbor 1.1.1.2 remote-as 300

route-map setmetricout permit 10
   set metric 120

RTD#
   router bgp 300
   neighbor 3.3.3.2 remote-as 100
   neighbor 3.3.3.2 route-map setmetricout out
   neighbor 1.1.1.1 remote-as 300

route-map setmetricout permit 10
   set metric 200

RTB#
   router bgp 400
   neighbor 4.4.4.4 remote-as 100
   neighbor 4.4.4.4 route-map setmetricout out

route-map setmetricout permit 10
   set metric 50
```

この設定では、他のアトリビュートがすべて同じである事実が考慮され、RTA はネクストホップとして RTC を選択します。 RTB がメトリックの比較に含まれるようにするには、RTA を次のように設定します。

```
RTA#
router bgp 100
neighbor 2.2.21 remote-as 300
neighbor 3.3.3.3 remote-as 300
neighbor 4.4.4.3 remote-as 400
bgp always-compare-med
```

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。

このセクションの例では、RTB が AS100 にスタティック経由でネットワークを注入すると仮定しています。 次に設定を示します。

```
RTB#
router bgp 400
redistribute static
default-metric 50
```

```
ip route 180.10.0.0 255.255.0.0 null 0
!--- This causes RTB to send out 180.10.0.0 with a metric of 50.
```

コミュニティ属性

コミュニティ アトリビュートは、推移的なオプションのアトリビュートで、0 ~ 4,294,967,200 の範囲内です。 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で分類し、そのコミュニティに応じてルーティングの決定を適用できます。 ルーティングの決定は、許容、優先、再配布などがあります。

コミュニティ アトリビュートを設定するには、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されている事前定義済みの代表的なコミュニティを次にいくつか示します。

no-export：eBGP ピアにアドバタイズしない。 このルートは AS 内に維持する。

no-advertise：このルートは内部または外部のいずれのピアにもアドバタイズしない。

internet：このルートを、Internet コミュニティにアドバタイズする。 すべてのルータがこのコミュニティに所属します。

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 件、次に示します。

- ```
  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場合、コミュニティに 200 が追加されます。 コミュニティ アトリビュートを設定しても、このアトリビュートはデフォルトでは隣接ルータに送信されません。 アトリビュートを隣接ルータに送信するには、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。

```
RTA#
router bgp 100
```

```
neighbor 3.3.3.3 remote-as 300
neighbor 3.3.3.3 send-community
neighbor 3.3.3.3 route-map setcommunity out
```
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 小数点、16 進法および 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: NN は、**ip bgp-community new-format global configuration コマンドを**発行します。 AA の最初の一部: NN は AS 数を表し、第 2 一部は 2 バイト数を表します。

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 アトリビュート値が 6553620 として表示されています。

```
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 global configuration コマンドを使って、AA のコミュニティ値 ディスプレイ: NN 形式。 この例では、コミュニティ値は **show ip bgp 6.0.0.0** コマンドの出力に 100:20 として表示されています。

```
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

BGP アップデートの送受信の制御には、数多くのフィルタ方式を使用できます。 BGP アップデートは、ルート情報、パス情報、またはコミュニティに基づいてフィルタリングできます。 どの方式を使用しても同じ結果が得られます。 その中でどの方式を選択するかは、具体的なネットワーク構成によって決まります。

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、特定の隣接ルータ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、隣接ルータとの間で送受信されるアップデートにアクセス リストを適用します。 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 はネットワーク 160.10.0.0 を生成して、アップデートを RTC に送信しています。 RTC から AS100 にアップデートを伝搬しないようにする場合は、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 リストを適用します。

```
RTC#
```

```
router bgp 300
network 170.10.0.0
neighbor 3.3.3.3 remote-as 200
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 distribute-list 1 out

access-list 1 deny 160.10.0.0 0.0.255.255

access-list 1 permit 0.0.0.0 255.255.255.255
!--- Filter out all routing updates about 160.10.x.x.
```

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。

このセクションの例では、RTB に 160.10.x.x という複数のサブネットがあると仮定しています。目的は、アップデートをフィルタリングして 160.0.0.0/8 だけをアドバタイズすることです。

注: /8 という表記は、サブネット マスクとして IP アドレスの一番左から 8 ビットを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 と同じです。

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 だけに制限するには、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.
```

このリストでは 160.0.0.0/8 だけが許可されます。

BGP ピアからネットワークをフィルタリングする方法の設定例は、『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。 このセクションの図では、160.10.0.0 に関するアップデートが AS100 に到達するのをブロックできます。 アップデートをブロックするには、AS200 から生成されたアップデートが AS100 に送信されないようにするアクセス 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 が 160.10.0.0 に関するアップデートを RTA に送信するのが停止されます。

```
RTC#
router bgp 300
neighbor 3.3.3.3 remote-as 200
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 filter-list 1 out
!--- The 1 is the access list number below. ip as-path access-list 1 deny ^200$ ip as-path
access-list 1 permit .*
```

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」であり、^ は「で始まる」、$ 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に、200 で始まり、200 で終わるパス情報を含めて送信するため、このアップデートはアクセス リストに一致します。 アクセス リストでは、これらのアップデートが拒否されます。

. には、*もう一つの正規表現はあります 「任意の文字」を意味し、* は「その文字の繰り返し」を意味しています。 従って*割り当てに必要他のすべての更新の伝達であるパス情報を示します。

^200$ を使用する代わりに ^200 を使用するとどうなるでしょうか。 AS400 が存在する場合は、このセクションの図に示すように、AS400 から生成され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。 このパス情報では、前半が 200 であり、後半が 400 になります。 このパス情報は 200 で始まるため、これらのアップデートはアクセス リスト ^200 と一致します。 そのため、このアクセス リストは、これらのアップデートが RTA へ送信されるのをブロックします。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。 このコマンドを使用すると、正規表現の設定に一致するパスがすべて表示されます。

AS 正規表現

このセクションでは、正規表現の作成方法について説明しています。

正規表現とは、入力文字列と照合するためのパターンです。 正規表現を作成する際には、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から成る文字列を指定します。

「パス フィルタリング」セクションの例では、^200$ という文字列を指定しました。 到達したアップデートに含まれるパス情報は、その文字列と照合されて判断が行われます。

正規表現は、次の要素で構成されます。

範囲

レンジとは、左角カッコと右角カッコの間に含まれる文字列です。 例：[abcd]

アトム

アトムとは、単一の文字です。 次に例を示します。

.

. には、 任意の単一の文字が一致します。

^

^ には、入力文字列の先頭が一致します。

$

$ には、入力文字列の末尾が一致します。

\

\ には、文字が一致します。

-

_ には、コンマ (,) 、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、またはスペースが一致します。

ピース

ピースとは、次のいずれかの記号であり、アトムの後に続きます。

*

* には、0 個以上のアトム シーケンスが一致します。

+

+ には、1 個以上のアトム シーケンスが一致します。

?

原子かヌルストリングと一致します。

ブランチ

ブランチとは、ピースが 0 個以上連結したものです。

次に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、「a」が存在しない場合も含みます。

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

この表現には、「aa」や「aba」が一致します。

_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 の起点 (origin) 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

このドキュメントでは、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。

この例では、RTB から RTC にアドバタイズされたルートを RTC が外部ピアに伝搬しないようにするため、RTB によってアドバタイズされる BGP ルートにコミュニティ アトリビュートを設定します。 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。

```
RTB#
router bgp 200
network 160.10.0.0
neighbor 3.3.3.1 remote-as 300
neighbor 3.3.3.1 send-community
neighbor 3.3.3.1 route-map setcommunity out

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

access-list 1 permit 0.0.0.0 255.255.255.255
```

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 はアトリビュート NO_EXPORT を含むアップデートを取得した場合、そのアップデートを外部ピア RTA には伝搬しません。

次の例では、RTB のコミュニティ アトリビュートを **100 200 additive** に設定しています。 この設定により、RTC へ送信される前に、既存のコミュニティ値に値 100 200 が追加されます。

```
RTB#
   router bgp 200
   network 160.10.0.0
   neighbor 3.3.3.1 remote-as 300
   neighbor 3.3.3.1 send-community
   neighbor 3.3.3.1 route-map setcommunity out

route-map setcommunity
```

```
match ip address 2
set community 100 200 additive

access-list 2 permit 0.0.0.0 255.255.255.255
```

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フィルタリングやアトリビュートの設定を行うことができます。

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

たとえば、次のルート マップ、**match-on-community** 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 リストを使用すると、コミュニティ値に基づいて、アップデートに含まれる特定のパラメータ (ウェイトやメトリックなど) をフィルタリングしたり、設定したりすることができます。 このセクションの 2 つ目の例では、RTB はコミュニティ 100 200 で RTC にアップデートを送信しました。 RTC がこれらの値に基づいてウェイトを設定するためには、次のように設定します。

```
RTC#
router bgp 300
neighbor 3.3.3.3 remote-as 200
neighbor 3.3.3.3 route-map check-community in

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

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

route-map check-community permit 30
match community 3

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

この例では、コミュニティ アトリビュートとして 100 を持つルートは、リスト 1 と一致します。 このルートのウェイトは 20 に設定されます。 コミュニティとして 200 だけを持つルートはすべてリスト 2 と一致し、ウェイトは 20 になります。 **exact** というキーワードは、コミュニティが 200 だけで構成されていることを示します。 最後のコミュニティ リストは、他のアップデートが廃棄されないようにするためにこの位置にあります。 デフォルトでは、一致しないものはすべて廃棄されるため、注意が必要です。 キーワード **internet** はすべてのルートを示します。これは、どのルートも Internet コミュニティのメンバであるためです。

詳細は、『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 隣接ルータとルート 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*
このセクションの図では、AS200 に対してローカルなネットワークに関する情報だけを AS200 から RTC が学習すると仮定します。 また、許容されたルートのウェイトを 20 に設定するものとします。 **neighbor** と **as-path** アクセス リストを組み合せて使用します。

```
RTC#
   router bgp 300
   network 170.10.0.0
   neighbor 3.3.3.3 remote-as 200
   neighbor 3.3.3.3 route-map stamp in

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

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

AS200 から生成されるアップデートには、200 で始まり 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ます。 その他のアップデートは廃棄されます。

次のように仮定します。

AS200 から生成され、ウェイトが 20 に設定されているアップデートは受け入れる。

AS400 から生成されたアップデートは廃棄する。

その他のアップデートはウェイトを 10 に設定する。

```
RTC#
router bgp 300
network 170.10.0.0
neighbor 3.3.3.3 remote-as 200
neighbor 3.3.3.3 route-map stamp in

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

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

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

```
ip as-path access-list 2 permit ^200 600 .*
```
この文では、AS200 に対してローカルであるアップデートのウェイトが 20 に設定されます。 またこの文では、AS400 の背後から到達するアップデートのウェイトが 10 に設定され、AS400 から到達するアップデートは廃棄されます。

set as-path prepend コマンドの使用

特定の状況下では、BGP 決定プロセスを操作するために、パス情報を操作する必要があります。その場合は、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 隣接ルータとルート マップ」セクションの図で、RTC が自身のネットワーク 170.10.0.0 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 に伝搬されるとき、AS600 のルータに 2 つの異なるルーティングで 170.10.0.0 についてのネットワーク到達可能性情報があります。 1 つは AS100 を経由したパス (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。 他のアトリビュートがすべて同じである場合、AS600 は最も短いパスを選択するため、AS100 を経由するルートが選択されます。

AS300 は、AS100 経由でそのトラフィックをすべて受信します。 AS300 の側からこの決定を制御する場合は、AS100 を経由するパスの方が、AS400 を経由するパスよりも長いように見せかけることができます。 これを行うには、AS100 にアドバタイズされる既存のパス情報の先頭に AS 番号を追加します。 よく使用されるのは、次のように自身の AS 番号を繰り返して追加する方法です。

```
RTC#
router bgp 300
network 170.10.0.0
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-map SETPATH out

route-map SETPATH
set as-path prepend 300 300
```
この設定により、AS600 は 170.10.0.0 に関するアップデートを AS100 経由で受信します。これらのアップデートには、以下のパス情報が含まれます。 (100、300、300、300)。 このパス情報は、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 グループは、同じアップデート ポリシーを持つ BGP 隣接ルータのグループです。 アップデート ポリシーは通常、ルート マップ、配布リスト、フィルタ リストなどで設定されます。 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 グループのメンバは、所属するピア グループの設定オプションをすべて受け継ぎます。 これらのオプションが発信アップデートに対して作用しない場合は、各メンバでそのオプションを上書きするように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だけです。

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の BGP 隣接ルータに適用します。

```
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 5.5.5.2 peer-group internalmap
   neighbor 5.6.6.2 peer-group internalmap
   neighbor 3.3.3.2 peer-group internalmap
   neighbor 3.3.3.2 filter-list 3 in
```

この設定では、**internalmap** という名前のピア グループを定義しています。 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、すべての内部隣接ルータ、RTE、RTF、および RTG に適用しています。 また、この設定では、隣接ルータ RTE 用に別のフィルタ リスト 3 を定義しています。 これは、ピア グループ内部でフィルタ リスト 2 を上書きします。

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だけです。

次に、外部隣接ルータに対してピア グループをどのように使用できるのかを見てみます。 このセクションの同じ図で、RTC にピア グループ **externalmap** を設定し、そのピア グループを外部隣接ルータに適用します。

```
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 2.2.2.2 remote-as 100
   neighbor 2.2.2.2 peer-group externalmap
   neighbor 4.4.4.2 remote-as 600
   neighbor 4.4.4.2 peer-group externalmap
   neighbor 1.1.1.2 remote-as 200
   neighbor 1.1.1.2 peer-group externalmap
   neighbor 1.1.1.2 filter-list 3 in
```

注: これらの設定では、ピア グループの外側で **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 を割り当てることにより、隣接ルータ 1.1.1.2 の着信アップデートを上書きしています。

ピア グループの詳細は、『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 リリースでも利用できます。 この機能では、同じ発信ポリシーを共有する隣接ルータのアップデート 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。 これらの隣接ルータは、同じアップデート メッセージを共有できます。 以前のリリースの Cisco IOS ソフトウェアでは、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。 この方式でアップデートをグループ化することにより、発信ポリシーおよび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 グループ機能を使用すると、アップデー

ト グループの複製がピア グループ設定から分離されます。 これにより、コンバージェンス時間および隣接ルータ設定の柔軟性が向上します。 詳細は、『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、Classless Interdomain Routing (CIDR; クラスレス ドメイン間ルーティング) です。 CIDR またはスーパーネッティングは、IP アドレスの新しい参照方法です。 CIDR を使うと、クラスの概念が、クラス A のような、B ありません、またはたとえば C.はかつて、ネットワーク 192.213.0.0 不正なクラスCネットワークでした。 現在、このネットワークは 192.213.0.0/16 で表される正当なスーパーネットになりました。 「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。 次の例では、RTB はネットワーク 160.10.0.0 を生成しています。 ここで、RTC を設定して、このルートのスーパーネット 160.0.0.0 が RTA に伝搬されるようにします。

```
RTB#
router bgp 200
neighbor 3.3.3.1 remote-as 300
network 160.10.0.0

#RTC
router bgp 300
neighbor 3.3.3.3 remote-as 200
neighbor 2.2.2.2 remote-as 100
network 170.10.0.0
aggregate-address 160.0.0.0 255.0.0.0
```

RTC は、集約アドレス 160.0.0.0 を RTA に伝搬します。

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。 アドレスを意図したとおりに集約するには、各コマンドの動作について十分に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。

aggregate-address *address-mask*
このコマンドは、プレフィクス ルートと、すべてのより詳細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 を伝搬しますが、160.10.0.0 が RTA に伝搬されるのをブロックしません。 そのため、ネットワーク 160.0.0.0 と 160.10.0.0 がどちらも RTA に伝搬されています。これにより、プレフィクス ルートと、より詳細なルートの両方がアドバタイズされます。

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は、そのアドレスを集約できません。

たとえば、RTB は BGP テーブル内に 160.0.0.0 のより詳細なエントリが存在しない場合、RTB は 160.0.0.0 の集約を生成できません。 より詳細なルートを BGP テーブルに注入することは可能です。 ルートの注入は次の状況で発生することがあります。

他の AS からの受信アップデート

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 だけを伝搬し、より詳細なルートは伝搬**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、プレフィクスのみをアドバタイズします。 このコマンドは、より詳細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 を伝搬し、より詳細なルート 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」セクションの例は、このような状況を説明しています。

aggregate-address *address-mask* **as-set**
このコマンドは、プレフィクス ルートと、より詳細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）」セクションは、このコマンドについて説明しています。

集約を行う際により詳細なルートを抑制するには、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、抑制対象のより詳細なルートを選択できます。

aggregate-address *address-mask* **suppress-map** *map-name*
このコマンドは、プレフィクス ルートと、より詳細なルートをアドバタイズしますが、 ルート マップに基づいてアドバタイズメントを抑制します。 「CIDR と集約アドレス」セクションの図に基づき、160.0.0.0 を集約し、より詳細なルート 160.20.0.0 を抑制して、160.10.0.0 の伝搬を許可すると仮定します。 その場合は、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 160.20.0.0 0.0.255.255
access-list 1 deny 0.0.0.0 255.255.255.255
```

suppress-map の定義により、アクセス リストで許可されるパケットのアップデートは抑制されます。

次に、このルート マップを **aggregate** 文に適用します。

```
RTC#
router bgp 300
neighbor 3.3.3.3 remote-as 200
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 remote-as 100
network 170.10.0.0
aggregate-address 160.0.0.0 255.0.0.0 suppress-map CHECK
```

次に別のバリエーションを示します。

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

このコマンドを使用すると、集約が送信される際に、メトリックなどのアトリビュートを設定できます。 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。

```
route-map SETMETRIC
set origin igp

aggregate-address 160.0.0.0 255.0.0.0 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

Request： RTB がプレフィクス 160.0.0.0 をアドバタイズし、より詳細なルートすべてを抑制するようにします。 この要求の問題は、ネットワーク 160.10.0.0 が AS200 にとってローカルである、つまり、AS200 が 160.10.0.0 の生成元である点です。 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 の生成元であるため、RTB は両方のネットワークを生成します。 この問題には次の 2 つの解決策があります。

第 1 の解決策は、スタティック ルートを使用し、それを BGP に再配布することです。 こうすれば、RTB は「incomplete（？）」の起点（origin）を含めて集約をアドバタイズします。

```
RTB#
router bgp 200
neighbor 3.3.3.1 remote-as 300
redistribute static
!--- This generates an update for 160.0.0.0 !--- with the origin path as "incomplete". ip route
160.0.0.0 255.0.0.0 null0
```

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。 このエントリを追加した場合、アップデートの起点（origin）が IGP に設定される点以外は同じ結果が得られます。

```
RTB#
router bgp 200
network 160.0.0.0 mask 255.0.0.0
!--- This entry marks the update with origin IGP. neighbor 3.3.3.1 remote-as 300 redistribute
static ip route 160.0.0.0 255.0.0.0 null0
```

CIDR 例 2（as-set）

パス情報のサイズを縮小するには、集約で **as-set** 文を使用します。 **as-set** を使用すると、集約対

象の複数パスに同じ AS 番号が何回登場しても AS 番号は一度しかリストされません。 **aggregate as-set** コマンドは、情報を集約することでパス アトリビュートに関する情報が失われるような場合に使用します。 次の例では、RTC は、RTA から 160.20.0.0 に関するアップデートを受信し、RTB から 160.10.0.0 に関するアップデートを受信します。 RTC でネットワーク 160.0.0.0/8 を集約し、そのネットワークを RTD に送信すると仮定します。 RTD では、そのルートの起点（origin）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。 どのパスが最初に到達したかに関係なく、パス情報はすべてこの集合に含められます。

```
RTB#
router bgp 200
network 160.10.0.0
neighbor 3.3.3.1 remote-as 300

RTA#
router bgp 100
network 160.20.0.0
neighbor 2.2.2.1 remote-as 300
```

ケース 1：

RTC には **as-set** 文が設定されていません。 RTC は、160.0.0.0/8 のアップデートを RTD に送信する際、そのルートが AS300 から生成されたものであるかのように、パス情報（300）を設定します。

```
RTC#
router bgp 300
neighbor 3.3.3.3 remote-as 200
neighbor 2.2.2.2 remote-as 100
neighbor 4.4.4.4 remote-as 400
aggregate 160.0.0.0 255.0.0.0 summary-only
!--- This command causes RTC to send RTD updates about 160.0.0.0/8 !--- with no indication that
160.0.0.0 actually comes from two different ASs. !--- This may create loops if RTD has an entry
back into AS100 or AS200.
```

Case 2:

```
RTC#
   router bgp 300
   neighbor 3.3.3.3 remote-as 200
   neighbor 2.2.2.2 remote-as 100
   neighbor 4.4.4.4 remote-as 400
   aggregate 160.0.0.0 255.0.0.0 summary-only
   aggregate 160.0.0.0 255.0.0.0 as-set
!--- This command causes RTC to send RTD updates about 160.0.0.0/8 !--- with an indication that
160.0.0.0 belongs to a set {100 200}.
```

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 内部の 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を実装すると、AS 内の iBGP メッシュが減少します。 BGP コンフェデレーションでは、1 つの AS を複数の AS に分割し、グループ全体を 1 つのコンフェデレーションに割り当てます。 それぞれの AS は単独でフル メッシュ構造の iBGP を持ち、コンフェデレーション内部の他の AS に接続されます。 これらの AS はコンフェデレーション内の AS への

eBGP ピアを持ちますが、iBGP を使用する場合と同様に AS はルーティングを交換します。 つまり、コンフェデレーションでは、ネクストホップ、メトリック、および ローカル プリファレンス情報が保持されます。 外部から見ると、コンフェデレーションは 1 つの 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識別子は、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部での複数の 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 つの BGP スピーカで構成される AS500 があるとします。 他の BGP 以外のスピーカも存在しますが、ここでは他の AS への eBGP 接続を持つ BGP スピーカだけを取り上げます。 AS500 の内部にフル iBGP メッシュを作成する必要がある場合、ルータごとに 9 つのピア接続が必要となります。 8 つの iBGP ピアと外部 AS への 1 つの eBGP ピアが必要です。

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識別子として 500 を割り当てます。 外部からは 1 つの AS (AS500) しか見えません。 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。

ルータ RTC、RTD、および RTA の設定例を次に示します。

注: RTA は AS50、AS60、AS70 を認識していません。 RTA は AS500 だけを認識しています。

```
RTC#
router bgp 50
bgp confederation identifier 500
bgp confederation peers 60 70
neighbor 128.213.10.1 remote-as 50 (IBGP connection within AS50)
neighbor 128.213.20.1 remote-as 50 (IBGP connection within AS50)
neighbor 129.210.11.1 remote-as 60 (BGP connection with confederation peer 60)
neighbor 135.212.14.1 remote-as 70 (BGP connection with confederation peer 70)
neighbor 5.5.5.5 remote-as 100 (EBGP connection to external AS100)

RTD#
router bgp 60
bgp confederation identifier 500
bgp confederation peers 50 70
neighbor 129.210.30.2 remote-as 60 (IBGP connection within AS60)
neighbor 128.213.30.1 remote-as 50(BGP connection with confederation peer 50)
neighbor 135.212.14.1 remote-as 70 (BGP connection with confederation peer 70)
neighbor 6.6.6.6 remote-as 600 (EBGP connection to external AS600)

RTA#
   router bgp 100
   neighbor 5.5.5.4 remote-as 500 (EBGP connection to confederation 500)
```

ルート リフレクタ

AS 内での iBGP ピアリングの増大に対するソリューションとしては、Route Reflector (RR; ルート リフレクタ) もあります。 「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。 この制約を若干緩め、制御を追加すると、ルータは iBGP で学習したルートを他の iBGP スピーカにアドバタイズ (リフレクト) できるようになります。 このルート リフレクションにより、AS 内の iBGP ピア数が低減されます。

通常は、AS100 内の RTA、RTB、RTC 間でフル iBGP メッシュを維持する必要があります。 RR の概念を利用することにより、RTC を RR として選択することが可能になります。 その場合、RTC は RTA および RTB との間で部分 iBGP ピアリングを実現します。 RTA と RTB の間のピアリングは必要ありません。これは、RTC が、RTA および RTB から到達するアップデートの RR になるためです。

neighbor route-reflector-client
このコマンドを設定したルータが RR になり、コマンドで指定された隣接ルータがその 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、RTA、RTB、および RTC は、AS100 内部で 1 つの RR を持つ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。

AS には複数の RR を設置できます。 この場合、RR は他の RR を、他のすべての iBGP スピーカとまったく同様に扱います。 他の RR が同じクラスタ (クライアント グループ) に属することも、他のクラスタに属することもできます。 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 で他の RR をフルメッシュ トポロジでの非クライアント ピアとして設定します。 クライアントは、クライアント クラスタの外部にある iBGP スピーカとはピア関係を確立しません。

この図について考えてみます。 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 から見ると、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 はクライアント RTE および RTF の RR です。 RTG は 3 番目のクラスタの 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 がルートを受信すると、RR のルートは次のようになります。 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 ピアにリフレクトし、クライアント ピアにもリフレクトする。
- eBGP ピアからのルート：クライアント ピアと非クライアント ピアすべてにアップデートを送信する。

ルータ RTC、RTD、および RTB の BGP 設定の関連部分を次に示します。

```
RTC#

router bgp 100
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-reflector-client
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 route-reflector-client
neighbor 7.7.7.7 remote-as 100
neighbor 4.4.4.4 remote-as 100
neighbor 8.8.8.8 remote-as 200


RTB#

router bgp 100
neighbor 3.3.3.3 remote-as 100
neighbor 12.12.12.12 remote-as 300


RTD#

router bgp 100
neighbor 6.6.6.6 remote-as 100
neighbor 6.6.6.6 route-reflector-client
neighbor 5.5.5.5 remote-as 100
neighbor 5.5.5.5 route-reflector-client
neighbor 7.7.7.7 remote-as 100
neighbor 3.3.3.3 remote-as 100
```

iBGP で学習されたルートがリフレクトされているので、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 方式では、このループを回避するために次のような手法が用意されています。

originator-id：これはオプションの非経過的な BGP アトリビュートであり、長さは 4 バイトです。 このアトリビュートは RR によって作成されます。 このアトリビュートは、ローカル AS 内でのルートの生成元ルータの Router-ID（RID）を伝達します。 設定が適切でないためにルーティング情報が生成元ルータに戻っても、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」セクション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 1 つの RR があります。 この場合、クラスタは RR のルータ ID によって識別されます。 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 が RR クライアント

からクラスタ外部の非クライアントにルートをリフレクトした場合、クラスタ リストにローカル クラスタ ID がアペンドされます。 このアップデートのクラスタ リストが空である場合は、RR によってクラスタ リストが新規に作成されます。 RR は、このアトリビュートを使用して、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判断します。 クラスタ リスト内にローカル クラスタ ID がある場合、アドバタイズメントは無視されます。

このセクションの図では、RTD、RTE、RTF および RTH は 1 つのクラスタに属しています。 RTD と RTH は両方とも、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 の設定を次に示します。

```
RTH#

router bgp 100
neighbor 4.4.4.4 remote-as 100
neighbor 5.5.5.5 remote-as 100
neighbor 5.5.5.5 route-reflector-client
neighbor 6.6.6.6 remote-as 100
neighbor 6.6.6.6 route-reflector-client
neighbor 7.7.7.7 remote-as 100
neighbor 3.3.3.3 remote-as 100
neighbor 9.9.9.9 remote-as 300
bgp cluster-id 10


RTD#

router bgp 100
neighbor 10.10.10.10 remote-as 100
neighbor 5.5.5.5 remote-as 100
neighbor 5.5.5.5 route-reflector-client
neighbor 6.6.6.6 remote-as 100
neighbor 6.6.6.6 route-reflector-client
neighbor 7.7.7.7 remote-as 100
neighbor 3.3.3.3 remote-as 100
neighbor 11.11.11.11 remote-as 400
bgp cluster-id 10


RTF#

router bgp 100
neighbor 10.10.10.10 remote-as 100
neighbor 4.4.4.4 remote-as 100
neighbor 13.13.13.13 remote-as 500


RTC#

router bgp 100
```

```
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 route-reflector-client
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-reflector-client
neighbor 4.4.4.4 remote-as 100
neighbor 7.7.7.7 remote-as 100
neighbor 10.10.10.10 remote-as 100
neighbor 8.8.8.8 remote-as 200
```

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 グループを使用していません。 クラスタ内部のクライアントが互いの間に直接 iBGP ピアを持たず、RR を通じてクライアントが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 上でのルートの生成元への withdrawal (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。 この送信により、問題が発生する可能性があります。

RR では、ルータ サブコマンド bgp client-to-client reflection がデフォルトで有効になっています。 RR で bgp client-to-client reflection を無効にして、クライアント間で冗長 BGP ピアリングが構成されている場合は、ピア グループを安全に使用できます。 詳細については同位グループの制限を参照して下さい。

RR と従来型 BGP スピーカ

AS 内には 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、このようなルータのことを「従来型 BGP スピーカ」と呼びます。 RR スキームでは、このような従来型 BGP スピーカの共存が可能になります。 このようなルータは、クライアント グループのメンバになることも、非クライアント グループのメンバになることも可能です。 これらのルータが存在すると、現在の iBGP モデルから RR モデルへの移行を簡単かつ段階的に行えます。 クラスタを作成するには、まず単一のルータを RR として設定し、他の RR と RR クライアントを通常の iBGP ピアにします。 その後、段階的にクラスタの数を増やしていきます。

この図では、RTD、RTE、および RTF はルート 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 との間では、通常の iBGP メッシュを実現可能です。 その後、アップグレードの準備が完了したら、RTC を RR にして、RTA と RTB をそのクライアントにすることが可能です。 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 の設定を次に示します。

```
RTD#

router bgp 100
neighbor 6.6.6.6 remote-as 100
neighbor 6.6.6.6 route-reflector-client
neighbor 5.5.5.5 remote-as 100
neighbor 5.5.5.5 route-reflector-client
neighbor 3.3.3.3 remote-as 100
neighbor 2.2.2.2 remote-as 100
neighbor 1.1.1.1 remote-as 100
neighbor 13.13.13.13 remote-as 300
```

```
RTC#

router bgp 100
neighbor 4.4.4.4 remote-as 100
neighbor 2.2.2.2 remote-as 100
neighbor 1.1.1.1 remote-as 100
neighbor 14.14.14.14 remote-as 400
```

RTC をアップデートして、RTC を RR にする準備ができたら、iBGP のフル メッシュ構造を削除し、RTA および 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。

ループを制御するためのもう 1 つの手段は、発信ルート マップの **set** 句でより多くの制約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作用しません。

また、隣接ルータごとの設定オプションである **nexthop-self** でも制約を加えることができます。RR で **nexthop-self** を使用すると、この句は eBGP で学習されたルートのネクストホップだけに作用します。これは、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からです。

ルート フラップ ダンプニング

ルート ダンプニングは、Cisco IOS ソフトウェア リリース 11.0 で導入されました。 ルート ダンプニングとは、ルートのフラッピングが原因で不安定な状態が発生するのを最小限に抑えるメカニズムです。 ルート ダンプニングは、ネットワーク上のゆらぎも減らします。 そのために、動作が正常でないルートを特定するための基準を定義します。 フラップしているルートには、1 回のフラップごとに 1000 のペナルティが与えられます。 ペナルティの累計が事前に定義した「suppress limit」に到達すると、その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、事前に設定した「half-life time」に基づいて指数関数的に減衰します。 ペナルティが事前に定義した「reuse limit」以下に減少すると、その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 を介して学習された AS 外部のルートには適用されません。 したがって、ルート ダンプニングを使用すると、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 秒単位で減衰します。 ルートの抑制は、10 秒単位で解除されます。 ルータは、ペナルティが「reuse limit」の半分より小さくなるまでダンプニング情報を保持します。 この条件を満たすと、ルータからダンプニング情報が削除されます。

初期設定では、ダンプニングはデフォルトでオフになっています。 必要な場合、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。 ルート ダンプニングを制御するには、次のコマンドを使用します。

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*：half-life-time を変更する。

すべてのパラメータを一度に設定する場合は、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：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 203.250.15.2 255.255.255.252

interface Serial1
 ip address 192.208.10.6 255.255.255.252

router bgp 100
 bgp dampening
 network 203.250.15.0
 neighbor 192.208.10.5 remote-as 300


RTD#
hostname RTD

interface Loopback0
ip address 192.208.10.174 255.255.255.192

interface Serial0/0
 ip address 192.208.10.5 255.255.255.252

router bgp 300
 network 192.208.10.0
 neighbor 192.208.10.6 remote-as 100
```

RTB には、ルート ダンプニングがデフォルト パラメータで設定されています。 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なります。

```
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なります。

```
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。 これは、そのルートへの最適パスはないものの、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
```

ルートはフラッピングに対するペナルティを受け取りますが、ペナルティはまだ「suppress limit」より下です。 デフォルトは 2000 です。 ルートの抑制は、まだ発生していません。 ルート フラップがさらに数回起こると、次のようになります。

```
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。 ペナルティが「reuse value」に達すると、ルートは再利用されます。 この例では、「reuse value」はデフォルトの 750 です。 ペナルティが「reuse limit」の半分より小さくなると、ダンプニング情報は削除されます。 この場合は、ペナルティが 375（750/2=375）になると、ダンプニング情報が削除されます。 フラップ統計情報を表示およびクリアするには、次のコマンドを使用します。

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] ：隣接ルータ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：隣接ルータ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 のパス選択方法

BGP のアトリビュートと用語について十分に理解できたところで、『BGP で最適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

実用的な設計例

このセクションでは、Cisco ルータで実際に表示される設定とルーティング テーブルを使用した設計例を示します。

このセクションでは、この設定を段階的に構築し、発生し得る問題について考察します。 AS が eBGP 経由で 2 つの ISP に接続されている場合は、ルートをより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、iBGP を AS100 内部の RTA と RTB との間で実行し、OSPF を IGP として実行しています。 2 つの ISP (AS200 および AS300) に接続していると仮定した場合、 すべてのルータの最初の設定は次のようになります。

注: 次の設定は、最終的な設定ではありません。

```
RTA#
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 203.250.13.41 255.255.255.0
```

```
interface Ethernet0
ip address 203.250.14.1 255.255.255.0

interface Serial0
 ip address 128.213.63.1 255.255.255.252

router ospf 10
 network 203.250.0.0 0.0.255.255 area 0

router bgp 100
 network 203.250.13.0
 network 203.250.14.0
 neighbor 128.213.63.2 remote-as 200
 neighbor 203.250.15.2 remote-as 100
 neighbor 203.250.15.2 update-source Loopback0

RTF#
hostname RTF

ip subnet-zero

interface Ethernet0
 ip address 203.250.14.2 255.255.255.0

interface Serial1
 ip address 203.250.15.1 255.255.255.252

router ospf 10
 network 203.250.0.0 0.0.255.255 area 0

RTB#
hostname RTB

ip subnet-zero

interface Serial0
 ip address 203.250.15.2 255.255.255.252

interface Serial1
 ip address 192.208.10.6 255.255.255.252

router ospf 10
 network 203.250.0.0 0.0.255.255 area 0

router bgp 100
network 203.250.15.0
 neighbor 192.208.10.5 remote-as 300
 neighbor 203.250.13.41 remote-as 100

RTC#
hostname RTC

ip subnet-zero
```

```
interface Loopback0
 ip address 128.213.63.130 255.255.255.192

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

router bgp 200
 network 128.213.0.0
 neighbor 128.213.63.1 remote-as 100
 neighbor 128.213.63.6 remote-as 400

RTD#
hostname RTD

ip subnet-zero

interface Loopback0
ip address 192.208.10.174 255.255.255.192

interface Serial0/0
 ip address 192.208.10.5 255.255.255.252
!
interface Serial0/1
 ip address 192.208.10.2 255.255.255.252

router bgp 300
 network 192.208.10.0
 neighbor 192.208.10.1 remote-as 500
 neighbor 192.208.10.6 remote-as 100

RTE#
hostname RTE

ip subnet-zero

interface Loopback0
ip address 200.200.10.1 255.255.255.0

interface Serial0
 ip address 195.211.10.2 255.255.255.252

interface Serial1
 ip address 128.213.63.6 255.255.255.252
 clockrate 1000000

router bgp 400
 network 200.200.10.0
 neighbor 128.213.63.5 remote-as 200
 neighbor 195.211.10.1 remote-as 500
```

```
RTG#
hostname RTG

ip subnet-zero

interface Loopback0
 ip address 195.211.10.174 255.255.255.192

interface Serial0
 ip address 192.208.10.1 255.255.255.252

interface Serial1
 ip address 195.211.10.1 255.255.255.252

router bgp 500
 network 195.211.10.0
 neighbor 192.208.10.2 remote-as 300
 neighbor 195.211.10.2 remote-as 400
```

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。 この方式は、IGP を BGP に再配布する方法よりも適しています。 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。 次に RTB の BGP テーブルを示します。

```
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：パス情報の起点（origin）が IGP であることを示します。

Path 情報：この情報は直感的なものです。 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 を介して学習されます。

注: ローカルで生成されたエントリ（203.250.15.0 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ていることを示します。 BGP は、『BGP で最適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 は宛先への到達に最適なパスを選択し、そのパスを IP ルーティング テーブル内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 アトリビュートに注目してください。 RTB は、iBGP に伝送された eBGP ネクストホップであるネクストホップ 128.213.63.2 を介して、128.213.0.0 に関する情報を得て

います。

次に、IP ルーティング テーブルを示します。

```
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 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
```

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ないように見えます。 ここには、2 つの問題が存在します。

最初の問題は、これらのエントリのネクストホップである 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はありません。 RTB は、OSPF 経由では 128.213.63.0 に関し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 128.213.63.2 への到達方法を認識するようになります。 次に RTA の設定を示します。

```
RTA#
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 203.250.13.41 255.255.255.0

interface Ethernet0
ip address 203.250.14.1 255.255.255.0

interface Serial0
 ip address 128.213.63.1 255.255.255.252

router ospf 10
 passive-interface Serial0
 network 203.250.0.0 0.0.255.255 area 0
 network 128.213.0.0 0.0.255.255 area 0

router bgp 100
 network 203.250.0.0 mask 255.255.0.0
 neighbor 128.213.63.2 remote-as 200
 neighbor 203.250.15.2 remote-as 100
 neighbor 203.250.15.2 update-source Loopback0
```

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なります。

```
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: すべてのエントリに、BGP がネクストホップに到達可能であることを意味する **>** があります。

次に、ルーティング テーブルを示します。

```
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 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 番目の問題は、ルーティング テーブルに BGP エントリがまだ存在しないことです。 唯一の違いは、128.213.63.0 が OSPF 経由で到達可能になっているという点です。 これは同期の問題です。 BGP は、これらのエントリをルーティング テーブルに入れておらず、エントリを BGP アップデートで送信しません。これは、BGP が IGP と同期されていないためです。

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 192.208.10.0 および 195.211.10.0 を認識していないことに注意してください。

このシナリオでは、同期をオフにした場合、エントリはルーティング テーブル内に表示されます。 ただし、接続は切断されたままです。

RTB で同期をオフにすると、次のようになります。

```
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へ到達する方法はありません。 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、問題はまだ残ったままになります。 ただし、後で他の問題を扱う際に、同期が必要になります。 BGP を RTA の OSPF に、メトリック 2000 で再配布します。

```
RTA#
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 203.250.13.41 255.255.255.0

interface Ethernet0
ip address 203.250.14.1 255.255.255.0

interface Serial0
 ip address 128.213.63.1 255.255.255.252

router ospf 10
 redistribute bgp 100 metric 2000 subnets
 passive-interface Serial0
 network 203.250.0.0 0.0.255.255 area 0
 network 128.213.0.0 0.0.255.255 area 0

router bgp 100
 network 203.250.0.0 mask 255.255.0.0
 neighbor 128.213.63.2 remote-as 200
 neighbor 203.250.15.2 remote-as 100
 neighbor 203.250.15.2 update-source Loopback0
```

ルーティング テーブルは次のようになります。

```
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 には iBGP より適した距離があるため、BGP エントリは表示されなくなりました。 OSPF の距離は 110 で、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 がマスクの相違により OSPF とは同期しないためです。 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、上の場合と同様です。

ここで、RTB の s1 インターフェイスを表示してルートを調べます。 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します。 この手順により、RTA が IGP 経由でネクストホ

ップ 192.208.10.5 に関する情報を得られるようになります。 この手順を踏まないと、ネクストホップ 192.208.10.5 に到達するために、eBGP 経由で反対のルートを通る必要があるため、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 203.250.13.41 255.255.255.0

interface Ethernet0
ip address 203.250.14.1 255.255.255.0

interface Serial0
 ip address 128.213.63.1 255.255.255.252

router ospf 10
 redistribute bgp 100 metric 2000 subnets
 passive-interface Serial0
 network 203.250.0.0 0.0.255.255 area 0
 network 128.213.0.0 0.0.255.255 area 0

router bgp 100
 no synchronization
 network 203.250.13.0
 network 203.250.14.0
 neighbor 128.213.63.2 remote-as 200
 neighbor 203.250.15.2 remote-as 100
 neighbor 203.250.15.2 update-source Loopback0

RTB#
hostname RTB

ip subnet-zero

interface Serial0
 ip address 203.250.15.2 255.255.255.252

interface Serial1
 ip address 192.208.10.6 255.255.255.252

router ospf 10
 redistribute bgp 100 metric 1000 subnets
 passive-interface Serial1
 network 203.250.0.0 0.0.255.255 area 0
 network 192.208.0.0 0.0.255.255 area 0

router bgp 100
 no synchronization
 network 203.250.15.0
 neighbor 192.208.10.5 remote-as 300
```

```
 neighbor 203.250.13.41 remote-as 100
```

BGP テーブル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
```

ネットワークを 2 つの異なる ISP である AS200 と AS300 との対話ができるように設計するには、複数の方法があります。 その 1 つとして、プライマリ ISP とバックアップ ISP を設定する方法があります。 ISP の 1 つから部分ルートが学習でき、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 からローカル ルートだけを受信します。 RTA と RTB の両方が OSPF へのデフォルト ルートを生成し、よりメトリックが小さいため RTB の方が優先されます。 このようにして、2 つの ISP 間で発信トラフィックの均衡をとることができます。

RTA から発信されたトラフィックが RTB 経由で戻ると、潜在的な非対称が発生する可能性があります。 この状況が発生する可能性があるのは、2 つの ISP との対話中に同じ IP アドレスのプール (同じメジャー ネット) を使用する場合です。 集約のため、AS 全体が外部には 1 つのエンティティのように見えることがあります。 ネットワークへの入口点は RTA または RTB 経由で発生する可能性があります。 インターネットへのポイントが複数ある場合でも、AS への着信トラフィックはすべて単一ポイント経由で到着しているのが見つかる場合があります。 この例では、2 つの ISP との対話時に 2 つの異なるメジャー ネットを使用しています。

非対称が発生する潜在的な理由としては、AS へ到達するためにアドバタイズされたパスの長さが異なることが挙げられます。 特定の宛先には、一方のサービス プロバイダーが近いはずです。この例では、パスがより短いため、ユーザのネットワーク宛ての AS400 からのトラフィックは常に RTA 経由で到達します。 この決定を調整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、ウェイトなどのアトリビュートが含まれてる場合は、AS400 によって出口ポイントが AS200 に設定されている場合があります。 この場合、対処法はありません。

次の設定は、すべてのルータの最終的な設定です。

```
RTA#
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 203.250.13.41 255.255.255.0

interface Ethernet0
 ip address 203.250.14.1 255.255.255.0
```

```
interface Serial0
 ip address 128.213.63.1 255.255.255.252

router ospf 10
 redistribute bgp 100 metric 2000 subnets
 passive-interface Serial0
 network 203.250.0.0 0.0.255.255 area 0
 network 128.213.0.0 0.0.255.255 area 0
 default-information originate metric 2000

router bgp 100
 no synchronization
network 203.250.13.0
 network 203.250.14.0
 neighbor 128.213.63.2 remote-as 200
 neighbor 128.213.63.2 route-map setlocalpref in
 neighbor 203.250.15.2 remote-as 100
 neighbor 203.250.15.2 update-source Loopback0

ip classless
ip default-network 200.200.0.0

route-map setlocalpref permit 10
 set local-preference 200
```

RTA では、AS200 から到達するルートのローカル プリファレンスは 200 に設定されています。また、ネットワーク 200.200.0.0 がデフォルト候補の選択肢となっています。 **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。

また、この例では、default-information originate コマンドを OSPF で使用して、デフォルト ルートを OSPF ドメイン内部に注入しています。 また、この例では、Intermediate System-to-Intermediate System Protocol（IS-IS プロトコル）および BGP に対しても、このコマンドを使用しています。 RIP では、追加の設定なしで、0.0.0.0 が RIP へ自動的に再配布されます。 IGRP および EIGRP では、BGP が IGRP および EIGRP に再配布された後、IGP ドメイン内にデフォルト情報が注入されます。 また、IGRP および EIGRP では、0.0.0.0 へのスタティック ルートを IGP ドメインに再配布できます。

```
RTF#
hostname RTF

ip subnet-zero

interface Ethernet0
 ip address 203.250.14.2 255.255.255.0

interface Serial1
 ip address 203.250.15.1 255.255.255.252

router ospf 10
 network 203.250.0.0 0.0.255.255 area 0

ip classless
```

```
RTB#
hostname RTB

ip subnet-zero

interface Loopback1
 ip address 203.250.15.10 255.255.255.252

interface Serial0
 ip address 203.250.15.2 255.255.255.252
!
interface Serial1
 ip address 192.208.10.6 255.255.255.252

router ospf 10
 redistribute bgp 100 metric 1000 subnets
 passive-interface Serial1
 network 203.250.0.0 0.0.255.255 area 0
 network 192.208.10.6 0.0.0.0 area 0
 default-information originate metric 1000
!
router bgp 100
 no synchronization
 network 203.250.15.0
 neighbor 192.208.10.5 remote-as 300
 neighbor 192.208.10.5 route-map localonly in
 neighbor 203.250.13.41 remote-as 100
!
ip classless
ip default-network 192.208.10.0
ip as-path access-list 1 permit ^300$

route-map localonly permit 10
 match as-path 1
set local-preference 300
```

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高い値です。 したがって、AS100 では AS300 のローカル ルートとして RTB が選択されます。 RTB のその他のルート（存在する場合）はすべて、内部ではローカル 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 より低い値です。 そのため、RTA が優先されます。

注: ここでアドバタイズしているのは、AS300 のローカル ルートだけです。 ^300$ と一致しないパス情報は、すべて廃棄されます。 ISP のカスタマーであるローカル ルートと隣接ルートをアドバタイズする必要がある場合は、^300_[0-9]* を使用します。

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 192.208.10.174 255.255.255.192
!
interface Serial0/0
 ip address 192.208.10.5 255.255.255.252
!
interface Serial0/1
 ip address 192.208.10.2 255.255.255.252

router bgp 300
 network 192.208.10.0
 neighbor 192.208.10.1 remote-as 500
 neighbor 192.208.10.6 remote-as 100

RTG#
hostname RTG

ip subnet-zero

interface Loopback0
 ip address 195.211.10.174 255.255.255.192

interface Serial0
 ip address 192.208.10.1 255.255.255.252

interface Serial1
 ip address 195.211.10.1 255.255.255.252

router bgp 500
 network 195.211.10.0
 aggregate-address 195.211.0.0 255.255.0.0 summary-only
 neighbor 192.208.10.2 remote-as 300
 neighbor 192.208.10.2 send-community
 neighbor 192.208.10.2 route-map setcommunity out
 neighbor 195.211.10.2 remote-as 400
!
ip classless
access-list 1 permit 195.211.0.0 0.0.255.255
access-list 2 permit any
route-map setcommunity permit 20
```

```
 match ip address 2
!
route-map setcommunity permit 10
 match ip address 1
 set community no-export
```

コミュニティ フィルタリングの使用例は、RTG でのものです。 **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。 このようにすると、RTD はそのルートを RTB にエクスポートしません。 ただし、その場合、RTB はこれらのルートを受信しません。

```
RTE#
hostname RTE

ip subnet-zero

interface Loopback0
 ip address 200.200.10.1 255.255.255.0

interface Serial0
 ip address 195.211.10.2 255.255.255.252

interface Serial1
 ip address 128.213.63.6 255.255.255.252

router bgp 400
 network 200.200.10.0
 aggregate-address 200.200.0.0 255.255.0.0 summary-only
 neighbor 128.213.63.5 remote-as 200
 neighbor 195.211.10.1 remote-as 500

ip classless
```

RTE は 200.200.0.0/16 を集約します。 次に、最終的な BGP と、RTA、RTF、および RTB のルーティング テーブルを示します。

```
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) に到達するためのルートが、RTB を経由していることを示しています。 その他の既知のネットワーク (200.200.0.0 など) へのルートは、RTA を経由します。 ラスト リゾートのゲートウェイは RTB に設定されます。 RTB と RTD との間の接続に何かが発生した場合は、RTA によってアドバタイズされたデフォルトが、メトリック 2000 で採用されます。

```
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