BGP の実装

ボーダー ゲートウェイ プロトコル（BGP）は、自律システム間にループフリーのドメイン間ルーティングを作成可能なエクステリア ゲートウェイ プロトコル（EGP）です。自律システムは、単一の技術管理に基づくルータのまとまりです。自律システム内のルータは、複数の内部ゲートウェイ プロトコル（IGP）を使用して自律システム内のルーティング情報を交換し、EGP を使用して自律システム外でパケットをルーティングします。

ここでは、Cisco IOS XR ソフトウェアでの BGP の概念と設定情報を説明します。

（注） BGP の詳細とこのモジュールに示す BGP コマンドの詳細な説明については、このモジュールの関連資料（217 ページ）の項を参照してください。設定作業の実行中に必要になることのある他のコマンドのドキュメントを見つけるには、Cisco ASR 9000 シリーズ ルータ ソフトウェア マスター コマンド索引で、オンライン検索してください。

BGP の実装の機能履歴

リリース	変更内容
リリース 3.7.2	この機能が導入されました。
リリース 3.9.0	次の機能がサポートされました。 • BGP プレフィックス独立コンバージェンス ユニパス プライマリ バックアップ • BGP Local Label Retention • 4 バイト自律システム番号の asplain 表記 • BGP ノンストップルーティング • BGP コマンドに対するコマンドライン インターフェイス（CLI）の一貫性 • L2VPN アドレス ファミリ コンフィギュレーション モード

リリース	変更内容
リリース 4.0.0	次の機能がサポートされました。 • BGP Add Path アドバタイズメント • 累積 iGP（AiGP） • プレルート • IPv4 BGP-Policy Accounting • IPv6 uRPF
リリース 4.1.0	5000 BGP NSR セッションのサポートの追加
リリース 4.1.1	次の機能が追加されました。 • BGP Accept Own • 不等コストの連続ロードバランシングに対する BGP DMZ リンク帯域幅
リリース 4.2.0	次の機能がサポートされました。 • 選択的 VRF ダウンロード • BGP Multi-Instance/Multi-AS • BGP の BFD マルチホップ サポート • BGP のエラー処理 分散 BGP（bgp 分散スピーカー）の設定のサポートが削除されました。
リリース 4.2.1	次の機能がサポートされました。 • グローバル プレフィックス用 BGP 3107 PIC アップデート • RIB および FIB 用 BGP プレフィックス独立コンバージェンス • RPKI に基づく BGP プレフィックスの発信元検証
リリース 4.2.3	BGP 属性のフィルタリング機能が追加されました。
リリース 4.3.0	アップデート生成のための BGP-RIB のフィードバック メカニズム機能が追加されました。
リリース 4.3.1	次の機能がサポートされていました。 • BGP VRF ダイナミック ルートのリーク **label-allocation-mode** コマンドは **label mode** コマンドに名前が変更されています。

リリース	変更内容
リリース 4.3.2	次の機能がサポートされました。 • ネイバー単位のリンク帯域幅
リリース 5.3.1	次の機能がサポートされました。 • L3VPN iBGP-PE-CE の設定 • 送信元ベースのフロータグ • 過剰パスの破棄
リリース 5.3.2	次の機能がサポートされました。 • グレースフルメンテナンス • ネイバー単位の TCP MSS • BGP DMZ 総帯域幅
リリース 6.0.1	次の機能がサポートされました。 • 過剰パントフロートラップ処理 • BGP 用 64-ECMP

• BGP の実装の前提条件 （3 ページ）
• BGP の実装に関する概要 （4 ページ）
• BGP Monitoring Protocol の概要 （92 ページ）
• BGP の実装方法 （93 ページ）
• BGP の実装の設定例 （200 ページ）
• フロー タグの伝達 （216 ページ）
• 次の作業 （216 ページ）
• その他の参考資料 （216 ページ）

BGP の実装の前提条件

適切なタスク ID を含むタスク グループに関連付けられているユーザ グループに属している必要があります。このコマンド リファレンスには、各コマンドに必要なタスク ID が含まれます。ユーザ グループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA 管理者に連絡してください。

BGP の実装に関する概要

BGP を実装するには、次の概念を理解する必要があります。

BGP 機能の概要

BGP はトランスポート プロトコルとして TCP を使用します。2 台の BGP ルータが互いの間に TCP 接続を形成し（ピア ルータ）、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達するためにルートで経由する必要のあるフル パス（BGP 自律システム番号）を示します。この情報は、ループ フリーである自律システムや、ルーティング動作に制限が適用されるルーティング ポリシーを表すグラフの作成に役立ちます。

TCP 接続を確立して BGP ルーティング情報を交換している 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 BGP ルーティング テーブル全体を交換します。この交換の後、ルーティング テーブルが変更されたとき差分更新が送信されます。BGP は BGP テーブルのバージョン番号を保存します。これはすべての 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。BGP ピア間の接続が維持されていることを確認するキープアライブ パケットが送信され、エラーまたは特殊な状態に応じて通知パケットが送信されます。

（注） `address-family ipv4 rtfilter` コマンドを使用して、RTC（ルートターゲット制約）を有効にするだけです。BGP EVPN の RTC を有効にするためには個別の設定は必要ありません。

（注） マルチ プロトコル ラベル スイッチング（MPLS）レイヤ 3 バーチャル プライベート ネットワーク（VPN）情報を配信するように BGP を設定する方法については、『*Cisco ASR 9000 Series Aggregation Services Router MPLS Configuration Guide*』を参照してください。

BGP による双方向フォワーディング検出（BFD）のサポートについては、『*Cisco ASR 9000 Series Aggregation Services Router Interface and Hardware Configuration Guide*』および『*Cisco ASR 9000 Series Aggregation Services Router Interface and Hardware Command Reference*』を参照してください。

BGP ルータ ID

ネイバー間に BGP セッションを確立するには、BGP にルータ ID を割り当てる必要があります。ルータ ID は、BGP セッションが確立されると、OPEN メッセージに含めて BGP ピアに送信されます。

BGP は次の方法（プリファレンス順）でルータ ID の取得を試みます。

- ルータ コンフィギュレーション モードで **bgp router-id** コマンドを使用して設定されたアドレスを使用する。
- 保存されたループバック アドレス設定を使用してルータがブートされた場合に、システムのループバック インターフェイス上の最大の IPv4 アドレスを使用する。
- 保存された設定に存在しない場合に、設定される最初のループバック アドレスのプライマリ IPv4 アドレスを使用する。

このいずれの方法でもルータ ID を取得できない場合、BGP はルータ ID を持たず、BGP ネイバーとのピアリング セッションを確立できません。そのような場合は、エラー メッセージがシステム ログに記録され、**show bgp summary** コマンドでは、ルータ ID として 0.0.0.0 が表示されます。

ルータ ID を取得した BGP では、さらに適したルータ ID が使用可能になっても、同じルータ ID の使用を続行します。この使用方法によって、いずれの BGP セッションでも不要なフラッピングが発生しないようにします。一方、現在使用中のルータ ID が無効になった場合（インターフェイスがダウンするか、設定が変更されたことによる）、BGP では新しいルータ ID を選択し（上記のルールを使用）、確立したすべてのピアリングセッションをリセットします。

（注） ルータ ID の不要な変更（およびそれによる BGP セッションのフラッピング）を避けるために、**bgp router-id** コマンドを設定することを、強く推奨します。

BGP 最大プレフィックス：過剰パスの破棄

IOS XR BGP の最大プレフィックス機能では、特定のアドレスファミリのネイバーから受信されるプレフィックスの数に上限が課されます。受信されるプレフィックスの数が設定した最大数を超えると、停止通知がネイバーに送信された後、BGPセッションが終了します（これはデフォルト動作です）。手動によるクリアがユーザによって実行されるまで、セッションはダウンしたままになります。セッションは、**clear bgp** コマンドを使用して再開できます。**restart** キーワードを指定した **maximum-prefix** コマンドを使用して、セッションが自動的に起動されるまでの期間を設定できます。プレフィックスの上限はユーザが設定できます。ユーザがそのアドレスファミリに対するプレフィックスの最大数を設定していない場合は、デフォルトの制限値が使用されます。デフォルトの制限については、BGP のデフォルト制限 （6 ページ）を参照してください。

過剰パスの破棄

追加パスを廃棄するオプションが、最大プレフィックス設定に追加されました。過剰パスの破棄オプションを設定すると、プレフィックスが設定した最大値を超えた場合に、ネイバーから受信された過剰なプレフィックスはすべて廃棄されます。ただし、この廃棄によってセッション フラップが発生することはありません。

過剰パスの破棄オプションの利点は次のとおりです。

- BGP のメモリ フットスタンプが制限されます。
- パスが設定された制限を超えるとピアのフラッピングが停止します。

過剰パスの破棄設定が削除されると、BGP は更新機能をサポートしている場合にルート更新メッセージをネイバーに送信します。それ以外の場合、セッションはフラップします。

同じ回線で、最大プレフィックス値が変更された場合のアクションを次に示します。

- 最大値が単独で変更されると、必要に応じてルート更新メッセージが送信されます。
- 新しい最大値が現在のプレフィックス カウント ステートよりも大きい場合、新しいプレフィックス ステートが保存されます。
- 新しい最大値が現在のプレフィックス カウント ステートより小さい場合、新しく設定されたステートの値に一致するように、既存のプレフィックスが一部削除されます。

どのプレフィックスを削除するかを制御する方法は現在ありません。

詳細な設定手順については、過剰パスの破棄の設定 （106 ページ）を参照してください。

機能制限

これらの制約事項は、過剰パスの破棄機能に適用されます。

- ルータがプレフィックスを廃棄すると、ネットワークの残りと一致せず、ルーティングループが起きる可能性があります。
- プレフィックスが廃棄されると、スタンバイおよびアクティブ状態の BGP セッションが別のプレフィックスを廃棄する可能性があります。その結果、NSR スイッチオーバーによって BGP テーブルの矛盾が生じます。
- 過剰パスの破棄設定は、ソフト再設定構成と共存できません。

BGP のデフォルト制限

Cisco IOS XR BGP では、ルータに設定できるネイバーの最大数、および特定のアドレス ファミリのピアから受け入れるプレフィックスの最大数に制限を設定しています。この制限は、ルータにとって、ローカルまたはリモート ネイバーのいずれかの設定ミスに起因する、リソースの枯渇に対する予防措置となります。BGP 設定には、次の制限が適用されます。

- 設定できるピアのデフォルトの最大数は 4000 です。このデフォルトは、**bgp maximum neighbor** コマンドを使用して変更できます。制限の範囲は 1 ～ 15000 です。 最大制限値を超えてさらにピアを設定しようとしたり、現在設定されているピアの数未満の最大制限値を設定しようとしたりすると失敗します。
- アドバタイズメントによりピアが BGP をフラッディングしないようにするために、サポートされているアドレス ファミリごとに、1 つのピアから受け入れるプレフィックスの数に対する制限が課されます。デフォルトの制限値は、該当するアドレス ファミリのピアに対して maximum-prefix *limit* コマンドを設定することにより、上書きできます。ユーザがそ

のアドレス ファミリに対するプレフィックスの最大数を設定していない場合は、次のデフォルト制限値が使用されます。

- IPv4 ユニキャスト：1048576
- IPv4 ラベル付きユニキャスト：131072
- IPv4 トンネル：1048576
- IPv6 ユニキャスト：524288
- IPv6 ラベル付きユニキャスト：131072
- IPv4 マルチキャスト：131072
- IPv6 マルチキャスト：131072
- IPv4 MVPN：2097152
- VPNv4 ユニキャスト：2097152
- IPv4 MDT：131072
- VPNv6 ユニキャスト：1048576
- L2VPN EVPN：2097152

特定のアドレス ファミリのピアから受信したプレフィックスの数が、このアドレス ファミリに対する最大制限値（デフォルト設定またはユーザ設定のいずれかによる）を超えると、停止通知メッセージがそのネイバーに送信され、このネイバーとのピアリングが終了されます。

特定のアドレスファミリのネイバーとのピアリングが確立され、そのネイバーから一定数のプレフィックスをすでに受信した後で、そのネイバーのプレフィックスの最大数が設定されていることがあります。設定されたプレフィックスの最大数が、アドレスファミリのネイバーからすでに受信したプレフィックスの数よりも小さい場合は、設定直後に停止通知メッセージがそのネイバーに送信され、そのネイバーとのピアリングが終了されます。

BGP ネクスト ホップ トラッキング

ネクストホップ情報が変更されると、BGPはルーティング情報ベース（RIB）から通知を受信します（イベント駆動型の通知）。BGP は RIB からネクスト ホップ情報を取得して次の処理を行います。

- ネクスト ホップが到達可能であるかどうかを確認する。
- ネクスト ホップへの完全再帰 IGP メトリックを見つける（最適パス計算で使用）。
- 受信したネクスト ホップを検証する。
- 発信ネクスト ホップを計算する。

- ネイバーの到達可能性および接続を確認する。

BGP は、次のいずれかのイベントが発生したときに通知を受けます。

- ネクスト ホップが到達不能になった。
- ネクスト ホップが到達可能になった。
- ネクスト ホップへの完全な繰り返し IGP メトリックが変更される。
- ファースト ホップの IP アドレスまたはファースト ホップのインターフェイスが変更される。
- ネクスト ホップが接続された。
- ネクスト ホップが接続解除された。
- ネクスト ホップがローカル アドレスになった。
- ネクスト ホップが非ローカル アドレスになった。

（注） 到達可能性および再帰メトリック イベントは、最適パスの再計算をトリガーします。

RIB からのイベント通知は、クリティカルおよび非クリティカルとして分類されます。クリティカルおよび非クリティカルイベントの通知は、別々のバッチで送信されます。ただし、非クリティカルイベントが保留中であり、クリティカルイベントを読み込む要求がある場合は、非クリティカル イベントがクリティカル イベントとともに送信されます。

- クリティカル イベントは、ネクスト ホップの到達可能性（到達可能と到達不能）、接続性（接続と非接続）、および局在性（ローカルと非ローカル）に関係があります。これらのイベントの通知は遅延しません。
- 非クリティカルイベントには、IGP メトリックの変更のみが含まれます。これらのイベントは 3 秒の間隔で送信されます。メトリック変更イベントは最後の 1 つが送信されてから 3 秒後にバッチ処理され、送信されます。

クリティカルおよび非クリティカル イベントのネクスト ホップ トリガー遅延は、nexthop trigger-delay コマンドを使用して、クリティカルおよび非クリティカル イベントの最小バッチ間隔を指定するように設定できます。トリガー遅延は、アドレス ファミリに依存します。

BGP ネクストホップ トラッキング機能では、次の特性を持つルートを持つネクスト ホップだけを BGP ルートの解決に使用するように指定することができます。

- 集約ルートを回避するために、プレフィックスの長さは指定された値よりも長くなっている。
- 振動につながる可能性のあるネクストホップの解決に BGP ルートが使用されないように、選択したリストにソース プロトコルが含まれている。

このルート ポリシーのフィルタリングが可能なのは、RIB により、ネクスト ホップを解決するルートのソース プロトコル、およびこのルートに関連付けられているマスクの長さが特定されるからです。nexthop route-policy コマンドは、ルート ポリシーを指定するために使用します。

ネクストホップ接続点を使用したネクストホップのルートポリシーのフィルタリングについては、『Cisco ASR 9000 シリーズ アグリゲーション サービス ルータ ルーティング設定ガイド』の「Cisco ASR 9000 シリーズ ルータ での ルーティング ポリシー言語の実装」のモジュールを参照してください。

範囲を指定した IPv4/VPNv4 テーブル ウォーク

処理するアドレス ファミリを判別するために、ネクスト ホップと関連付けられたゲートウェイ コンテキストを逆参照し、次に、ゲートウェイ コンテキストを調べてそのゲートウェイ コンテキストを使用しているアドレス ファミリを判別することにより、ネクスト ホップ通知が受信されます。IPv4 ユニキャストと VPNv4 ユニキャスト アドレス ファミリは、RIB 内の IPv4 ユニキャスト テーブルに登録されるため、同じゲートウェイ コンテキストを共有します。その結果、RIB から IPv4 ユニキャスト ネクスト ホップ通知を受信したときは、グローバル IPv4 ユニキャスト テーブルと VPNv4 テーブルの両方が処理されます。ネクスト ホップでマスクを保持することで、そのネクスト ホップが、IPv4 ユニキャストまたは VPNv4 ユニキャスト、あるいはその両方に属しているかどうかを示します。この範囲を指定したテーブル ウォークにより、適切なアドレス ファミリ テーブル内に処理が限定されます。

アドレス ファミリ処理の並べ替え

Cisco IOS XR ソフトウェア では、アドレス ファミリの数値に基づいてアドレス ファミリ テーブルを探索します。ネクスト ホップ通知バッチを受信すると、アドレス ファミリ処理の順序が、次の順序に並べ替えられます。

- IPv4 トンネル
- VPNv4 ユニキャスト
- IPv4 ラベル付きユニキャスト
- IPv4 ユニキャスト
- IPv4 マルチキャスト
- IPv6 ユニキャスト

ネクスト ホップ処理の新規スレッド

spkr プロセスの critical-event スレッドでは、ネクスト ホップ、双方向フォワーディング検出（BFD）、および高速外部フェールオーバー（FEF）の通知のみを処理します。この critical-event スレッドによって、BGP コンバージェンスは、大量の時間を必要とするおそれのある他のイベントによる悪影響が確実に受けなくなります。

show、clear、debug コマンド

show bgp nexthops コマンドは、ネクスト ホップ通知に関する統計情報、この通知の処理に費やした時間、およびRIBに登録されている各ネクストホップに関する詳細を表示します。**clear bgp nexthop performance-statistics** コマンドは、モニタリングを容易にするために、ネクストホップの **show** コマンドの処理部分に関する累積統計情報をクリアします。**clear bgp nexthop registration** コマンドは、ネクスト ホップを RIB に非同期的に登録します。ネクストホップの **show** コマンドおよび **clear** コマンドについては、 *Routing Command Reference for Cisco ASR 9000 Series Routers*の「*BGP Commands on* Cisco ASR 9000 シリーズ ルータ 」のモジュールを参照してください。

debug bgp nexthop コマンドは、ネクスト ホップ処理の情報を表示します。**out** キーワードを指定すると、RIB に登録されている BGP のネクスト ホップに関するデバッグ情報のみが表示されます。 **in** キーワードを指定した場合は、RIB から受信したネクストホップ通知に関するデバッグ情報が表示されます。 **out** キーワードでは、RIB に送信されたネクストホップ通知に関するデバッグ情報が表示されます。『 *Cisco ASR 9000 Series Aggregation Services Router Routing Debug Command Reference*』の「*BGP Debug Commands on Cisco ASR 9000 Series Aggregation Services Router* 」のモジュールを参照してください。

BGP の自律システム番号形式

自律システム番号（ASN）は、自律システム（AS）を識別するために使用されるグローバルに一意な識別子であり、これにより、AS では、ネイバー AS との間で外部ルーティング情報を交換できるようになります。一意の ASN は、BGP ルーティングで使用するために各 AS に割り当てられます。BGP では、ASN を 2 バイトの番号および 4 バイトの番号としてエンコードします。

2 バイト自律システム番号形式

2 バイト ASN は asplain 表記で表されます。2 バイトの範囲は 1 ～ 65535 です。

4 バイト自律システム番号形式

2 バイト自律システム番号（ASN）がいつか枯渇するときに備えて、BGP では 4 バイト ASN をサポートしています。4 バイト ASN は、asplain 表記と asdot 表記の両方で表されます。

asplain 表記での 4 バイト ASN のバイトの範囲は 1 ～ 4294967295 です。AS は 4 バイトの 10 進数として表されます。4 バイト ASN の asplain 表現は draft-ietf-idr-as-representation-01.txt で定義されています。

asdot 形式の 4 バイト ASN の場合は、4 バイトの範囲は 1.0 ～ 65535.65535 で、次の形式になります。

high-order-16-bit-value-in-decimal . low-order-16-bit-value-in-decimal

BGP の 4 バイト ASN 機能は、4 バイト AS 番号をサポートしていない BGP スピーカーをまたがって、4 バイトをベースとする AS パス情報を伝播するために使用されます。ASN のサイズを 2 バイトから 4 バイトに拡張するための情報については、draft-ietf-idr-as4bytes-12.txt を参照してください。AS は 4 バイトの 10 進数として表されます。

as-format コマンド

as-format コマンドは、ASN 表記を asdot に設定します。**as-format** コマンドを設定していない場合のデフォルト値は asplain です。

BGP の設定

Cisco IOS XR ソフトウェアでの BGP は、特定のネイバーに対するすべての設定を、ネイバー設定の下の 1 箇所にまとめる必要のある、ネイバーベースの設定モデルに従っています。ネイバー間での設定の共有と、アップデート メッセージの共有のいずれについても、ピア グループはサポートされていません。ピア グループの概念は、BGP 設定でテンプレートとして使用する一連の設定グループおよびネイバー間でアップデート メッセージを共有するために自動生成されるアップデート グループによって置き換えられました。

コンフィギュレーション モード

BGP コンフィギュレーションは、モードにグループ化されています。ここではいくつかの BGP コンフィギュレーション モードの開始方法について説明します。現行のモードで **?** コマンドを入力すると、そのモードで使用可能なコマンドを表示できます。

ルータ コンフィギュレーション モード

次に、ルータ コンフィギュレーション モードを開始する例を示します。

```
RP/0/RSP0/cpu 0: router# configuration
RP/0/RSP0/cpu 0: router(config)# router bgp 140
RP/0/RSP0/cpu 0: router(config-bgp)#
```

ルータ アドレス ファミリ コンフィギュレーション モード

次に、ルータ アドレス ファミリ コンフィギュレーション モードを開始する例を示します。

```
RP/0/RSP0/cpu 0: router(config)# router bgp 112
RP/0/RSP0/cpu 0: router(config-bgp)# address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-af)#
```

ネイバー コンフィギュレーション モード

次に、ネイバー コンフィギュレーション モードを開始する例を示します。

```
RP/0/RSP0/cpu 0: router(config)# router bgp 140
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 10.0.0.1
RP/0/RSP0/cpu 0: router(config-bgp-nbr)#
```

ネイバー アドレス ファミリ コンフィギュレーション モード

次に、ネイバー アドレス ファミリ コンフィギュレーション モードを開始する例を示します。

```
RP/0/RSP0/cpu 0: router(config)# router bgp 112
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 10.0.0.1
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)#
```

VRF コンフィギュレーション モード

次に、VPN ルーティングおよび転送（VRF）コンフィギュレーション モードを開始する例を示します。

```
RP/0/RSP0/cpu 0: router(config)# router bgp 140
RP/0/RSP0/cpu 0: router(config-bgp)# vrf vrf_A
RP/0/RSP0/cpu 0: router(config-bgp-vrf)#
```

VRF アドレス ファミリ コンフィギュレーション モード

次に、VRF アドレス ファミリ コンフィギュレーション モードを開始する例を示します。

```
RP/0/RSP0/cpu 0: router(config)# router bgp 112
RP/0/RSP0/cpu 0: router(config-bgp)# vrf vrf_A
RP/0/RSP0/cpu 0: router(config-bgp-vrf)# address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-vrf-af)#
```

VRF アドレスファミリでの復元力のある *CE* 単位のラベルモードの設定

VRF アドレスファミリに復元力のある CE 単位のラベルモードを設定するには、次のタスクを実行します。

（注） 復元力のある CE 単位の 6PE ラベル割り当ては、CRS-1 ルータと CRS-3 ルータではサポートされていません。ASR 9000 ルータでのみサポートされています。

手順の概要

1. **configure**
2. **router bgp***as-number*
3. **vrf***vrf-instance*
4. **address-family {ipv4 | ipv6} unicast**
5. **label mode per-ce**
6. 次のいずれかを実行します。
   - **end**
   - **commit**

手順の詳細

ステップ 1 **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
RP/0/RSP0/cpu 0: router(config)#
```

グローバル コンフィギュレーション モードを開始します。

ステップ 2 **router bgp***as-number*

例：

```
RP/0/RSP0/cpu 0: router(config)# router bgp 666
RP/0/RSP0/cpu 0: router(config-bgp)#
```

自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。

ステップ 3 **vrf***vrf-instance*

例：

```
RP/0/RSP0/cpu 0: router(config-bgp)# vrf vrf-pe
RP/0/RSP0/cpu 0: router(config-bgp-vrf)#
```

VRF インスタンスを設定します。

ステップ 4 **address-family {ipv4 | ipv6} unicast**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp-vrf)# address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-vrf-af)#
```

IPv4 または IPv6 のいずれかのアドレス ファミリ ユニキャストを指定し、アドレス ファミリのコンフィギュレーション サブモードを開始します。

ステップ 5 **label mode per-ce**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp-vrf-af)# label mode per-ce
RP/0/RSP0/cpu 0: router(config-bgp-vrf-af)#
```

復元力のある CE 単位のラベルモードを設定します。

ステップ 6 次のいずれかを実行します。

- **end**
- **commit**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp-vrf-af)# end
```

または

```
RP/0/RSP0/cpu 0: router(config-bgp-vrf-af)# commit
```

設定変更を保存します。

- **end** コマンドを実行すると、変更をコミットするように要求されます。

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:
```

  - **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
  - **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
  - **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。

- 実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

ルートポリシーを使用した復元力のある *CE* 単位のラベルモードの設定

ルートポリシーを使用して復元力のある CE 単位のラベルモードを設定するには、次のタスクを実行します。

（注） 復元力のある CE 単位の 6PE ラベル割り当ては、CRS-1 ルータと CRS-3 ルータではサポートされていません。ASR 9000 ルータでのみサポートされています。

手順の概要

1. **configure**
2. **route-policy** *policy-name*
3. **set label mode per-ce**
4. 次のいずれかを実行します。
   - **end**
   - **commit**

手順の詳細

ステップ 1 **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
RP/0/RSP0/cpu 0: router(config)#
```

グローバル コンフィギュレーション モードを開始します。

ステップ 2 **route-policy** *policy-name*

例：

```
RP/0/RSP0/cpu 0: router(config)# route-policy route1
RP/0/RSP0/cpu 0: router(config-rpl)#
```

ルート ポリシーを作成し、ルート ポリシー コンフィギュレーション モードを開始します。

ステップ 3 **set label mode per-ce**

例：

```
RP/0/RSP0/cpu 0: router(config-rpl)# set label mode per-ce
RP/0/RSP0/cpu 0: router(config-rpl)#
```

復元力のある CE 単位のラベルモードを設定します。

ステップ 4 次のいずれかを実行します。

- **end**
- **commit**

例：

```
RP/0/RSP0/cpu 0: router(config-rpl)# end
```

または

```
RP/0/RSP0/cpu 0: router(config-rpl)# commit
```

設定変更を保存します。

- **end** コマンドを実行すると、変更をコミットするように要求されます。

  ```
  Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:
  ```

  - **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
  - **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。

- **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。

- 実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

VRF ネイバー コンフィギュレーション モード

次に、VRF ネイバー コンフィギュレーション モードを開始する例を示します。

```
Router(config)# router bgp 140
Router(config-bgp)# vrf vrf_A
Router(config-bgp-vrf)# neighbor 11.0.1.2
Router(config-bgp-vrf-nbr)#
```

VRF ネイバー アドレス ファミリ コンフィギュレーション モード

次に、VRF ネイバー アドレス ファミリ コンフィギュレーション モードを開始する例を示します。

```
RP/0/RSP0/cpu 0: router(config)# router bgp 112
RP/0/RSP0/cpu 0: router(config-bgp)# vrf vrf_A
RP/0/RSP0/cpu 0: router(config-bgp-vrf)# neighbor 11.0.1.2
RP/0/RSP0/cpu 0: router(config-bgp-vrf-nbr)# address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-vrf-nbr-af)#
```

VPNv4 アドレス ファミリ コンフィギュレーション モード

次に、VPNv4 ネイバー アドレス ファミリ コンフィギュレーション モードを開始する例を示します。

```
RP/0/RSP0/cpu 0: router(config)# router bgp 152
RP/0/RSP0/cpu 0: router(config-bgp)# address-family vpnv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-af)#
```

L2VPN アドレス ファミリ コンフィギュレーション モード

次に、L2VPN ネイバー アドレス ファミリ コンフィギュレーション モードを開始する例を示します。

```
RP/0/RSP0/cpu 0: router(config)# router bgp 100
RP/0/RSP0/cpu 0: router(config-bgp)# address-family l2vpn vpls-vpws
RP/0/RSP0/cpu 0: router(config-bgp-af)#
```

ネイバーサブモード

Cisco IOS XR BGP では、ネイバーサブモードを使用することにより、 **neighbor** キーワードおよびネイバーアドレスによってすべての設定にプレフィックスを付けることなく、設定を入力できます。

- Cisco IOS XR ソフトウェアにはネイバー用のサブモードがあり、このモードではすべてのコマンドに「neighbor *x.x.x.x*」というプレフィックスを付ける必要がなくなります。

  Cisco IOS XR ソフトウェアでの設定は次のとおりです。

```
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 192.23.1.2
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# remote-as 2002
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# address-family ipv4 unicast
```

- ネイバー コンフィギュレーション サブモード内のアドレス ファミリ コンフィギュレーションサブモードは、アドレスファミリ固有のネイバー設定の入力に使用できます。Cisco IOS XR ソフトウェアでの設定は次のとおりです。

```
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 2002::2
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# remote-as 2023
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# address-family ipv6 unicast
RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# next-hop-self
RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# route-policy one in
```

- ネイバーアドレスファミリ コンフィギュレーションサブモードで、ネイバー固有の IPv4、IPv6、VPNv4、または VPNv6 コマンドを入力する必要があります。Cisco IOS XR ソフトウェア での設定は次のとおりです。

```
RP/0/RSP0/cpu 0: router(config)# router bgp 109
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 192.168.40.24
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# remote-as 1
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# maximum-prefix 1000
```

- VRF ネイバー アドレス ファミリ コンフィギュレーション サブモードで、ネイバー固有の IPv4 および IPv6 コマンドを入力する必要があります。Cisco IOS XR ソフトウェア での設定は次のとおりです。

```
RP/0/RSP0/cpu 0: router(config)# router bgp 110
RP/0/RSP0/cpu 0: router(config-bgp)# vrf vrf_A
RP/0/RSP0/cpu 0: router(config-bgp-vrf)# neighbor 11.0.1.2
RP/0/RSP0/cpu 0: router(config-bgp-vrf-nbr)# address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-vrf-nbr-af)# route-policy pass all in
```

コンフィギュレーション テンプレート

af-group、**session-group**、および **neighbor-group** コンフィギュレーション コマンドは、Cisco IOS XR ソフトウェアでのネイバー設定にテンプレートのサポートを提供します。

af-group コマンドは、アドレスファミリ固有のネイバーコマンドを IPv4、IPv6、または IPNv4、アドレスファミリ内でグループ化するために使用します。同じアドレス ファミリ コンフィギュレーションを持つネイバーは、アドレス ファミリ固有の設定のアドレス ファミリ グループ（af-group）の名前を使用できます。ネイバーは、**use** コマンドを使用してアドレスファミリグループから設定を継承します。ネイバーがアドレス ファミリ グループを使用するように設定してある場合、ネイバーでは（デフォルトで）アドレス ファミリ グループから設定全体を継承します。ただし、そのネイバーに対して明示的に設定されている項目がある場合、ネイバーでは、設定の一部をアドレス ファミリ グループから継承しません。アドレス ファミリ グループ コンフィギュレーションは、BGP ルータ コンフィギュレーション モードで入力します。次に、アドレス ファミリ グループ コンフィギュレーション モードを開始する例を示します。

```
RP/0/RSP0/cpu 0: router(config)# router bgp 140
RP/0/RSP0/cpu 0: router(config-bgp)# af-group afmcast1 address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-afgrp)#
```

アドレスファミリに依存しないコンフィギュレーションをネイバーが継承してくるセッショングループを作成するには、**session-group** コマンドを使用します。ネイバーは、**use** コマンドを使用してセッショングループから設定を継承します。ネイバーがセッション グループを使用するように設定してある場合、ネイバーでは（デフォルトで）セッション グループの設定全体を継承します。そのネイバーに直接設定されている場合、ネイバーでは一部の設定をセッション グループから継承しません。次に、セッション グループ コンフィギュレーション モードを開始する例を示します。

```
RP/0/RSP0/cpu 0: router# router bgp 140
RP/0/RSP0/cpu 0: router(config-bgp)# session-group session1
RP/0/RSP0/cpu 0: router(config-bgp-sngrp)#
```

neighbor-group コマンドを使用すると、1 つ以上のネイバーに同一の設定を適用しやすくなります。ネイバー グループにはセッション グループとアドレス ファミリ グループを含めることができ、またネイバーに対する全体的な設定を含めることができます。ネイバーグループを設定すると、**use** コマンドを使用してネイバーはグループの設定を継承できます。ネイバー グループを使用するようにネイバーを設定してある場合、ネイバーでは、ネイバー グループの BGP 設定全体を継承します。

次に、ネイバー グループ コンフィギュレーション モードを開始する例を示します。

```
RP/0/RSP0/cpu 0: router(config)# router bgp 123
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor-group nbrgroup1
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)#
```

次に、ネイバー グループ アドレス ファミリ コンフィギュレーション モードを開始する例を示します。

```
RP/0/RSP0/cpu 0: router(config)# router bgp 140
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor-group nbrgroup1
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp-af)#
```

- ただし、そのネイバーに対して明示的に設定されている項目がある場合、ネイバーでは、設定の一部をネイバー グループから継承しません。また、セッション グループまたはアドレス ファミリ グループも使用されている場合は、ネイバー グループの設定の一部が非表示になることがあります。

Cisco IOS XR ソフトウェア での設定のグループ化は、次の効果を持ちます。

- セッション グループ レベルでコマンドを入力すると、アドレス ファミリに依存しないコマンドが定義されます（ネイバー サブモードでの同じコマンドと同様）。
- アドレス ファミリ グループ レベルでコマンドを入力すると、指定したアドレス ファミリに対するアドレス ファミリ依存のコマンドが定義されます（ネイバー アドレス ファミリ コンフィギュレーション サブモードでの同じコマンドと同様）。
- ネイバーグループレベルでコマンドを入力すると、アドレスファミリに依存しないコマンドと、アドレスファミリ依存するコマンドが各アドレスファミリに定義され（使用可能なすべての **neighbor** コマンドと同様）、アドレスファミリグループのコマンドとセッショングループのコマンドに **use** コマンドが定義されます。

テンプレート継承ルール

Cisco IOS XR ソフトウェアの場合、BGP ネイバーまたはグループは、他の設定グループから設定を継承します。

アドレス ファミリに依存しない設定

- ネイバーは、セッション グループおよびネイバー グループから継承できます。
- ネイバー グループは、セッション グループおよび他のネイバー グループから継承できます。
- セッション グループは、他のセッション グループから継承できます。
- セッション グループとネイバー グループを使用しているネイバーの場合は、セッション グループでの設定が、ネイバー グループのグローバル アドレス ファミリ設定よりも優先されます。

アドレス ファミリ依存の設定

- アドレス ファミリ グループは、他のアドレス ファミリ グループから継承できます。
- ネイバー グループは、アドレス ファミリ グループおよび他のネイバー グループから継承できます。
- ネイバーは、アドレス ファミリ グループおよびネイバー グループから継承できます。

設定グループ継承ルールは、次のように優先順位付けされます。

1. 項目がネイバーに直接設定されている場合は、その値が使用されます。次の例では、ネイバー グループとネイバー設定の両方にアドバタイズメント間隔が設定されており、ネイバー設定からのアドバタイズメント間隔が使用されています。

```
RP/0/RSP0/cpu 0: router(config)# router bgp 140
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor-group AS_1
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# advertisement-interval 15
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# exit
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 10.1.1.1
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# remote-as 1
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# use neighbor-group AS_1
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# advertisement-interval 20
```

show bgp neighbors コマンドからの次の出力は、使用されたアドバタイズメント間隔が 20 秒であることを示しています。

```
RP/0/RSP0/cpu 0: router# show bgp neighbors 10.1.1.1

BGP neighbor is 10.1.1.1, remote AS 1, local AS 140, external link
 Remote router ID 0.0.0.0
  BGP state = Idle
  Last read 00:00:00, hold time is 180, keepalive interval is 60 seconds
  Received 0 messages, 0 notifications, 0 in queue
  Sent 0 messages, 0 notifications, 0 in queue
  Minimum time between advertisement runs is 20 seconds

 For Address Family: IPv4 Unicast
  BGP neighbor version 0
  Update group: 0.1
  eBGP neighbor with no inbound or outbound policy; defaults to 'drop'
  Route refresh request: received 0, sent 0
  0 accepted prefixes
  Prefix advertised 0, suppressed 0, withdrawn 0, maximum limit 524288
  Threshold for warning message 75%

  Connections established 0; dropped 0
  Last reset 00:00:14, due to BGP neighbor initialized
  External BGP neighbor not directly connected.
```

2. 上記と異なり、セッション グループまたはネイバー グループから継承する設定と、ネイバー上での直接設定のある項目の場合は、ネイバー上の設定が使用されます。セッション グループまたはアドレス ファミリ グループから継承するように設定されている一方で、直接設定されている値のないネイバーの場合は、セッション グループまたはアドレス ファミリ グループにある値が使用されます。次の例では、ネイバー グループとセッション グループにアドバタイズメント間隔が設定されており、セッション グループからのアドバタイズメント間隔値が使用されています。

```
RP/0/RSP0/cpu 0: router(config)# router bgp 140
RP/0/RSP0/cpu 0: router(config-bgp)# session-group AS_2
RP/0/RSP0/cpu 0: router(config-bgp-sngrp)# advertisement-interval 15
RP/0/RSP0/cpu 0: router(config-bgp-sngrp)# exit
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor-group AS_1
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# advertisement-interval 20
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# exit
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 192.168.0.1
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# remote-as 1
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# use session-group AS_2
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# use neighbor-group AS_1
```

show bgp neighbors コマンドからの次の出力は、使用されたアドバタイズメント間隔が 15 秒であることを示しています。

```
RP/0/RSP0/cpu 0: router# show bgp neighbors 192.168.0.1

BGP neighbor is 192.168.0.1, remote AS 1, local AS 140, external link
 Remote router ID 0.0.0.0
  BGP state = Idle
  Last read 00:00:00, hold time is 180, keepalive interval is 60 seconds
  Received 0 messages, 0 notifications, 0 in queue
  Sent 0 messages, 0 notifications, 0 in queue
  Minimum time between advertisement runs is 15 seconds

 For Address Family: IPv4 Unicast
  BGP neighbor version 0
  Update group: 0.1
  eBGP neighbor with no inbound or outbound policy; defaults to 'drop'
  Route refresh request: received 0, sent 0
  0 accepted prefixes
  Prefix advertised 0, suppressed 0, withdrawn 0, maximum limit 524288
  Threshold for warning message 75%

  Connections established 0; dropped 0
  Last reset 00:03:23, due to BGP neighbor initialized
  External BGP neighbor not directly connected.
```

3. 上記の例と異なり、ネイバー グループを使用し、セッション グループもアドレス ファミリ グループも使用しないネイバーの場合は、直接または継承によってネイバー グループから設定値を取得できます。次の例では、ネイバーに直接設定されておらず、セッション グループを使用していないため、ネイバー グループからのアドバタイズメント間隔が使用されます。

```
RP/0/RSP0/cpu 0: router(config)# router bgp 150
RP/0/RSP0/cpu 0: router(config-bgp)# session-group AS_2
RP/0/RSP0/cpu 0: router(config-bgp-sngrp)# advertisement-interval 20
RP/0/RSP0/cpu 0: router(config-bgp-sngrp)# exit
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor-group AS_1
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# advertisement-interval 15
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# exit
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 192.168.1.1
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# remote-as 1
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# use neighbor-group AS_1
```

show bgp neighbors コマンドからの次の出力は、使用されたアドバタイズメント間隔が 15 秒であることを示しています。

```
 RP/0/RSP0/cpu 0: router# show bgp neighbors 192.168.1.1

BGP neighbor is 192.168.2.2, remote AS 1, local AS 140, external link
 Remote router ID 0.0.0.0
  BGP state = Idle
  Last read 00:00:00, hold time is 180, keepalive interval is 60 seconds
  Received 0 messages, 0 notifications, 0 in queue
  Sent 0 messages, 0 notifications, 0 in queue
  Minimum time between advertisement runs is 15 seconds

 For Address Family: IPv4 Unicast
```

```
 BGP neighbor version 0
 Update group: 0.1
 eBGP neighbor with no outbound policy; defaults to 'drop'
 Route refresh request: received 0, sent 0
 Inbound path policy configured
 Policy for incoming advertisements is POLICY_1
 0 accepted prefixes
 Prefix advertised 0, suppressed 0, withdrawn 0, maximum limit 524288
 Threshold for warning message 75%

 Connections established 0; dropped 0
 Last reset 00:01:14, due to BGP neighbor initialized
 External BGP neighbor not directly connected.
```

同じルールを説明するために、次の例では、アドバタイズメント間隔に 15（セッション グループから）および25（ネイバー グループから）を設定する方法を示します。セッション グループのアドバタイズメント間隔設定は、ネイバー グループの設定よりも優先されます。インバウンド ポリシーには、ネイバー グループから POLICY_1 が設定されます。

```
RP/0/RSP0/cpu 0: routerconfig)# router bgp 140
RP/0/RSP0/cpu 0: router(config-bgp)# session-group ADV
RP/0/RSP0/cpu 0: router(config-bgp-sngrp)# advertisement-interval 15
RP/0/RSP0/cpu 0: router(config-bgp-sngrp)# exit
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor-group ADV_2
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# advertisement-interval 25
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp-af)# route-policy POLICY_1 in
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp-af)# exit
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# exit
RP/0/RSP0/cpu 0: router(config-bgp)# exit
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 192.168.2.2
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# remote-as 1
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# use session-group ADV
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# use neighbor-group ADV_2
```

show bgp neighbors コマンドからの次の出力は、使用されたアドバタイズメント間隔が 15 秒であることを示しています。

```
RP/0/RSP0/cpu 0: router# show bgp neighbors 192.168.2.2

BGP neighbor is 192.168.2.2, remote AS 1, local AS 140, external link
 Remote router ID 0.0.0.0
  BGP state = Idle
  Last read 00:00:00, hold time is 180, keepalive interval is 60 seconds
  Received 0 messages, 0 notifications, 0 in queue
  Sent 0 messages, 0 notifications, 0 in queue
  Minimum time between advertisement runs is 15 seconds

 For Address Family: IPv4 Unicast
  BGP neighbor version 0
  Update group: 0.1
  eBGP neighbor with no inbound or outbound policy; defaults to 'drop'
  Route refresh request: received 0, sent 0
  0 accepted prefixes
  Prefix advertised 0, suppressed 0, withdrawn 0, maximum limit 524288
  Threshold for warning message 75%

  Connections established 0; dropped 0
  Last reset 00:02:03, due to BGP neighbor initialized
```

```
  External BGP neighbor not directly connected.
```

4. 指定しない場合は、デフォルト値が使用されます。次の例では、ネイバー設定とネイバーグループ設定のいずれも使用するようにネイバーに設定されていないため、ネイバー 10.0.101.5 のアドバタイズメントの最小実行時間間隔は、30 秒（デフォルト）に設定されています。

```
RP/0/RSP0/cpu 0: router(config)# router bgp 140
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor-group AS_1
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# remote-as 1
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# exit
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor-group adv_15
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# remote-as 10
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# advertisement-interval 15
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# exit
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 10.0.101.5
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# use neighbor-group AS_1
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# exit
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 10.0.101.10
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# use neighbor-group adv_15
```

show bgp neighbors コマンドからの次の出力は、使用されたアドバタイズメント間隔が 30 秒であることを示しています。

```
RP/0/RSP0/cpu 0: router# show bgp neighbors 10.0.101.5

BGP neighbor is 10.0.101.5, remote AS 1, local AS 140, external link
 Remote router ID 0.0.0.0
  BGP state = Idle
  Last read 00:00:00, hold time is 180, keepalive interval is 60 seconds
  Received 0 messages, 0 notifications, 0 in queue
  Sent 0 messages, 0 notifications, 0 in queue
  Minimum time between advertisement runs is 30 seconds

 For Address Family: IPv4 Unicast
  BGP neighbor version 0
  Update group: 0.2
  eBGP neighbor with no inbound or outbound policy; defaults to 'drop'
  Route refresh request: received 0, sent 0
  0 accepted prefixes
  Prefix advertised 0, suppressed 0, withdrawn 0, maximum limit 524288
  Threshold for warning message 75%
Connections established 0; dropped 0
  Last reset 00:00:25, due to BGP neighbor initialized
  External BGP neighbor not directly connected.
```

グループが他のグループから設定を継承する場合に使用される継承ルールは、グループから継承するネイバーに対して適用されるルールと同じです。

継承した設定の表示

BGP によって継承された設定を表示するには、次の **show** コマンドを使用します。

show bgp neighbors

show bgp neighbors コマンドは、ネイバーの BGP 設定に関する情報を表示する場合に使用します。

- このネイバーで使用されるセッショングループ、ネイバーグループ、またはアドレスファミリグループから継承するすべての設定など、ネイバーの有効な設定を表示するには、**configuration** キーワードを使用します。
- このネイバーで設定を継承できる、セッショングループ、ネイバーグループ、およびアドレスファミリグループを表示するには、**inheritance** キーワードを使用します。

次に示す **show bgp neighbors** コマンドの例は、この設定例に基づいています。

```
RP/0/RSP0/cpu 0: router(config)# router bgp 142
RP/0/RSP0/cpu 0: router(config-bgp)# af-group GROUP_3 address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-afgrp)# next-hop-self
RP/0/RSP0/cpu 0: router(config-bgp-afgrp)# route-policy POLICY_1 in
RP/0/RSP0/cpu 0: router(config-bgp-afgrp)# exit
RP/0/RSP0/cpu 0: router(config-bgp)# session-group GROUP_2
RP/0/RSP0/cpu 0: router(config-bgp-sngrp)# advertisement-interval 15
RP/0/RSP0/cpu 0: router(config-bgp-sngrp)# exit
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor-group GROUP_1
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# use session-group GROUP_2
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# ebgp-multihop 3
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp-af)# weight 100
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp-af)# send-community-ebgp
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp-af)# exit

RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# exit
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 192.168.0.1
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# remote-as 2
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# use neighbor-group GROUP_1
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# use af-group GROUP_3
RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# weight 200
```

次に、 **inheritance** キーワードを指定した **show bgp neighbors** コマンドの出力例を示します。この例は、ネイバーが、ネイバー グループ GROUP_1 からセッション パラメータを継承すること、ネイバー グループ GROUP_1 はセッション グループ GROUP_2 から継承していることを示します。このネイバーは、IPv4 ユニキャスト パラメータをアドレス ファミリ グループ GROUP_3 から継承し、IPv4 マルチキャスト パラメータをネイバー グループ GROUP_1 から継承します。

```
RP/0/RSP0/cpu 0: router# show bgp neighbors 192.168.0.1 inheritance

  Session:        n:GROUP_1 s:GROUP_2
  IPv4 Unicast:   a:GROUP_3
  IPv4 Multicast: n:GROUP_1
```

次に、 **configuration** キーワードを指定した **show bgp neighbors** コマンドの出力例を示します。この例は、設定の各項目の継承元を示すか、ネイバーへの直接設定（継承元が [] ）を示し

ます。たとえば、**ebgp-multihop 3** コマンドはネイバーグループ GROUP_1 から継承されており、**next-hop-self** コマンドはアドレスファミリグループ GROUP_3 から継承されています。

```
RP/0/RSP0/cpu 0: router# show bgp neighbors 192.168.0.1 configuration

neighbor 192.168.0.1
 remote-as 2                    []
 advertisement-interval 15      [n:GROUP_1 s:GROUP_2]
 ebgp-multihop 3                [n:GROUP_1]
 address-family ipv4 unicast    []
  next-hop-self                 [a:GROUP_3]
  route-policy POLICY_1    in   [a:GROUP_3]
  weight 200                    []
 address-family ipv4 multicast [n:GROUP_1]
  default-originate             [n:GROUP_1]
```

show bgp af-group

アドレスファミリグループを表示するには、**show bgp af-group** コマンドを使用します。

- このアドレスファミリグループで使用されるアドレスファミリグループから継承したすべての設定など、アドレスファミリグループの有効な設定を表示するには、**configuration** キーワードを使用します。
- このアドレスファミリグループで設定を継承できるアドレスファミリグループを表示するには、**inheritance** キーワードを使用します。
- このアドレスファミリグループから設定を継承するネイバー、ネイバーグループ、アドレスファミリ グループを表示するには、**users** キーワードを使用します。

次に示す **show bgp af-group** コマンドの例は、この設定例に基づいています。

```
RP/0/RSP0/cpu 0: router(config)# router bgp 140
RP/0/RSP0/cpu 0: router(config-bgp)# af-group GROUP_3 address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-afgrp)# remove-private-as
RP/0/RSP0/cpu 0: router(config-bgp-afgrp)# route-policy POLICY_1 in
RP/0/RSP0/cpu 0: router(config-bgp-afgrp)# exit
RP/0/RSP0/cpu 0: router(config-bgp)# af-group GROUP_1 address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-afgrp)# use af-group GROUP_2
RP/0/RSP0/cpu 0: router(config-bgp-afgrp)# maximum-prefix 2500 75 warning-only
RP/0/RSP0/cpu 0: router(config-bgp-afgrp)# default-originate
RP/0/RSP0/cpu 0: router(config-bgp-afgrp)# exit
RP/0/RSP0/cpu 0: router(config-bgp)# af-group GROUP_2 address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-afgrp)# use af-group GROUP_3
RP/0/RSP0/cpu 0: router(config-bgp-afgrp)# send-community-ebgp
RP/0/RSP0/cpu 0: router(config-bgp-afgrp)# send-extended-community-ebgp
RP/0/RSP0/cpu 0: router(config-bgp-afgrp)# capability orf prefix both
```

次に、 **show bgp af-group** コマンドで **configuration** キーワードを指定した場合の出力例を示します。この例では、各設定項目がどこから継承されたかを表しています。**default-originate** コマンドは、このアドレスファミリグループで直接設定されています（[] で示されています）。**remove-private-as** コマンドは、アドレスファミリグループ GROUP_2 から継承されています。アドレスファミリグループ GROUP_2 は、アドレスファミリグループ GROUP_3 から継承されています。

```
RP/0/RSP0/cpu 0: router# show bgp af-group GROUP_1 configuration

af-group GROUP_1 address-family ipv4 unicast
  capability orf prefix-list both           [a:GROUP_2]
  default-originate                         []
  maximum-prefix 2500 75 warning-only       []
  route-policy POLICY_1 in                  [a:GROUP_2 a:GROUP_3]
  remove-private-AS                         [a:GROUP_2 a:GROUP_3]
  send-community-ebgp                       [a:GROUP_2]
  send-extended-community-ebgp              [a:GROUP_2]
```

次に、 **users** キーワードを指定した **show bgp af-group** コマンドの出力例を示します。

```
RP/0/RSP0/cpu 0: router# show bgp af-group GROUP_2 users

IPv4 Unicast: a:GROUP_1
```

次に、 **inheritance** キーワードを指定した **show bgp af-group** コマンドの出力例を示します。これは、指定されたアドレス ファミリ グループ GROUP_1 は、GROUP_2 アドレス ファミリ グループを直接使用しており、さらに GROUP_2 で GROUP_3 アドレス ファミリ グループを使用していることを示しています。

```
RP/0/RSP0/cpu 0: router# show bgp af-group GROUP_1 inheritance

IPv4 Unicast: a:GROUP_2 a:GROUP_3
```

show bgp session-group

セッショングループを表示するには、**show bgp session-group** コマンドを使用します。

- このセッショングループで使用されるセッショングループから継承したすべての設定など、セッショングループの有効な設定を表示するには、**configuration** キーワードを使用します。
- このセッショングループで設定を継承できるセッショングループを表示するには、**inheritance** キーワードを使用します。
- このセッショングループから設定を継承するセッショングループ、ネイバーグループ、ネイバーを表示するには、**users** キーワードを使用します。

show bgp session-group コマンドの出力は、次のセッショングループ設定に基づいています。

```
RP/0/RSP0/cpu 0: router(config)# router bgp 113
RP/0/RSP0/cpu 0: router(config-bgp)# session-group GROUP_1
RP/0/RSP0/cpu 0: router(config-bgp-sngrp)# use session-group GROUP_2
RP/0/RSP0/cpu 0: router(config-bgp-sngrp)# update-source Loopback 0
RP/0/RSP0/cpu 0: router(config-bgp-sngrp)# exit
RP/0/RSP0/cpu 0: router(config-bgp)# session-group GROUP_2
RP/0/RSP0/cpu 0: router(config-bgp-sngrp)# use session-group GROUP_3
RP/0/RSP0/cpu 0: router(config-bgp-sngrp)# ebgp-multihop 2
RP/0/RSP0/cpu 0: router(config-bgp-sngrp)# exit
```

```
RP/0/RSP0/cpu 0: router(config-bgp)# session-group GROUP_3
RP/0/RSP0/cpu 0: router(config-bgp-sngrp)# dmz-link-bandwidth
```

次に、EXEC コンフィギュレーション モードで **configuration** キーワードを指定した **show bgp session-group** コマンドの出力例を示します。

```
RP/0/RSP0/cpu 0: router# show bgp session-group GROUP_1 configuration

session-group GROUP_1
 ebgp-multihop 2          [s:GROUP_2]
 update-source Loopback0 []
 dmz-link-bandwidth       [s:GROUP_2 s:GROUP_3]
```

次に示す **inheritance** キーワードを指定した **show bgp session-group** の出力例では、GROUP_1 セッショングループが GROUP_3 セッショングループと GROUP_2 セッショングループからセッションパラメータを継承することを示しています。

```
RP/0/RSP0/cpu 0: router# show bgp session-group GROUP_1 inheritance

Session: s:GROUP_2 s:GROUP_3
```

次に示す **users** キーワードを指定した **show bgp session-group** の出力例では、GROUP_1 セッショングループと GROUP_2 セッショングループが GROUP_3 セッショングループからセッションパラメータを継承することを示しています。

```
RP/0/RSP0/cpu 0: router# show bgp session-group GROUP_3 users

Session: s:GROUP_1 s:GROUP_2
```

show bgp neighbor-group

ネイバーグループを表示するには、**show bgp neighbor-group** コマンドを使用します。

- このネイバーグループで使用されるネイバーグループから継承したすべての設定など、ネイバーグループの有効な設定を表示するには、**configuration** キーワードを使用します。
- このネイバーファミリグループで設定を継承できるアドレスファミリグループ、セッショングループ、およびネイバーグループを表示するには、**inheritance** キーワードを使用します。
- このネイバーグループから設定を継承するネイバーおよびネイバーグループを表示するには、**users** キーワードを使用します。

この例は、次のグループ設定に基づいています。

```
RP/0/RSP0/cpu 0: router(config)# router bgp 140
RP/0/RSP0/cpu 0: router(config-bgp)# af-group GROUP_3 address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-afgrp)# remove-private-as
RP/0/RSP0/cpu 0: router(config-bgp-afgrp)# soft-reconfiguration inbound
RP/0/RSP0/cpu 0: router(config-bgp-afgrp)# exit
RP/0/RSP0/cpu 0: router(config-bgp)# af-group GROUP_2 address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-afgrp)# use af-group GROUP_3
```

```
RP/0/RSP0/cpu 0: router(config-bgp-afgrp)# send-community-ebgp
RP/0/RSP0/cpu 0: router(config-bgp-afgrp)# send-extended-community-ebgp
RP/0/RSP0/cpu 0: router(config-bgp-afgrp)# capability orf prefix both
RP/0/RSP0/cpu 0: router(config-bgp-afgrp)# exit
RP/0/RSP0/cpu 0: router(config-bgp)# session-group GROUP_3
RP/0/RSP0/cpu 0: router(config-bgp-sngrp)# timers 30 90
RP/0/RSP0/cpu 0: router(config-bgp-sngrp)# exit
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor-group GROUP_1
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# remote-as 1982
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# use neighbor-group GROUP_2
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp-af)# exit
RP/0/RSP0/cpu 0: router(config-nbrgrp)# exit
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor-group GROUP_2
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# use session-group GROUP_3
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# address-family ipv4 unicast
RP/0/RSP0/cpu 0: routerconfig-bgp-nbrgrp-af)# use af-group GROUP_2
RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp-af)# weight 100
```

次に、 **configuration** キーワードを指定した **show bgp neighbor-group** コマンドの出力例を示します。構成セット ソースが各コマンドの右側に表示されます。上記の出力で、リモート自律システムは、ネイバー グループ GROUP_1 に直接設定されており、送信コミュニティ設定はネイバー グループ GROUP_2 から継承されています。ネイバー グループ GROUP_2 では、アドレス ファミリ グループ GROUP_3 から設定を継承しています。

```
RP/0/RSP0/cpu 0: router# show bgp neighbor-group GROUP_1 configuration

   neighbor-group GROUP_1
    remote-as 1982                   []
    timers 30 90                     [n:GROUP_2 s:GROUP_3]
    address-family ipv4 unicast      []
     capability orf prefix-list both [n:GROUP_2 a:GROUP_2]
     remove-private-AS               [n:GROUP_2 a:GROUP_2 a:GROUP_3]
     send-community-ebgp             [n:GROUP_2 a:GROUP_2]
     send-extended-community-ebgp    [n:GROUP_2 a:GROUP_2]
     soft-reconfiguration inbound    [n:GROUP_2 a:GROUP_2 a:GROUP_3]
     weight 100                      [n:GROUP_2]
```

次の例は、**inheritance** キーワードを指定した場合の **show bgp neighbor-group** コマンドの出力を示しています。この出力は、指定したネイバー グループ GROUP_1 が、ネイバー グループ GROUP_2 からセッション（アドレス ファミリ独立）設定パラメータを継承していることを示しています。ネイバー グループ GROUP_2 はセッション グループ GROUP_3 からセッション パラメータを継承しました。また、GROUP_1 ネイバー グループは GROUP_2 ネイバー グループから IPv4 ユニキャスト設定パラメータを継承し、さらに GROUP_2 ネイバー グループが GROUP_2 アドレス ファミリ グループから継承し、GROUP_2 アドレス ファミリ グループ自体は GROUP_3 アドレス ファミリ グループから継承していることも示しています。

```
RP/0/RSP0/cpu 0: router# show bgp neighbor-group GROUP_1 inheritance

    Session:      n:GROUP-2 s:GROUP_3
    IPv4 Unicast: n:GROUP_2 a:GROUP_2 a:GROUP_3
```

次に、 **users** キーワードを指定した **show bgp neighbor-group** コマンドの出力例を示します。この出力は、GROUP_1 ネイバー グループが GROUP_2 ネイバー グループからセッション（アドレス ファミリ独立）設定パラメータを継承していることを示しています。GROUP_1 ネイバー グループは GROUP_2 ネイバー グループから IPv4 ユニキャスト設定パラメータも継承しています。

```
RP/0/RSP0/cpu 0: router# show bgp neighbor-group GROUP_2 users

Session:      n:GROUP_1
IPv4 Unicast: n:GROUP_1
```

デフォルトのアドレス ファミリはない

BGP では、デフォルト アドレス ファミリの概念に対応していません。アドレス ファミリを BGP でアクティブにするには、このアドレス ファミリを BGP ルータ コンフィギュレーションで明示的に設定する必要があります。同様に、このアドレス ファミリの BGP セッションをアクティブにするには、ネイバーでそのアドレス ファミリを明示的に設定する必要があります。ネイバーを設定するために、BGP ルータ コンフィギュレーション レベルでアドレス ファミリを設定する必要はありません。ただし、ネイバーにアドレス ファミリを設定するには、BGP ルータ コンフィギュレーション レベルでそのアドレス ファミリを設定する必要があります。

ネイバーアドレスファミリの組み合わせ

デフォルトの VRF の場合、Cisco IOS XR ソフトウェア リリース 6.2.x 以降では、IPv4 ユニキャスト アドレス ファミリと IPv4 ラベル付きユニキャスト アドレス ファミリの両方が同じネイバーでサポートされています。

デフォルト以外の VRF では、IPv4 ユニキャスト アドレス ファミリと IPv4 ラベル付きユニキャスト アドレス ファミリはどちらも同じネイバーでサポートされません。ただし、次のエラーが発生した場合は、 Cisco ASR 9000 シリーズルータでこの設定が受け入れられます。

```
bgp[1051]: %ROUTING-BGP-4-INCOMPATIBLE_AFI : IPv4 Unicast and IPv4 Labeled-unicast Address
 families together are not supported under the same neighbor.
```

1 つの BGP セッションに IPv4 ユニキャストと IPv4 ラベル付きユニキャスト AFI/SAF の両方がある場合、ルーティング動作は非決定的になります。したがって、プレフィックスが正しくアドバタイズされない場合があります。プレフィックスが正しくアドバタイズされないと、到達可能性の問題が発生します。このような到達可能性の問題を回避するには、IPv4 ユニキャストまたは IPv4 ラベル付きユニキャスト アドレス ファミリのいずれかを介してプレフィックスをアドバタイズするルート ポリシーを明示的に設定する必要があります。

ルーティング ポリシーの強制適用

外部 BGP（eBGP）ネイバーには、インバウンドおよびアウトバウンドのポリシーを設定する必要があります。ポリシーが設定されていない場合、そのネイバーからのルートは受け入れられず、いずれのルートもそのネイバーにアドバタイズされません。この付加的なセキュリティ

手段によって、設定を誤って省略した場合に、ルートが偶然受け入れられたり、アドバタイズされたりすることが決してなくなります。

（注） この制約は eBGP ネイバー（このルータと異なる自律システムに属すネイバー）だけに適用されます。内部 BGP（iBGP）ネイバー（同じ自律システム内のネイバー）の場合は、ポリシーがなければ、すべてのルートが受け入れられるか、アドバタイズされます。

次の例では、すべてのルートが変更なしで許可およびアドバタイズされる場合に、eBGP ネイバーに対して単純な pass-all ポリシーが設定されています。

```
RP/0/RSP0/cpu 0: router(config)# route-policy pass-all
RP/0/RSP0/cpu 0: router(config-rpl)# pass
RP/0/RSP0/cpu 0: router(config-rpl)# end-policy
RP/0/RSP0/cpu 0: router(config)# commit
```

ネイバーに pass-all ポリシーを適用するには、ネイバー アドレス ファミリ コンフィギュレーション モードで **route-policy (BGP)** コマンドを使用します。次の例は、ネイバー 192.168.40.42 からの受信と、このネイバーに対するすべての IPv4 ユニキャスト ルートのアドバタイズを、すべての IPv4 ユニキャスト ルートに許可する方法を示します。

```
RP/0/RSP0/cpu 0: router(config)# router bgp 1
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 192.168.40.24
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# remote-as 21
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# route-policy pass-all in
RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# route-policy pass-all out
RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# commit
```

すべてのアクティブ アドレス ファミリに対するインバウンドとアウトバウンドの両方のポリシーを持っていない eBGP ネイバーを表示するには、**show bgp summary** コマンドを使用します。次の例の出力では、該当する eBGP ネイバーが感嘆符（!）によって示されています。

```
RP/0/RSP0/cpu 0: router# show bgp all all summary

Address Family: IPv4 Unicast
============================

BGP router identifier 10.0.0.1, local AS number 1
BGP generic scan interval 60 secs
BGP main routing table version 41
BGP scan interval 60 secs
BGP is operating in STANDALONE mode.

Process         RecvTblVer    bRIB/RIB  SendTblVer
Speaker                41          41          41

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  St/PfxRcd
10.0.101.1        0     1     919     925       41    0    0 15:15:08       10
10.0.101.2        0     2       0       0        0    0    0 00:00:00 Idle


Address Family: IPv4 Multicast
```

```
==============================

BGP router identifier 10.0.0.1, local AS number 1
BGP generic scan interval 60 secs
BGP main routing table version 1
BGP scan interval 60 secs
BGP is operating in STANDALONE mode.

Process         RecvTblVer    bRIB/RIB  SendTblVer
Speaker                 1           1           1

Some configured eBGP neighbors do not have both inbound and
outbound policies configured for IPv4 Multicast address family.
These neighbors will default to sending and/or receiving no
routes and are marked with '!' in the output below. Use the
'show bgp neighbor <nbr_address>' command for details.

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  St/PfxRcd
10.0.101.2        0     2       0       0        0    0    0 00:00:00 Idle!


Address Family: IPv6 Unicast
============================

BGP router identifier 10.0.0.1, local AS number 1
BGP generic scan interval 60 secs
BGP main routing table version 2
BGP scan interval 60 secs
BGP is operating in STANDALONE mode.

Process         RecvTblVer    bRIB/RIB  SendTblVer
Speaker                 2           2           2

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  St/PfxRcd
2222::2           0     2     920     918        2    0    0 15:15:11        1
2222::4           0     3       0       0        0    0    0 00:00:00 Idle


Address Family: IPv6 Multicast
==============================

BGP router identifier 10.0.0.1, local AS number 1
BGP generic scan interval 60 secs
BGP main routing table version 1
BGP scan interval 60 secs
BGP is operating in STANDALONE mode.

Process         RecvTblVer    bRIB/RIB  SendTblVer
Speaker                 1           1           1

Some configured eBGP neighbors do not have both inbound and
outbound policies configured for IPv6 Multicast address family.
These neighbors will default to sending and/or receiving no
routes and are marked with '!' in the output below. Use the
'show bgp neighbor <nbr_address>' command for details.

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  St/PfxRcd
2222::2           0     2     920     918        0    0    0 15:15:11        0
2222::4           0     3       0       0        0    0    0 00:00:00 Idle!
```

テーブル ポリシー

BGP のテーブル ポリシー機能を使用すると、ルートのトラフィック索引の値をグローバル ルーティング テーブルにインストールされるときに設定できます。この機能を有効にするには **table-policy** コマンドを使用します。また BGP ポリシーアカウンティング機能もサポートされています。

BGP ポリシー アカウンティングでは、BGP ルートに設定されたトラフィック索引を使用してさまざまなカウンタをトラックします。テーブルポリシーの使用法の詳細については、『*Routing Configuration Guide for Cisco ASR 9000 Series Routers*』の「 *Implementing Routing Policy on Cisco ASR 9000 Series Router* 」のモジュールを参照してください。BGP ポリシーアカウンティングの詳細については、『*IP Addresses and Services Command Reference for Cisco ASR 9000 Series Routers*』の「 *Cisco Express Forwarding Commands on Cisco ASR 9000 Series Router* 」のモジュールを参照してください。

テーブル ポリシーを使用すると、一致基準に基づいて RIB からのルートをドロップすることもできます。この機能は特定のアプリケーションにおいて有用ですが、BGP がグローバル ルーティングおよびフォワーディング テーブルにインストールしていないネイバーに対して、BGP がルートをアドバタイズするところに、簡単にルーティング「ブラック ホール」が作成されてしまうため、注意して使用する必要があります。

アップデート グループ

BGP アップデート グループ機能には、アウトバウンド ポリシーを共有し、アップデート メッセージを共有できるネイバーのアップデート グループをダイナミックに計算し、最適化する新しいアルゴリズムが含まれています。BGP アップデート グループ機能では、アップデート グループ レプリケーションはピア グループ コンフィギュレーションから分離されるため、ネイバー コンフィギュレーションのコンバージェンス時間が短縮され、柔軟性が高まります。

この機能を使用するには、次の概念を理解しておく必要があります。

関連トピック

BGP アップデートの生成およびアップデート グループ （32 ページ）
BGP アップデート グループ （32 ページ）

BGP アップデートの生成およびアップデート グループ

BGP アップデート グループ機能により、BGP アップデートの生成がネイバー設定から分離されます。BGP アップデート グループ機能により、アウトバウンド ルーティング ポリシーに基づいて BGP アップデート グループ メンバーシップを動的に計算するアルゴリズムが導入されます。この機能に対してネットワーク オペレータによる設定は不要です。アップデート グループをベースとするメッセージ生成は自動的かつ個別に行われます。

BGP アップデート グループ

設定の変更があった場合、ルータでは、アップデート グループ メンバーシップを自動的に再計算し、変更を適用します。

BGP アップデート グループの生成を最適化するには、ネットワーク オペレータは、類似するアウトバウンド ポリシーを持つネイバーのアウトバウンド ルーティング ポリシーを同じものにしておくことを推奨します。この機能には、BGP アップデート グループを監視するためのコマンドが含まれます。

BGP コスト コミュニティ

BGP コスト コミュニティは非過渡的な拡張コミュニティ属性で、内部 BGP（iBGP）およびコンフェデレーション ピアへ渡されますが、外部 BGP（eBGP）ピアへは渡されません。コスト コミュニティ機能により、コスト値を特定のルートに割り当てることで、ローカル ルート プリファレンスをカスタマイズし、最適パス選択プロセスに反映させることができます。拡張コミュニティ形式は、最適パスアルゴリズムの異なるポイントでの最適パスの決定に影響する標準の挿入ポイント（POI）を定義します。

コスト コミュニティ属性は、ルート ポリシーで **set extcommunity cost** コマンドを設定することにより、内部ルートに適用されます。**set extcommunity cost** コマンドについては、『*Cisco ASR 9000 Series Aggregation Services Router Routing Command Reference*』の「*Routing Policy Language Commands on Cisco ASR 9000* シリーズ ルータ 」のモジュールを参照してください。cost community set 句は、コスト コミュニティ ID 番号（0 ～ 255）およびコスト コミュニティ番号（0 ～ 4294967295）を使用して設定されます。コストコミュニティ番号によってパスの優先度が判断されます。最も低いコストコミュニティ番号を持つパスが優先されます。コストコミュニティ番号を具体的に設定していないパスには、デフォルトのコストコミュニティ番号である 2147483647（0 ～ 4294967295 の中央値）が割り当てられ、最適パス選択プロセスにより評価されます。2つのパスが同じコストコミュニティ番号を使用して設定されている場合、パス選択プロセスでは最も低いコスト コミュニティ ID のパスが優先されます。このコスト拡張コミュニティ リンク属性は、拡張コミュニティ交換がイネーブルなとき、iBGP ピアに伝播します。

次のコマンドには **route-policy** キーワードが含まれています。このキーワードは、cost community set 句で設定されるルート ポリシーを適用するために使用できます。

- **aggregate-address**
- **redistribute**
- **network**

BGP コスト コミュニティはどのように最適パス選択プロセスに影響するか

BGP最適パス選択プロセスは、挿入ポイント（POI）においてコストコミュニティ属性の影響を受けます。デフォルトでは、POI は、内部ゲートウェイ プロトコル（IGP）メトリック比較に従います。同一の宛先に向かう複数のパスを受信したとき、BGP では最適パス選択プロセスを使用して、いずれのパスが最適パスであるのかを決定します。最良パスは BGP により自動的に決定され、ルーティングテーブルにインストールされます。複数の等コストパスが使用可能な場合、POI で個別のパスにプリファレンスを割り当てることができます。ローカルの最適パス選択で POI が有効でない場合は、コスト コミュニティ属性は暗黙的に無視されます。

コスト コミュニティは、最初に POI で、次にコミュニティ ID でソートされます。コスト コミュニティ属性を使用して、同一の POI に対し複数のパスを設定できます。最も低いコスト コミュニティ ID を持つパスが最優先で検討されます。つまり、特定の POI に対するすべてのコスト コミュニティ パスは、最も低いコスト コミュニティを持つパスから検討されていきます。コスト コミュニティ コストを持たないパス（評価中の POI およびコミュニティ ID）には、デフォルトのコミュニティ コスト値（2147483647）が割り当てられます。コスト コミュニティ値が等しい場合、コスト コミュニティ比較は、その POI で次に低いコミュニティ ID に進みます。

最も低いコスト コミュニティを持つパスを選択するには、両方のパスのコスト コミュニティを同時に探索します。これを行うには、コストコミュニティのチェーンにポインタを 2 つ設定し、各パスに 1 つずつ割り当て、POI に対する探索の各ステップでコミュニティ ID の順に両方のポインタを次のコストコミュニティに進め、最良のパスが選ばれたとき、または比較して順位が付かなくなったときに終了します。探索の各ステップで、次のチェックが実行されます。

```
If neither pointer refers to a cost community,
     Declare a tie;

  Elseif a cost community is found for one path but not for the other,
     Choose the path with cost community as best path;
  Elseif the Community ID from one path is less than the other,
     Choose the path with the lesser Community ID as best path;
  Elseif the Cost from one path is less than the other,
     Choose the path with the lesser Cost as best path;
  Else Continue.
```

（注） パスにコストコミュニティ属性が設定されていない場合、最適パス選択プロセスはそのパスにデフォルトのコスト値（最大値 4294967295 の半分である 2147483647）が割り当てられているものと見なします。

POI でコストコミュニティ属性を適用することで、ローカルの自律システムまたはコンフェデレーションにおける任意の部分にあるピアを起点とするか、このピアで学習したパスに、値を割り当てることができるようになります。コストコミュニティは、最適パス選択プロセス中の「タイブレーカー」として使用できます。同一の自律システムまたはコンフェデレーションにおける別個の等コストパスに対し、コストコミュニティのインスタンスを複数設定できます。たとえば、複数の等コスト出口ポイントがあるネットワークにおいて、特定の出口パスに、より低いコスト コミュニティ値を適用すれば、その出口パスは BGP 最適パス選択プロセスにより優先されることになります。マルチエグジット IGP ネットワークにおけるルート プリファレンスの反映 （36 ページ）に記載されているシナリオを参照してください。

（注） BGP では、コストコミュニティの比較がデフォルトで有効になっています。比較を無効にするには、**bgp bestpath cost-community ignore** コマンドを使用します。

BGP 最適パス選択処理については、BGP 最適パス アルゴリズム （41 ページ）を参照してください。

集約ルートおよびマルチパスに対するコスト コミュニティのサポート

BGP コスト コミュニティ機能では、集約ルートおよびマルチパスをサポートしています。コスト コミュニティ属性は、いずれかのルートのタイプに適用できます。コスト コミュニティ属性は、コスト コミュニティ属性を伝送するコンポーネント ルートから集約ルートまたはマルチパス ルートに渡されます。一意の ID のみが渡され、いずれの個別コンポーネント ルートについても、最大のコストのみが、ID ごとの集約に対して適用されます。複数のコンポーネント ルートに同一の ID が含まれる場合は、設定されている最大のコストがルートに適用されます。たとえば、次の 2 つのコンポーネント ルートは、インバウンド ルート ポリシーを使用してコスト コミュニティ属性が設定されています。

- 10.0.0.1
  - POI=IGP
  - コスト コミュニティ ID=1
  - コスト番号=100
- 192.168.0.1
  - POI=IGP
  - コスト コミュニティ ID=1
  - コスト番号=200

これらのコンポーネント ルートを集約するか、マルチパスとして設定した場合は、コスト値 200 が最大のコストであるため、この値がアドバタイズされます。

1 つ以上のコンポーネント ルートがコスト コミュニティ属性を伝送しない場合、またはこれらのコンポーネント ルートに異なる ID が設定されている場合は、デフォルト値（2147483647）が、集約ルートまたはマルチパス ルートに対してアドバタイズされます。たとえば、次の 3 つのコンポーネント ルートは、インバウンド ルート ポリシーを使用してコスト コミュニティ属性が設定されています。ただし、これらのコンポーネント ルートには 2 つの異なる ID が設定されています。

- 10.0.0.1
  - POI=IGP
  - コスト コミュニティ ID=1
  - コスト番号=100
- 172.16.0.1
  - POI=IGP
  - コスト コミュニティ ID=2

- コスト番号=100

- 192.168.0.1
  - POI=IGP
  - コスト コミュニティ ID=1
  - コスト番号=200

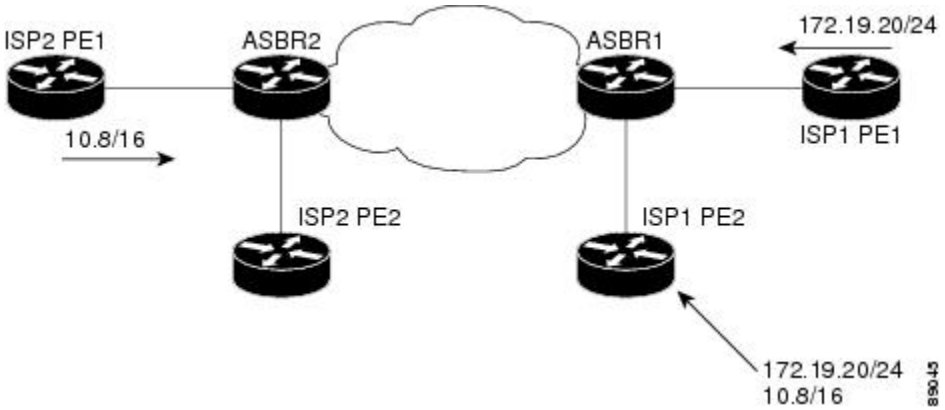
アドバタイズされる単一のパスには、次のような集約コスト コミュニティなどがあります。

{POI=IGP, ID=1, Cost=2147483647} {POI-IGP, ID=2, Cost=2147483647}

マルチエグジット IGP ネットワークにおけるルート プリファレンスの反映

次の図は、エッジに 2 つの自律システム境界ルータ（ASBR）がある IGP ネットワークを示します。各 ASBR は、ネットワーク 10.8/16 に対して等コスト パスを持ちます。

図 *1:* マルチエグジット ポイントの *IGP* ネットワーク



BGP では、両パスは等しいと見なされます。マルチパス ロード シェアリングが設定されている場合は、ルーティングテーブルへの両方のパスが組み込まれ、トラフィックの負荷を分散するために使用されます。マルチパス ロードバランシングが設定されていない場合、BGP により最初に最適パスであると学習されたパスが選択され、ルーティングテーブルに組み込まれます。この動作は、一部の条件下では望ましくない場合があります。たとえば、パスは最初に ISP1 PE2 から学習されますが、ISP1 PE2 と ASBR1 間のリンクは低速リンクです。

コスト コミュニティ属性のコンフィギュレーションを使用して ASBR2 が学習したパスにより低いコスト コミュニティ値を適用することで、BGP 最適パス選択プロセスに影響を与えることができます。たとえば、次のコンフィギュレーションは ASBR2 に適用されています。

```
RP/0/RSP0/cpu 0: router(config)# route-policy ISP2_PE1
RP/0/RSP0/cpu 0: router(config-rpl)# set extcommunity cost (1:1)
```

上記のルート ポリシーでは、コスト コミュニティ番号値の 1 がルート 10.8.0.0 に適用されます。デフォルトでは、ASBR1 で学習したパスにはコスト コミュニティ番号 2147483647 が割り当てられます。ASBR2 から学習したパスのコスト コミュニティ番号の方が小さいため、このパスが優先されます。

バックドア リンクを持つ EIGRP MPLS VPN PE-CE に対する BGP コスト コミュニティ サポート

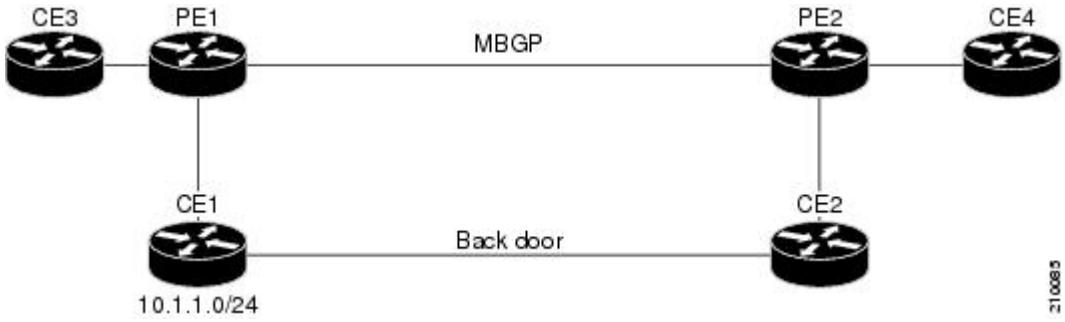
バックドア リンクの方が先に学習された場合、BGP では、EIGRP MPLS VPN トポロジのバックドア リンクを優先します。（バックドア リンクまたはルートは、リモート サイトとメイン サイトの間の VPN の外部に設定される接続で、たとえば、リモート サイトを企業ネットワークへ接続する WAN 専用線などがあります）。

BGP コスト コミュニティの「準最適パス」挿入ポイント（POI）機能は、VPN およびバックドア リンクが混在する EIGRP VPN ネットワーク トポロジをサポートしています。この POI は BGP に再配布される EIGRP ルートに自動的に適用されます。「準最適パス」POI は、EIGRP のルート タイプおよびメトリックを伝送します。この POI は、BGP がその他のあらゆる比較ステップの前にこの POI を検討するように影響を与えておくことで、最適パス計算プロセスに作用します。設定は必要ありません。PE、CE、またはバックドアルータに Cisco IOS XR ソフトウェア がインストールされている場合、この機能は、EIGRP VPN サイトについて自動的に有効にされます。

EIGRP MPLS VPN の設定については、 *MPLS Configuration Guide for Cisco ASR 9000 Series RoutersMPLS Configuration Guide for Cisco NCS 560 Series Routers*を参照してください。

図 *2:* コスト コミュニティを使用してバックドア リンクをサポートする方法を示すネットワーク

この図は、コスト コミュニティを使用して、ネットワークのバックドア リンクをサポートする方法を示します。



次に、PE1 におけるイベント シーケンスを示します。

1. PE1 では、仮想ルーティングおよび転送（VRF）インスタンスを実行している CE1 から EIGRP を介して IPv4 プレフィックス 10.1.1.0/24 を学習します。EIGRP は最適パスを選択して、RIB に組み込みます。コスト拡張コミュニティのエンコードと、RIB に対するこの情報の追加も行います。

2. ルートは BGP に再配布されます（IGP-to-BGP 再配布が設定されていることを想定）。BGP では、再配布プロセスを介して、ルートからコスト拡張コミュニティも受け取ります。
3. 新しく再配布されたプレフィックスの最適パスを BGP が判別すると、そのパスは PE ピア（PE2）にアドバタイズされます。
4. PE2 では、BGP VPNv4 プレフィックス route_distinguisher:10.1.1.0/24 をコスト コミュニティとともに受信します。CE2 では、おそらくは、EIGRP を介して PE2 に同じプレフィックスをアドバタイズします（CE1 と CE2 の間にバックドア リンクがあるため）。通常、PE2 BGP では、再配布プロセスによって、コスト コミュニティ値とともに CE ルートをすでに学習しています。
5. PE2 には、BGP のパスが 2 つあります。マルチパス BGP を介するコスト コミュニティ cost1 のパス（PE1）と、EIGRP ネイバーを介するコスト コミュニティ cost2 の別のパス（CE2）です。
6. PE2 では、拡張された BGP 最適パス計算を実行します。
7. PE2 は、適切なコスト コミュニティ値を渡して、RIB に最適パスを組み込みます。
8. PE2 RIB には、10.1.1.0/24 に対するパスが 2 つあります。EIGRP によって追加されたコスト コミュニティ cost2 のパスと、BGP によって追加されたコスト コミュニティ cost1 の別のパスです。両方のルートパスがコストコミュニティを持つため、RIB では、まずコストを比較します。BGP パスのコストコミュニティの方が低いため、これが選択されて、RIB にダウンロードされます。
9. PE2 RIB では、VRF を介して BGP パスを EIGRP に再配布します。パスが 2 個あるため EIGRP は拡散更新アルゴリズム（DUAL）を実行し、BGP 再配布パスを選択します。
10. PE2 EIGRP は、このパスを CE2 にアドバタイズします。これにより、このパスは、MPLS ネットワークを介してトラフィックを送信するために、このプレフィックス対して使用されるネクスト ホップになります。

ルーティング情報ベースへのルートの追加

最適パス計算の後で、ソーシングされていないパスが最適パスになった場合、BGP では、このルートをルーティング情報ベース（RIB）に追加し、他の IGP 拡張コミュニティと一緒にコスト コミュニティを渡します。

パスを含むルートがプロトコルによって RIB に追加される場合、RIB では、現在の最適パスを調べてルートを確認し、追加されたパスを調べてコスト拡張コミュニティを確認します。コスト拡張コミュニティが見つかった場合、RIB では、コストコミュニティの設定を比較します。比較して順位が付く場合は、適切な最適パスが選択されます。比較して順位が付かない場合、RIB では、最適パス アルゴリズムの残りの手順に進みます。現在の最適パスと追加されたパスのいずれにもコスト コミュニティがない場合、RIB では、最適パス アルゴリズムの残りの手順を続行します。BGP 最適パス アルゴリズムについては、BGP 最適パス アルゴリズム （41 ページ）を参照してください。

BGP DMZ 総帯域幅

BGP は、内部 BGP（iBGP）ピアへのルートをアドバタイズするときに、外部 BGP（eBGP）マルチパスの *dmz-link bandwidth* 値の集約をサポートしています。

帯域幅を集約するための明示的なコマンドはありません。帯域幅は、次の条件を満たしている場合に集約されます。

- ネットワークにはマルチパスがあり、すべてのマルチパスにはリンク帯域幅の値があります。
- next-hop-self に設定されたネクストホップ属性。指定されたネイバーにアドバタイズされるすべてのルートのネクストホップ属性をローカルルータのアドレスに設定します。
- dmz-link bandwidth の値を変更する可能性があるアウトバウンドポリシーは設定されていません。

（注）

- マルチパス（eBGP または iBGP）のいずれかの *dmz-link bandwidth* 値が不明な場合、ベストパスを含むすべてのマルチパスの *dmz-link* 値はルーティング情報ベース（RIB）にダウンロードされません。
- iBGP マルチパスの *dmz-link bandwidth* 値は、集約時に考慮されません。
- 集約値でアドバタイズされるルートは、ベストパスまたは追加パスにすることができます。
- 追加パスは、ネクストホップが維持されるため、DMZ リンクの帯域幅集約には適していません。追加パスの next-hop-self の設定はサポートされていません。
- VPNv4 および VPNv6 afi の場合、f *dmz link-bandwidth* 値はアウトバウンドルート ポリシーを使用し、ルートテーブルを指定するか、または **additive** キーワードを使用して設定されます。また、これによってピアの受信端でルートがインポートされなくなります。

```
extcommunity-set bandwidth dmz_ext
   1:8000
 end-set
 !
 route-policy dmz_rp_vpn
   set extcommunity bandwidth dmz_ext additive     <<< 'additive' keyword.
   pass
 end-policy
```

例

ネットワーク内の内部ルータに接続されたルタ 1 とルータ 2 の 2 台のルータについて検討してみましょう。ルータ 1 は、2 つの異なる ISP から 50 と 20 の帯域幅をアドバタイズします。ルータ 2 は、2 つの異なる ISP から 60 と 30 の帯域幅をアドバタイズします。ベストパスアルゴリズムを使用すると、内部ルータに対してルータ 1 は 50 の帯域幅をアドバタイズし、ルータ 2 は 60 の帯域幅をアドバタイズします。これによ

り、トラフィックフローが削減されます。ただし、帯域幅を集約することで、ルータ 1 は 70（50＋20）の帯域幅をアドバタイズし、ルータ 2 は 90（60＋30）の帯域幅をアドバタイズします。これにより、トラフィックフローが増加します。

BGP DMZ 総帯域幅の設定：例

次に、ボーダー ゲートウェイ プロトコルの緩衝地帯（BGP DMZ）リンク帯域幅の設定例を示します。トポロジ、R1---(iBGP)---R2---(iBGP)---R3 について検討してみましょう。

1. R1 では次のようになります。

```
bgp: prefix p/n has:
path 1(bestpath)       with LB value 100
path 2(ebgp multipath) with LB value 30
path 3(ebgp multipath) with LB value 50
```

   ベストパスが R2 にアドバタイズされると、集約された DMZ リンクの帯域幅の値 180 を送信します。パス 1、2、および 3 の集約値。

2. R2 では次のようになります。

```
bgp: prefix p/n has:
path 1(bestpath)       with LB value 60
path 2(ebgp multipath) with LB value 200
path 3(ebgp multipath) with LB value 50
```

   ベストパスが R3 にアドバタイズされると、集約された DMZ リンクの帯域幅の値 310 を送信します。パス 1、2、および 3 の集約値。

3. R3 では次のようになります。

```
bgp: prefix p/n has:
path 1(bestpath)       with LB 180 {learned from R1}
path 2(ibgp multipath) with LB 310 {learned from R2}
```

ポリシーベースのリンク帯域幅の設定：例

次に、ポリシーベースの DMZ リンク帯域幅を設定する例を示します。リンク帯域幅の拡張コミュニティは、ネイバーインまたはネイバーアウトのポリシー接続点で、パスごとに設定できます。*dmz-link-bandwidth* ノブは、eBGP ネイバー コンフィギュレーション モードで設定されます。この特定のネイバーから受信したすべてのパスは、iBGP ピアに送信されるときに、リンク帯域幅拡張コミュニティでマークされます。

1. インバウンドまたはアウトバウンドのルートポリシーを設定します。

```
extcommunity-set bandwidth dmz_ext
  1:1290400000
end-set
!
route-policy dmz_rp
  set extcommunity bandwidth dmz_ext
  pass
end-policy
!

 neighbor 10.0.101.1
  remote-as 1001
  address-family ipv4 unicast
```

```
 route-policy dmz_rp in          <<< Inbound route-policy.
 route-policy pass out
!
```

2. BGP ネイバーで *dmz-link-bandwidth* を設定します。

```
neighbor 10.0.101.2
  remote-as 1001
  dmz-link-bandwidth              <<< Under neighbor.
  address-family ipv4 unicast
   route-policy pass in
   route-policy pass out
  !
```

ポリシーベースの拡張コミュニティセットの詳細については、『*Cisco ASR 9000* シリーズ アグリゲーション サービス ルータ ルーティング設定ガイド』の「ルーティング ポリシーの実装」の章を参照してください。

BGP 用 64-ECMP のサポート

IOS XR では、BGP に最大 64 の等コストマルチパス（ECMP）ネクストホップを設定できます。過負荷状態のルータが 64 を超える LSP のトラフィックをロードバランシングできる場合、ネットワークに 64-ECMP が必要です。

BGP 最適パス アルゴリズム

BGP ルータは、通常は同じ宛先に対する複数のパスを受信します。BGP の最適パス アルゴリズムは、IP ルーティング テーブルに格納し、トラフィックの転送に使用する最適なパスを決めるものです。この項では、インターネット技術特別調査委員会（IETF）のネットワーク ワーキング グループによる draft-ietf-idr-bgp4-24.txt 資料の 9.1 項で指定されている BGP 最適パス アルゴリズムの Cisco IOS XR ソフトウェア実装について説明します。

BGP 最適パス アルゴリズムは、次の 3 つのパートに分かれて実行されます。

- パート 1：2 つのパスを比較して、いずれが優れているのかを判別します。
- パート 2：すべてのパスを順に処理し、全体として最適なパスを選択するためにパスを比較する順序を決定します。
- パート 3：新しい最適パスを使用するに足るだけの差が新旧の最適パスにあるかどうかを判別します。

（注） 比較演算が推移的ではないため、パート 2 で決定された比較の順序は重要です。つまり、3 つのパス、A、B、C がある場合、A と B を比較したときに A の方が優れていて、B と C と比較したときに B の方が優れている場合、A と C を比較したときに必ずしも A が優れているとは限りません。この非推移性は、Multi Exit Discriminator（MED）が、すべてのパス間ではなく、同じネイバー自律システム（AS）からのパス間のみで比較されるために生じます。

パスのペアの比較

2 つのパスを比較して、優れたパスを判別するには、次の手順を実行します。

1. いずれかのパスが無効な場合（可能な最大MED値を持つパス、到達不能なネクストホップを持つパスなど）、もう一方のパスが選択されます（そのパスが有効な場合）。
2. パスの準最適パス コスト コミュニティが等しくない場合は、準最適パス コスト コミュニティの低いパスが最適パスとして選択されます。
3. パスの重みが等しくない場合は、重みが最大のパスが選択されます。

（注） 重みは完全にルータにローカルであり、weight コマンドまたはルーティング ポリシーを使用して設定できます。

4. パスのローカル プリファレンスが等しくない場合は、ローカル プリファレンスが高い方のパスが選択されます。

（注） パスとともにローカル プリファレンス属性を受信したか、ルーティング ポリシーによって設定された場合は、その値が、この比較で使用されます。それ以外の場合は、デフォルト ローカル プリファレンス値の 100 が使用されます。デフォルト値は、bgp default local-preference コマンドを使用して変更できます。

5. パスの 1 つが再配布されたパス、つまり **redistribute** コマンドまたは **network** コマンドによるパスの場合は、そのパスが選択されます。それ以外の場合、パスの 1 つがローカルで作成された集約パスのとき、つまり aggregate-address コマンドによるパスのときは、そのパスが選択されます。

（注） ステップ 1 ～ ステップ 4 では、RFC 1268 の「Path Selection with BGP」を実装します。

6. パス間で AS パスの長さが異なる場合は、AS パスの短い方のパスが選択されます。このステップは、bgp bestpath as-path ignore コマンドが設定されている場合は省略されます。

（注） AS パスの長さを計算する場合は、コンフェデレーション セグメントは無視され、AS セットは 1 としてカウントされます。

（注） eiBGP は、内部および外部の BGP マルチパス ピアを指定します。eiBGP では、内部および外部のパスを同時に使用できます。

7. パス間で起点が異なる場合は、起点の値が低い方のパスが選択されます。内部ゲートウェイ プロトコル（IGP）は EGP よりも低く、EGP は INCOMPLETE より低いと見なされます。

8. 該当する場合は、パスの MED が比較されます。等しくない場合は、MED の低いパスが選択されます。

このステップが実行されるかどうかに影響するコンフィギュレーション オプションは多数あります。一般に、MED はパスが両方のパスが同じ AS にあるネイバーから受信された場合に比較され、それ以外の場合は MED 比較はスキップされます。ただし、この動作は特定のコンフィギュレーション オプションによって変更され、考慮すべきいくつかの場合があります。

bgp bestpath med always コマンドが設定されている場合、MED 比較は、パス内のネイバー AS にかかわらず、常に実行されます。それ以外の場合、MED 比較は、次のように、比較する 2 つのパスの AS パスによって異なります。

- パスに AS パスがない場合、または AS パスが AS_SET で始まる場合、パスは内部と見なされ、MED は他の内部パスと比較されます。
- AS パスが AS_SEQUENCE で開始されている場合、ネイバー AS は、シーケンスの最初の AS 番号であり、MED は、同じネイバー AS を持つ他のパスと比較されます。
- AS パスがコンフェデレーション セグメントのみを含むか、コンフェデレーション セグメントで開始されて AS_SET が続く場合、MED は、他のいずれのパスとも比較されません。ただし、**bgp bestpath med confed** コマンドが設定されている場合を除きます。その場合、パスは内部であると見なされ、MED は他の内部パスと比較されます。
- AS パスがコンフェデレーション セグメントとそれに続く AS_SEQUENCE で開始している場合、ネイバー AS は AS_SEQUENCE の最初の AS 番号であり、MED は同じネイバー AS を持つ他のパスと比較されます。

（注） パスとともに MED 属性を受信しなかった場合、MED は 0 であると見なされます。ただし、**bgp bestpath med missing-as-worst** コマンドが設定されている場合を除きます。この場合、MED 属性が受信されていない場合、MED は最高値と見なされます。

9. パスの 1 つを外部ピアから受信し、もう 1 つを内部（またはコンフェデレーション）ピアから受信した場合は、外部ピアからのパスが選択されます。

10. パスのネクスト ホップへの IGP メトリックが異なる場合、IGP メトリックが小さい方のパスが選択されます。

11. パスの IP コスト コミュニティが等しくない場合は、IP コスト コミュニティの低いパスが最適パスとして選択されます。

12. ステップ 1 ～ステップ 10 ですべてのパス パラメータが一致している場合は、ルータ ID が比較されます。送信元属性付きでパスを受信した場合は、この属性が比較対象のルー

タ ID として使用されます。それ以外の場合は、パスの受信元ネイバーのルータ ID が使用されます。パス間でルータ ID が異なる場合は、ルータ ID の小さい方のパスが選択されます。

（注） 送信元をルータ ID として使用する場合は、2 つのパスが同じルータ ID を持つことがあります。同じピア ルータと 2 つの BGP セッションを持つこともでき、したがって、同じルータ ID を持つ 2 つのパスを受信することがあります。

13. パス間でクラスタ長が異なる場合は、クラスタ長の小さい方のパスが選択されます。クラスタ リスト属性なしでパスを受信した場合、クラスタの長さは 0 であると見なされます。
14. 最後に、IP アドレスの小さいネイバーから受信したパスが選択されます。ローカル生成されたパス（たとえば、再配布されたパス）は、ネイバー IP アドレスが 0 であると見なされます。

比較の順序

BGP 最適パス アルゴリズム実装のパート 2 では、パスの比較順序を決定します。比較順序は次のように決定されます。

1. 各グループ内のすべてのパス間で MED を比較できるように、パスがグループ分けされます。2 つのパス間で MED を比較できるかどうかは、#unique_73 と同じルールを使用して決定されます。通常、この比較の結果は、ネイバー AS ごとに 1 グループになります。**bgp bestpath med always** コマンドが設定されている場合は、パスを含む 1 グループだけがあります。
2. 各グループ内の最適パスが決定されます。最適パスは、グループ内のすべてのパスを反復処理し、その時点までの最適なパスを追跡することによって決定されます。各パスが、この時点までの最適なパスと比較され、より適していれば新しいこの時点までの最適なパスになって、グループ内の次のパスと比較されます。
3. ステップ 2 の各グループから選択した最適パスで構成される、パスのセットを形成します。このパス セットに対してステップ 2 と同様の比較を繰り返すことによって、全体としての最適パスを選択します。

最適パスの変更の抑制

実装のパート 3 では、最適パスの変更を抑制するかどうか、つまり、新しい最適パスを使用するのか、既存の最適パスの使用を続行するのかを決定します。最適パス選択アルゴリズムが任意性を持つ部分まで、新規の最適パスと一致している場合は（ルータ ID が同一であることが前提）、引き続き既存の最適パスを使用できます。既存の最適パスの使用を続行すると、ネットワークでのチャーンを回避できます。

（注） この抑制動作は、IETF ネットワーキング ワーキング グループの draft-ietf-idr-bgp4-24.txt 資料に準拠していませんが、IETF ネットワーキング ワーキング グループの draft-ietf-idr-avoid-transition-00.txt 資料に指定されています。

この抑制動作は、**bgp bestpath compare-routerid** コマンドを設定してオフにできます。このコマンドを設定すると、新しい最適パスが常に既存の最適パスよりも優先されます。

それ以外の場合は、次の手順を使用して、最適パスの変更を抑制するかどうかが決定されます。

1. 既存の最適パスが有効でなくなった場合は、変更を抑制できません。
2. 既存または新規の最適パスを内部（またはコンフェデレーション）ピアから受信したか、ローカルで生成した（再配布によるなど）場合は、変更を抑制できません。つまり、抑制は、両方のパスを外部ピアから受信した場合のみ可能です。
3. パスを同じピアから受信した場合（通常はパスのルータ ID が同一）は、変更を抑制できません。ルータ ID は、#unique_73 のルールを使用して計算されます。
4. パスの重み、ローカル プリファレンス、起点、またはネクスト ホップへの IGP メトリックが異なる場合は、変更を抑制できません。このすべての値は、#unique_73 のルールを使用して計算されます。
5. パスの AS パス長が異なり、**bgp bestpath as-path ignore** コマンドが設定されていない場合は、変更を抑制できません。この場合もやはり、AS パスの長さは、#unique_73 のルールを使用して計算されます。
6. パスの MED を比較でき、MED が異なる場合は、変更を抑制できません。MED を比較できるかどうかは、#unique_73 で説明されている MED 値の計算とまったく同じルールによって判定されます。
7. ステップ 1 ～ステップ 6 のすべてのパス パラメータに該当しない場合は、変更を抑制できます。

アドミニストレーティブ ディスタンス

アドミニストレーティブディスタンスは、ルーティング情報源の信頼性を示す評価基準です。通常は、値が大きいほど、信頼性の格付けが下がります。BGP のアドミニストレーティブディスタンスを指定する方法については、*Routing Command Reference for Cisco ASR 9000 Series Routers* の「BGP コマンド」のモジュールを参照してください。

一般的にルートは複数のプロトコルによって検出されます。アドミニストレーティブディスタンスは、複数のプロトコルから学習したルートを区別するために使用されます。最もアドミニストレーティブ ディスタンスが低いルートが IP ルーティング テーブルに組み込まれます。BGP はデフォルトで、表 1 : デフォルトの BGP アドミニストレーティブ ディスタンス （46 ページ）のアドミニストレーティブ ディスタンスを使用します。

表 *1:* デフォルトの *BGP* アドミニストレーティブ ディスタンス

ディスタンス	デフォルト値	機能
外部	20	eBGP から学習したルートに適用されます。
内部	200	iBGP から学習したルートに適用されます。
ローカル	200	ルータを起点とするルートに適用されます。

（注） ディスタンスは BGP パス選択アルゴリズムに影響しませんが、BGP で学習されたルートを IP ルーティング テーブルに組み込むかどうかを左右します。

通常、eBGP を介して学習されたルートは、ディスタンス（20）を理由として IP ルーティング テーブルに組み込まれます。ただし、2 つの AS には IGP-learned バックドア ルートと eBGP-learned のルートがあります。ポリシーは、IGP-learned パスを優先パスとして使用し、IGP パスが停止しているときに eBGP-learned パスを使用するなどの内容になります。図 3 : バックドアの例 （46 ページ） を参照してください。

図 *3:* バックドアの例

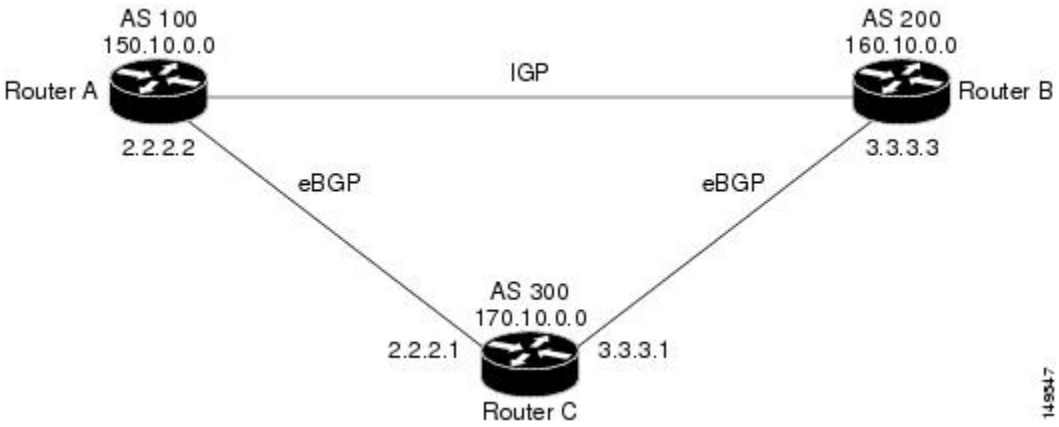


図 3 : バックドアの例 （46 ページ） では、ルータ A と C、ルータ B と C が eBGP を実行しています。ルータ A および B は、IGP を実行しています（ルーティング情報プロトコル（RIP）、Enhanced Interior Gateway Routing Protocol（IGRP）、Enhanced IGRP、または Open Shortest Path First（OSPF）など）。RIP、IGRP、Enhanced IGRP、および OSPF のデフォルト ディスタンスは、それぞれ、120、100、90、および 110 です。これらの距離はすべて eBGP のデフォルト ディスタンス（20）よりも長くなります。通常は、ディスタンスの一番小さいルートが優先されます。

ルータ A は、160.10.0.0 に関するアップデートを、eBGP と IGP の 2 つのルーティング プロトコルから受信します。eBGP のデフォルトのディスタンスが IGP のデフォルトのディスタンスよりも低いので、ルータ A はルータ C からの eBGP-learned ルートを選択します。ルータ A に

ルータ B（IGP）からの 160.10.0.0 について学習させる場合は、BGP バック ドアを確立します。を参照してください。

次の例では、ネットワーク バック ドアが設定されています。

```
RP/0/RSP0/cpu 0: router(config)# router bgp 100
RP/0/RSP0/cpu 0: router(config-bgp)# address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-af)# network 160.10.0.0/16 backdoor
```

ルータ A では、eBGP-learned ルートをローカルとして扱い、ディスタンス 200 で IP ルーティング テーブルに組み込みます。このネットワークは Enhanced IGRP を介しても学習しているため（ディスタンスは 90）、Enhanced IGRP ルートは、IP ルーティング テーブルに正常に組み込まれ、トラフィックの転送に使用されます。Enhanced IGRP-learned ルートが停止すると、eBGP-learned ルートが IP ルーティング テーブルに組み込まれ、トラフィックの転送に使用されます。

Although BGP ではネットワーク 160.10.0.0 をローカル エントリとして扱いますが、通常、ローカル エントリをアドバタイズするようにネットワーク 160.10.0.0 をアドバタイズすることはありません。

マルチプロトコル BGP

マルチプロトコル BGP は、BGP の拡張バージョンで、複数のネットワーク層プロトコル、および IP マルチキャスト ルートに関するルーティング情報を伝送します。BGP は、ユニキャスト ルーティングのセットと、マルチキャスト ルーティングのセットの 2 つのルート セットを伝送します。マルチキャスト ルーティングと関連付けられたルートは、データ分散ツリーを構築するためにプロトコル独立マルチキャスト（PIM）機能で使用されます。

マルチプロトコル BGP は、トラフィックの種類別に使用するリソースを制限するなどの目的で、マルチキャスト トラフィック専用のリンクが必要な場合に役立ちます。マルチプロトコル BGP を使用すると、マルチキャスト ルーティング トポロジとは異なるユニキャスト ルーティング トポロジによって、ネットワークおよびリソースの制御を向上できます。

BGP でドメイン間マルチキャスト ルーティングを実行する唯一の方法は、ユニキャスト ルーティングに対応できる BGP インフラストラクチャを使用することでした。通常は、すべてのマルチキャスト トラフィックを 1 つのネットワーク アクセス ポイント（NAP）で交換します。これらのルータがマルチキャスト対応でないか、マルチキャスト トラフィックのフローに適用するさまざまなポリシーがある場合は、マルチプロトコル BGP なしでマルチキャスト ルーティングをサポートできません。

（注） ユニキャストとマルチキャストの両方のネットワーク層到達可能性情報（NLRI）を交換する BGP ピアを設定することはできますが、マルチ プロトコル BGP クラウドと BGP クラウドを接続することはできません。つまり、マルチプロトコル BGP ルートを BGP に再配布できません。

図 4 : 不一致のユニキャスト ルートおよびマルチキャスト ルート （48 ページ）に、一致しておらず、したがって、マルチプロトコル BGP なしでは実現できない、単純なユニキャストとマルチキャストのトポロジを示します。

自律システム 100、200、および 300 は、FDDI リングである 2 つの NAP にそれぞれ接続しています。1 つはユニキャスト ピアリング（ユニキャスト トラフィックの交換）に使用されます。Multicast Friendly Interconnect（MFI）リングは、マルチキャスト ピアリング（マルチキャスト トラフィックの交換）に使用されます。各ルータは、ユニキャストおよびマルチキャスト対応です。

図 *4:* 不一致のユニキャスト ルートおよびマルチキャスト ルート

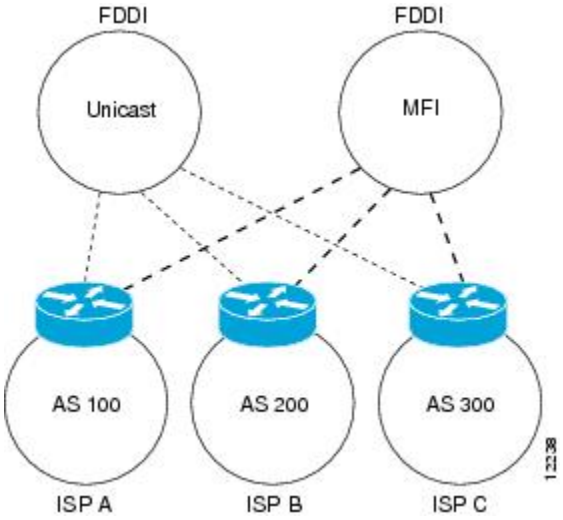


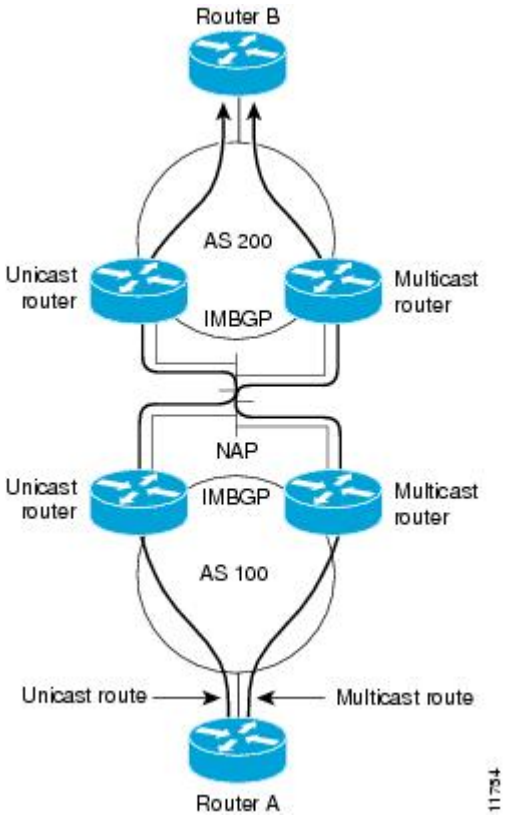
図 5 : マルチキャスト BGP 環境 （49 ページ） は、ユニキャストだけに対応したルータおよびマルチキャストだけに対応したルータのトポロジです。左側にある 2 つのルータはユニキャストだけに対応しています（マルチキャスト ルーティングをサポートしていないか、マルチキャスト ルーティングを実行するよう設定されていない）。右側にある 2 つのルータはマルチキャストだけに対応したルータです。ルータ A および B は、ユニキャストおよびマルチキャスト ルーティングの両方をサポートしています。ユニキャストだけに対応したルータおよびマルチキャストだけに対応したルータは、1 つの NAP に接続されています。

図 5 : マルチキャスト BGP 環境 （49 ページ） では、ユニキャスト トラフィックだけがルータ A からユニキャスト ルータを経由してルータ B との間を行き来できます。マルチキャスト トラフィックは、このパス上を流れることができないため、別のルーティング テーブルが必要です。マルチキャスト トラフィックは、ルータ A からマルチキャスト ルータを経由してルータ B との間を行き来するパスを使用します。

図 5 : マルチキャスト BGP 環境 （49 ページ） に、ルータ A からルータ B へのユニキャスト ルートおよびマルチキャスト ルートを別々に持つマルチプロトコル BGP 環境を示します。マルチプロトコル BGP では、これらのルートが不一致であることが許可されています。この図では、両方の自律システムに内部マルチプロトコル BGP（IMBGP）が設定されている必要があります。

PIM などのマルチキャスト ルーティング プロトコルでは、マルチキャスト BGP データベースを使用して、マルチキャスト対応の送信元に対する Reverse Path Forwarding（RPF）検索を実行します。したがって、マルチキャスト トポロジ上ではパケットの送信と受け入れが可能ですが、ユニキャスト トポロジ上ではできません。

図 *5:* マルチキャスト *BGP* 環境



ルート ダンプニング

ルート ダンプニングは、インターネットワーク上でのフラッピング ルートの伝搬を最小限に抑える BGP 機能です。ルートの状態が使用可能、使用不可能、使用可能、使用不可能という具合に、繰り返し変化する場合、ルートはフラッピングと見なされます。

たとえば、自律システム 1、自律システム 2、および自律システム 3 の 3 つの BGP 自律システムがあるネットワークについて考えます。自律システム 1 のネットワーク A へのルートがフラッピングする（利用できなくなる）と仮定します。ルート ダンプニングがない状況では、自律システム 1 から自律システム 2 への eBGP ネイバーは、取り消しメッセージを自律システム 2 に送信します。次に自律システム 2 内の境界ルータは、取り消しメッセージを自律システム 3 に伝播します。ネットワーク A へのルートが再出現したとき、自律システム 1 は自律システム 2 に、自律システム 2 は自律システム 3 にアドバタイズメント メッセージを送信します。ネットワーク A へのルートが利用可能になったり不可になったりを繰り返す場合、取り消しメッセージおよびアドバタイズメント メッセージが多数送信されます。ルート フラッピング

は、インターネットに接続されたインターネットワークでの問題です。インターネットのバックボーンでルートのフラッピングが生じると、通常、多くのルートに影響を与えるからです。

フラッピングの最小化

ルート ダンプニング機能は、次のようにしてフラッピングの問題を最小限に抑えます。ここでも、ネットワーク A へのルートがフラッピングしたと仮定します。（ルート ダンプニングがイネーブルになっている）自律システム 2 内のルータは、ネットワーク A にペナルティ 1000 を割り当てて、履歴状態に移行させます。自律システム 2 内のルータは、引き続きネイバーにルートのステータスをアドバタイズします。ペナルティは累積されます。ルート フラップが非常に頻繁に発生し、ペナルティが設定可能な抑制制限を超える場合は、フラップの発生回数に関係なく、ルータはネットワーク A へのルートのアドバタイズを停止します。このようにして、ルート ダンプニングが発生します。

ネットワーク A に課されたペナルティは再使用制限に達するまで減衰し、達すると同時にそのルートは再びアドバタイズされます。再使用制限の半分の時点で、ネットワーク A へのルートのダンプニング情報が削除されます。

（注） ルート ダンプニングがイネーブルの場合は、リセットによってルートが取り消されるときでも、BGP ピアのリセットにペナルティは適用されません。

BGP ルーティング ドメイン コンフェデレーション

iBGP メッシュを削減する方法の 1 つとして、ある自律システムを複数の副自律システムに分割し、単一のコンフェデレーションにグループ化することがあげられます。外部からは、このコンフェデレーションは単一の自律システムであるかのように見えます。各自律システムは内部で完全にメッシュ化されていて、同じコンフェデレーション内の他の自律システムとの間には数本の接続があります。異なる自律システム内にあるピアは eBGP セッションを持ちますが、ルーティング情報は iBGP ピアと同様な方法で交換されます。具体的には、ネクスト ホップ、MED、およびローカル プリファレンス情報は維持されます。この機能により、自律システムすべてに対して単一の IGP を保持できます。

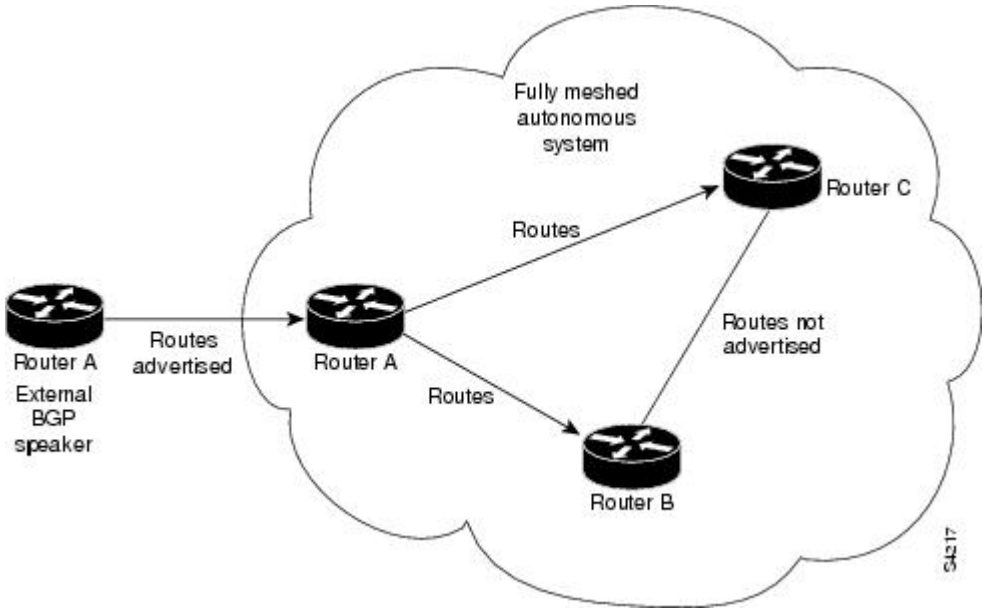
BGP ルート リフレクタ

BGP を使用するには、すべての iBGP スピーカーが完全メッシュ化されている必要があります。ただし、iBGP スピーカーの数が多い場合、この要件には適切な拡張性がありません。コンフェデレーションを設定する代わりに、ルート リフレクタ設定を使用すると iBGP メッシュを削減できます。

図 6 : 完全メッシュ化された 3 つの iBGP スピーカー （51 ページ） に、3 つの iBGP スピーカー（ルータ A、B、C）を持つ、単純な iBGP 設定の例を示します。ルート リフレクタがない場合、ルータ A は外部ネイバーからルートを受け取ると、そのルートをルータ B と C の両方にアドバタイズする必要があります。ルータ B と C は iBGP が学習したルートを他の iBGP ス

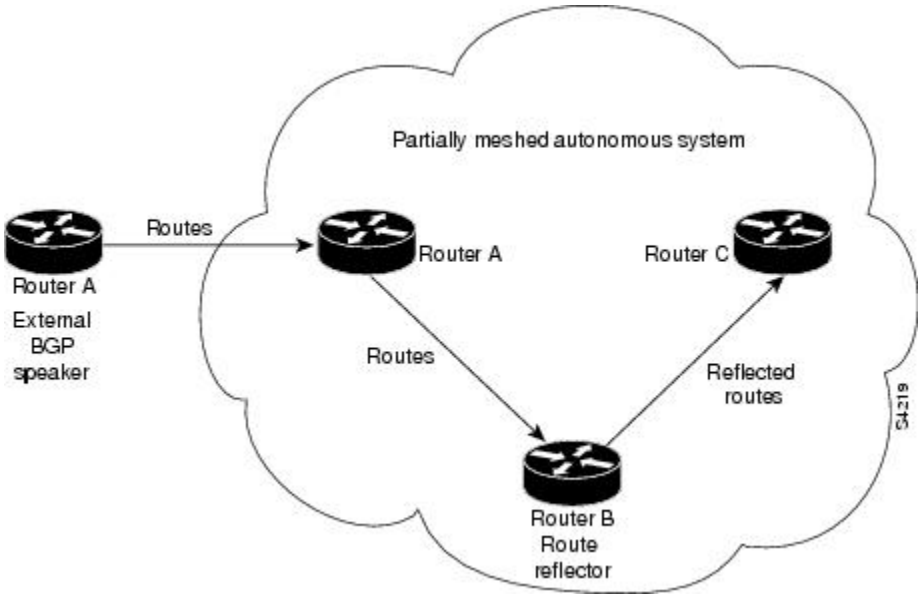
ピーカーに再アドバタイズしません。これは、これらのルータが内部ネイバーから他の内部ネイバーに学習したルートを渡さないことで、ルーティング情報のループを防ぐためです。

図 *6*:完全メッシュ化された *3* つの *iBGP* スピーカー



ルートリフレクタがある場合は、学習したルートをネイバーに渡す方法があるため、すべての iBGP スピーカーを完全にメッシュ化する必要はありません。このモデルでは、iBGP が学習したルートを一連の iBGP ネイバーに渡す役割を持つルート リフレクタとして、1 つの iBGP ピアを設定しています。図 7 : ルート リフレクタのある単純な BGP モデル （52 ページ） では、ルータ B がルートリフレクタとして設定されています。ルータ A からアドバタイズされたルートをルートリフレクタが受信すると、ルータ C にアドバタイズします。逆の場合も同じです。このスキームにより、ルータ A とルータ C 間の iBGP セッションは不要になります。

図 *7:* ルート リフレクタのある単純な *BGP* モデル



ルート リフレクタの内部ピアは、次の 2 種類のグループに分けられます。クライアントのピアと、自律システム内の他の全ルータ（非クライアント ピア）です。ルート リフレクタは、これらの 2 つのグループ間でルートを反映させます。ルート リフレクタおよびそのクライアント ピアは、クラスタを形成します。非クライアント ピアは相互に完全メッシュ構造にする必要がありますが、クライアント ピアはその必要はありません。クラスタ内のクライアントは、クラスタ外の iBGP スピーカーとは通信しません。

図 *8 :* より複雑な *BGP* ルート リフレクタのモデル

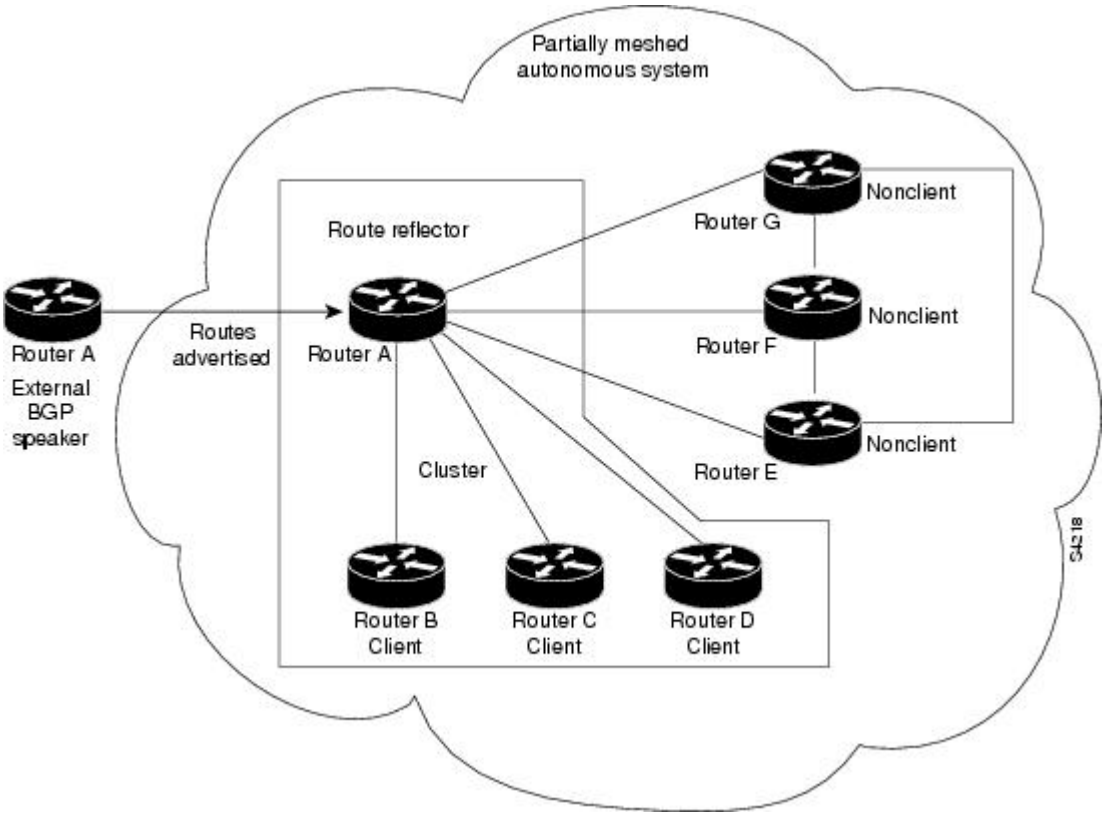


図 8 : より複雑な BGP ルート リフレクタのモデル （53 ページ） に、より複雑なルート リフレクタのスキームを示します。ルータ A は、ルータ B、C、および D があるクラスタ内のルート リフレクタです。ルータ E、F、および G は完全にメッシュ化された非クライアント ルータです。

ルート リフレクタがアドバタイズされたルートを受信すると、ネイバーに応じて、次のようなアクションを取ります。

- 外部 BGP スピーカーからのルートをすべてのクライアントおよび非クライアント ピアにアドバタイズします。
- 非クライアント ピアからのルートをすべてのクライアントにアドバタイズします。
- クライアントからのルートをすべてのクライアントおよび非クライアント ピアにアドバタイズします。したがって、クライアントを完全メッシュ構造にする必要はありません。

ルート リフレクタ対応の BGP スピーカーとともに、ルート リフレクタの概念に対応していない BGP スピーカーを併用することもできます。これらは、クライアントまたは非クライアント グループのメンバとなることができます。したがって、旧 BGP モデルからルート リフレクタ モデルへ、簡単に順次移行できます。たとえば、最初に、ルート リフレクタおよびいくつかのクライアントを持つ単一のクラスタを作成します。他のすべての iBGP スピーカーはルート リフレクタに対して非クライアント ピアとすることができ、クラスタを作成して徐々に追加します。

自律システムは複数のルート リフレクタを持つことができます。ルート リフレクタは、他のルートリフレクタを他の iBGP スピーカーと同様に扱います。ルートリフレクタは、他のルート リフレクタをクライアント グループまたは非クライアント グループに含むように設定できます。単純な設定では、バックボーンを多数のクラスタに分割してもかまいません。各ルートリフレクタは、非クライアント ピアとして他のルート リフレクタとともに設定されます（このため、すべてのルートリフレクタは完全メッシュ化されます）。クライアントは、所属するクラスタのルート リフレクタとだけ、iBGP セッションを維持するように設定されます。

通常、クライアントのクラスタには、ルートリフレクタが 1 つ存在します。その場合、クラスタはルートリフレクタのルータ ID で識別されます。冗長性を向上させ、シングル ポイント障害を避けるために、クラスタは複数のルートリフレクタを含むことがあります。この場合、クラスタ内のすべてのルート リフレクタにクラスタ ID を設定し、ルートリフレクタが同一クラスタ内のルートリフレクタからのアップデートを識別できるようにする必要があります。クラスタに機能を提供しているルートリフレクタはすべて完全メッシュ化され、同一のクライアントおよび非クライアント ピアのセットを持っている必要があります。

デフォルトでは、ルートリフレクタのクライアントは完全メッシュ化されている必要はなく、クライアントからのルートは他のクライアントに反映されます。ただし、クライアントが完全メッシュ化されている場合は、ルートリフレクタはルートをクライアントに反映する必要はありません。

iBGP が学習したルートが反映されるため、ルーティング情報がループする場合があります。ルート リフレクタ モデルには、ルーティングのループを防ぐ、次のようなメカニズムがあります。

- 送信元 ID は、任意で非過渡的な BGP 属性です。これは 4 バイトの属性で、ルート リフレクタにより作成されます。この属性は、ローカル自律システムのルートの送信元のルータ ID を保持します。したがって、設定ミスによりルーティング情報が送信元に戻ってくる場合、その情報は無視されます。
- クラスタ リストは任意で非過渡的な BGP 属性です。これは、ルートが渡したクラスタ ID のシーケンスです。ルートリフレクタでは、クライアントから非クライアント ピアにルートを反映するとき（およびその逆のとき）、ローカル クラスタ ID をクラスタ リストに付加します。クラスタ リストが空の場合は、新規のクラスタ リストが作成されます。ルートリフレクタでは、この属性を使用して、設定ミスによりルーティング情報が同じクラスタにループ バックしているかどうかを識別できます。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。

RPL：プレフィックスが is-best-path/is-best-multipath の場合

ボーダー ゲートウェイ プロトコル（BGP）ルータは、同じ宛先への複数のパスを受信します。標準として、デフォルトでは、BGP ベストパスアルゴリズムが IP ルーティングテーブルにインストールする最適なパスを決定します。これはトラフィックの転送に使用されます。

BGP は、最初の有効なパスを現在のベストパスとして割り当てます。次に、BGP は、ベストパスとリスト内の次のパスとを比較します。このプロセスは、BGP が有効なパスのリストの最後に到達するまで継続されます。これには、ベストパスの決定に使用されるすべてのルールが

含まれます。指定されたアドレスプレフィックスに複数のパスがある場合、BGPは次のように処理します。

- ベストパス選択ルールに従って、パスの 1 つをベストパスとして選択します。
- 転送テーブルにベストパスをインストールします。各 BGP スピーカーは、ピアへのベストパスのみをアドバタイズします。

（注） ベストパスのみを送信するアドバタイズメントルールは、そのピアに対して BGP スピーカ上に存在する宛先の完全なルーティング状態を伝達しません。

BGPスピーカがピアのいずれかからパスを受信した後、ピアがそのパスをパケットの転送に使用します。他のすべてのピアは、このピアから同じパスを受信します。これにより、BGPネットワークでの一貫したルーティングが実現します。リンク帯域幅使用率を向上させるには、ほとんどの BGP 実装では、特定の条件を満たす追加パスをマルチパスとして選択し、それらを転送テーブルにインストールします。このような着信パケットは、ベストパスとマルチパス上でロードバランシングされます。ピアにアドバタイズされていない転送テーブルにパスをインストールできます。RR ルートリフレクタは、ベストパスとマルチパスを検出します。このようにして、ルートリフレクタはベストパスとマルチパスに異なるコミュニティを使用します。この機能を使用すると、RR または境界ルータによって実行されるローカルの決定を BGP で通知できます。この新機能を使用した場合は、コミュニティストリングを使用して RR によって選択されました（たとえば、is-best-path の場合は community 100:100）。コントローラは、どのベストパスがすべての R に送信されるかを確認します。ボーダー ゲートウェイ プロトコル ルータは、同じ宛先への複数のパスを受信します。ベストパスの計算を実行している間は、1 つのベストパスが存在し、場合によっては同等のパスおよび同等でない若干数のパスが存在します。したがって、abest-path と is-equal-best-path の要件です。

BGP のベストパスアルゴリズムは、IP ルーティングテーブル内でベストパスを決定し、トラフィックの転送に使用します。RPL内のこの機能拡張により、決定を行うためのポリシーを作成できます。ベストパスのローカル選択のためのコミュニティストリングの追加。BGP追加パス（Add Path）の導入により、BGP はベストパスよりも多くを通知するようになりました。BGP はベストパスと、ベストパスと同等のパス全体を通知できます。これは、BGP マルチパスルールとすべてのバックアップパスに従っています。

RPLネクストホップ破棄設定を使用したリモートトリガ型ブラックホールのフィルタリング

リモートトリガ型ブラックホール（RTBH）フィルタリングは、保護されたネットワークに入る前に望ましくないトラフィックをドロップする機能を提供する技術です。RTBH フィルタリングは、null0 インターフェイスに転送することによって、送信元アドレスまたは宛先アドレスのいずれかに基づいて、ネットワークのエッジで望ましくないトラフィックをすばやくドロップする方法を提供します。宛先アドレスに基づく RTBH フィルタリングは、一般に宛先

ベースの RTBH フィルタリングと呼ばれます。一方、送信元アドレスに基づく RTBH フィルタリングは、送信元ベースの RTBH フィルタリングと呼ばれます。

RTBH フィルタリングは、セキュリティツールキットの多くの技術の 1 つであり、次の方法でネットワークセキュリティを強化するために一緒に使用できます。

- DDoS 攻撃とワーム攻撃を効果的に軽減する
- 攻撃下でターゲットを宛先とするすべてのトラフィックを隔離する
- ブロックリスト フィルタリングの適用

宛先ベースの RTBH フィルタリングの設定

RTBH は、**set next-hop discard** コマンドを使用して、ネクストホップで望ましくないトラフィックを破棄するルートポリシー（RPL）を定義することによって実装されます。

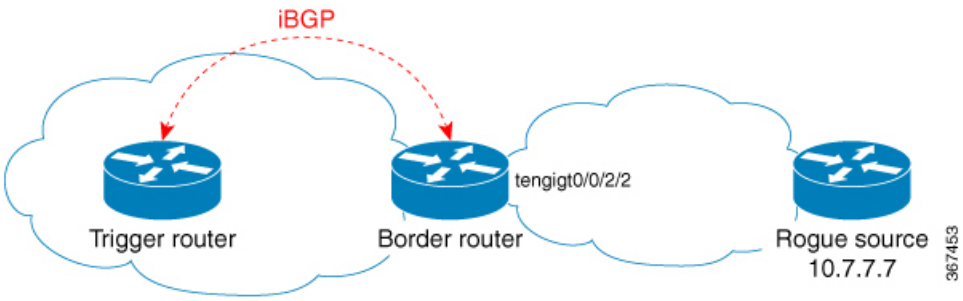
RTBH フィルタリングは、対象のプレフィックスのネクストホップをヌルインターフェイスに設定します。対象を宛先とするトラフィックは、入力時にドロップされます。

set next-hop discard 設定は、ネイバー インバウンド ポリシーで使用されます。この設定がパスに適用されている場合、プライマリネクストホップは実際のパスに関連付けられますが、Null0 に設定されたネクストホップで RIB が更新されます。受信したプライマリネクストホップが到達不能であっても、RTBH パスは到達可能と見なされ、ベストパス選択プロセスの候補となります。RTBH パスは、通常の BGP アドバタイズメントルールに基づいて、受信したネクストホップまたは nexthop-self のいずれかを持つ他のピアに再度アドバタイズされます。

RTBH フィルタリングの一般的な展開シナリオでは、アクセスおよび集約ポイントで内部ボーダー ゲートウェイ プロトコル（iBGP）を実行し、トリガーとして動作するようにネットワーク オペレーション センター（NOC）で個別のデバイスを設定する必要があります。トリガー側のデバイスは、iBGP 更新をエッジに送信します。これにより、望ましくないトラフィックが null0 インターフェイスに転送され、ドロップされます。

次に、不正ルータが境界ルータにトラフィックを送信しているトポロジを示します。

図 *9: RTBH* フィルタリングを実装するためのトポロジ



トリガールータに適用される設定

特殊なタグでマークされた静的ルートにコミュニティを設定し、BGP に適用する静的ルート再配布ポリシーを設定します。

```
route-policy RTBH-trigger
  if tag is 777 then
    set community (1234:4321, no-export) additive
    pass
  else
    pass
  endif
  end-policy

router bgp 65001
 address-family ipv4 unicast
  redistribute static route-policy RTBH-trigger
 !
 neighbor 192.168.102.1
  remote-as 65001
  address-family ipv4 unicast
   route-policy bgp_all in
   route-policy bgp_all out
```

ブラックホール化させる必要がある送信元プレフィックスの特殊なタグを使用して静的ルートを設定します。

```
router static
 address-family ipv4 unicast
 10.7.7.7/32 Null0 tag 777
```

ボーダールータに適用される設定

トリガールータのコミュニティセットと一致するルートポリシーを設定し、次のように set next-hop discard を設定します。

```
route-policy RTBH
  if community matches-any (1234:4321) then
    set next-hop discard
  else
    pass
  endif
end-policy
```

次のように、ルートポリシーを iBGP ピアに適用します。

```
router bgp 65001
 address-family ipv4 unicast
 !
 neighbor 192.168.102.2
  remote-as 65001
  address-family ipv4 unicast
   route-policy RTBH in
   route-policy bgp_all out
```

確認

境界ルータで、プレフィックス 10.7.7.7/32 に Nexthop-discard というフラグが付けられます。

```
RP/0/RSP0/CPU0:router#show bgp
BGP router identifier 10.210.0.5, local AS number 65001
BGP generic scan interval 60 secs
BGP table state: Active
Table ID: 0xe0000000   RD version: 12
```

```
BGP main routing table version 12
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
N>i10.7.7.7/32        192.168.102.2            0    100      0 ?
```

```
RP/0/RSP0/CPU0:router#show bgp 10.7.7.7/32
BGP routing table entry for 10.7.7.7/32
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                12          12
Last Modified: Jul  4 14:37:29.048 for 00:20:52
Paths: (1 available, best #1, not advertised to EBGP peer)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  Local
    192.168.102.2 (discarded) from 192.168.102.2 (10.210.0.2)
      Origin incomplete, metric 0, localpref 100, valid, internal best, group-best
      Received Path ID 0, Local Path ID 1, version 12
      Community: 1234:4321 no-export
```

```
RP/0/RSP0/CPU0:router#show route 10.7.7.7/32

Routing entry for 10.7.7.7/32
  Known via "bgp 65001", distance 200, metric 0, type internal
  Installed Jul 4 14:37:29.394 for 01:47:02
  Routing Descriptor Blocks
    directly connected, via Null0
      Route metric is 0
  No advertising protos.
```

show コマンドのデフォルトのアドレス ファミリ

show コマンドのほとんどは、アドレスファミリ（AFI）およびサブアドレスファミリ（SAFI）の引数を使用します（AFI および SAFI については、RFC 1700 および RFC 2858 を参照してください）。Cisco IOS XR ソフトウェアパーサーには、afi および safi を設定して、**show** コマンドの実行時には指定する必要がないようにする機能があります。次のパーサー コマンドがあります。

- **set default-afi** { **ipv4** | **ipv6** | **all** }
- **set default-safi** { **unicast** | **multicast** | **all** }

パーサーでは、デフォルト afi 値が **ipv4** に、デフォルト safi 値が **unicast** に自動的に設定されます。デフォルトの afi 値を **ipv4** から変更する、あるいはデフォルトの safi 値を **unicast** から変更する場合、使用する必要があるのはパーサーコマンドのみです。**show** コマンドに指定された **afi** または **safi** キーワードは、パーサーコマンドを使用して設定した値を上書きします。afi および safi に現在設定されている値を確認するには、次の **show default-afi-safi-vrf** コマンドを使用します。

TCP Maximum Segment Size

最大セグメントサイズ（MSS）は、コンピュータまたは通信デバイスが単一のフラグメント化されていないTCPセグメントで受信できるデータの最大量です。すべてのTCPセッションは、単一のパケットで転送可能なバイト数に関する制限によってバインドされます。この制限がMSS です。TCP は、パケットを IP レイヤに渡す前に、送信キューでパケットをチャンクに分割します。

TCP MSS 値は、インターフェイスの最大伝送ユニット（MTU）に依存します。これは、1 つのインスタンスでプロトコルによって送信可能なデータの最大長です。最大 TCP パケット長は、TCP セットアップ プロセス中に、送信元デバイスのアウトバウンド インターフェイスの MTU と宛先デバイスによって知らされる MSS の両方によって決まります。MSS が MTU に近づくほど、BGP メッセージの転送がより効率的になります。データフローの各方向に異なる MSS 値を使用できます。

ネイバー単位の TCP MSS

ネイバー単位の TCP MSS 機能を使用すると、ネイバーごとに一意の TCP MSS プロファイルを作成できます。ネイバー単位の TCP MSS は、ネイバーグループとセッショングループの 2 つのモードでサポートされています。以前は、TCP MSS 設定は、BGP 設定のグローバルレベルでのみ使用できるようになっていました。

ネイバー単位の TCP MSS 機能では、以下を行えます。

- ネイバー単位の TCP MSS 設定を有効にする。
- **inheritance-disable** コマンドを使用して、ネイバーグループまたはセッショングループの特定のネイバーの TCP MSS を無効にする。
- TCP MSS 値の設定を解除する。設定解除時に、プロトコル制御ブロック（PCB）の TCP MSS 値がデフォルト値に設定されます。

（注） デフォルトの TCP MSS 値は 536（オクテット単位）または 1460（バイト単位）です。MSS のデフォルトの 1460 は、TCP がパケットを IP レイヤに渡す前に、送信キュー内のデータを 1460 バイトのチャンクにセグメント化することを意味します。

ネイバー単位の TCP MSS を設定するには、ネイバー単位、ネイバーグループまたはセッショングループの設定で **tcp mss** コマンドを使用します。

詳細な設定手順については、ネイバー単位の TCP MSS の設定 （107 ページ）を参照してください。

ネイバー単位の TCP MSS を無効にする詳細な手順については、ネイバー単位の TCP MSS の無効化 （109 ページ）を参照してください。

MPLS VPN Carrier Supporting Carrier

Carrier Supporting Carrier（CSC）は、サービス プロバイダーの 1 つが別のサービス プロバイダーに自社のバックボーン ネットワークのセグメントの使用を許可する状況を記述した用語です。他のプロバイダーにバックボーン ネットワークのセグメントを提供するサービス プロバイダーは、バックボーン キャリアと呼ばれます。バックボーン ネットワークのセグメントを使用するサービス プロバイダーは、カスタマー キャリアと呼ばれます。

バックボーン キャリアは、ボーダー ゲートウェイ プロトコル/マルチプロトコル ラベル スイッチング（BGP/MPLS）VPN サービスを提供します。カスタマー キャリアは、次のいずれかになります。

- インターネット サービス プロバイダー（ISP）（定義上、ISP は VPN サービスを提供しません）
- BGP/MPLS VPN サービス プロバイダー

BGP をイネーブルにするように CSC ネットワークを設定して、バックボーン キャリア プロバイダー エッジ（PE）ルータとカスタマー キャリア カスタマー エッジ（CE）ルータ間のルートおよび MPLS ラベルを、複数パスを使用して転送できます。BGP を使用して IPv4 ルートと MPLS ラベル ルートを配布する利点を次に示します。

- BGPは、VPNルーティング/転送（VRF）テーブル内で内部ゲートウェイ プロトコル（IGP）およびラベル配布プロトコル（LDP）の代わりをします。BGPを使用して、ルートおよび MPLS ラベルを配布できます。2 つではなく単一のプロトコルを使用すると、設定およびトラブルシューティングが簡単になります。
- BGP は、2 つの ISP を接続する場合の優先ルーティング プロトコルです。主な理由は、そのルーティング ポリシーと拡張性です。ISP では、通常、2 つのプロバイダー間で BGP を使用します。この機能を使用すると、これらの ISP は BGP を使用できます。

BGP を使用した MPLS VPN CSC の設定の詳細については、 *MPLS Configuration Guide for Cisco ASR 9000 Series RoutersMPLS Configuration Guide for Cisco NCS 560 Series Routers*の「*Implementing MPLS Layer 3 VPNs on* Cisco ASR 9000 シリーズ ルータ 」のモジュールを参照してください。

BGP キーチェーン

BGP キーチェーンを使用すると、2 つの BGP ピア間のキーチェーン認証がイネーブルになります。BGP のエンドポイントは、どちらも draft-bonica-tcp-auth-05.txt を順守する必要があり、一方のエンドポイントのキーチェーンと、もう一方のエンドポイントのパスワードは機能しません。

キーチェーン管理の詳細については、 *System Security Configuration Guide for Cisco ASR 9000 Series Routers*を参照してください。

BGP では、認証にこのキーチェーンを使用して、ヒットレス キー ロールオーバーを実装できます。キー ロールオーバーの仕様は時間に基づいているため、ピア間で時計のずれがあるとロールオーバーのプロセスに影響します。許容値の指定を設定できるため、承認時間枠をその

分だけ（前後に）拡張できます。この承認時間枠により、アプリケーション（ルーティングプロトコルおよび管理プロトコルなど）のヒットレス キー ロールオーバーが容易になります。

キーのロール オーバーは、エンドポイントでのキーチェーン設定の不一致が原因でセッショントラフィック（送信または受信）で使用する共通のキーがない場合を除き、BGP セッションには影響しません。

BGP ノンストップ ルーティング

ボーダー ゲートウェイ プロトコル（BGP）のノンストップ ルーティング（NSR）とステートフル スイッチオーバー（SSO）機能を使用すると、すべての bgp ピアリングで BGP 状態を維持し、サービスを中断させるおそれのあるイベントの実行中にも連続的なパケット転送を行えるようになります。NSR の下では、サービスを中断するおそれのあるイベントは、ピア ルータに表示されません。プロトコル セッションは中断されず、ルーティング ステートはプロセスの再起動とスイッチオーバーをまたがって維持されます。

BGP NSR では、次のイベントの際のノンストップ ルーティングを実現します。

- ルート プロセッサ スイッチオーバー
- BGP または TCP でのプロセスのクラッシュまたはプロセス障害

（注） BGP NSR は、デフォルトで有効になっています。BGP NSR を無効にするには、**nsr disable** コマンドを使用します。また、無効になっている BGP NSR を有効に戻すには、**no nsr disable** コマンドを使用します。

プロセスのクラッシュまたはプロセス障害が発生した場合、NSR は **nsr process-failures switchover** コマンドが設定されている場合にのみ維持されます。アクティブなインスタンスのプロセス障害が発生した場合は、**nsr process-failures switchover** により復旧処理としてフェールオーバーが設定され、スタンバイ ルート プロセッサ（RP）またはスタンバイ分散型ルートプロセッサ（DRP）にスイッチ オーバーが行われることで、NSR が維持されます。コンフィギュレーション コマンドの一例として、RP/0/RSP0/CPU0:router(config) # nsr process-failures switchover があります。

nsr process-failures switchover コマンドは、BGP または TCP プロセスがクラッシュした場合に NSR セッションと BGP セッションの両方を維持します。この設定を行わないと、BGP プロセスまたは TCP プロセスがクラッシュした場合に BGP ネイバー セションがフラップします。この設定は、BGP ネイバーのフラップが予想される場合に BGP プロセスまたは TCP プロセスが再起動する場合は役立ちません。

ルート プロセッサ スイッチオーバーおよびインサービス システムのアップグレード（ISSU）の間、NSR は TCP と BGP の両方のステートフル スイッチオーバー（SSO）によって実現されます。

NSR では、ネットワーク内の他のルータ上でソフトウェア アップグレードを強要せず、NSR をサポートするためにピア ルータは必要ありません。

障害に起因するルート プロセッサ スイッチオーバーが発生した場合、TCP 接続および BGP セッションはトランスペアレントにスタンバイ ルート プロセッサに移行され、スタンバイ ルート プロセッサがアクティブになります。既存のプロトコル ステートは、アクティブになるスタンバイ ルート プロセッサ上で維持されて、ピアによるプロトコル ステートのリフレッシュは不要です。

ソフト再設定やポリシーの変更などのイベントにより、BGP の内部状態が変化することがあります。このようなイベントの際に、アクティブとスタンバイの BGP プロセスの間でステートの一貫性を確保するために、同期ポイントとして機能する、ポストイットの概念が導入されています。

BGP NSR には次の機能があります。

- NSR 関連のアラームおよび通知
- 設定され、動作している NSR の状態は、個別に追跡される
- NSR 統計情報の収集
- **show** コマンドを使用した NSR 統計情報の表示
- XML スキーマのサポート
- アクティブとスタンバイのインスタンス間のステート同期を検証する監査メカニズム
- NSR をイネーブルおよびディセーブルにする CLI コマンド
- 5000 NSR セッションのサポート

BGP Local Label Retention

プライマリ PE-CE リンクが故障した場合、BGP では、プライマリ パスに対応するルートおよびこのルートのローカル ラベルを取り消し、デフォルトでは、ルーティング情報ベース（RIB）および転送情報ベース（FIB）にバックアップ パスをプログラムします。

ただし、プライマリ PE のすべての内部ピアがバックアップ パスを新しい最適パスとして使用するように再コンバージェンスするまで、トラフィックは、プライマリ パスに割り当てられたローカル ラベルとともに、引き続きプライマリ PE に転送されます。したがって、プライマリ パスに前に割り当てられていたローカル ラベルは、再コンバージェンス後、設定可能な期間、プライマリ PE 上で保持する必要があります。BGP Local Label Retention 機能を使用すると、ローカル ラベルを指定期間保持できます。時間を指定していない場合、ローカル ラベルは、デフォルト値の 5 分間保持されます。

retain local-label コマンドを使用すると、ネットワークがコンバージェンスされるまで、ローカル ラベルを保持できます。

BGP コマンドに対するコマンドライン インターフェイス（CLI）の一貫性

Cisco IOS XR リリース 3.9.0 以降、ボーダー ゲートウェイ プロトコル（BGP）コマンドでは、**disable** キーワードを使用して、機能を無効にします。キーワード **inheritance-disable** では、親レベルからの機能プロパティの継承が無効になります。

BGP の追加パス

ボーダーゲートウェイ プロトコル（BGP）の追加パス機能では、1 つのプレフィックスに対して複数のパスを送信できるように、BGP スピーカーの BGP プロトコル機械を変更します。これにより、ネットワークに「パスの多様性」が生まれます。追加パスにより、エッジルータでの BGP プレフィックス独立コンバージェンス（PIC）が可能になります。

（注） BGP 追加パス機能は、VRF ではサポートされていません。

BGP 追加パスでは、iBGP ネットワーク内の追加パス アドバタイズメントが可能になり、プレフィックスに対する次のタイプのパスがアドバタイズされます。

- バックアップ パス：高速コンバージェンスおよび接続の回復をイネーブルにします。
- グループ最適パス：ルート振動を解決します。
- すべてのパス：iBGP フル メッシュをエミュレートします。

（注） 追加パスは、MDT、トンネル、および L2VPN アドレス ファミリと eBGP ピアリングでは、サポートされていません。

iBGP マルチパス ロード シェアリング

ローカル ポリシーが設定されていないボーダー ゲートウェイ プロトコル（BGP）対応ルータが複数のネットワーク層到達可能性情報（NLRI）を同じ宛先の内部 BGP（iBGP）から受信すると、このルータは 1 つの iBGP パスを最適パスとして選択します。この最適パスは、次にこのルータの IP ルーティング テーブルに組み込まれます。

iBGP のマルチパス ロード シェアリング機能を使用すると、BGP 対応ルータでは、複数の iBGP パスを宛先への最適パスとして選択できます。この最適パスまたはマルチパスは、次にこのルータの IP ルーティング テーブルに組み込まれます。

eBGP から取得した到達可能性情報を持つ複数の境界 BGP ルータがあり、ローカル ポリシーが適用されていない場合、境界ルータでは、eBGPパスを最適パスとして選択します。境界ルータでは、この最適パスをISPネットワークの内部にアドバタイズします。コアルータの場合、同じ宛先に対し複数のパスがある場合がありますが、1つのパスのみを最適パスとして選択し、そのパスを転送用に使用します。iBGPマルチパス ロードシェアリングでは、複数の等距離パス間でロード シェアリングを可能にする機能が追加されます。

複数の iBGP の最適パスを設定すると、ルータでは、特定のサイトを宛先とするトラフィックを均等に負担できるようになります。

iBGP のマルチパス ロード シェアリング機能は、サービス プロバイダー バックボーンを持つマルチプロトコルラベルスイッチング（MPLS）バーチャルプライベートネットワーク（VPN）と同様に機能します。

同じ宛先への複数のパスをマルチパスと見なすには、次の基準を満たす必要があります。

- すべての属性が同じである必要があります。属性には、重み、ローカルプリファレンス、自律システム パス（長さだけでなく属性全体）、発信元コード、Multi Exit Discriminator（MED）、および Interior Gateway Protocol（IGP）距離が含まれます。
- 各マルチパスのネクスト ホップ ルータが異なっている必要があります。

基準を満たしていて、複数のパスがマルチパスと見なされても、BGP対応ルータは、引き続きマルチパスの 1 つをベスト パスに指定し、このベスト パスをそのネイバーにアドバタイズします。

（注） マルチパスの変更後、eiBGPマルチパス候補の評価中にIGPメトリックは考慮されず、また、最適でないパスを使用できます。

内部および外部の BGP マルチパスが設定されている Carrier Supporting Carrier（CSC）ネットワークでは、VRF 単位のラベルモードはサポートされていません。

VRF単位のラベルモードは、ループを引き起こす可能性があるため、eiBGPマルチパスがあるBGPPICエッジには使用できません。プレフィックス単位のラベルのみが、VRF単位のラベルモードをサポートしています。

BGP 選択的マルチパス

従来の BGP マルチパス機能を使用すると、同じ宛先への並列パスを受信するルータは、ルーティングテーブルに複数のパスをインストールできます。デフォルトでは、このマルチパス機能は設定されているすべてのピアに適用されます。BGP選択的マルチパスでは、選択したピアのみにマルチパス機能を適用できます。

複数のパスを受信する BGP ルータは、**maximum-paths ... selective** オプションを使用して設定されます。複数のパスを共有する iBGP/eBGP ネイバーは、**multipath** オプションを使用して設定され、BGP ルータ上にネイバーとして追加されます。

（注） マルチパスをアドバタイズする前にマルチホップ情報を上書きしないようにするには、**next-hop-unchanged multipath** コマンドを使用します。

BGP 選択的マルチパスの使用時には、次の動作に注意してください。

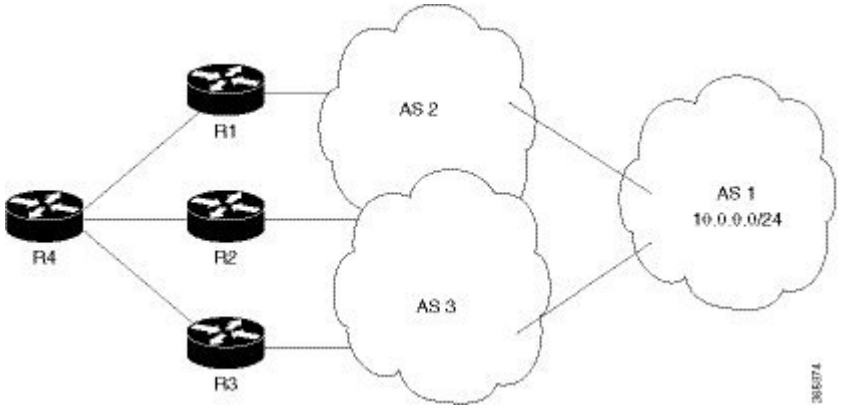
- BGP選択的マルチパスは、ベストパスの計算には影響しません。ベストパスは、マルチパスのセットに常に含まれています。
- VPN プレフィックスの場合、PE パスは「常に」マルチパスの対象となります。

maximum-paths コマンドと **multipath** コマンドについては、『*Cisco ASR 9000 Series Aggregation Services Router Routing Command Reference*』を参照してください。

トポロジ

次の図に、この項で使用する設定を図示したトポロジの例を示します。

図 *10 : BGP* 選択的マルチパス



ルータ R4 は、ルータ R1、R2、および R3 から同じ宛先への並列パスを受信します。ルータ R1 と R2 がルータ R4 上の選択的マルチパスネイバーとして設定されている場合、これらのルータからの並列パスだけがルータ R4 のルーティングテーブルにインストールされます。

コンフィギュレーション

（注） この機能を設定する前に、ルータ上で実行されている iBGP/eBGP を使用してネットワークトポロジを設定します。

ルータ R4 上に BGP 選択的マルチパスを設定するには、次の手順を実行します。

1. トポロジ内の選択した複数のパスを受け入れるようにルータ R4 を設定します。

```
/* To configure selective multipath for iBGP/eBGP
```

```
RP/0/RSP0/cpu 0: router(config)# router bgp 1
RP/0/RSP0/cpu 0: router(config-bgp)# address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-af)# maximum-paths ibgp 4 selective
RP/0/RSP0/cpu 0: router(config-bgp-af)# maximum-paths ebgp 5 selective
RP/0/RSP0/cpu 0: router(config-bgp-af)# commit

/* To configure selective multipath for eiBGP
RP/0/RSP0/cpu 0: router(config)# router bgp 1
RP/0/RSP0/cpu 0: router(config-bgp)# address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-af)# maximum-paths eibgp 6 selective
RP/0/RSP0/cpu 0: router(config-bgp-af)# commit
```

2. ルータ R4 のネイバーを設定します。

ルータ R1（1.1.1.1）および R2（2.2.2.2）は、**multipath** オプションを使用してネイバーとして設定されます。

ルータ R3（3.3.3.3）は **multipath** オプションを使用せずにネイバーとして設定されているため、このルータからのルートをマルチパスとして選択することはできません。

```
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 1.1.1.1
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# multipath
RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# commit

RP/0/RSP0/cpu 0: router(config-bgp-nbr)# neighbor 2.2.2.2
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# multipath
RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# commit

RP/0/RSP0/cpu 0: router(config-bgp-nbr)# neighbor 3.3.3.3
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# commit
```

BGP 選択的マルチパス機能が正常に設定されました。

累積内部ゲートウェイ プロトコル属性

累積内部ゲートウェイ プロトコル（AiGP）属性は、オプションで非推移的な BGP パス属性です。AiGP 属性の属性タイプ コードは、IANA によって割り当てられます。AiGP 属性の値フィールドは、タイプ、長さ、値（TLV）の要素として定義されます。AiGP TLV には、累積 IGP メトリックが含まれます。

AiGP 機能は 3107 ネットワークに必要であり、パスに関連付けられた距離を計算する現在の OSPF の動作をシミュレートします。OSPF/LDP では、プレフィックスおよびラベル情報をローカル領域だけに入れて伝送します。次に、BGP では、エリア境界にある BGP にルートを再配布することにより、すべてのリモートエリアにプレフィックスおよびラベルを伝送します。次に、ルートおよびラベルが、LSP を使用してアドバタイズされます。ルートのネクストホップはローカル ルータに対する各 ABR で変更されます。これによって、エリア境界を越えて OSPF ルートをリークする必要がなくなります。各コア リンクで使用可能な帯域幅が OSPF コストにマップされます。したがって、BGP では、各 PE 間でこのコストを正しく伝送する必要があります。この機能は、AiGP を使用して実現されています。

IPv6 プロバイダー エッジの VRF ごとおよび CE ごとのラベル

IPv6 のための VRF ごとおよび CE ごとのラベルの機能により、デフォルト VRF ごとまたは CE ネクスト ホップごとにラベルを割り当てることにより、ラベル スペースを節約できるようになります。

デフォルトでは、すべての IPv6 プロバイダー エッジ（6PE）ラベルは、プレフィックスごとに割り当てられます。VRF インスタンスに属する各プレフィックスは1つのラベルを使ってアドバタイズされます。これは、パケットのカスタマー エッジ（CE）ネクスト ホップを決定するために、VRF フォワーディング テーブルでさらにルックアップが行われる原因になります。

ただし、**per-ce** キーワードまたは **per-vrf** キーワードを指定して **label mode** コマンドを使用すると、PE ルータ上での追加のルックアップが回避され、ラベル スペースが節約されます。

一意のカスタマー エッジ（CE）ピア ルータからアドバタイズされたすべてのルートで同じラベルを使用するように指定するには、**per-ce** キーワードを使用します。一意の VRF からアドバタイズされたすべてのルートで同じラベルを使用するように指定するには、**per-vrf** キーワードを使用します。

Cisco ASR 9000 の A9K-SIP-700 での IPv4 BGP ポリシー アカウンティング

ボーダー ゲートウェイ プロトコル（BGP）ポリシー アカウンティングは、異なるピア間で送受信される IP トラフィックを測定および分類します。ポリシーアカウンティングは個々の入力または出力インターフェイス単位で有効になります。IP トラフィックを識別するために、コミュニティ リスト、自律システム番号、または自律システム パスなどのパラメータに基づくカウンタが割り当てられます。

BGP ポリシー アカウンティングを使用して、通過するルートに基づいてトラフィックのアカウンティングを行うことができます。サービスプロバイダーは、すべてのトラフィックをカスタマー別に識別してアカウンティングを実施し、それに応じて課金できます。

BGP ポリシーアカウンティングと、BGP ポリシーアカウンティングの設定方法については、『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Configuration Guide*』の「*Implementing Cisco Express Forwarding*」のモジュールを参照してください。

Cisco ASR 9000 の A9K-SIP-700 での IPv6 ユニキャスト ルーティング

Cisco ASR 9000 の A9K-SIP-700 には、すべてのインターネット プロトコル バージョン 6（IPv6）ユニキャスト機能が備わっています。

IPv6 ユニキャスト アドレスは、単一ノード上の単一インターフェイスの識別子です。ユニキャスト アドレスに送信されたパケットは、そのアドレスが示すインターフェイスに配信されます。Cisco IOS XR ソフトウェアでは、次の IPv6 ユニキャスト アドレス タイプがサポートされます。

- 集約可能グローバル アドレス
- サイトローカル アドレス

- リンクローカル アドレス
- IPv4 互換 IPv6 アドレス

IPv6 ユニキャスト アドレッシングの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Configuration Guide*』の「*Implementing Network Stack IPv4 and IPv6*」モジュールを参照してください。

Cisco ASR 9000 の A9K-SIP-700 での IPv6 uRPF サポート

ユニキャスト IPv6 リバース パス転送（uRPF）は、検証可能な IP 送信元アドレスを欠いている IP パケットを廃棄することにより、不正な形式の IP 送信元アドレスまたはスプーフィングされた IP 送信元アドレスがネットワークに侵入した場合に生じる問題を軽減します。ユニキャスト RPF はシスコ エクスプレス フォワーディング（CEF）テーブルで逆ルックアップを実行することで、この処理を行います。このため、uRPF が可能になるのは、ルータで CEF が有効になっている場合だけです。

IPV6 uRPF を有効にするには、インターフェイス コンフィギュレーション モードで **ipv6 verify unicast source reachable-via {any | rx} [allow-default] [allow-self-ping]** コマンドを使用します。

IPv6 uRPF の詳細については、 *IP Addresses and Services Command Reference for Cisco ASR 9000 Series Routers*の「*Implementing Cisco Express Forwarding*」のモジュールを参照してください。

BGP の AS パスからのプライベート AS 番号の削除および置換

プライベート自律システム番号（ASN）は、グローバルに一意な AS 番号を保護するために、インターネット サービス プロバイダー（ISP）およびお客様のネットワークで使用されます。プライベート AS 番号は一意でないため、グローバル インターネットへのアクセスには使用できません。AS 番号はルーティング アップデートの eBGP AS パスに表示されます。プライベート ASN を使用している場合にグローバル インターネットにアクセスするには、AS パスからプライベート ASN を削除する必要があります。

パブリックな AS 番号は、InterNIC によって割り当てられ、グローバルに一意です。範囲は 1 ～ 64511 です。プライベート AS 番号は、グローバルに一意な AS 番号（有効な範囲は 64512 ～ 65535）を保護するために使用されます。プライベート AS 番号はグローバル BGP ルーティング テーブルにリークできません。プライベート AS 番号は一意ではなく、BGP 最適パスの計算には一意の AS 番号が必要であるからです。そのため、ルートが BGP ピアに伝播される前に、AS パスからプライベート AS 番号を削除する必要がある可能性があります。

外部 BGP（eBGP）では、グローバルなインターネットへのルーティングで、グローバルに一意な AS 番号を使用する必要があります。プライベート AS 番号（これは一意でない）を使用すると、グローバルなインターネットにアクセスできません。BGP の AS パスからプライベート ASN を削除および交換する機能によって、プライベート AS に属するルータがグローバルなインターネットにアクセスできるようになりました。ネットワーク管理者は、発信アップデート メッセージに含まれる AS パスからプライベート AS を削除するようにルータを設定します。場合によっては、これらの番号をローカル ルータの ASN で置き換えて、AS パス長が変化しないようにします。

AS パスからプライベート ASN を削除および交換する機能は、次のように拡張されました。

- **remove-private-as** コマンドでは、次の処理が行われます。
  - AS パスにパブリックとプライベートの両方の ASN が含まれる場合も、AS パスからプライベート AS 番号を削除します。
  - AS パスにプライベート AS 番号のみが含まれる場合も、プライベート AS 番号を削除します。このコマンドは eBGP ピアのみに適用され、その場合、eBGP ピアではローカル ルータの AS 番号が AS パスに付加されるため、長さ 0 の AS パスにはなることはありません。
  - AS パスでコンフェデレーションセグメントの前にプライベート ASN が出現する場合でも、プライベート AS 番号を削除します。
- **replace-as** コマンドは、パスから削除されるプライベート AS 番号をローカル AS 番号に置き換えることで、AS パスを同じ長さに保ちます。

この機能は、アドレス ファミリ コンフィギュレーション モードでネイバーに適用できます。そのため、アドレスファミリ内のネイバーにこの機能を適用すると、アウトバウンドの更新メッセージのみが影響を受けます。

プライベート AS 番号が削除または置換されたことを確認するには、**show bgp neighbors** コマンドおよび **show bgp update-group** コマンドを使用します。

選択的 VRF ダウンロード

選択的 VRF ダウンロード（SVD）機能を使用すると、ラインカード経由でのトラフィックの転送に必要なラインカードに、これらのプレフィックスおよびラベルだけをダウンロードできるようになります。

統合エッジ MSE プラットフォームにおける要件を満たすために、VRF の数、VRF インターフェイスの数、およびプレフィックス容量が増大しています。コンバージェンスのタイミングは、ラインカードのエンジンによって異なります。コンバージェンスのタイミングを決定する重要な要因の 1 つが、プレフィックスとそれに関連付けられたデータ構造を操作およびプログラムするのにかかる時間です。プレフィックスとラベルの数が少ないほど、コンバージェンスのタイミングが向上します。VRF ルートの選択的ダウンロードを有効にすると、SVD ではレイヤ 3 VPN（L3VPN）のスケーラビリティが高くなり、コンバージェンスの問題が緩和されます。

選択的 VRF ダウンロードでのラインカードのロールとフィルタ

選択的 VRF ダウンロード（SVD）コンテキストでは、ラインカードに次のロールがあります。

- コア LC：コアに接するインターフェイス（他の P/PE に接続するインターフェイス）のみを持つラインカード
- カスタマー LC：カスタマーに接するインターフェイス（異なる VRF の CE に接続するインターフェイス）を 1 つ以上持つラインカード

ラインカードでは、次のプレフィックスを処理します。

- ローカル プレフィックス：設定された VRF コンテキスト内のルータに接続されている CE から受信するプレフィックス
- リモート プレフィックス：別の PE から受信され、設定されている VRF にインポートされたプレフィックス

これらのフィルタは、ラインカード タイプごとに適用できます。

- ラベルや IP フォワーディングを正しく設定できるように、コア LC には、すべてのローカル プレフィックスおよび VRF ラベルが必要です。
- カスタマー LC には、接続されているすべての VRF と、接続されている VRF に依存関係がある他の VRF に対するローカルおよびリモートのプレフィックスが必要です。これはインポートおよびエクスポートの RT コンフィギュレーションに基づきます。VRF「A」に VRF「B」からインポートされたルートがある場合、VRF「A」のインポートされたルートは、VRF「B」にあるネクスト ホップを指します。ルート解決のためには、VRF「A」インターフェイスを持つ各ラインカードに VRF「B」ルートをダウンロードする必要があります。
- ラインカードにコアに接するインターフェイスとカスタマーに接するインターフェイスの両方がある場合は、フィルタリングを実行する必要はありません。このようなラインカードには、すべてのテーブルとすべてのルートがあります。これらのラインカードには「標準」というロールがあります。すべての RP および DRP は、標準ロールがあります。
- L3VPN のルートを正しく解決するために、すべてのノードに IPv4 のデフォルト テーブルがある必要があります。ただし、ラインカードに IPv6 インターフェイスがない場合は、すべての IPv6 テーブルとルートをフィルタで除外できます。このような場合、このラインカードは IPv6 AFI に「関係していない」と見なすことができます。この後、このラインカードは IPv6 をサポートしていないように動作します。

選択的 VRF ダウンロードの無効化

デフォルトでは、選択的 VRF ダウンロード（SVD）機能は無効になっています。SVD を有効にするには、**svd platform enable** コマンドを管理コンフィギュレーションモードで設定し、**reload location all** コマンドを使用してシャーシをリロードします。すでに有効になっている SVD を無効にするには、**no svd platform enable** コマンドを使用し、**reload location all** コマンドでシャーシをリロードします。

SVD の使用または不使用によるラインカードにダウンロードされたルートの計算

選択的 VRF ダウンロードオプションを使用したか、または使用しなかった場合にラインカードにダウンロードされるルートの数は、次に示すラインカードタイプ別にダウンロードされたテーブルとルートの総数に従って計算できます。

次の表に、各 SVD カードタイプのラインカードにダウンロードされたルートとテーブルの総数をまとめます。SVD なしの行の数値の差異によって、節減数を計算できます。

表 *2:* ラインカードタイプ別にダウンロードされたテーブルとルートの総数

カード タイプ	ダウンロードされたテーブル	ダウンロードされたルート
カスタマー	(o+Y)	(o+Y)R
コア	n	nxR
SVD なし	n	nR

- n は、存在する VRF の合計数です。
- o は、カード上で直接プロビジョニング/設定された VRF の数です（n は o 以上）。
- R は VRF ごとのルートの数です。
- x は、SVD ローカルとルート総数の比率です。
- Y は、直接プロビジョニングされた VRF（o）に依存する VRF の数です（Y は 0 以上）。

次に、計算の例を示します。

顧客はシステムに 100 の VRF を設定していて、ラインカードは 5 枚使用しています。IPv4 アドレスファミリの場合、4 枚のラインカードが同等の VRF 分布でカスタマー向けに動作していますが、1 枚はコア向けです。テーブル間の依存関係は存在しません。この例では、n = 100、o = 25、x = 3/10、Y = 0、R = 1000 となっています。

ダウンロードされたルートの数は次のとおりです。

- SVD なし：(nR) = 100,000
- カスタマー向けカード：(o + Y) R = 25,000
- コア向けカード：(nxR) = 30,000

この例では、SVD 機能によって 70 %近く削減されています。

存在する VRF の総数（n）は、RSP カード上で **show cef tables summary location** *node-id* コマンドを使用して検出できます。

```
RP/0/RSP0/cpu 0: router#show cef tables summary location 0/rsp0/cpu0

Role change timestamp     : Apr  3 07:21:46.759
Current Role              : Core
No. of times Eod received : 2
Eod received              : Apr  3 07:21:46.980

No. of Tables                 :       106
No. of Converged Tables       :       106
No. of Deleted Tables         :         0
No. of Bcdl Subscribed Tables :       106
No. of Marked Tables          :         0
```

ラインカード上でプロビジョニングされている VRF の数（o）は、**show cef tables summary location 0/0/cpu0**の「No. Of Tables」フィールドから導出されます。これにより、ラインカード 0/0/cpu0 に固有のテーブルが提供されます。

VRF あたりのルート（R）は、**show cef tables location** *node-id* コマンドを使用して検出できます。

```
RP/0/RSP0/cpu 0: router#show cef tables location 0/1/CPU0
Sat Apr  6 01:22:32.471 UTC

Codes:   L - SVD Local Routes, R - SVD Remote Routes
         T - Total Routes
         C - Table Converged,  D - Table Deleted
         M - Table Marked,     S - Table Subscribed

Table           Table ID             L          R           T C D M S
default         0xe0000000           9          3          23 Y N N Y
**nVSatellite   0xe0000010           1          0           6 Y N N Y
cdn             0xe0000011           0          0           5 Y N N Y
oir             0xe0000012           0          0           5 Y N N Y
vrf1            0xe0000013           3          1          11 Y N N Y
```

VRF 「vrf1」の場合、合計ルートは「T」列 (11) になります。そのため、VRF ごとのルートの数がすべての VRF と同じでなかった場合は、「デフォルト以外の VRF のルート」の総数を計算し、VRF の数で割って、VRF ごとの平均ルート数に達するようにする必要があります。

SVD ローカルの比率：ルートの総数（x）は、SVD ローカルルートの数と特定の VRF のルートの総数を使用して検出できます。たとえば、前出の **show cef tables location 0/1/CPU0**の出力例では、L 列の数字はローカルルートの数、T 列の数字はその VRF のルートの総数を表しています。したがって、L 列と T 列の数値の比率によって、特定の VRF の比率が得られます。この比率がすべての VRF で同じでない場合は、すべての VRF で平均化する必要があります。

直接プロビジョニングされた VRF（Y）に依存する VRF の数は、ルータの設定によって異なるため、手動で計算する必要があります。たとえば、ルートインポートが、他の VRF によってエクスポートされたルートから依存型の VRF インポートをターゲットにしている場合などです。VRF は、直接プロビジョニングされる他の何らかの VRF に存在するネクストホップに依存している場合に、依存型になります。Y を自動的に計算する show コマンドはありません。これは、さまざまな VRF にルートをインポートするためのルートの設定方法に完全に依存しているためです。

BGP Accept Own

BGP Accept Own 機能を使用すると、自動送信 VPN ルート（BGP スピーカーがルートリフレクタ（RR）から受信するルート）を処理できるようになります。「自動送信」ルートは、スピーカー自体によって最初にアドバタイズされたルートです。BGP プロトコル（RFC4271）に従って、BGP スピーカーは、スピーカー自体によって送信されたアドバタイズメントを拒否します。ただし、BGP Accept Own メカニズムを使用すると、プレフィックスの特定の属性を変更するルートリフレクタから反映された場合に、ルータは自身がアドバタイズしたプレフィックスを受け入れることが可能になります。ACCEPT-OWN と呼ばれる特別なコミュニティがルートリフレクタによってプレフィックスに付加されます。これは ORIGINATOR_ID および

NEXTHOP/MP_REACH_NLRI チェックをバイパスするための受信側ルータに対する信号です。通常、BGP スピーカーは自動送信されたプレフィックスを自動送信チェック（ORIGINATOR_ID、NEXTHOP/MP_REACH_NLRI）によって検出し、受信した更新をドロップします。ただし、更新に Accept Own コミュニティがあれば、BGP スピーカーはそのルートを処理します。

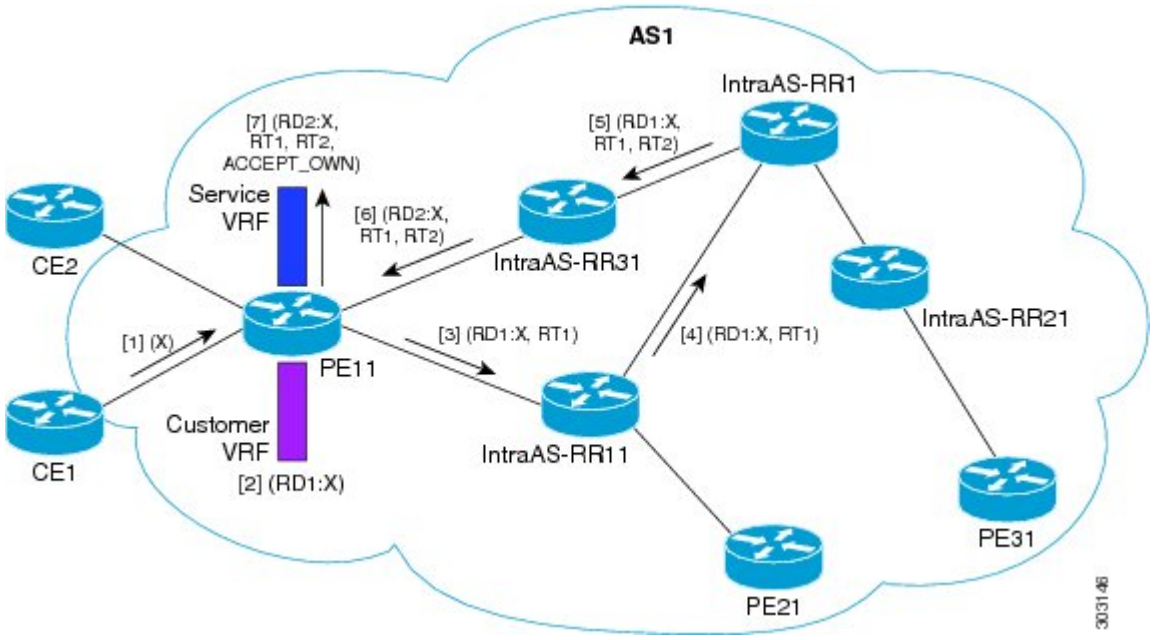
BGP Accept Own の応用例の 1 つは、MPLS VPN ネットワーク内のエクストラネットの自動設定です。エクストラネットの設定では、ある VRF にあるルートは同じ PE の別の VRF にインポートされます。通常、エクストラネットのメカニズムでは、別の VRF からのプレフィックスのインポートを制御するために、エクストラネット VRF のインポート RT またはインポートポリシーを編集する必要があります。ただし、Accept Own 機能を使用すると、ルート リフレクタは、PE で設定変更することなく、その制御をアサートできます。このように Accept Own 機能によって、異なる VRF 間でのルートのインポートの制御を集中管理できます。

BGP Accept Own 機能は、ネイバー コンフィギュレーション モードの VPNv4 および VPNv6 アドレス ファミリ向けにのみサポートされています。

Accept Own コミュニティと RT を処理するルートリフレクタ

ACCEPT_OWN コミュニティは、InterAS ルートリフレクタ（InterAS-RR）によってアウトバウンド ルート ポリシーを使用して発信されます。ACCEPT_OWN のコミュニティ属性を持つプレフィックスの伝搬を最小限に抑えるために、この属性は送信元 PE に対するアウトバウンド ルート ポリシーを使用して InterAS-RR に付加されます。InterAS-RR は、ACCEPT-OWN コミュニティを追加して RT を変更した後、仲介 RR を通じて新しい Accept Own ルートを、接続されている PE（送信元など）に送信します。ルートは、ルート ポリシーによって変更されます。

Accept Own の設定例



この設定例の内容は次のとおりです。

- PE11 にカスタマー VRF とサービス VRF が設定されています。
- OSPF は IGP として使用されます。
- VPNv4 ユニキャストおよび VPNv6 ユニキャストのアドレス ファミリが PE ネイバーと RR ネイバーとの間でイネーブルになっており、IPv4 および IPv6 が PE ネイバーと CE ネイバーとの間でイネーブルになっています。

Accept Own の設定は次のように動作します。

1. CE1 がプレフィックス X を発信します。
2. プレフィックス X は、カスタマー VRF に (RD1:X) として設定されています。
3. プレフィックス X は IntraAS-RR11 に (RD1:X, RT1) としてアドバタイズされます。
4. IntraAS-RR11 が InterAS-RR1 に X を (RD1:X, RT1) としてアドバタイズします。
5. InterAS-RR1 はインバウンドのプレフィックス X とアウトバウンドの ACCEPT_OWN コミュニティに RT2 を付加し、IntraAS-RR31 にプレフィックス X をアドバタイズします。
6. IntraAS-RR31 が PE11 に X をアドバタイズします。
7. PE11 は X をサービス VRF に (RD2:X,RT1, RT2, ACCEPT_OWN) としてインストールします。

リモート PE：Accept Own ルートの処理

リモート PE（送信元 PE 以外の PE）は、すべての同等ルート間の最適パスを計算します。この最適パス アルゴリズムは、Accept Own パスが Accept Own でないパスよりも優先されるように変更されています。最適パスの比較は IGP メトリックの比較の直前に実行されます。リモート PE がルートリフレクタ 1 から Accept Own パスを受信し、ルートリフレクタ 2 から Accept Own でないパスを受信し、これらのパスが同一であった場合は、Accept Own パスが優先されます。そのためインポートは Accept Own パスで実行されます。

不等コストの連続ロードバランシングに対する BGP DMZ リンク帯域幅

不等コストの連続ロードバランシングに対するボーダー ゲートウェイ プロトコル非武装地帯（BGP DMZ）リンク帯域幅により、BGP DMZ リンク帯域幅を使用して、ローカル ノード上で連続プレフィックスに対する不等コスト ロードバランシングをサポートできます。不均等ロードバランシングは、BGP ネイバー コンフィギュレーション モードの **dmz-link-bandwidth** コマンドと、インターフェイス コンフィギュレーション モードの **bandwidth** コマンドを使用して実行します。

BGP の BFD マルチホップ サポート

BGP では、双方向フォワーディング検出マルチホップ（BFD-MH）のサポートが有効になっています。BFD マルチホップでは複数のネットワーク ホップにまたがることのある 2 つのアドレス間に BFD セッションを確立します。Cisco IOS XR ソフトウェア BFD マルチホップは RFC

5883 に基づきます。BFD マルチホップの詳細については、 *Interface and Hardware Component Configuration Guide for Cisco ASR 9000 Series Routers*および *Interface and Hardware Component Command Reference for Cisco ASR 9000 Series Routers*を参照してください。

BGP Multi-Instance および Multi-AS

自律システム（AS）に対応するルータでは、複数の BGP インスタンスがサポートされています。各 BGP インスタンスは、同じまたは異なる RP/DRP ノードで実行される独立したプロセスです。BGP インスタンス間ではプレフィックス テーブルは共有されません。分散 BGP と同様に、共通 adj-rib-in（bRIB）は不要です。BGP インスタンスは互いに通信することはなく、また互いにピアリングを設定することもありません。個々のインスタンスは他のルータとのピアリングを独立して設定できます。

Multi-AS BGP を使用すると、Multi-Instance BGP の各インスタンスに異なる AS 番号を設定できるようになります。

Multi-Instance および Multi-AS BGP は次の機能を備えています。

- 共通ルーティング インフラストラクチャを使用して、複数のルータによって提供されるサービスを単一の IOS-XR ルータに統合するメカニズム。
- 異なる BGP インスタンスに異なる AF を設定することにより、AF の分離を実現するメカニズム。
- 複数のインスタンス間でピアリング セッション全体を分散させることによって、セッションのスケールを高めることができる手段。
- 個々のインスタンスに異なる BGP テーブルを伝送させることにより、プレフィックスのスケール（特に RR で）を高めることができるメカニズム。
- 特定の状況における BGP コンバージェンスの改善。
- NSR を含むすべての BGP 機能は、すべてのインスタンスに対応しています。
- ロードおよびコミット ルータ レベルの操作は、以前に確認または適用された構成上で実行できます。

制約事項

- ルータは最大 4 つの BGP インスタンスをサポートします。
- 各 BGP インスタンスには、固有の ルータ ID が必要です。
- 各 BGP インスタンスで設定できるアドレス ファミリは 1 つだけです（VPNv4、VPNv6 および RT 制約は複数の BGP インスタンスで設定できます）。
- IPv4/IPv6 ユニキャストは、IPv4/IPv6 ラベル付きユニキャストが設定されている同じ BGP インスタンス内にある必要があります。
- IPv4/IPv6 マルチキャストは、IPv4/IPv6 ユニキャストが設定されている同じ BGP インスタンス内にある必要があります。

- 単一の BGP インスタンスに対するすべての設定変更を同時にコミットすることができます。ただし、複数のインスタンスに対する設定変更は同時にコミットできません。
- 同じリモート ルータとのピアリング時に、BGP の update-source をすべてのインスタンスのデフォルト VRF で一意にすることが推奨されます。

RPKI に基づく BGP プレフィックスの発信元検証

BGP ルートは、BGP アナウンスメントの形で、プレフィックスが経由したドメイン間パスを識別する自律システム（AS）の設定と、アドレス プレフィックスを関連付けます。この設定は、BGP 内で AS_PATH 属性として表され、プレフィックスを発信した AS で開始されます。

誤ったプレフィックスのアナウンス、中間者攻撃など、BGP に対する既知の脅威を低減しやすくするためのセキュリティ要件の 1 つは、BGP ルートの発信元 AS を検証する能力です。アドレス プレフィックスの発信元であるとする AS 番号（BGP ルートの AS_PATH 属性から導出）は、プレフィックスの所有者によって検証および許可される必要があります。

Resource Public Key Infrastructure（RPKI）は、IP アドレスとリソースとしての AS 番号の公的で検証可能なデータベースを構築するためのアプローチです。RPKI は、BGP（インターネット）プレフィックスから許可された元の AS 番号への情報マッピングなどの情報を含む、グローバルに分散されたデータベースです。BGP を実行しているルータは、RPKI に接続して、BGP パスの元の AS を検証できます。

BGP RPKI の送信元バインド機能を使用すると、RPKI サーバ接続に使用する送信元の IP アドレスとインターフェイスを指定できます。たとえば、この機能では、ループバック インターフェイスから送信元となる RPKI セッションを設定できます。

BGP の origin-as 検証はデフォルトで有効になっています。

RPKI キャッシュ サーバの設定

リソース公開キー インフラストラクチャ（RPKI）キャッシュ サーバ パラメータを設定するには、次の作業を実行します。

RPKI サーバのコンフィギュレーション モードで RPKI キャッシュ サーバ パラメータを設定します。RPKI サーバ コンフィギュレーション モードを開始するには、ルータ BGP コンフィギュレーション モードで **rpki server** コマンドを使用します。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **rpki server** {*host-name* | *ip-address*}
4. **bind-source interface** *name*
5. 次のいずれかのコマンドを使用します。
   - **transport ssh port** *port_number*
   - **transport tcp port** *port_number*
6. （任意） **username** *user_name*

7. （任意） **password** *password*
8. **preference** *preference_value*
9. **purge-time** *time*
10. 次のいずれかのコマンドを使用します。
    - **refresh-time** *time*
    - **refresh-time off**
11. 次のいずれかのコマンドを使用します。
    - **response-time** *time*
    - **response-time off**
12. **commit**
13. （任意） **shutdown**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)#router bgp 100`	BGP AS 番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**rpki server** {*host-name* | *ip-address*} 例： `RP/0/RSP0/cpu 0: router(config-bgp)#rpki server 10.2.3.4`	RPKI サーバのコンフィギュレーション モードを開始し、RPKI のキャッシュ パラメータを設定します。
ステップ 4	**bind-source interface** *name* 例： `Router#(config-bgp)# bind-source interface Loopback2`	RPKI サーバ接続に使用する送信元インターフェイスとしてループバック インターフェイスを指定します。

ステップ 5	次のいずれかのコマンドを使用します。
	• **transport ssh port** *port_number*
	• **transport tcp port** *port_number*
	例：
	`RP/0/RSP0/cpu 0: router(config-bgp-rpki-server)#transport ssh port 22`
	または
	`RP/0/RSP0/cpu 0: router(config-bgp-rpki-server)#transport tcp port 2`
		RPKI キャッシュの転送方法を指定します。
		• **ssh**：SSH を使用して RPKI キャッシュに接続するには **ssh** を選択します。
		• **tcp**：TCP（暗号化されていない）を使用して RPKI キャッシュに接続するには **tcp** を選択します。
		• **port** *port_number*：指定した RPKI キャッシュ転送に使用するポート番号を指定します。TCP の場合、サポートされているポート番号の範囲は 1 ~ 65535 です。SSH の場合は、ポート番号 22 を使用します。

	コマンドまたはアクション	目的
		（注） SSH を介した RPKI キャッシュ転送の場合は、カスタム ポート番号を指定しないでください。SSH を介した RPKI にはポート 22 を使用する必要があります。 （注） transport には TCP と SSH のいずれかを設定できます。transport を変更すると、キャッシュ セッションがフラップします。

ステップ 6	（任意） **username** *user_name*
	例：
	`RP/0/RSP0/cpu 0:`
	`router(config-bgp-rpki-server)#username ssh_rpki_uname`
		RPKI キャッシュ サーバの（SSH）ユーザ名を指定します。

ステップ 7	（任意） **password** *password*
	例：
	`RP/0/RSP0/cpu 0:`
	`router(config-bgp-rpki-server)#password ssh_rpki_pass`
		RPKI キャッシュ サーバの（SSH）パスワードを指定します。
		（注） 「username」と「password」の設定は、SSH 転送方式がアクティブな場合にのみ適用されます。

ステップ 8	**preference** *preference_value*
	例：
	`RP/0/RSP0/cpu 0:`
	`router(config-bgp-rpki-server)#preference 1`
		RPKI キャッシュのプリファレンス値を指定します。プリファレンス値の範囲は 1 ～ 10 です。設定するプリファレンス値は低い方が適切です。

ステップ 9	**purge-time** *time*
	例：
	`RP/0/RSP0/cpu 0:`
	`router(config-bgp-rpki-server)#purge-time 30`
		キャッシュ セッションのドロップ後に、BGP がキャッシュからのルートを保持するまで待機する時間を設定します。破棄時間は秒単位で設定します。破棄時間の範囲は 30 ～ 360 秒です。

ステップ 10	次のいずれかのコマンドを使用します。
	• **refresh-time** *time*
	• **refresh-time off**
	例：
	`RP/0/RSP0/cpu 0:`
	`router(config-bgp-rpki-server)#refresh-time 20`
	または
	`RP/0/RSP0/cpu 0:`
	`router(config-bgp-rpki-server)#refresh-time off`
		キャッシュへの定期的なシリアル クエリー送信操作の間に BGP が待機する時間を設定します。リフレッシュの時間を秒単位で設定します。リフレッシュの時間の範囲は 15 ～ 3600 秒です。
		シリアルクエリーを定期的に送信しないように指定するには、**off** オプションを設定します。

	コマンドまたはアクション	目的

ステップ 11	次のいずれかのコマンドを使用します。
	• **response-time** *time*
	• **response-time off**
	例：
	`RP/0/RSP0/cpu 0:`
	`router(config-bgp-rpki-server)#response-time 30`
	または
	`RP/0/RSP0/cpu 0:`
	`router(config-bgp-rpki-server)#response-time off`
		シリアルまたはリセットのクエリーを送信した後に BGP が応答を待機する時間を設定します。応答時間を秒の単位で設定します。応答時間の範囲は 15 ～ 3600 秒です。
		応答を無期限に待機するには、**off** オプションを設定します。

ステップ 12	**commit**	

ステップ 13	（任意） **shutdown**
	例：
	`RP/0/RSP0/cpu 0:`
	`router(config-bgp-rpki-server)#shutdown`
		RPKIキャッシュのシャットダウンを設定します。

RPKI 最適パス計算の設定

RPKI 最適パス計算オプションを設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **bgp bestpath origin-as use validity**
4. **bgp bestpath origin-as allow invalid**
5. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)#router bgp 100`	BGP AS 番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGPルーティング プロセスを設定できます。
ステップ 3	**bgp bestpath origin-as use validity** 例： `RP/0/RSP0/cpu 0: router(config-bgp)#bgp bestpath origin-as use validity`	BGPベストパス処理でのパスのプリファレンスに影響するBGPパスの有効性状態を有効にします。この設定は、ルータ BGP アドレス ファミリ サブモードでも設定できます。

	コマンドまたはアクション	目的
ステップ 4	**bgp bestpath origin-as allow invalid** 例： `RP/0/RSP0/cpu 0: router(config-bgp)#bgp bestpath origin-as allow invalid`	すべての「無効な」パスをBGP最適パス計算対象にします。 （注） この設定はグローバルアドレスファミリ、ネイバー、およびネイバー アドレス ファミリの各サブモードでも設定できます。ルータ BGP とアドレスファミリサブモードで bgp bestpath origin-as allow invalid を設定すると、すべての「無効な」パスがBGP最適パス計算対象になります。デフォルトではこのようなパスは最適パス候補になりません。ネイバーまたはネイバー アドレスファミリ サブモードで bgp bestpath origin-as を設定すると、その特定のネイバーまたはネイバーアドレスファミリのすべての「無効な」パスがベストパス候補として見なされます。このネイバーは eBGP ネイバーでなければなりません。 この設定は、**bgp bestpath origin-as use validity** 設定がイネーブルの場合にのみ有効になります。
ステップ 5	**commit**	

グローバル プレフィックス用 BGP 3107 PIC アップデート

グローバル プレフィックス用 BGP 3107 PIC アップデート機能は、MPLS VPN プロバイダーネットワークでの、グローバル IPv4 および IPv6 プレフィックス用のプレフィックス独立コンバージェンス（PIC）アップデートをサポートします。この機能は、BGP を使用した、グローバルな IPv4 または IPv6 のプレフィックス用の MPLS ラベルの配布について記述した、RFC 3107 に基づいています。これにより、IGP の拡張性が向上され、高速コンバージェンスのための PIC アップデートも実現されます。

RFC 3107 により、ルートおよびラベルを BGP で伝送できるようになります。特定のルートへの配布に BGP を使用する場合は、このルートにマッピングされている MPLS ラベルの配布にも使用できます。特定の 1 つのルートに対するラベルマッピング情報は、ルート自体の配布に使用される、同じ BGP アップデート メッセージに同梱されます。RFC 3107 では、OSPF からネクスト ホップ ループをフィルタリングでき、LDP によってアドバタイズされるラベルを削減できます。この実装によって、OSPF および LDP データベースが大幅に削減されます。

3107 PIC 実装では、additional-path 設定を含めて、次のアドレス ファミリをサポートしています。

- address-family ipv4 unicast
- address-family ipv6 unicast

- address-family vpnv4 unicast
- address-family vpnv6 unicast

（注） address-family l2vpn vpls-vpws では、additional-path をサポートしていません。したがって、address-family l2vpn vpls-vpws を使用する L2VPN サービスでは、PIC コンバージェンス時間が保証されません。

3107 PIC 実装では、次の Cisco IOS XR 機能をサポートしています。

- 3107 の PIC エッジ
- Traffic Engineering Fast-reroute（TE FRR）：逐語的なトンネルを使用して、コア リンク障害に対する 50 ミリ秒以内のトラフィック コンバージェンスが保証されます。
- L2VPN サービス（VPWS）
- L3VPN VPNv4 サービス
- 6 PE サービス
- 6 VPE サービス
- VPLS サービス

グローバル プレフィックス用 BGP 3107 PIC アップデート実装では、Light-Weight Recursive（LW-RLDI）オブジェクトの代わりに共有 Recursive Load Info（RLDI）転送オブジェクトを使用します。RLDI は複数リーフ間で共有されますが、LW-RLDI はリーフごとにインスタンス化されます。処理がプレフィックスに依存しなくなるため、共有は PIC アップデートの処理で有用です。

RIB および FIB 用 BGP プレフィックス独立コンバージェンス

RIB および FIB 用 BGP PIC によって、PE-CE としてのスタティック再帰のサポートおよび Fast Re-Route トリガーを使用したバックアップ アクティベーションの高速化のサポートが加わります。

RIB および FIB 用 BGP PIC 機能では、次の要素をサポートしています。

- コンバージェンス時間をさらに削減する、高速な PE-CE リンク ダウン検出用の FRR に似たトリガー（PIC エッジの高速アクティベーション）。
- スタティック再帰ルートのための PIC エッジ。
- 明示的に /32 スタティック ルートを設定しない、PIC エッジのための BFD シングルホップ トリガー。
- 第 1（IGP）レベルでの失敗トリガーの際の第 3 レベル以降での再帰 PIC アクティベーション。

- BGP ネクスト ホップの解決に関して、FIB が BGP と同期していることを保証する、FIB での BGP パス再帰制約。

BGP PIC エッジが設定されている場合、**neighbor shutdown** コマンドを設定しても、バックアップパスに切り替える CEF はトリガーされません。代わりに、BGP はルーティングテーブルの最上位プレフィックスから最後まで 1 つずつ CEF のフィードを再開するため、時間遅延が発生します。

⚠

注意 この時間遅延によって、ネットワーク内でブラックホールが発生します。回避策として、**neighbor shutdown** コマンドを設定する前に、トラフィックをバックアップパスに手動でルーティングしておく必要があります。

BGP アップデート メッセージのエラー処理

BGP アップデート メッセージのエラー処理によって、セッションのリセットを避けるためにエラー アップデート メッセージの処理における BGP の動作が変わります。IETF IDR *I-D:draft-ietf-idr-error-handling* で説明されているアプローチに基づいて、Cisco IOS XR BGP アップデート メッセージのエラー処理を実装すると、重大度、更新エラーが発生する可能性、属性のタイプなどの要素に基づいて、BGP 更新エラーはさまざまなカテゴリに分類されます。各カテゴリで発生したエラーは、ドラフトに沿って処理されます。セッションのリセットは、エラーの処理プロセス中は可能な限り回避されます。一部のカテゴリのエラー処理は、デフォルトの動作を有効または無効にする設定コマンドによって制御されます。

基本の BGP 仕様に応じて、不正な属性を含むアップデート メッセージを受信した BGP スピーカは、不正な属性が受信されたセッションをリセットする必要があります。セッションのリセットは、不正な属性があるルートだけでなくセッションを介して交換される他の有効なルートにも影響するので、この動作は好ましくありません。

BGP 属性のフィルタリング

BGP 属性フィルタ機能によって、BGP アップデート メッセージ内の BGP アップデートの整合性を確認し、無効な属性を検出したときには最適な対応を行います。BGP アップデート メッセージには、必須およびオプションの属性のリストが含まれています。アップデート メッセージのこれらの属性には、MED、LOCAL_PREF、COMMUNITY などが含まれています。属性の形式が正しくない場合は、ルータの受信側でこれらの属性をフィルタ処理する必要があります。BGP 属性フィルタ機能では、着信アップデート メッセージで受信した属性をフィルタリングします。属性フィルタは、受信側ルータで好ましくない動作を引き起こす可能性のある属性を排除するためにも使用できます。

BGP アップデートの中には、ネットワーク層到達可能性情報（NLRI）またはアップデート メッセージ内の他のフィールドなどの誤った形式の属性のために、形式が不正になるものがあります。これらの不正なアップデートを受信すると、受信側ルータで好ましくない動作が発生します。このような不正な動作は、アップデート メッセージの解析時や、受信した NLRI の再

アドバタイズ時に発生することがあります。このような場合に備えて、受信側でこれらの破損した属性をフィルタ処理することが重要です。

BGP 属性のフィルタリングのアクション

属性フィルタリングを設定するには、1 つまたはある範囲の属性コードと対応するアクションを指定します。実行できるアクションには次のものがあります。

- Treat-as-withdraw：対応する IPv4 ユニキャストまたは MP_REACH NLRI があれば、ネイバーの Adj-RIB-In から取り消します。
- Discard Attribute：一致した部分の属性は廃棄され、アップデート メッセージの残りの部分は正常に処理されます。

受信したアップデート メッセージに 1 つ以上のフィルタされた属性が含まれている場合、メッセージに対して設定されたアクションが実行されます。オプションで、さらに詳細なデバッグを行うためにアップデート メッセージを保存して、コンソールに syslog メッセージを表示することもできます。

属性がフィルタと一致した場合は、属性のその後の処理は停止され、対応するアクションが実行されます。

属性フィルタ グループ コマンド モードを開始するには、**attribute-filter group** コマンドを使用します。属性を破棄または更新メッセージを「取り消し」アクションとして処理するには、属性フィルタ グループ コマンド モードで **attribute** コマンドを使用します。

BGP のエラー処理と属性フィルタリングの syslog メッセージ

不正な形式のアップデート パケットをルータが受信すると、ROUTING-BGP-3-MALFORM_UPDATE タイプの ios_msg がコンソールに出力されます。このレートは、すべてのネイバーで 1 分間に 1 つのメッセージになるよう制限されています。不正なパケットが「Discard Attribute」（A5）または「Local Repair」（A6）アクションの対象になった場合は、ネイバー 1 つおよびアクション 1 つごとに ios_msg メッセージが出力されます。これは、ネイバーが直前の「Established」状態に到達して以降に受信した不正な形式のアップデートの数とは関係ありません。

BGP エラー処理の syslog メッセージの例を次に示します。

```
%ROUTING-BGP-3-MALFORM_UPDATE : Malformed UPDATE message received from neighbor 13.0.3.50
 - message length 90 bytes,
 error flags 0x00000840, action taken "TreatAsWithdraw".
Error details: "Error 0x00000800, Field "Attr-missing", Attribute 1 (Flags 0x00, Length
 0), Data []"
```

これは「Discard Attribute」アクションに対する BGP 属性フィルタリングの syslog メッセージの例です。

```
[4843.46]RP/0/0/CPU0:Aug 21 17:06:17.919 : bgp[1037]: %ROUTING-BGP-5-UPDATE_FILTERED :
One or more attributes were filtered from UPDATE message received from neighbor 40.0.101.1
 - message length 173 bytes,
```

```
 action taken "DiscardAttr".
 Filtering details: "Attribute 16 (Flags 0xc0): Action "DiscardAttr"". NLRIs: [IPv4
Unicast] 88.2.0.0/17
```

これは「Treat-as-withdraw」アクションに対する BGP 属性フィルタリングの syslog メッセージの例です。

```
[391.01]RP/0/0/CPU0:Aug 20 19:41:29.243 : bgp[1037]: %ROUTING-BGP-5-UPDATE_FILTERED :
 One or more attributes were filtered from UPDATE message received from neighbor 40.0.101.1
 - message length 166 bytes,
 action taken "TreatAsWdr".
 Filtering details: "Attribute 4 (Flags 0xc0): Action "TreatAsWdr"". NLRIs: [IPv4 Unicast]
 88.2.0.0/17
```

BGP リンクステート

BGP リンクステート（LS）は、BGP を介して内部ゲートウェイ プロトコル（IGP）リンクステートデータベースを伝えるために定義されたアドレス ファミリ識別子（AFI）およびサブアドレス ファミリ識別子（SAFI）です。BGP LS は、ネットワーク トポロジ情報をトポロジ サーバおよびアプリケーション層トラフィック最適化（ALTO）サーバに提供します。BGP LS では、集約、情報の非表示、および抽象化に対するポリシー ベースの制御が可能です。BGP LS は、IS-IS および OSPFv2 をサポートしています。

（注） IGP は、リモート ピアからの BGP LS データを使用しません。BGP は、ルータの他のコンポーネントに受信した BGP LS データをダウンロードしません。

BGP パーマネント ネットワーク

BGP パーマネント ネットワーク機能は、BGP 経由のスタティック ルーティングをサポートしています。（ルート ポリシーで識別された）IPv4 または IPv6 宛先への BGP ルートは、管理用に作成して、BGP ピアに選択的にアドバタイズできます。これらのルートは、管理上削除されるまでルーティング テーブルに残ります。

パーマネント ネットワークは、プレフィックスのセットを永続的なものとして定義するために使用されます。つまり、プレフィックスのセットのアップストリームにおいて BGP のアドバタイズメントまたは取り消しは 1 回しかありません。プレフィックス セットの各ネットワークに対し、BGP 固定パスが作成され、優先度はそのピアから受信される他の BGP パスよりも低く扱われます。BGP 固定パスが最適パスである場合は RIB にダウンロードされます。

グローバル アドレス ファミリ コンフィギュレーション モードの **permanent-network** コマンドは、ルートポリシーを使用して固定パスが設定されるプレフィックス（ネットワーク）のセットを識別します。ネイバー アドレスファミリ コンフィギュレーション モードの **advertise permanent-network** コマンドは、固定パスをアドバタイズする必要があるピアの識別に使用されます。別の最適パスが使用可能であっても、固定パスは常にアドバタイズパーマネントネッ

トワーク設定を持つピアにアドバタイズされます。固定パスは、固定パスを受信するように設定されていないピアにはアドバタイズされません。

パーマネント ネットワーク機能は、デフォルトの仮想ルーティングおよび転送（VRF）下の IPv4 ユニキャストおよび IPv6 ユニキャスト アドレス ファミリ内のプレフィックスのみをサポートします。

制約事項

次の制限は、パーマネント ネットワークの設定時に適用されます。

- パーマネント ネットワーク プレフィックスは、グローバル アドレス ファミリでルート ポリシーによって指定する必要があります。
- グローバル アドレス ファミリ コンフィギュレーション モードでルート ポリシーを使用してパーマネント ネットワークを構成し、それをネイバー アドレス ファミリ コンフィギュレーション モードで設定する必要があります。
- パーマネント ネットワーク設定を削除する場合は、ネイバー アドレス ファミリ コンフィギュレーション モードの設定を削除してから、グローバル アドレス ファミリ コンフィギュレーション モードから削除します。

アップデート生成のための BGP-RIB のフィードバック メカニズム

アップデート生成機能のためのボーダー ゲートウェイ プロトコル ルーティング情報ベース（BGP-RIB）のフィードバック メカニズムによって、ネットワークで不完全なルート アドバタイズメントが行われて、それによってパケット損失が発生するのを防ぐことができます。このメカニズムによって、ルートがネイバーにアドバタイズされる前にローカルに組み込まれるようになります。

BGP は RIB からのフィードバックを待ちます。このフィードバックには、BGP によって RIB に組み込まれたルートが、BGP がネイバーにアップデートを送信する前に転送情報ベース（FIB）に組み込まれたことが示されています。RIB は BCDL のフィードバック メカニズムを使用して、そのバージョンのルートが FIB によって使用されたかを判断し、BGP をそのバージョンで更新します。BGP がアップデートを送信するのは、FIB が組み込んだバージョン以下のバージョンのルートだけです。この選択的な更新によって、BGP が不完全なアップデートを送信しないようになり、ルータのリロード、LC OIR、または代替パスが使用可能になるリンク フラップ後にデータ プレーンがプログラミングされる前であっても、トラフィックの引き込みが行われるようになります。

BGP が RIB に組み込んだルートが FIB に組み込まれたことを示す RIB からのフィードバックを BGP が待機し、その後で BGP がネイバーにアップデートを送信するように設定するには、ルータ アドレスファミリ IPv4 またはルータ アドレスファミリ VPNv4 コンフィギュレーション モードで **update wait-install** コマンドを使用します。**show bgp**、**show bgp neighbors**、および **show bgp process performance-statistics** コマンドを実行すると、update wait-install 設定の情報が表示されます。

BGP VRF ダイナミック ルートのリーク

Border Gateway Protocol（BGP）ダイナミック ルートのリーク機能では、デフォルトの VRF（グローバル VRF）とその他すべての非デフォルト VRF 間にルートをインポートできるようにし、グローバルと VPN ホスト間に接続を提供します。インポート プロセスによって VRF テーブルにインターネット ルートが組み込まれるか、またはインターネット テーブルに VRF ルートが組み込まれて、接続を提供します。

（注）

- 直接接続されたルートは、デフォルトの VRF から非デフォルトの VRF に BGP VRF ダイナミックルートリークを使用してリークできません。
- リークされたルートは、宛先 VRF 内のルートを対象にしたり、上書きしたりすることはできません。たとえば、2 台の接続されたルータ R1 （宛先 VRF 「dest-vrf」）と R2（送信元 VRF 「source-vrf」）を考えてみましょう。ルート CR-1 に接続された source-vrf が dest-vrf にリークされます。この場合、dest-vrf からのルートは、source-vrf からリークされたルート CR-1 の対象となるか、または上書きされます。

ダイナミック ルート リークは次の方法で有効になります。

- VRF アドレスファミリ コンフィギュレーション モードで **import from default-vrf route-policy** *route-policy-name* ［**advertise-as-vpn**］コマンドを使用して、デフォルト VRF から非デフォルト VRF にインポートする。

  advertise-as-vpn オプションが設定されている場合、デフォルト VRF から非デフォルト VRF にインポートしたパスは、PE と CE にアドバタイズされます。**advertise-as-vpn** オプションが設定されていない場合、デフォルト VRF から非デフォルト VRF にインポートされたパスは PE にアドバタイズされません。ただし、この場合も CE にはパスがアドバタイズされます。

- VRF アドレスファミリ コンフィギュレーション モードで **export to default-vrf route-policy** *route-policy-name* コマンドを使用して、非デフォルト VRF からデフォルト VRF にインポートする。

インポートしたルートをフィルタリングするには、ルート ポリシーが必要です。これにより、インターネット テーブルと VRF テーブル間でのルートの意図せぬインポートや対応するセキュリティ問題を低減します。

インポートできるプレフィックスの数にハードリミットはありません。インポートによりインポート先の VRF に新しいプレフィックスが作成されるため、プレフィックスとパスの総数が増加します。ただし、グルーバル ルートをインポートしている各 VRF がグローバル テーブルを受け取るネイバーと同等のワークロードを追加します。これは、ユーザが一部を除くすべてのプレフィックスをフィルタリングした場合も同様です。したがって、インポートする VRF の適切な数は 5 ～ 10 個です。

ユーザ定義の Martian チェック

Cisco IOS XR ソフトウェア リリース 5.1.0 では、IP アドレスプレフィックスが次のものである場合、Martian チェックを無効にできます。

- IPv4 アドレス プレフィックス
  - 0.0.0.0/8
  - 127.0.0.0/8
  - 224.0.0.0/4
- IPv6 アドレス プレフィックス
  - ::
  - ::0002 - ::ffff
  - ::ffff:a.b.c.d
  - fe80:xxxx
  - ffxx:xxxx

復元力のある CE 単位のラベルモード

復元力のある CE 単位のラベルは、CE 単位のラベルモードの拡張機能で、プレフィックス独立コンバージェンス（PIC）とロードバランシングをサポートします。

現時点では、プレフィックスごと、CE ごと、および VRF 単位の 3 つのラベルモードに次の制限があります。

- ASR 9000 イーサネット ライン カードと A9K-SIP-700 に対するサポートなし
- PIC に対するサポートなし
- 複数の CE にわたるロード バランシングに対するサポートなし
- PIC をサポートするローカル トラフィックの迂回時の一時的な転送ループ
- EIBGP マルチパスのロード バランシングに対するサポートなし
- 転送パフォーマンスへの影響
- ネットワーク内の別のベンダーのルータでのプレフィックス単位のラベルモードによるスケールの問題

復元力のある CE 単位のラベルスキームでは、CE パスまたはネクストホップのそれぞれの一意のセットに対して BGP が LSD に一意の書き換えラベルをインストールします。BGP テーブルにこのラベルをポイントする 1 つ以上のプレフィックスが含まれている場合があります。また、BGP は CE パス（プライマリ）と、オプションのバックアップ PE パスを RIB にインストールします。FIB は LSD からラベル書き換え情報を、RIB から IP パスを学習します。

安定状態では、弾力性のある CE ごとのラベル宛のラベル付きのトラフィックには、すべての CE ネクスト ホップにわたってロード バランシングが行われます。すべての CE パスが失敗すると、そのラベル宛のすべてのトラフィックが IP ルックアップとなり、使用可能な場合は、バックアップ PE に転送されます。このアクションはラベルをポイントする可能性があるプレフィックスの数と関係なく、ラベル上で実行されるため、プライマリ パスの障害時は PIC の動作になります。

BGP と OSPF での過剰なパントフロートラップの実装

BGP と OSPF での過剰なパントフロートラップ（EPFT） 機能は、制御パケットトラフィックの割り当てられた共有よりも多くのリモートデバイスからの制御パケットトラフィックを識別し、軽減しようとします。リモートデバイスは、送信元の MAC アドレスで識別されます。リモートデバイスがルータに対して制御パケットトラフィックを送信すると、そのルータの CPU を保護するために、制御パケットは Local Packet Transport Service（LPTS） キューによってパントされ、ポリシングされます。1 台のデバイスから過剰なレートの制御パケットトラフィックが送信される場合は、ポリサーキューがいっぱいになり、多くのパケットがドロップされます。1 台の「バッドアクター」デバイスからのレートが他のデバイスのレートを大幅に超えている場合、他のデバイスのほとんどはルータまでの制御パケットをまったく取得できません。過剰なパントフロートラップ機能は、この状況に対処します。

過剰なパントフロートラップに関する情報

過剰なパントフロートラップ（EPFT）機能は、物理インターフェイス、サブインターフェイス、バンドルインターフェイス、およびバンドル サブインターフェイスからの制御パケットトラフィックをモニタします。この機能は、OSPF と BGP でバッドアクターを特定するのに役立ちます。EPFT は、送信元 MAC 単位で OSPF と BGP のルーティングプロトコルをモニタします。バッドアクターが検出されると、特定の期間にわたって制御パケットがドロップされ、送信元 MAC が一定の期間（デフォルトでは15分間）「ペナルティボックス」に配置されます。ペナルティタイムアウトの終了時に、特定の送信元MAC の TCAM エントリがドロップから除外されます。その後もリモートデバイスからの過剰なレートのパケットトラフィックが着信する場合は、リモートデバイスは再度トラップされます。

（注） 過剰なパントフロートラップ機能が有効になっていない場合でも、「バッドアクター」が他のデバイスのサービスのみに影響を与えることがあります。ただし、それらはルータをダウンさせることはできません。

EPFT 実装の制約事項

EPFT 機能の実装には、次の制約事項が適用されます。

- EPFT はサブスクライバ インターフェイスでは有効になっていません。
- BGP および OSPF ルーティングプロトコルのみがサポートされます。

- OSPFV3 はサポートされていません。
- OSPF および BGP パケットは、バッドアクターを特定した後、特定の期間（デフォルトは 15分）は完全にドロップされます。この場合はペナルティポリシングは行われません。
- サブスクライバインターフェイスまたはインターフェイスベースのフローが設定されている場合、**routing-protocol-enable** コマンドは設定できません。また、逆も同様です。つまり、 **routing-protocol-enable** コマンドが設定されている場合、サブスクライバ インターフェイスまたはインターフェイスベースのフローは設定できません。
- サテライト ICL インターフェイスは EPFT モニタリングから除外されます。

過剰なパントフロートラップ処理の有効化

OSPF または BGP プロトコルで過剰なパントフロートラップ（EPFT） 機能を有効にし、指定されたペナルティタイムアウト期間を使用するには、このタスクを実行します。

始める前に

EPFT は、サブスクライバ以外のインターフェイスでのみ有効にできます。

手順の概要

1. **configure**
2. **lpts punt excessive-flow-trap non-subscriber-interfaces mac**
3. **lpts punt excessive-flow-trap penalty-timeout** *protocol time*
4. **lpts punt excessive-flow-trap routing-protocols-enable**
5. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**lpts punt excessive-flow-trap non-subscriber-interfaces mac** 例： `RP/0/RSP0/cpu 0: router(config)# lpts punt excessive-flow-trap non-subscriber-interfaces mac`	過剰パントフロートラップ機能をサブスクライバ以外のターフェイスに対して有効にします。
ステップ 3	**lpts punt excessive-flow-trap penalty-timeout** *protocol time* 例： `RP/0/RSP0/cpu 0: router(config)# lpts punt excessive-flow-trap penalty-timeout bgp 10`	ペナルティタイムアウト値を設定します。これは、プロトコルのペナルティボックスに送信元 MAC トラップが配置される期間です。ペナルティタイムアウト値は分単位で、範囲は 1 ～ 1000 です。デフォルトのペナリティタイムアウト値は 15 分です。

	コマンドまたはアクション	目的
ステップ 4	**lpts punt excessive-flow-trap routing-protocols-enable** 例： `RP/0/RSP0/cpu 0: router(config)# lpts punt excessive-flow-trap routing-protocols-enable`	L3 ルーティングプロトコルで EPFT を有効にします。
ステップ 5	**commit**	

過剰なパントフロートラップ処理の有効化：例

次に、サブスクライバ以外のインターフェイスに対して過剰なパントフロートラップを有効にする例を示します。

```
configure
lpts punt excessive-flow-trap
 penalty-timeout ospf 20  <<optional>>
 penalty-timeout bgp 20   <<optional>>
 non-subscriber-interfaces mac  <<This is mandatory for routing protcols to be enabled>>
routing-protocols-enable
end
!!
```

過剰なパントフロートラップ機能に関して、バッドアクター、ペナルティステータス、およびその他の詳細情報を表示するには、次のいずれかの **show** コマンドを EXEC モードで使用します。

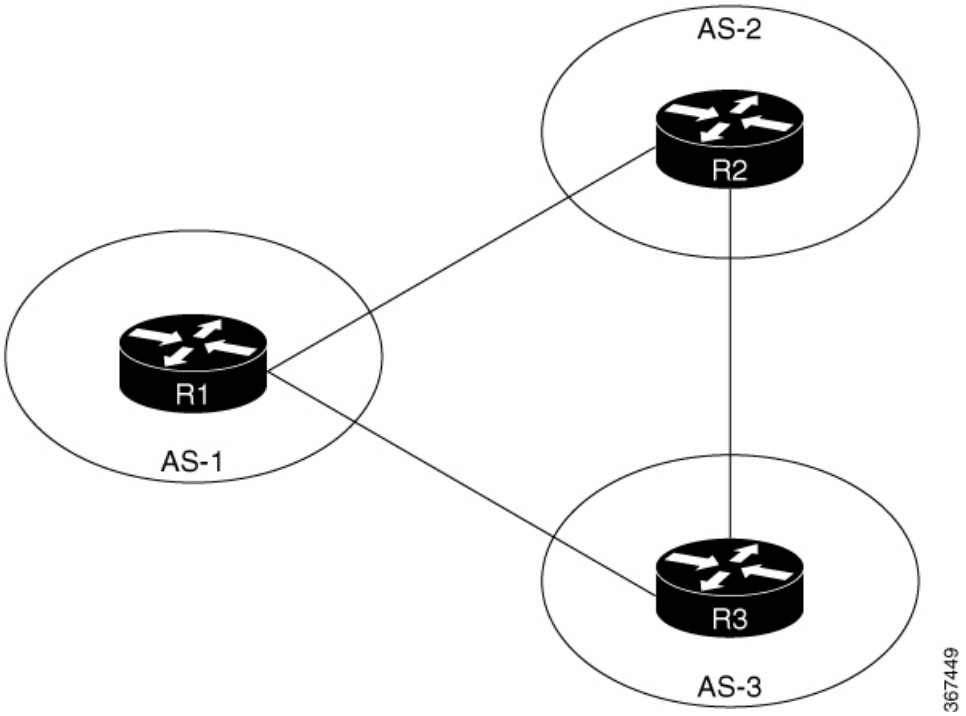
- **show lpts punt excessive-flow-trap** ［*protocol* ］
- **show lpts punt excessive-flow-trap all**

BGP マルチパスの機能拡張

- マルチパス プレフィックスのネクストホップ計算の上書きは許可されていません。**next-hop-unchanged multipath** コマンドを使用すると、マルチパス プレフィックスのネクストホップ計算の上書きが無効になります。
- マルチパスの計算時に as-path オンワードを無視する機能が追加されます。 **bgp multipath as-path ignore onwards** コマンドを使用すると、マルチパスの計算時に as-path オンワードが無視されます。

マルチパスを計算している間に複数の接続されたルータが以降の as-path を無視し始めると、ルーティングループが発生します。そのため、ループを形成する可能性があるルータでは **bgp multipath as-path ignore onwards** コマンドを設定しないでください。

図 *11* : ループの形成を示すトポロジ



異なる自律システム（AS-1、AS-2、および AS-3）内の 3 台のルータ、R1、R2、および R3 を想定します。これらのルータは相互に接続されています。R1 が R2 と R3 にプレフィックスをアナウンスします。R2 と R3 の両方がマルチパスを使用して設定されており、また、bgp multipath as-path で以降のコマンドが無視されます。R3 はマルチパスとして設定されているため、R2 はそのトラフィックの一部を R3 に送信します。同様に、R3 はそのトラフィックの一部を R2 に送信します。これにより、R3 と R2 の間に転送ループが発生します。そのため、このような転送ループを回避するには、接続されたルータで **bgp multipath as-path ignore onwards** コマンドを設定しないでください。

BGP SAFI-2 および SAFI-129 を使用した MVPN

BGP は、マルチキャスト VPN（MVPN）の後続のアドレスファミリ識別子（SAFI）-2 および SAFI-129 をサポートしています。

SAFI-129 は、コア IPv4 ネットワークでマルチキャストルーティングをサポートする機能を提供します。SAFI-129 は、BGP ベースの MVPN をサポートしています。SAFI-129 の追加により、マルチキャストで、ユニキャスト トポロジに依存しないこともあるアップストリーム マルチキャスト ホップを選択できるようになります。カスタマーエッジ（CE）ルータから学習したマルチキャストルートまたはリモートプロバイダーエッジ（PE）ルータから学習したマルチキャスト VPN ルートは、マルチキャストルーティング情報ベース（MuRIB）にインストールされます。この MuRIB には、マルチキャストに固有のルートが入力され、ユニキャスト転送では使用されません。PE-CE BGP プレフィックスは SAFI-2 を使用してアドバタイズされ、PE-PE のルートは SAFI-129 を使用してアドバタイズされます。

BGP Monitoring Protocol の概要

BGP Monitoring Protocol（BMP）機能により、BGP スピーカー（BMP クライアントという）をモニタできるようになります。BMP サーバとして機能するようにデバイスを設定して、複数のアクティブ ピア セッションが確立された 1 つまたは複数の BMP クライアントをモニタできます。また、1 つ以上の BMP サーバに接続するように BMP クライアントを設定することもできます。BMP 機能では、複数の BMP サーバ（プライマリ サーバとして設定）を、アクティブな状態で相互に独立して機能しながら BMP クライアントをモニタするように設定できます。

BMP プロトコルは、隣接するルーティング情報ベースやピアの着信 （Adj-RIB-In）テーブルへの継続的なアクセス、およびモニタリングステーションが詳細な分析のために使用できると特定の統計情報の定期的なダンプを提供します。BMP は、ピアの Adj-RIB-In テーブルをポリシーごとに表示します。

すべての BGP インスタンスに対して、グローバルに設定された複数の BMP サーバが存在する場合があります。設定された BMP サーバは複数のスピーカーインスタンス間で共通であり、インスタンス内の各 BGP ピアは、BMP サーバのすべてまたは一部によるモニタリング用に設定でき、BGP スピーカーの観点からは BGP ピアと BMP サーバ間の「Any-to-Any」マップを提供します。いずれかの BGP ピアが起動する前に BMP サーバが設定されている場合は、BGP ピアが起動するとすぐにモニタリングが開始されます。BMP サーバの設定は、その特定の BMP サーバによってモニタされるように設定されている BGP ピアがない場合にのみ削除できます。

BMP クライアントと BMP サーバ間のセッションは、プレーン TCP（暗号化/カプセル化なし）で動作します。BMP サーバとの TCP セッションが確立されていない場合、クライアントは 7 秒ごとに接続を再試行します。

BMP サーバは、そのクライアント（BGP スピーカー）にメッセージを送信しません。メッセージフローは一方向（BGP スピーカから BMP サーバへ）のみです。

Cisco NCS 5500 シリーズ ルータでは、最大 8 台の BMP サーバを設定できます。各 BMP サーバはサーバ ID で指定され、IP アドレス、ポート番号などの特定のパラメータを設定できます。ホストとポートの詳細を使用して BMP サーバを正常に設定すると、BGP スピーカーは BMP サーバへの接続を試行します。TCP 接続が設定されると、最初のメッセージとして開始メッセージが送信されます。

bmp server により、ユーザは複数（独立かつ非同期）の BMP サーバ接続の設定が可能になります。

BGP スピーカーのすべてのネイバーは、必ずしも BMP クライアントである必要はありません。BMP クライアントは、BMP サーバとの直接 TCP 接続を持っているクライアントです。これらの BGP スピーカーはそれぞれ、多数の BGP ネイバーまたはピアを持つことができます。BGP スピーカーの下で、そのネイバーのいずれかが BMP モニタリング用に設定されている場合、その特定のピアルータのメッセージのみが BMP サーバに送信されます。

BMP サーバへのセッション接続は、BMP クライアントでの初期遅延後に試行されます。この初期遅延は設定できます。初期遅延が設定されていない場合は、7 秒のデフォルトの接続遅延が使用されます。複数の BMP サーバの状態が厳密に切り替わり、リフレッシュ遅延が小さい特定の状況下で、初期遅延を設定することが重要になります。これにより、冗長なルートリフ

レッシュが生成される可能性があります。これにより、大量のネットワークトラフィックが発生し、デバイスに負荷がかかります。初期遅延が異なると、ネットワークとルータの負荷スパイクが低減される可能性があります。

初期遅延後、BMP サーバへの TCP 接続が試行されます。サーバ接続がアップ状態になると、モニタリングが有効になっているピアがあるかどうかが確認されます。すでにモニタされている BGP ピアが「ESTAB」状態になると、スピーカーはそのピアの「peer-up」メッセージを BMP サーバに送信します。BGP ピアがルートリフレッシュ要求を受信すると、ネイバーが更新を送信します。このルートリフレッシュは、各 BMP サーバに設定された遅延に基づいて開始されます。これをルートリフレッシュ遅延といいます。モニタするネイバが複数ある場合、それらが有効になっている BMP サーバに基づいて、各ネイバにはリフレッシ遅延が設定されます。すべての BGP ネイバーがリフレッシュ要求に応えて更新を送信すると、BMP サーバ内のテーブルが最新の状態になります。BMP モニタリングの開始後にネイバーが接続を確立する場合は、ルートリフレッシュ要求は必要ありません。そのネイバーから受信したすべてのルートが BMP サーバに送信されます。

（注） BMP プレインバウンドポリシーのルートモニタリングの場合、新しい BMP サーバが起動すると、BGP スピーカーによってルートリフレッシュ要求がピアルータに送信されます。ただし、BMP ポストインバウンドポリシーのルートモニタリングの場合、ルートリフレッシュ要求は、新しい BMP サーバが起動したときにピアルータに送信されません。これは、BMP テーブルが更新生成に使用されるためです。

複数の BMP サーバが立て続けにアクティブ化される場合は、BGP ピアにリフレッシュ要求をバッチ化すると便利です。**bmp server initial-refresh-delay** コマンドを使用して、最初の BMP サーバが起動したときにリフレッシュメカニズムをトリガーする際の遅延を設定できます。このタイムフレーム内に他の BMP サーバがオンラインになった場合は、1 セットのリフレッシュ要求のみが BGP ピアに送信されます。また、BGP スピーカーからのすべてのリフレッシュ要求をスキップし、ピアからのすべての着信メッセージだけをモニタするように、**bmp server initial-refresh-delay skip** コマンドを設定することもできます。

クライアントとサーバの設定では、デバイスのリソース負荷を最小限に抑え、過度なネットワーク トラフィックが発生しないようにすることが推奨されます。BMP 設定では、サーバとクライアントの間の接続でフラッピングが発生しないように、BMP サーバ上でさまざまな遅延タイマーを設定できます。

BGP の実装方法

BGP ルーティングのイネーブル化

BGP ルーティングをイネーブルにし、BGP ルーティング プロセスを設定するには、次の作業を実行します。BGP ネイバーの設定は、BGP ルーティングのイネーブル化の一部として含まれています。

（注） BGP ルーティングをイネーブルにするには、1 つ以上のネイバーおよび 1 つ以上のアドレス ファミリを設定する必要があります。**address family** コマンドおよび **remote as** コマンドを使用して、リモート AS とアドレス ファミリの両方を持つ 1 つ以上のネイバーをグローバルに設定する必要があります。

始める前に

BGP はルータ ID（設定済みループバック アドレスなど）を取得できなければなりません。1 つ以上のアドレス ファミリを BGP ルータ コンフィギュレーションに設定する必要があり、同じアドレス ファミリをネイバーの下にも設定する必要があります。

（注） ネイバーが外部 BGP（eBGP）ピアとして設定されている場合は、**route-policy** コマンドを使用して、インバウンドおよびアウトバウンドのルート ポリシーをネイバー上に設定する必要があります。

（注） 2 つのピア間で eBGP ネイバーシップを確立している間、BGP は 2 つのピアが直接接続されているかどうかをチェックします。ピアが直接接続されていない場合、デフォルトでは、BGP は関係を確立しようとしません。2 つの BGP ピアが直接接続されておらず、ルータのループバック間でピアリングが必要な場合は、**ignore-connected-check** コマンドを使用できます。このコマンドは、BGP 制御パケットの送信元 IP が宛先と同じネットワーク内にあるかどうかを確認するために BGP が実行するデフォルトのチェックを無効にします。このシナリオでは、TTL 値が 1 の場合 、**ignore-connected-check** が使用されていれば十分です。

egp-multihop *ttl* の設定は、ピアが直接接続されておらず、その間に多くのルータが存在する場合に必要です。**egp-multihop** *ttl* コマンドが設定されていない場合、デフォルトでは、eBGP は BGP メッセージを伝送するパケットの TTL を 1 に設定します。eBGP を複数ホップ離れているルータ間で設定する必要がある場合は、TTL 値を設定する必要があります。この TTL 値は、それらの間のホップ数以上にする必要があります。たとえば、2 つの BGP ピアリング ルータ R1 と R4 の間にホップが 2 つ（R2、R3）がある場合は、TTL 値を 3 に設定する必要があります。

手順の概要

1. **configure**
2. **route-policy** *route-policy-name*
3. **end-policy**
4. **commit**
5. **configure**
6. **router bgp** *as-number*
7. **bgp router-id** *ip-address*

8. **address-family** { **ipv4** | **ipv6** } **unicast**
9. **exit**
10. **neighbor** *ip-address*
11. **remote-as** *as-number*
12. **address-family** { **ipv4** | **ipv6** } **unicast**
13. **route-policy** *route-policy-name* { **in** | **out** }
14. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	

ステップ 2	**route-policy** *route-policy-name*
	例：
	`RP/0/RSP0/cpu 0: router(config)# route-policy drop-as-1234`
	`RP/0/RSP0/cpu 0: router(config-rpl)# if as-path passes-through '1234' then`
	`RP/0/RSP0/cpu 0: router(config-rpl)# apply check-communities`
	`RP/0/RSP0/cpu 0: router(config-rpl)# else`
	`RP/0/RSP0/cpu 0: router(config-rpl)# pass`
	`RP/0/RSP0/cpu 0: router(config-rpl)# endif`
		（任意）ルートポリシーを作成し、ルートポリシー コンフィギュレーション モードを開始します。このモードではルート ポリシーを定義できます。

ステップ 3	**end-policy** 例： `RP/0/RSP0/cpu 0: router(config-rpl)# end-policy`	（任意）ルート ポリシーの定義を終了し、ルート ポリシー コンフィギュレーション モードを終了します。
ステップ 4	**commit**	
ステップ 5	**configure**	
ステップ 6	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	BGP AS 番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 7	**bgp router-id** *ip-address* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# bgp router-id 192.168.70.24`	指定したルータ ID で、ローカル ルータを設定します。

	コマンドまたはアクション	目的
ステップ 8	**address-family** { **ipv4** | **ipv6** } **unicast** 例： `RP/0/RSP0/cpu 0: router(config-bgp)# address-family ipv4 unicast`	IPv4 または IPv6 のいずれかのアドレス ファミリを指定し、アドレスファミリのコンフィギュレーション サブモードを開始します。 このコマンドのすべてのキーワードと引数のリストを参照するには、CLI ヘルプ（?）を使用します。
ステップ 9	**exit** 例： `RP/0/RSP0/cpu 0: router(config-bgp-af)# exit`	現在のコンフィギュレーション モードを終了します。
ステップ 10	**neighbor** *ip-address* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 172.168.40.24`	BGP ルーティングのためにルータをネイバー コンフィギュレーション モードにして、ネイバーの IP アドレスを BGP ピアとして設定します。
ステップ 11	**remote-as** *as-number* 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# remote-as 2002`	ネイバーを作成し、リモート自律システム番号を割り当てます。
ステップ 12	**address-family** { **ipv4** | **ipv6** } **unicast** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# address-family ipv4 unicast`	IPv4 または IPv6 のいずれかのアドレス ファミリを指定し、アドレスファミリのコンフィギュレーション サブモードを開始します。 このコマンドのすべてのキーワードと引数のリストを参照するには、CLI ヘルプ（?）を使用します。
ステップ 13	**route-policy** *route-policy-name* { **in** | **out** } 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# route-policy drop-as-1234 in`	（任意）指定したポリシーを着信 IPv4 ユニキャスト ルートに適用します。
ステップ 14	**commit**	

特定の自律システムに対する複数の BGP インスタンスの設定

特定の自律システムに複数の BGP インスタンスを設定するには、次のタスクを実行します。

単一の BGP インスタンスに対するすべての設定変更を同時にコミットすることができます。ただし、複数のインスタンスに対する設定変更は同時にコミットできません。

手順の概要

1. **configure**
2. **router bgp** *as-number* [**instance** *instance name*]
3. **bgp router-id** *ip-address*
4. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* [**instance** *instance name*] 例： `RP/0/RSP0/CPU0:router(config)# router bgp 100 instance inst1`	ユーザが指定した BGP インスタンスに対し BGP コンフィギュレーション モードを開始します。
ステップ 3	**bgp router-id** *ip-address* 例： `RP/0/RSP0/CPU0:router(config-bgp)# bgp router-id 10.0.0.0`	BGP スピーキング ルータの固定ルータ ID（BGP インスタンス）を設定します。 （注） 各 BGP インスタンスに一意のルータ ID を手動で設定する必要があります。
ステップ 4	**commit**	

BGP のルーティング ドメイン コンフェデレーションの設定

BGP のルーティング ドメイン コンフェデレーションを設定するには、次の作業を実行します。これには、コンフェデレーション ID の指定と、コンフェデレーションに属す自律システムの指定を含みます。

ルーティング ドメイン コンフェデレーションを設定すると、自律システムを複数の自律システムに分割して、これを 1 つのコンフェデレーションにグループ化することによって、内部 BGP（iBGP）メッシュを削減することができます。それぞれの自律システムは、そのシステム自身内で完全にメッシュ化されていて、同じコンフェデレーションの別の自律システムとの接続を数個持ちます。このコンフェデレーションによりネクストホップおよびローカルプリファレンス情報が維持され、これにより、すべての自律システムに対して Interior Gateway Protocol（IGP）を 1 つ維持できるようになります。外部からは、このコンフェデレーションは単一の自律システムであるかのように見えます。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **bgp confederation identifier** *as-number*
4. **bgp confederation peers** *as-number*
5. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router# router bgp 120`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**bgp confederation identifier** *as-number* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# bgp confederation identifier 5`	BGP コンフェデレーション ID を指定します。

ステップ 4	**bgp confederation peers** *as-number*
	例：
	`RP/0/RSP0/cpu 0: router(config-bgp)# bgp confederation peers 1091`
	`RP/0/RSP0/cpu 0: router(config-bgp)# bgp confederation peers 1092`
	`RP/0/RSP0/cpu 0: router(config-bgp)# bgp confederation peers 1093`
	`RP/0/RSP0/cpu 0: router(config-bgp)# bgp confederation peers 1094`
	`RP/0/RSP0/cpu 0: router(config-bgp)# bgp confederation peers 1095`
	`RP/0/RSP0/cpu 0: router(config-bgp)# bgp confederation peers 1096`
		BGP 自律システムが指定された BGP コンフェデレーション ID に属することを指定します。例に示すように、複数の AS 番号を同じコンフェデレーション ID に関連付けることができます。

ステップ 5	**commit**	

リンク障害後の eBGP セッションの即時リセット

デフォルトでは、リンクがダウンすると、直接隣接する外部ピアの BGP セッションはすべて即時にリセットされます。自動リセットをディセーブルにするには **bgp fast-external-fallover disable** コマンドを使用します。自動リセットをイネーブルにするには **no bgp fast-external-fallover disable** コマンドを使用します。

BGP タイマー値が 10 および 30 に設定されているノードで eBGP セッションの数が 3500 に達すると、eBGP セッションはフラップします。3500 を超える数の eBGP セッションに対応するには、**lpts pifib hardware police location** *location-id* コマンドを使用してパケット レートを大きくします。eBGP セッションを増加する設定の例を次に示します。

```
RP/0/RSP0/cpu 0: router#configure
RP/0/RSP0/cpu 0: router(config)#lpts pifib hardware police location 0/2/CPU0
RP/0/RSP0/cpu 0: router(config-pifib-policer-per-node)#flow bgp configured rate 4000
RP/0/RSP0/cpu 0: router(config-pifib-policer-per-node)#flow bgp known rate 4000
RP/0/RSP0/cpu 0: router(config-pifib-policer-per-node)#flow bgp default rate 4000
RP/0/RSP0/cpu 0: router(config-pifib-policer-per-node)#commit
```

ネイバー変更のロギング

ネイバー変更のロギングはデフォルトでイネーブルになっています。ロギングをオフにするには、**log neighbor changes disable** コマンドを使用します。ロギングがディセーブルにされている場合にロギングを再びイネーブルにするには、**no log neighbor changes disable** コマンドを使用します。

BGP タイマーの調整

BGP ネイバーにタイマーを設定するには、次のタスクを実行します。

BGP は、定期実行アクティビティ（キープアライブ メッセージの送信、ネイバーがダウンしたと判断する条件となるそのネイバーからメッセージを受信しなかった期間など）を制御するために、特定のタイマーを使用します。ルータ コンフィギュレーション モードで timers bgp コマンドを使用して設定した値は、特定のネイバーでネイバー コンフィギュレーション モードで timers コマンドを使用すると上書きできます。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **timers bgp** *keepalive hold-time*
4. **neighbor** *ip-address*
5. **timers** *keepalive hold-time*
6. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 123`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**timers bgp** *keepalive hold-time* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# timers bgp 30 90`	すべてのネイバーのデフォルトのキープアライブ時間とデフォルトの保留時間を設定します。
ステップ 4	**neighbor** *ip-address* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 172.168.40.24`	BGP ルーティングのためにルータをネイバー コンフィギュレーション モードにして、ネイバーの IP アドレスを BGP ピアとして設定します。

	コマンドまたはアクション	目的
ステップ 5	**timers** *keepalive hold-time* 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# timers 60 220`	（任意）BGP ネイバーのキープアライブ タイマーと保持時間タイマーを設定します。
ステップ 6	**commit**	

BGP のデフォルト ローカル プリファレンス値の変更

BGP パスのデフォルト ローカル プリファレンス値を設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **bgp default local-preference** *value*
4. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**bgp default local-preference** *value* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# bgp default local-preference 200`	デフォルト値 100 以外のデフォルト ローカル プリファレンス値を設定します。100 より大きい値を設定して推奨度を上げるか、または 100 未満の値を設定して推奨度を低くすることができます。
ステップ 4	**commit**	

BGP の MED メトリックの設定

メトリックがまだ設定されていないルート（MED 属性が設定されていない、受信されたルート）をピアにアドバタイズするように Multi Exit Discriminator（MED）を設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **default-metric** *value*
4. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	自律システム番号を指定し、BGP コンフィギュレーションモードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**default-metric** *value* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# default metric 10`	まだメトリックが設定されていないルート（MED 属性を持たない、受信されたルート）をピアにアドバタイズするように MED を設定する場合に使用されるデフォルトのメトリックを設定します。
ステップ 4	**commit**	

BGP の重みの設定

ネイバーから受信したルートに重みを割り当てるには、次のタスクを実行します。重みとは、ベストパス選択プロセスを制御するためにパスに割り当てる数値です。ほとんどのトラフィックで特定のネイバーを優先する場合、**weight** コマンドを使用して、そのネイバーから学習したすべてのルートに大きい重みを割り当てることができます。

始める前に

（注） 新たに設定した重みを反映するには、**clear bgp** コマンドを使用する必要があります。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **neighbor** *ip-address*
4. **remote-as** *as-number*
5. **address-family** { **ipv4** | **ipv6** } **unicast**
6. **weight** *weight-value*
7. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**neighbor** *ip-address* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 172.168.40.24`	BGP ルーティングのためにルータをネイバー コンフィギュレーション モードにして、ネイバーの IP アドレスを BGP ピアとして設定します。
ステップ 4	**remote-as** *as-number* 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# remote-as 2002`	ネイバーを作成し、リモート自律システム番号を割り当てます。
ステップ 5	**address-family** { **ipv4** | **ipv6** } **unicast** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# address-family ipv4 unicast`	IPv4 または IPv6 のいずれかのアドレス ファミリを指定し、アドレス ファミリのコンフィギュレーション サブモードを開始します。 このコマンドのすべてのキーワードと引数のリストを参照するには、CLI ヘルプ（?）を使用します。
ステップ 6	**weight** *weight-value* 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# weight 41150`	ネイバーから学習したすべてのルートに重みを割り当てます。
ステップ 7	**commit**	

BGP 最適パス計算の調整

デフォルトの BGP 最適パスの計算の動作を変更するには、次の作業を実行します。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **bgp bestpath med missing-as-worst**
4. **bgp bestpath med always**
5. **bgp bestpath med confed**

6. **bgp bestpath as-path ignore**
7. **bgp bestpath compare-routerid**
8. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 126`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**bgp bestpath med missing-as-worst** 例： `RP/0/RSP0/cpu 0: router(config-bgp)# bgp bestpath med missing-as-worst`	このパスを最も必要のないパスにするために、このパス内の不明 MED 属性の値は無限であると見なすように、BGP ソフトウェアに指示します。
ステップ 4	**bgp bestpath med always** 例： `RP/0/RSP0/cpu 0: router(config-bgp)# bgp bestpath med always`	パスがどの自律システムから受信されたかに関係なく、すべてのパスの間でプレフィックスについて MED を比較するように、指定した自律システムの BGP スピーカーを設定します。
ステップ 5	**bgp bestpath med confed** 例： `RP/0/RSP0/cpu 0: router(config-bgp)# bgp bestpath med confed`	コンフェデレーション ピアから学習したパスについて MED 値を BGP ソフトウェアで比較できるようにします。
ステップ 6	**bgp bestpath as-path ignore** 例： `RP/0/RSP0/cpu 0: router(config-bgp)# bgp bestpath as-path ignore`	最適パスを選択するときに、自律システム パスの長さが無視されるように BGP ソフトウェアを設定します。
ステップ 7	**bgp bestpath compare-routerid** 例： `RP/0/RSP0/cpu 0: router(config-bgp)# bgp bestpath compare-routerid`	類似パスのルータ ID を比較するように自律システムの BGP スピーカーを設定します。
ステップ 8	**commit**	

BGP バックドア ルートの指定

外部ボーダーゲートウェイ プロトコル（eBGP）のアドミニストレーティブディスタンスに、ローカルにソースされたBGPルートのアドミニストレーティブディスタンスを設定し、Interior Gateway Protocol（IGP）ルートよりも推奨度を低くするには、次の作業を実行します。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **address-family** { **ipv4** | **ipv6** } **unicast**
4. **network** { *ip-address* / *prefix-length* | *ip-address mask* } **backdoor**
5. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**address-family** { **ipv4**	**ipv6** } **unicast** 例： `RP/0/RSP0/cpu 0: router(config-bgp)# address-family ipv4 unicast`	IPv4 または IPv6 のいずれかのアドレス ファミリを指定し、アドレスファミリのコンフィギュレーション サブモードを開始します。 このコマンドのすべてのキーワードと引数のリストを参照するには、CLI ヘルプ（?）を使用します。
ステップ 4	**network** { *ip-address* / *prefix-length*	*ip-address mask* } **backdoor** 例： `RP/0/RSP0/cpu 0: router(config-bgp-af)# network 172.20.0.0/16`	指定されたネットワークを作成してアドバタイズするようにローカル ルータを設定します。
ステップ 5	**commit**	

集約アドレスの設定

BGP ルーティング テーブルに集約エントリを作成するには、次の作業を実行します。

手順の概要

1. **configure**
2. **router bgp** *as-number*

3. **address-family** { **ipv4** | **ipv6** } **unicast**
4. **aggregate-address** *address/mask-length* [**as-set**] [**as-confed-set**] [**summary-only**] [**route-policy** *route-policy-name*]
5. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**address-family** { **ipv4**	**ipv6** } **unicast** 例： `RP/0/RSP0/cpu 0: router(config-bgp)# address-family ipv4 unicast`	IPv4 または IPv6 のいずれかのアドレス ファミリを指定し、アドレス ファミリのコンフィギュレーション サブモードを開始します。 このコマンドのすべてのキーワードと引数のリストを参照するには、CLI ヘルプ（?）を使用します。
ステップ 4	**aggregate-address** *address/mask-length* [**as-set**] [**as-confed-set**] [**summary-only**] [**route-policy** *route-policy-name*] 例： `RP/0/RSP0/cpu 0: router(config-bgp-af)# aggregate-address 10.0.0.0/8 as-set`	集約アドレスを作成します。このルートにアドバタイズされたパスは、集約されるすべてのパスに含まれるすべての要素で構成された自律システム セットです。 • **as-set** キーワードは、関係するパスから自律システム セット パス情報およびコミュニティ情報を生成します。 • **as-confed-set** キーワードは、関係するパスから自律システム コンフェデレーション セット パス情報を生成します。 • **summary-only** キーワードは、アップデートから具体的なルートをすべてフィルタリングします。 • **route-policy** *route-policy-name* キーワードおよび引数は、集約ルートの属性の設定に使用されるルート ポリシーを指定します。
ステップ 5	**commit**	

IGP への iBGP ルートの再配布

Intermediate System-to-Intermediate System（IS-IS）や Open Shortest Path First（OSPF）など、内部ゲートウェイ プロトコル（IGP）に iBGP ルートを再配布するには、次の作業を実行します。

（注） **bgp redistribute-internal** コマンドを使用するには、すべての BGP ルートを IP ルーティング テーブルに再インストールするために、**clear route *** コマンドを発行する必要があります。

注意 IGP への iBGP ルートの再配布は、自律システム内にルーティング ループが作成される原因となる可能性があります。このコマンドの使用には注意が必要です。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **bgp redistribute-internal**
4. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**bgp redistribute-internal** 例： `RP/0/RSP0/cpu 0: router(config-bgp)# bgp redistribute-internal`	IGP（IS-IS や OSPF など）への iBGP ルートの再配布を許可します。
ステップ 4	**commit**	

過剰パスの破棄の設定

BGP 最大プレフィックス過剰パスの破棄を設定するには、次のタスクを実行します。

手順の概要

1. **configure**

2. **router bgp** *as-number*
3. **neighbor** *ip-address*
4. **address-family** { **ipv4** | **ipv6** } **unicast**
5. **maximum-prefix** *maximum* **discard-extra-paths**
6. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/cpu 0: router# configure`	グローバル コンフィギュレーション モード を開始します。
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 10`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**neighbor** *ip-address* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 10.0.0.1`	BGP ルーティングのためにルータをネイバー コンフィギュレーション モードにして、ネイバーの IP アドレスを BGP ピアとして設定します。
ステップ 4	**address-family** { **ipv4** | **ipv6** } **unicast** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# address-family ipv4 unicast`	IPv4 または IPv6 のいずれかのアドレス ファミリを指定し、アドレスファミリのコンフィギュレーション サブモードを開始します。
ステップ 5	**maximum-prefix** *maximum* **discard-extra-paths** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# maximum-prefix 1000 discard-extra-paths`	許可されるプレフィックス数の制限を設定します。 最大プレフィックスの制限を超えると過剰パスを破棄するように過剰パスの破棄を設定します。
ステップ 6	**commit**	

ネイバー単位の TCP MSS の設定

ネイバーによって継承されるネイバーグループに TCP MSS を設定するには、次のタスクを実行します。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **address-family ipv4 unicast**
4. **exit**
5. **neighbor-group** *name*

6. **tcp mss** *segment-size*
7. **address-family ipv4 unicast**
8. **exit**
9. **exit**
10. **neighbor** *ip-address*
11. **remote-as** *as-number*
12. **use neighbor-group** *group-name*
13. **address-family ipv4 unicast**
14. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/cpu 0: router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 10`	自律システム番号を指定し、BGP コンフィギュレーションモードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**address-family ipv4 unicast** 例： `RP/0/RSP0/cpu 0: router(config-bgp)# address-family ipv4 unicast`	IPv4 アドレス ファミリ ユニキャストを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 4	**exit** 例： `RP/0/RSP0/cpu 0: router(config-bgp-af)# exit`	ルータ アドレス ファミリ コンフィギュレーション モードを終了し、BGP コンフィギュレーション モードに戻ります。
ステップ 5	**neighbor-group** *name* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# neighbor-group n1`	ネイバー グループ コンフィギュレーション モードを開始します。
ステップ 6	**tcp mss** *segment-size* 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# tcp mss 500`	TCP 最大セグメントサイズを設定します。範囲は 68 ～ 10000 です。
ステップ 7	**address-family ipv4 unicast** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# address-family ipv4 unicast`	IPv4 アドレス ファミリ ユニキャストを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 8	**exit** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp-af)# exit`	ルータ アドレス ファミリ コンフィギュレーション モードを終了します。
ステップ 9	**exit** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# exit`	ネイバー グループ コンフィギュレーション モードを終了します。
ステップ 10	**neighbor** *ip-address* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 10.0.0.2`	BGP ルーティングのためにルータをネイバー コンフィギュレーション モードにして、ネイバーの IP アドレスを BGP ピアとして設定します。
ステップ 11	**remote-as** *as-number* 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# remote-as 1`	ネイバーを作成し、リモート自律（AS）システム番号を割り当てます。 • 2 バイト自律システム番号（ASN）の範囲は 1 ～ 65535 です。 • asplain 形式の 4 バイト自律システム番号（ASN）の範囲は、1 ～ 4294967295 です。 • asdot 形式の 4 バイト自律システム番号（ASN）の範囲は、1.0 ～ 65535.65535 です。
ステップ 12	**use neighbor-group** *group-name* 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# use neighbor-group n1`	BGP ネイバーが指定されたネイバー グループから設定を継承することを指定します。

ステップ 13	**address-family ipv4 unicast**
	例：
	`RP/0/RSP0/cpu 0: router(config-bgp-nbr)# address-family ipv4 unicast`
	`RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)#`
		IPv4 アドレス ファミリ ユニキャストを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 14	**commit**	

ネイバー単位の TCP MSS の無効化

ネイバーグループの特定のネイバーに対する TCP MSS を無効にするには、このタスクを実行します。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **address-family ipv4 unicast**
4. **exit**
5. **neighbor-group** *name*
6. **tcp mss** *segment-size*
7. **address-family ipv4 unicast**
8. **exit**
9. **exit**
10. **neighbor** *ip-address*
11. **remote-as** *as-number*
12. **use neighbor-group** *group-name*
13. **tcp mss inheritance-disable**
14. **address-family ipv4 unicast**
15. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/cpu 0: router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 10`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**address-family ipv4 unicast** 例： `RP/0/RSP0/cpu 0: router(config-bgp)# address-family ipv4 unicast`	IPv4 アドレス ファミリ ユニキャストを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 4	**exit** 例： `RP/0/RSP0/cpu 0: router(config-bgp-af)# exit`	ルータ アドレス ファミリ コンフィギュレーション モードを終了し、BGP コンフィギュレーション モードに戻ります。
ステップ 5	**neighbor-group** *name* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# neighbor-group n1`	ネイバー グループ コンフィギュレーション モードを開始します。
ステップ 6	**tcp mss** *segment-size* 例：	TCP 最大セグメントサイズを設定します。範囲は 68 ～ 10000 です。

	コマンドまたはアクション	目的
	`RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# tcp mss 500`	
ステップ 7	**address-family ipv4 unicast** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# address-family ipv4 unicast`	IPv4 アドレス ファミリ ユニキャストを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 8	**exit** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp-af)# exit`	ルータ アドレス ファミリ コンフィギュレーション モードを終了します。
ステップ 9	**exit** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# exit`	ネイバー グループ コンフィギュレーション モードを終了します。
ステップ 10	**neighbor** *ip-address* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 10.0.0.2`	BGP ルーティングのためにルータをネイバー コンフィギュレーション モードにして、ネイバーの IP アドレスを BGP ピアとして設定します。
ステップ 11	**remote-as** *as-number* 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# remote-as 1`	ネイバーを作成し、リモート自律（AS）システム番号を割り当てます。 • 2 バイト自律システム番号（ASN）の範囲は 1 ～ 65535 です。 • asplain 形式の 4 バイト自律システム番号（ASN）の範囲は、1 ～ 4294967295 です。 • asdot 形式の 4 バイト自律システム番号（ASN）の範囲は、1.0 ～ 65535.65535 です。
ステップ 12	**use neighbor-group** *group-name* 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# use neighbor-group n1`	BGP ネイバーが指定されたネイバー グループから設定を継承することを指定します。
ステップ 13	**tcp mss inheritance-disable** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# tcp mss inheritance-disable`	ネイバーに対する TCP MSS を無効にします。

	コマンドまたはアクション	目的

ステップ 14	**address-family ipv4 unicast**
	例：
	`RP/0/RSP0/cpu 0: router(config-bgp-nbr)# address-family ipv4 unicast`
	`RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)#`
		IPv4 アドレス ファミリ ユニキャストを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 15	**commit**	

マルチプロトコル BGP へのプレフィックスの再配布

別のプロトコルからマルチプロトコル BGP へプレフィックスを再配布するには、次のタスクを実行します。

再配布とは、あるルーティング プロトコルから別のルーティング プロトコルへプレフィックスを挿入するプロセスです。ここでは、別のルーティングプロトコルのプレフィックスをマルチプロトコル BGP に挿入する方法について説明します。具体的には、**redistribute** コマンドを使用してマルチプロトコル BGP に再配布されるプレフィックスは、ユニキャストデータベースまたはマルチキャストデータベース、あるいはその両方に挿入されます。

（注） BGP は、VRF での ISIS ルートの再配布をサポートしていません。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **address-family** { **ipv4** | **ipv6** } **unicast**
4. 次のいずれかを実行します。
   - **redistribute connected** [**metric** *metric-value*] [**route-policy** *route-policy-name*]
   - **redistribute eigrp** *process-id* [**match** { **external** | **internal** }] [**metric** *metric-value*] [**route-policy** *route-policy-name*]
   - **redistribute ospf** *process-id* [**match** { **external** [**1** | **2**] | **internal** | **nssa-external** [**1** | **2**]]} [**metric** *metric-value*] [**route-policy** *route-policy-name*]
   - **redistribute ospfv3** *process-id* [**match** { **external** [**1** | **2**] | **internal** | **nssa-external** [**1** | **2**]]} [**metric** *metric-value*] [**route-policy** *route-policy-name*]
   - **redistribute rip** [**metric** *metric-value*] [**route-policy** *route-policy-name*]
   - **redistribute static** [**metric** *metric-value*] [**route-policy** *route-policy-name*]
5. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**address-family** { **ipv4** | **ipv6** } **unicast** 例： `RP/0/RSP0/cpu 0: router(config-bgp)# address-family ipv4 unicast`	IPv4 または IPv6 のいずれかのアドレス ファミリを指定し、アドレス ファミリのコンフィギュレーション サブモードを開始します。 このコマンドのすべてのキーワードと引数のリストを参照するには、CLI ヘルプ（?）を使用します。
ステップ 4	次のいずれかを実行します。 • **redistribute connected** [**metric** *metric-value*] [**route-policy** *route-policy-name*] • **redistribute eigrp** *process-id* [**match** { **external** | **internal** }] [**metric** *metric-value*] [**route-policy** *route-policy-name*] • **redistribute ospf** *process-id* [**match** { **external** [**1** | **2**] | **internal** | **nssa-external** [**1** | **2**]]} [**metric** *metric-value*] [**route-policy** *route-policy-name*] • **redistribute ospfv3** *process-id* [**match** { **external** [**1** | **2**] | **internal** | **nssa-external** [**1** | **2**]]} [**metric** *metric-value*] [**route-policy** *route-policy-name*] • **redistribute rip** [**metric** *metric-value*] [**route-policy** *route-policy-name*] • **redistribute static** [**metric** *metric-value*] [**route-policy** *route-policy-name*] 例： `RP/0/RSP0/cpu 0: router(config-bgp-af)# redistribute ospf 110`	指定したインスタンスからのルートがBGPに再配布されるようにします。
ステップ 5	**commit**	

BGP ルート ダンプニングの設定

BGP ルート ダンプニングを設定してモニタするには、次の作業を実行します。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **address-family** { **ipv4** | **ipv6** } **unicast**
4. **bgp dampening** [*half-life* [*reuse suppress max-suppress-time*] | **route-policy** *route-policy-name*]
5. **commit**
6. **show bgp** [**ipv4** { **unicast** | **multicast** | **labeled-unicast** | **all** } | **ipv6 unicast** | **all** { **unicast** | **multicast** | **all** | **labeled-unicast** } | **vpnv4 unicast** [**rd** *rd-address*] | **vrf** { *vrf-name* | **all** } [**ipv4** { **unicast** | **labeled-unicast** } | **ipv6 unicast**]] **flap-statistics**
7. **show bgp** [**ipv4** { **unicast** | **multicast** | **labeled-unicast** | **all** } | **ipv6 unicast** | **all** { **unicast** | **multicast** | **all** | **labeled-unicast** } | **vpnv4 unicast** [**rd** *rd-address*] | **vrf** { *vrf-name* | **all** } [**ipv4** { **unicast** | **labeled-unicast** } | **ipv6 unicast**]] **flap-statistics regexp** *regular-expression*
8. **show bgp** [**ipv4** { **unicast** | **multicast** | **labeled-unicast** | **all** } | **ipv6 unicast** | **all** { **unicast** | **multicast** | **all** | **labeled-unicast** } | **vpnv4 unicast** [**rd** *rd-address*] | **vrf** { *vrf-name* | **all** } [**ipv4** { **unicast** | **labeled-unicast** } | **ipv6 unicast**]] **route-policy** *route-policy-name*
9. **show bgp** [**ipv4** { **unicast** | **multicast** | **labeled-unicast** | **all** } | **ipv6 unicast** | **all** { **unicast** | **multicast** | **all** | **labeled-unicast** } | **vpnv4 unicast** [**rd** *rd-address*] | **vrf** { *vrf-name* | **all** } [**ipv4** { **unicast** | **labeled-unicast** } | **ipv6 unicast**]] { *mask* | */prefix-length* }}
10. **show bgp** [**ipv4** { **unicast** | **multicast** | **labeled-unicast** | **all** } | **ipv6 unicast** | **all** { **unicast** | **multicast** | **all** | **labeled-unicast** } | **vpnv4 unicast** [**rd** *rd-address*] | **vrf** { *vrf-name* | **all** } [**ipv4** { **unicast** | **labeled-unicast** } | **ipv6 unicast**]] **flap-statistics** { *ip-address* [{ *mask* | */prefix-length* } [**longer-prefixes**
11. **clear bgp** [**ipv4** { **unicast** | **multicast** | **labeled-unicast** | **all** } | **ipv6 unicast** | **all** { **unicast** | **multicast** | **all** | **labeled-unicast** } | **vpnv4 unicast** [**rd** *rd-address*] | **vrf** { *vrf-name* | **all** } [**ipv4** { **unicast** | **labeled-unicast** } | **ipv6 unicast**]] **flap-statistics**
12. **clear bgp** [**ipv4** { **unicast** | **multicast** | **labeled-unicast** | **all** } | **ipv6 unicast** | **all** { **unicast** | **multicast** | **all** | **labeled-unicast** } | **vpnv4 unicast** [**rd** *rd-address*] | **vrf** { *vrf-name* | **all** } [**ipv4** { **unicast** | **labeled-unicast** } | **ipv6 unicast**]] **flap-statistics regexp** *regular-expression*
13. **clear bgp** [**ipv4** { **unicast** | **multicast** | **labeled-unicast** | **all** } | **ipv6 unicast** | **all** { **unicast** | **multicast** | **all** | **labeled-unicast** } | **vpnv4 unicast** [**rd** *rd-address*] | **vrf** { *vrf-name* | **all** } [**ipv4** { **unicast** | **labeled-unicast** } | **ipv6 unicast**]] **route-policy** *route-policy-name*
14. **clear bgp** [**ipv4** { **unicast** | **multicast** | **labeled-unicast** | **all** } | **ipv6 unicast** | **all** { **unicast** | **multicast** | **all** | **labeled-unicast** } | **vpnv4 unicast** [**rd** *rd-address*] | **vrf** { *vrf-name* | **all** } [**ipv4** { **unicast** | **labeled-unicast** } | **ipv6 unicast**]] **flap-statistics** *network* / *mask-length*
15. **clear bgp** [**ipv4** { **unicast** | **multicast** | **labeled-unicast** | **all** } | **ipv6 unicast** | **all** { **unicast** | **multicast** | **all** | **labeled-unicast** } | **vpnv4 unicast** [**rd** *rd-address*] | **vrf** { *vrf-name* | **all** } [**ipv4** { **unicast** | **labeled-unicast** } | **ipv6 unicast**]] **flap-statistics** *ip-address* / *mask-length*
16. **show bgp** [**ipv4** { **unicast** | **multicast** | **labeled-unicast** | **all** } | **ipv6 unicast** | **all** { **unicast** | **multicast** | **all** | **labeled-unicast** } | **vpnv4 unicast** [**rd** *rd-address*] | **vrf** { *vrf-name* | **all** } [**ipv4** { **unicast** | **labeled-unicast** } | **ipv6 unicast**]] **dampened-paths**
17. **clear bgp** [**ipv4** { **unicast** | **multicast** | **labeled-unicast** | **all** } | **ipv6 unicast** | **all** { **unicast** | **multicast** | **all** | **labeled-unicast** } | **vpnv4 unicast** [**rd** *rd-address*] | **vrf** { *vrf-name* | **all** } [**ipv4** { **unicast** | **labeled-unicast** } | **ipv6 unicast**]] **dampening** *ip-address* / *mask-length*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**address-family** { **ipv4** | **ipv6** } **unicast** 例： `RP/0/RSP0/cpu 0: router(config-bgp)# address-family ipv4 unicast`	IPv4 または IPv6 のいずれかのアドレス ファミリを指定し、アドレス ファミリのコンフィギュレーション サブモードを開始します。 このコマンドのすべてのキーワードと引数のリストを参照するには、CLI ヘルプ（?）を使用します。
ステップ 4	**bgp dampening** [*half-life* [*reuse suppress max-suppress-time*] | **route-policy** *route-policy-name*] 例： `RP/0/RSP0/cpu 0: router(config-bgp-af)# bgp dampening 30 1500 10000 120`	指定したアドレス ファミリに対して BGP ダンプニングを設定します。
ステップ 5	**commit**	
ステップ 6	**show bgp** [**ipv4** { **unicast** | **multicast** | **labeled-unicast** | **all** } | **ipv6** **unicast** | **all** { **unicast** | **multicast** | **all** | **labeled-unicast** } | **vpnv4 unicast** [**rd** *rd-address*] | **vrf** { *vrf-name* | **all** } [**ipv4** { **unicast** | **labeled-unicast** } | **ipv6 unicast**]] **flap-statistics** 例： `RP/0/RSP0/cpu 0: router# show bgp flap statistics`	BGP フラップ統計情報を表示します。
ステップ 7	**show bgp** [**ipv4** { **unicast** | **multicast** | **labeled-unicast** | **all** } | **ipv6** **unicast** | **all** { **unicast** | **multicast** | **all** | **labeled-unicast** } | **vpnv4 unicast** [**rd** *rd-address*] | **vrf** { *vrf-name* | **all** } [**ipv4** { **unicast** | **labeled-unicast** } | **ipv6 unicast**]] **flap-statistics regexp** *regular-expression* 例： `RP/0/RSP0/cpu 0: router# show bgp flap-statistics regexp _1$`	正規表現に一致するすべてのパスの BGP フラップ統計情報を表示します。

	コマンドまたはアクション	目的
ステップ 8	**show bgp** [**ipv4** { **unicast** | **multicast** | **labeled-unicast** | **all** } | **ipv6** **unicast** | **all** { **unicast** | **multicast** | **all** | **labeled-unicast** } | **vpnv4 unicast** [**rd** *rd-address*] | **vrf** { *vrf-name* | **all** } [**ipv4** { **unicast** | **labeled-unicast** } | **ipv6 unicast**]] **route-policy** *route-policy-name* 例： `RP/0/RSP0/cpu 0: router(config)# show bgp flap-statistics route-policy policy_A`	指定されたルート ポリシーの BGP フラップ統計情報を表示します。
ステップ 9	**show bgp** [**ipv4** { **unicast** | **multicast** | **labeled-unicast** | **all** } | **ipv6** **unicast** | **all** { **unicast** | **multicast** | **all** | **labeled-unicast** } | **vpnv4 unicast** [**rd** *rd-address*] | **vrf** { *vrf-name* | **all** } [**ipv4** { **unicast** | **labeled-unicast** } | **ipv6 unicast**]] { *mask* | */prefix-length* }} 例： `RP/0/RSP0/cpu 0: router# show bgp flap-statistics 172.20.1.1`	指定されたプレフィックスの BGP フラップを表示します。
ステップ 10	**show bgp** [**ipv4** { **unicast** | **multicast** | **labeled-unicast** | **all** } | **ipv6** **unicast** | **all** { **unicast** | **multicast** | **all** | **labeled-unicast** } | **vpnv4 unicast** [**rd** *rd-address*] | **vrf** { *vrf-name* | **all** } [**ipv4** { **unicast** | **labeled-unicast** } | **ipv6 unicast**]] **flap-statistics** { *ip-address* [{ *mask* | */prefix-length* } [**longer-prefixes** 例： `RP/0/RSP0/cpu 0: router# show bgp flap-statistics 172.20.1.1 longer-prefixes`	指定された IP アドレスのより具体的なエントリの BGP フラップ統計情報を表示します。
ステップ 11	**clear bgp** [**ipv4** { **unicast** | **multicast** | **labeled-unicast** | **all** } | **ipv6** **unicast** | **all** { **unicast** | **multicast** | **all** | **labeled-unicast** } | **vpnv4 unicast** [**rd** *rd-address*] | **vrf** { *vrf-name* | **all** } [**ipv4** { **unicast** | **labeled-unicast** } | **ipv6 unicast**]] **flap-statistics** 例： `RP/0/RSP0/cpu 0: router# clear bgp all all flap-statistics`	すべてのルートの BGP フラップ統計情報をクリアします。
ステップ 12	**clear bgp** [**ipv4** { **unicast** | **multicast** | **labeled-unicast** | **all** } | **ipv6** **unicast** | **all** { **unicast** | **multicast** | **all** | **labeled-unicast** } | **vpnv4 unicast** [**rd** *rd-address*] | **vrf** { *vrf-name* | **all** } [**ipv4** {	指定された正規表現に一致するすべてのパスの BGP フラップ統計情報をクリアします。

	コマンドまたはアクション	目的
	unicast | **labeled-unicast** } | **ipv6 unicast**]] **flap-statistics regexp** *regular-expression* 例： `RP/0/RSP0/cpu 0: router# clear bgp ipv4 unicast flap-statistics regexp _1$`	
ステップ 13	**clear bgp** [**ipv4** { **unicast** | **multicast** | **labeled-unicast** | **all** } | **ipv6 unicast** | **all** { **unicast** | **multicast** | **all** | **labeled-unicast** } | **vpnv4 unicast** [**rd** *rd-address*] | **vrf** { *vrf-name* | **all** } [**ipv4** { **unicast** | **labeled-unicast** } | **ipv6 unicast**]] **route-policy** *route-policy-name* 例： `RP/0/RSP0/cpu 0: router# clear bgp ipv4 unicast flap-statistics route-policy policy_A`	指定されたルート ポリシーの BGP フラップ統計情報をクリアします。
ステップ 14	**clear bgp** [**ipv4** { **unicast** | **multicast** | **labeled-unicast** | **all** } | **ipv6 unicast** | **all** { **unicast** | **multicast** | **all** | **labeled-unicast** } | **vpnv4 unicast** [**rd** *rd-address*] | **vrf** { *vrf-name* | **all** } [**ipv4** { **unicast** | **labeled-unicast** } | **ipv6 unicast**]] **flap-statistics** *network* / *mask-length* 例： `RP/0/RSP0/cpu 0: router# clear bgp ipv4 unicast flap-statistics 192.168.40.0/24`	指定されたネットワークの BGP フラップ統計情報をクリアします。
ステップ 15	**clear bgp** [**ipv4** { **unicast** | **multicast** | **labeled-unicast** | **all** } | **ipv6 unicast** | **all** { **unicast** | **multicast** | **all** | **labeled-unicast** } | **vpnv4 unicast** [**rd** *rd-address*] | **vrf** { *vrf-name* | **all** } [**ipv4** { **unicast** | **labeled-unicast** } | **ipv6 unicast**]] **flap-statistics** *ip-address* / *mask-length* 例： `RP/0/RSP0/cpu 0: router# clear bgp ipv4 unicast flap-statistics 172.20.1.1`	指定されたネイバーから受信したルートの BGP フラップ統計情報をクリアします。
ステップ 16	**show bgp** [**ipv4** { **unicast** | **multicast** | **labeled-unicast** | **all** } | **ipv6 unicast** | **all** { **unicast** | **multicast** | **all** | **labeled-unicast** } | **vpnv4 unicast** [**rd** *rd-address*] | **vrf** { *vrf-name* | **all** } [**ipv4** { **unicast** | **labeled-unicast** } | **ipv6 unicast**]] **dampened-paths** 例：	抑制が解除されるまでの時間を含む、ダンプニングされたルートを表示します。

	コマンドまたはアクション	目的
	`RP/0/RSP0/cpu 0: router# show bgp dampened-paths`	
ステップ 17	**clear bgp** [**ipv4** { **unicast** | **multicast** | **labeled-unicast** | **all** } | **ipv6** **unicast** | **all** { **unicast** | **multicast** | **all** | **labeled-unicast** } | **vpnv4 unicast** [**rd** *rd-address*] | **vrf** { *vrf-name* | **all** } [**ipv4** { **unicast** | **labeled-unicast** } | **ipv6 unicast**]] **dampening** *ip-address* / *mask-length* 例： `RP/0/RSP0/cpu 0: router# clear bgp dampening`	ルート ダンプニング情報をクリアし、抑制されたルートを抑制解除します。 注意 **clear bgp dampening** コマンドは常に個々のアドレスファミリに対して使用してください。システムの通常動作中は、**clear bgp dampening** でアドレスファミリに対して **all** オプションを絶対に使用しないでください。たとえば `clear bgp ipv4 unicast dampening prefix x.x.x./y` を使用してください。

ルーティングテーブル更新時のポリシー適用

ルーティングテーブルにインストールされるルートにルーティングポリシーを適用するには、次の作業を実行します。

始める前に

テーブルポリシーのフィルタリングに使用可能なサポートされている属性と操作のリストについては、*Routing Configuration Guide for Cisco ASR 9000 Series Routers*（本書）の「でのルーティングポリシーの実装 Cisco ASR 9000 シリーズ ルータ 」のモジュールを参照してください。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **address-family** { **ipv4** \| **ipv6** } **unicast**
4. **table-policy** *policy-name*
5. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120.6`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。

	コマンドまたはアクション	目的
ステップ 3	**address-family** { **ipv4** | **ipv6** } **unicast** 例： `RP/0/RSP0/cpu 0: router(config-bgp)# address-family ipv4 unicast`	IPv4 または IPv6 のいずれかのアドレス ファミリを指定し、アドレス ファミリのコンフィギュレーション サブモードを開始します。 このコマンドのすべてのキーワードと引数のリストを参照するには、CLI ヘルプ（?）を使用します。
ステップ 4	**table-policy** *policy-name* 例： `RP/0/RSP0/cpu 0: router(config-bgp-af)# table-policy tbl-plcy-A`	ルーティング テーブルにインストールされるルートに、指定されたポリシーを適用します。
ステップ 5	**commit**	

BGP アドミニストレーティブ ディスタンスの設定

あるルートのクラスよりも別のルートのクラスを優先するために使用できるアドミニストレーティブ ディスタンスを使用することを指定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **address-family** { **ipv4** | **ipv6** } **unicast**
4. **distance bgp** *external-distance internal-distance local-distance*
5. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**address-family** { **ipv4** | **ipv6** } **unicast** 例： `RP/0/RSP0/cpu 0: router(config-bgp)# address-family ipv4 unicast`	IPv4 または IPv6 のいずれかのアドレス ファミリ ユニキャストを指定し、アドレス ファミリのコンフィギュレーション サブモードを開始します。 このコマンドのすべてのキーワードと引数のリストを参照するには、CLI ヘルプ（?）を使用します。

	コマンドまたはアクション	目的
ステップ 4	**distance bgp** *external-distance internal-distance local-distance* 例： `RP/0/RSP0/cpu 0: router(config-bgp-af)# distance bgp 20 20 200`	あるルートのクラスよりも別のルートのクラスを優先するために外部、内部、およびローカルのアドミニストレーティブディスタンスを設定します。値が高いほど、信頼性のランクは低くなります。
ステップ 5	**commit**	

BGP ネイバー グループおよびネイバーの設定

BGP ネイバー グループを設定し、ネイバーにネイバー グループの設定を適用するには、次の作業を実行します。ネイバー グループは、ネイバーに関連するアドレス ファミリから独立した設定とアドレス ファミリ固有の設定を持つテンプレートです。

ネイバー グループを設定すると、各ネイバーは、**use** コマンド経由で設定を継承できるようになります。ネイバー グループを使用するように設定されているネイバーは、デフォルトでネイバー グループの設定すべて（アドレス ファミリに依存しない設定とアドレス ファミリ固有の設定を含む）を継承します。継承された設定を上書きするには、ネイバーに対して直接コマンドを設定するか、または **use** コマンドを使用して、セッション グループ、またはアドレス ファミリ グループを設定します。

ネイバー グループではアドレス ファミリに依存しない設定を行うことができます。アドレス ファミリ固有の設定では、アドレス ファミリ サブモードを開始するようにネイバー グループのアドレス ファミリを設定する必要があります。

ネイバー グループ コンフィギュレーション モードでは、ネイバー グループについて、アドレス ファミリに依存しないパラメータを設定できます。ネイバー グループ コンフィギュレーション モードで **address-family** コマンドを使用します。

neighbor group コマンドを使用してネイバー グループ名を指定した後で、オプションをそのネイバー グループに割り当てることができます。

（注） 指定されたネイバー グループで設定できるコマンドはすべて、ネイバーでも設定できます。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **address-family** { **ipv4** | **ipv6** } **unicast**
4. exit
5. **neighbor-group** *name*
6. **remote-as** *as-number*
7. **address-family** { **ipv4** | **ipv6** } **unicast**

8. **route-policy** *route-policy-name* { **in** | **out** }
9. **exit**
10. **exit**
11. **neighbor** *ip-address*
12. **use neighbor-group** *group-name*
13. **remote-as** *as-number*
14. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**address-family** { **ipv4** | **ipv6** } **unicast** 例： `RP/0/RSP0/cpu 0: router(config-bgp)# address-family ipv4 unicast`	IPv4 または IPv6 のいずれかのアドレス ファミリ ユニキャストを指定し、アドレス ファミリのコンフィギュレーション サブモードを開始します。 このコマンドのすべてのキーワードと引数のリストを参照するには、CLI ヘルプ（?）を使用します。
ステップ 4	exit 例： `RP/0/RSP0/cpu 0: router(config-bgp-af)# exit`	現在のコンフィギュレーション モードを終了します。
ステップ 5	**neighbor-group** *name* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# neighbor-group nbr-grp-A`	ルータをネイバー グループ コンフィギュレーション モードにします。
ステップ 6	**remote-as** *as-number* 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# remote-as 2002`	ネイバーを作成し、リモート自律システム番号を割り当てます。
ステップ 7	**address-family** { **ipv4** | **ipv6** } **unicast** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# address-family ipv4 unicast`	IPv4 または IPv6 のいずれかのアドレス ファミリ ユニキャストを指定し、アドレス ファミリのコンフィギュレーション サブモードを開始します。 このコマンドのすべてのキーワードと引数のリストを参照するには、CLI ヘルプ（?）を使用します。

	コマンドまたはアクション	目的
ステップ 8	**route-policy** *route-policy-name* { **in** | **out** } 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp-af)# route-policy drop-as-1234 in`	（任意）指定したポリシーを着信 IPv4 ユニキャスト ルートに適用します。
ステップ 9	**exit** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp-af)# exit`	現在のコンフィギュレーション モードを終了します。
ステップ 10	**exit** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# exit`	現在のコンフィギュレーション モードを終了します。
ステップ 11	**neighbor** *ip-address* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 172.168.40.24`	BGP ルーティングのためにルータをネイバー コンフィギュレーション モードにして、ネイバーの IP アドレスを BGP ピアとして設定します。
ステップ 12	**use neighbor-group** *group-name* 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# use neighbor-group nbr-grp-A`	（任意）BGP ネイバーが指定されたネイバー グループから設定を継承することを指定します。
ステップ 13	**remote-as** *as-number* 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# remote-as 2002`	ネイバーを作成し、リモート自律システム番号を割り当てます。
ステップ 14	**commit**	

BGP のルートリフレクタの設定

BGP のルート リフレクタを設定するには、次の作業を実行します。

route-reflector-client コマンドで設定されるネイバーはすべてクライアント グループのメンバーであり、その他の iBGP ピアはローカル ルータ リフレクタの非クライアント グループのメンバーです。

ルート リフレクタは、そのクライアントとあわせてクラスタを形成します。クライアントからなるクラスタには通常、ルート リフレクタが 1 つ存在します。このようなインスタンスでは、

クラスタはソフトウェアにより、ルート リフレクタのルータ ID と認識されます。冗長性を高め、ネットワークでのシングルポイント障害を回避するために、クラスタに複数のリフレクタが含まれていることもあります。この場合、このクラスタのルート リフレクタはすべて、同じ 4 バイトのクラスタ ID を使って設定する必要があります。これはルート リフレクタが、同じクラスタに属する別のルート リフレクタからのアップデートを認識できるようにするためです。クラスタに複数のルータ リフレクタがある場合にクラスタ ID を設定するには、**bgp cluster-id** コマンドを使用します。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **bgp cluster-id** *cluster-id*
4. **neighbor** *ip-address*
5. **remote-as** *as-number*
6. **address-family** { **ipv4** | **ipv6** } **unicast**
7. **route-reflector-client**
8. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**bgp cluster-id** *cluster-id* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# bgp cluster-id 192.168.70.1`	クラスタに対応するルート リフレクタの 1 つとして、ローカル ルータを設定します。クラスタを識別するために、指定したクラスタ ID を設定します。
ステップ 4	**neighbor** *ip-address* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 172.168.40.24`	BGP ルーティングのためにルータをネイバー コンフィギュレーション モードにして、ネイバーの IP アドレスを BGP ピアとして設定します。
ステップ 5	**remote-as** *as-number* 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# remote-as 2003`	ネイバーを作成し、リモート自律システム番号を割り当てます。

	コマンドまたはアクション	目的
ステップ 6	**address-family** { **ipv4** | **ipv6** } **unicast** 例： `RP/0/RSP0/cpu 0: router(config-nbr)# address-family ipv4 unicast`	IPv4 または IPv6 のいずれかのアドレス ファミリ ユニキャストを指定し、アドレス ファミリのコンフィギュレーション サブモードを開始します。 このコマンドのすべてのキーワードと引数のリストを参照するには、CLI ヘルプ（?）を使用します。
ステップ 7	**route-reflector-client** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# route-reflector-client`	BGP ルート リフレクタとしてルータを設定し、そのクライアントとしてネイバーを設定します。
ステップ 8	**commit**	

ルート ポリシーによる BGP ルート フィルタリングの設定

ルート ポリシーによる BGP ルーティング フィルタリングを設定するには、次の作業を実行します。

始める前に

インバウンドおよびアウトバウンドのネイバー ポリシー フィルタリングで使用可能なサポートされている属性と操作のリストについては、『*Cisco ASR 9000* シリーズ アグリゲーション サービス ルータ ルーティング設定ガイド』（本書）の「*Cisco IOS XR* ソフトウェア Cisco ASR 9000 シリーズ ルータ」のモジュールを参照してください。

手順の概要

1. **configure**
2. **route-policy** *name*
3. **end-policy**
4. **router bgp** *as-number*
5. **neighbor** *ip-address*
6. **address-family** { **ipv4** \| **ipv6** } **unicast**
7. **route-policy** *route-policy-name* { **in** \| **out** }
8. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	

	コマンドまたはアクション	目的

ステップ 2	**route-policy** *name*
	例：
	`RP/0/RSP0/cpu 0: router(config)# route-policy drop-as-1234`
	`RP/0/RSP0/cpu 0: router(config-rpl)# if as-path passes-through '1234' then`
	`RP/0/RSP0/cpu 0: router(config-rpl)# apply check-communities`
	`RP/0/RSP0/cpu 0: router(config-rpl)# else`
	`RP/0/RSP0/cpu 0: router(config-rpl)# pass`
	`RP/0/RSP0/cpu 0: router(config-rpl)# endif`
		（任意）ルートポリシーを作成し、ルートポリシーコンフィギュレーションモードを開始します。このモードではルート ポリシーを定義できます。

ステップ 3	**end-policy** 例： `RP/0/RSP0/cpu 0: router(config-rpl)# end-policy`	（任意）ルート ポリシーの定義を終了し、ルート ポリシー コンフィギュレーション モードを終了します。
ステップ 4	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	自律システム番号を指定し、BGP コンフィギュレーションモードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 5	**neighbor** *ip-address* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 172.168.40.24`	BGP ルーティングのためにルータをネイバー コンフィギュレーション モードにして、ネイバーの IP アドレスを BGP ピアとして設定します。
ステップ 6	**address-family** { **ipv4** | **ipv6** } **unicast** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# address-family ipv4 unicast`	IPv4 または IPv6 のいずれかのアドレス ファミリ ユニキャストを指定し、アドレスファミリのコンフィギュレーション サブモードを開始します。 このコマンドのすべてのキーワードと引数のリストを参照するには、CLI ヘルプ（?）を使用します。
ステップ 7	**route-policy** *route-policy-name* { **in** | **out** } 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# route-policy drop-as-1234 in`	指定されたポリシーをインバウンドルートに適用します。
ステップ 8	**commit**	

BGP 属性フィルタリングの設定

BGP 属性フィルタリングを設定するには、次のタスクを実行します。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **attribute-filter group** *attribute-filter group name*
4. **attribute** *attribute code* { **discard** | **treat-as-withdraw** }

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 100`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**attribute-filter group** *attribute-filter group name* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# attribute-filter group ag_discard_med`	属性フィルタ グループ名を指定し、属性フィルタ グループ コンフィギュレーション モードを開始することで、BGP ネイバーに特定の属性フィルタ グループを設定できます。
ステップ 4	**attribute** *attribute code* { **discard** | **treat-as-withdraw** } 例： `RP/0/RSP0/cpu 0: router(config-bgp-attrfg)# attribute 24 discard`	単一またはある範囲の属性コードと関連するアクションを指定します。実行できるアクションには次のものがあります。 • Treat-as-withdraw：アップデート メッセージを取り消すかを検討します。対応する IPv4 ユニキャストまたは MP_REACH NLRI があれば、ネイバーの Adj-RIB-In から取り消します。 • Discard Attribute：この属性を廃棄します。一致した部分の属性は廃棄され、アップデート メッセージの残りの部分は正常に処理されます。

BGP ネクスト ホップ トリガー遅延の設定

BGP ネクスト ホップ トリガー遅延を設定するには、次の作業を実行します。ルーティング情報ベース（RIB）では変更の重大度に基づいてダンプニング通知が分類されます。イベント通知はクリティカルおよび非クリティカルとして分類されます。この作業では、クリティカルイベントと非クリティカル イベントの最小バッチ間隔を指定できます。

手順の概要

1. **configure**
2. **router bgp** *as-number*

3. **address-family** { **ipv4** | **ipv6** } **unicast**
4. **nexthop trigger-delay** { **critical** *delay* | **non-critical** *delay* }
5. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**address-family** { **ipv4**	**ipv6** } **unicast** 例： `RP/0/RSP0/cpu 0: router(config-bgp)# address-family ipv4 unicast`	IPv4 または IPv6 のいずれかのアドレス ファミリ ユニキャストを指定し、アドレス ファミリのコンフィギュレーション サブモードを開始します。 このコマンドのすべてのキーワードと引数のリストを参照するには、CLI ヘルプ（?）を使用します。
ステップ 4	**nexthop trigger-delay** { **critical** *delay*	**non-critical** *delay* } 例： `RP/0/RSP0/cpu 0: router(config-bgp-af)# nexthop trigger-delay critical 15000`	重要なネクスト ホップ トリガー遅延を設定します。
ステップ 5	**commit**	

BGP 更新でのネクストホップ処理の無効化

ネイバーに対するネクスト ホップの計算をディセーブルにし、BGP アップデートのネクスト ホップ フィールドにユーザ自身のアドレスを挿入するには、次の作業を実行します。ルートをアドバタイズするときに使用する最適なネクスト ホップの計算をディセーブルにすると、すべてのルートがネットワーク デバイスによってネクスト ホップとしてアドバタイズされます。

（注） ネクスト ホップ処理は、アドレス ファミリ グループ、ネイバー グループ、またはネイバー アドレス ファミリに対して無効にすることができます。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **neighbor** *ip-address*

4. **remote-as** *as-number*
5. **address-family** { **ipv4** | **ipv6** } **unicast**
6. **next-hop-self**
7. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**neighbor** *ip-address* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 172.168.40.24`	BGP ルーティングのためにルータをネイバー コンフィギュレーション モードにして、ネイバーの IP アドレスを BGP ピアとして設定します。
ステップ 4	**remote-as** *as-number* 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# remote-as 206`	ネイバーを作成し、リモート自律システム番号を割り当てます。
ステップ 5	**address-family** { **ipv4** | **ipv6** } **unicast** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# address-family ipv4 unicast`	IPv4 または IPv6 のいずれかのアドレス ファミリ ユニキャストを指定し、アドレス ファミリのコンフィギュレーション サブモードを開始します。 このコマンドのすべてのキーワードと引数のリストを参照するには、CLI ヘルプ（?）を使用します。
ステップ 6	**next-hop-self** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# next-hop-self`	指定されたネイバーにアドバタイズされるすべてのルートのネクスト ホップ属性をローカル ルータのアドレスに設定します。ルートをアドバタイズするときに使用する最適なネクストホップの計算をディセーブルにすると、すべてのルートがローカル ネットワーク デバイスによってネクスト ホップとしてアドバタイズされます。
ステップ 7	**commit**	

BGPコミュニティおよび拡張コミュニティアドバタイズメントの設定

コミュニティ属性および拡張コミュニティ属性を eBGP ネイバーに送信することを指定するには、次の作業を実行します。これらの属性は、デフォルトでは eBGP ネイバーに送信されません。これに対して、iBGP ネイバーには常に送信されます。ここでは、コミュニティ属性を送信できるようにする方法の例を示します。拡張コミュニティを送信できるようにするには、**send-community-ebgp** キーワードを **send-extended-community-ebgp** キーワードで置き換えます。

send-community-ebgp コマンドをネイバー グループまたはアドレス ファミリ グループに対して設定すると、このグループを使用するすべてのネイバーが設定を継承します。あるネイバーに対して特別にこのコマンドを設定すると、継承された値が上書きされます。

（注） BGP コミュニティと拡張コミュニティ フィルタリングは、iBGP ネイバーには設定できません。コミュニティと拡張コミュニティは、VPNv4、MDT、IPv4、および IPv6 アドレス ファミリでは常に iBGP ネイバーに送信されます。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **neighbor** *ip-address*
4. **remote-as** *as-number*
5. **address-family**{**ipv4** {**labeled-unicast** | **unicast** | **mdt** | **multicast** | **mvpn** | **tunnel**} | **ipv6** {**labeled-unicast** | **mvpn** | **unicast**}}
6. 次のいずれかのコマンドを使用します。
   - **send-community-ebgp**
   - **send-extended-community-ebgp**
7. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**neighbor** *ip-address* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 172.168.40.24`	BGP ルーティングのためにルータをネイバー コンフィギュレーション モードにして、ネイバーの IP アドレスを BGP ピアとして設定します。

	コマンドまたはアクション	目的
ステップ 4	**remote-as** *as-number* 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# remote-as 2002`	ネイバーを作成し、リモート自律システム番号を割り当てます。
ステップ 5	**address-family**{**ipv4** {**labeled-unicast** | **unicast** | **mdt** | **multicast** | **mvpn** | **tunnel**} | **ipv6** {**labeled-unicast** | **mvpn** | **unicast**}} 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# address-family ipv6 unicast`	指定のアドレス ファミリに対応しネイバー アドレス ファミリ コンフィギュレーション モードを開始します。**ipv4** または **ipv6** アドレス ファミリ キーワードと、指定したアドレス ファミリ サブ モード ID の 1 つを使用します。 IPv6 アドレス ファミリ モードでは、次のサブモードをサポートしています。 • **labeled-unicast** • **mvpn** • **unicast** IPv4 アドレス ファミリ モードでは、次のサブモードをサポートしています。 • **labeled-unicast** • **mdt** • **multicast** • **mvpn** • **rt-filter** • **tunnel** • **unicast** アドレス ファミリ サブモードのサポートの詳細については、 *Routing Command Reference for Cisco ASR 9000 Series Routers*の「*BGP Commands*」のモジュールの **address-family (BGP)** コマンドを参照してください。
ステップ 6	次のいずれかのコマンドを使用します。 • **send-community-ebgp** • **send-extended-community-ebgp** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# send-community-ebgp` または	ルータが（デフォルトでは eBGP ネイバーでディセーブルにされている）コミュニティ属性と拡張コミュニティ属性を指定された eBGP ネイバーに送信することを指定します。

	コマンドまたはアクション	目的
	`RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# send-extended-community-ebgp`	
ステップ 7	**commit**	

BGP コスト コミュニティの設定

BGP コスト コミュニティを設定するには、次のタスクを実行します。

BGP は同一宛先への複数のパスを受信し、最適パス アルゴリズムを使用して RIB にインストールする最適なパスを決定します。ユーザが部分比較後に出力点を決定できるようにするため、最適パス選択処理で同等パスのタイブレークのためにコスト コミュニティが定義されます。

手順の概要

1. **configure**
2. **route-policy** *name*
3. **set extcommunity cost** { *cost-extcommunity-set-name* | *cost-inline-extcommunity-set* } [**additive**]
4. **end-policy**
5. **router bgp** *as-number*
6. 次のいずれかを実行します。
   - **default-information originate**
   - **aggregate-address** *address/mask-length* [**as-set**] [**as-confed-set**] [**summary-only**] [**route-policy** *route-policy-name*]
   - **address-family** { **ipv4 unicast** | **ipv4 multicast** | **ipv4 tunnel** | **ipv6 unicast** | **vpnv4 unicast** } **redistribute connected** [**metric** *metric-value*] [**route-policy** *route-policy-name*]
   - **address-family** { **ipv4 unicast** | **ipv4 multicast** | **ipv4 tunnel** | **ipv6 unicast** | **vpnv4 unicast** } **redistribute eigrp** *process-id* [**match** { **external** | **internal** }] [**metric** *metric-value*] [**route-policy** *route-policy-name*]
   - **address-family** { **ipv4 unicast** | **ipv4 multicast** | **ipv4 tunnel** | **ipv6 unicast** | **vpnv4 unicast** } **redistribute isis** *process-id* [**level** { **1** | **1-inter-area** | **2** }] [**metric** *metric-value*] [**route-policy** *route-policy-name*]
   - **address-family** { **ipv4 unicast** | **ipv4 multicast** | **ipv4 tunnel** | **ipv6 unicast** | **vpnv4 unicast** } **redistribute ospf** *process-id* [**match** { **external** [**1** | **2**] | **internal** | **nssa-external** [**1** | **2**]}] [**metric** *metric-value*] [**route-policy** *route-policy-name*]
7. 次のいずれかを実行します。
   - **address-family** { **ipv4 unicast** | **ipv4 multicast** | **ipv4 tunnel** | **ipv4 mdt** | **ipv6 unicast** | **ipv6 multicast** | **vpnv4 unicast** | **vpnv6 unicast** } **redistribute ospfv3** *process-id* [**match** { **external** [**1** | **2**] | **internal** | **nssa-external** [**1** | **2**]}] [**metric** *metric-value*] [**route-policy** *route-policy-name*]
   - **address-family** { **ipv4 unicast** | **ipv4 multicast** | **ipv4 tunnel** | **ipv4 mdt** | **ipv6 unicast** | **ipv6 multicast** | **vpnv4 unicast** | **vpnv6 unicast** } **redistribute rip** [**metric** *metric-value*] [**route-policy** *route-policy-name*]

- **address-family** { **ipv4 unicast** | **ipv4 multicast** | **ipv4 tunnel** | **ipv4 mdt** | **ipv6 unicast** | **ipv6 multicast** | **vpnv4 unicast** | **vpnv6 unicast** } **redistribute static** [**metric** *metric-value*] [**route-policy** *route-policy-name*]
- **address-family** { **ipv4 unicast** | **ipv4 multicast** | **ipv4 tunnel** | **ipv4 mdt** | **ipv6 unicast** | **ipv6 multicast** | **vpnv4 unicast** | **vpnv6 unicast** } **network** { *ip-address/prefix-length* | *ip-address mask* } [**route-policy** *route-policy-name*]
- **neighbor** *ip-address* **remote-as** *as-number* **address-family** { **ipv4 unicast** | **ipv4 multicast** | **ipv4 tunnel** | **ipv4 mdt** | **ipv6 unicast** | **ipv6 multicast** | **vpnv4 unicast** | **vpnv6 unicast** }
- **route-policy** *route-policy-name* { **in** | **out** }

8. **commit**
9. **show bgp** [**vrf** *vrf-name*] *ip-address*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**route-policy** *name* 例： `RP/0/RSP0/cpu 0: router(config)# route-policy costA`	ルート ポリシー コンフィギュレーション モードに切り替え、設定するルート ポリシーの名前を指定します。
ステップ 3	**set extcommunity cost** { *cost-extcommunity-set-name* | *cost-inline-extcommunity-set* } [**additive**] 例： `RP/0/RSP0/cpu 0: router(config)# set extcommunity cost cost_A`	コストの BGP 拡張コミュニティ属性を指定します。
ステップ 4	**end-policy** 例： `RP/0/RSP0/cpu 0: router(config)# end-policy`	ルート ポリシーの定義を終了して、ルート ポリシー コンフィギュレーション モードを終了します。
ステップ 5	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	BGP コンフィギュレーション モードを開始します。このモードでは BGP ルーティング プロセスを設定できます。
ステップ 6	次のいずれかを実行します。 • **default-information originate** • **aggregate-address** *address/mask-length* [**as-set**] [**as-confed-set**] [**summary-only**] [**route-policy** *route-policy-name*] • **address-family** { **ipv4 unicast** | **ipv4 multicast** | **ipv4 tunnel** | **ipv6 unicast** | **vpnv4 unicast** }	コスト コミュニティを付加ポイント（ルート ポリシー）に適用します。

	コマンドまたはアクション	目的
	redistribute connected [**metric** *metric-value*] [**route-policy** *route-policy-name*] • **address-family** { **ipv4 unicast** | **ipv4 multicast** | **ipv4 tunnel** | **ipv6 unicast** | **vpnv4 unicast** } **redistribute eigrp** *process-id* [**match** { **external** | **internal** }] [**metric** *metric-value*] [**route-policy** *route-policy-name*] • **address-family** { **ipv4 unicast** | **ipv4 multicast** | **ipv4 tunnel** | **ipv6 unicast** | **vpnv4 unicast** } **redistribute isis** *process-id* [**level** { **1** | **1-inter-area** | **2** }] [**metric** *metric-value*] [**route-policy** *route-policy-name*] • **address-family** { **ipv4 unicast** | **ipv4 multicast** | **ipv4 tunnel** | **ipv6 unicast** | **vpnv4 unicast** } **redistribute ospf** *process-id* [**match** { **external** [**1** | **2**] | **internal** | **nssa-external** [**1** | **2**]}] [**metric** *metric-value*] [**route-policy** *route-policy-name*]	
ステップ 7	次のいずれかを実行します。 • **address-family** { **ipv4 unicast** | **ipv4 multicast** | **ipv4 tunnel** | **ipv4 mdt** | **ipv6 unicast** | **ipv6 multicast** | **vpnv4 unicast** | **vpnv6 unicast** } **redistribute ospfv3** *process-id* [**match** { **external** [**1** | **2**] | **internal** | **nssa-external** [**1** | **2**]}] [**metric** *metric-value*] [**route-policy** *route-policy-name*] • **address-family** { **ipv4 unicast** | **ipv4 multicast** | **ipv4 tunnel** | **ipv4 mdt** | **ipv6 unicast** | **ipv6 multicast** | **vpnv4 unicast** | **vpnv6 unicast** } **redistribute rip** [**metric** *metric-value*] [**route-policy** *route-policy-name*] • **address-family** { **ipv4 unicast** | **ipv4 multicast** | **ipv4 tunnel** | **ipv4 mdt** | **ipv6 unicast** | **ipv6 multicast** | **vpnv4 unicast** | **vpnv6 unicast** } **redistribute static** [**metric** *metric-value*] [**route-policy** *route-policy-name*] • **address-family** { **ipv4 unicast** | **ipv4 multicast** | **ipv4 tunnel** | **ipv4 mdt** | **ipv6 unicast** | **ipv6 multicast** | **vpnv4 unicast** | **vpnv6 unicast** } **network** { *ip-address/prefix-length* | *ip-address mask* } [**route-policy** *route-policy-name*] • **neighbor** *ip-address* **remote-as** *as-number* **address-family** { **ipv4 unicast** | **ipv4 multicast** | **ipv4 tunnel** | **ipv4 mdt** | **ipv6 unicast** | **ipv6 multicast** | **vpnv4 unicast** | **vpnv6 unicast** } • **route-policy** *route-policy-name* { **in** | **out** }	

	コマンドまたはアクション	目的
ステップ 8	**commit**	
ステップ 9	**show bgp** [**vrf** *vrf-name*] *ip-address* 例： `RP/0/RSP0/cpu 0: router# show bgp 172.168.40.24`	コスト コミュニティを次の形式で表示します。 Cost: *POI* : *cost-community-ID* : *cost-number*

ネイバーからのソフトウェアツーストア更新の設定

ネイバーからソフトウェアツーストア更新を受信するように設定するには、次の作業を実行します。

ネイバーがルート リフレッシュに対応している場合は、soft-reconfiguration inbound コマンドによって、ルート リフレッシュ要求がネイバーに送信されるようになります。ネイバーがルート リフレッシュに対応していない場合は、ネイバーが受信ルートを再学習するようにするため、**clear bgp soft** コマンドを使用してネイバーをリセットする必要があります。BGP インバウンド ソフト リセットを使用したネイバーのリセット （168 ページ）を参照してください。

（注） ネイバーからのアップデートの保存は、ネイバーがルート リフレッシュに対応しているか、soft-reconfiguration inbound コマンドが設定されている場合にだけ機能します。ネイバーがルート リフレッシュに対応しており、**soft-reconfiguration inbound** コマンドが設定されていても、このコマンドで **always** オプションが使用されていない場合は元のルートは格納されません。元のルートはルート リフレッシュ要求によって容易に復元できます。ルート リフレッシュは、ルーティング情報を再送信するためにピアに要求を送信します。soft-reconfiguration inbound コマンドは、変更されていない形式でピアから受信したすべてのパスを保存し、クリアする際にこれらの保存されたパスを参照します。ソフト再設定はメモリに負荷がかかる処理です。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **neighbor** *ip-address*
4. **address-family** { **ipv4** | **ipv6** } **unicast**
5. **soft-reconfiguration inbound** [**always**]
6. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	

	コマンドまたはアクション	目的
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**neighbor** *ip-address* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 172.168.40.24`	BGP ルーティングのためにルータをネイバー コンフィギュレーション モードにして、ネイバーの IP アドレスを BGP ピアとして設定します。
ステップ 4	**address-family** { **ipv4** | **ipv6** } **unicast** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# address-family ipv4 unicast`	IPv4 または IPv6 のいずれかのアドレス ファミリ ユニキャストを指定し、アドレスファミリのコンフィギュレーション サブモードを開始します。 このコマンドのすべてのキーワードと引数のリストを参照するには、CLI ヘルプ（?）を使用します。
ステップ 5	**soft-reconfiguration inbound** [**always**] 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# soft-reconfiguration inbound always`	指定したネイバーから受信したアップデートを格納するようにソフトウェアを設定します。ソフト再設定インバウンドを設定すると、ソフトウェアは変更またはフィルタ処理されたルートのほかに、元の変更されていないルートを格納することになります。これにより、インバウンドポリシーの変更後に「ソフト クリア」を実行できるようになります。 ソフト再設定により、ピアがルート フレッシュに対応していない場合、ソフトウェアはポリシー適用前に受信した更新を格納できます（対応している場合は更新のコピーが格納されます）。 **always** キーワードを使用すると、ルート リフレッシュがピアでサポートされている場合でも、ソフトウェアにコピーが格納されます。
ステップ 6	**commit**	

BGP パーシステンス

BGPパーシステンスにより、ローカルルータは、ネイバーセッションがダウンした後でも、設定されたネイバーから学習したルートを保持できます。BGPパーシステンスは、長期的グレースフル リスタート（LLGR）とも呼ばれます。LLGRはグレースフルリスタート（GR）が終了した後、またはGR が有効になっていない場合はただちに有効になります。LLGR は、LLGR の失効タイマーが期限切れになったとき、またはネイバーがルートを改訂した後にEnd-of-RIB マーカーを送信したときに終了します。ネイバーのLLGRが終了すると、そのネイバーからのルートのうち、失効状態のままであるルートはすべて削除されます。LLGR機能は、ネイバー

に設定されている場合、BGP OPEN メッセージでそのネイバーに通知されます。LLGR は、グレースフルリスタートとは次のように異なります。

- GR よりも長時間有効にできます。
- LLGR 失効ルートはルート選択（ベストパス計算）時の優先順位が最も低くなります。
- LLGR 失効ルートは、ベストパスとして選択されている場合に、接続されている LLGR_STALE コミュニティを使用してアドバタイズされます。LLGR に対応していないルータには、まったくアドバタイズされません。
- ネイバーへの転送パスがダウンしていることが検出された場合、LLGR 失効ルートは削除されません。
- ネイバーがルートを再度アドバタイズしない場合でも、ネイバーへの BGP セッションが複数回ダウンしても LLGR 失効ルートは削除されません。
- NO_LLGR コミュニティを持つルートは保持されません。

BGPは、ネイバーがBGPパーシステンス機能をネゴシエートするまで、コミュニティ65535:6、65535:7 を含む更新をネイバーに渡しません。コミュニティ 65535:6 と 65535:7 はそれぞれ LLGR_STALE と NO_LLGR 用に予約されていますが、リリース5.2.2 より前にこれらのコミュニティを設定している場合は、BGP の動作が予測できない場合があります。コミュニティ 65535:6 と 65535:7 は設定しないことをお勧めします。

BGP パーシステンス機能は次の AFI でのみサポートされています。

- VPNv4 と VPNv6
- RT 制約
- フロー スペック（IPv4、IPv6、VPNv4、VPNv6）
- プライベート IPv4 および IPv6（IPv4/v6 アドレスファミリ内部 VRF）

BGP 永続化設定：例

次に、BGP ネイバー3.3.3.3 で長時間グレースフルリスタート（LLGR）の失効時間を16777215 に設定する例を示します。

```
router bgp 100
 neighbor 3.3.3.3
  remote-as 30813
  update-source Loopback0
  graceful-restart stalepath-time 150
  address-family vpnv4 unicast
   long-lived-graceful-restart capable
   long-lived-graceful-restart stale-time send 16777215 accept 16777215
  !
  address-family vpnv6 unicast
   long-lived-graceful-restart capable
   long-lived-graceful-restart stale-time send 16777215 accept 16777215
```

BGP グレースフルメンテナンス

BGP リンクまたはルータがダウンすると、ネットワーク内の他のルータは、障害が発生したルータを通過していたトラフィックに代替パスがある場合は、そのパスを検索します。関係するすべてのルータが代替パスに関して一致するまでに必要な時間をコンバージェンス時間といいます。コンバージェンス時間の間に、ダウンしているルータまたはリンクに送信されるトラフィックがドロップされます。BGP グレースフルメンテナンス機能により、ルータまたはリンクが動作を停止する前に、ネットワークでコンバージェンスを実行できます。ネットワークが代替パスにトラフィックを再ルーティングする間、ルータまたはリンクはサービス状態に維持されます。影響を受けるルータまたはリンクにまだ到達していないトラフィックは、以前と同様に引き続き配信されます。すべてのトラフィックが再ルーティングされた後は、ルータまたはリンクを安全に動作を停止させることができます。

グレースフルメンテナンス機能は、代替パスが存在し、プライマリパスが取り消された時点でこれらの代替パスがルータにとって不明である場合に役立ちます。この機能は、プライマリパスが取り消される前に、これらの代替パスを提供します。この機能は、コンバージェンス時間が長いネットワークに最適です。大規模なルーティングテーブルやルートリフレクタの存在などのいくつかの要因によって、コンバージェンス時間が長くなる可能性があります。

BGP ルータまたはリンクがサービスに組み込まれると、コンバージェンス中にトラフィックが失われる可能性もありますが、ルータまたはリンクが動作を停止した場合よりも低くなります。BGP グレースフルメンテナンス機能はこのシナリオでも使用できます。

BGP グレースフルメンテナンスの制約事項

BGP グレースフルメンテナンスには、次の制約事項が適用されます。

- 影響を受けるルータが GSHUT コミュニティ属性を送信するように設定されている場合、そのルータを受信するネットワーク内の他のルータは、それを解釈するように設定する必要があります。コミュニティとルーティングポリシーを一致させ、より低い優先順位を設定する必要があります。
- LOCAL_PREF 属性は別の AS には送信されません。そのため、eBGP リンクでは LOCAL_PREF オプションを使用できません。

（注） この制約事項は、AS コンフェデレーションのメンバと AS 間の eBGP リンクには適用されません。

- 代替ルートがネットワーク内に存在している必要があります。存在していない場合は、低い優先順位をアドバタイズしても効果はありません。たとえば、代替ルートを持たないシングルホーム接続のカスタマールータのグレースフルメンテナンスを設定する利点はありません。
- 送信側ルータの出力または受信側ルータの入力のいずれかで、時間を消費するポリシーが存在する場合は、グレースフルメンテナンス動作は時間がかかることがあります。

- eBGP ASBR ネイバーを設定すると、BGP を介して直接接続されたルートに対して暗黙的ヌルラベルがアドバタイズされます。ユーザが eBGP ネイバーをシャットダウンした場合、システムの取り消しがネイバー状態の変更を書き換えるため、ラベルは再プログラミングされません。暗黙的ヌルラベル機能のサポートにより、ネイバーフラップの上書きの追加または削除の観点から、チャーンを回避するのに役立ちます。

グレースフルメンテナンスの動作

グレースフルメンテナンスがアクティブになると、影響を受けるルートは優先順位を下げて再度アドバタイズされます。そのため、隣接するルータが代替ルートを選択することになります。次のいずれかの方法を使用して、ルート優先順位の低下を通知します。

- **GSHUT コミュニティの追加**：リモートルータが優先順位を自由に設定できるようにするには、この方法を使用します。受信側ルータは、ポリシー内のこのコミュニティと一致しており、それ自体の優先順位を設定する必要があります。

- **LOCAL_PREF 値の低減**：内部 BGP ネイバーに対して機能します。リモートルータが GSHUT コミュニティと一致しない場合は、この方法を使用します。

- **AS パスを前に付加**：内部および外部の両方の BGP ネイバーに対して機能します。リモートルータが GSHUT コミュニティと一致しない場合は、この方法を使用します。

グレースフルメンテナンスが BGP 接続でアクティブになると、次の 2 つの動作が発生します。

1. 接続から受信したすべてのルートが優先順位の低い他のネイバーに再度アドバタイズされます。これは、実際に他のネイバーにアドバタイズされたルートに対してのみ実行されます。受信したルートがベストパスとして選択されていないためアドバタイズされていなかった可能性があります。この場合、再アドバタイズされません。
2. 接続にアドバタイズされたすべてのルートが優先順位の低いものから再アドバタイズされます。

最初の動作が実行されるようにするために、接続から受信したすべてのルートが graceful-shut という内部属性でタグ付けされます。この属性は、ルータにのみ内部的に分類され、BGP はアドバタイズしません。この属性は、**show bgp** コマンドを使用してルートを表示した場合に表示されます。これは GSHUT コミュニティとは異なります。GSHUT コミュニティは BGP によってアドバタイズされ、**show bgp** コマンドを使用してルートを表示したときにコミュニティリストに表示されます。

graceful-shut 属性を持つすべてのルートには、ルート選択時に最低の優先順位が与えられます。グレースフルメンテナンスでの BGP セッションで送信または受信した新しいルート更新も、前述のように処理されます。

相互自律システム

パブリックインターネット内の別の AS に低い優先順位をアドバタイズすると、遠くのネットワークで不要なルーティングアドバタイズメントが発生する場合がありますが、これは望まし

くありません。ルータが GSHUT コミニティから eBGP ネイバーへ発信するには、ネイバーアドレスファミリへの追加設定（**send-community-gshut-ebgp**）が必要です。

（注） これは、受信した時点でこのコミュニティをすでに備えているルータの GSHUT コミニュティには影響しません。このルータが GSHUT を追加したときにのみ、そのコミニュティに影響を与えます。

自動シャットダウンなし

グレースフルメンテナンス機能は、シャットダウンを実行しません。グレースフルメンテナンスが設定されている場合は、システムの再起動によっても設定されたままになります。これは、ルータまたはBGPネイバーのシャットダウンとともに使用することを目的としています。オペレータは、必要な場合は常に明示的にシャットダウンする必要があります。グレースフルメンテナンスが不要になったら、オペレータが明示的にグレースフルメンテナンスを非アクティブ化する必要があります。グレースフルメンテナンスは、シャットダウンが完了した後か、または非アクティブ化されたファシリティが再び起動した後に、非アクティブにできます。起動操作によってグレースフルメンテナンスを有効にしたままにするかどうかは、起動操作時に一時的なルーティングが問題であるかどうかによって異なります。

グレースフルメンテナンス後のシャットダウンのタイミング

グレースフルメンテナンスのアクティブ化の結果として、ネットワークが収束した後にルータまたはリンクをシャットダウンできます。コンバージェンスに 1 秒未満しかかからない場合と、1 時間以上かかる場合があります。残念ながら、単一のルータは、ネットワーク全体がいつ収束したかを認識できません。グレースフルメンテナンスのアクティブ化の後、更新の送信を開始するまでに数秒かかることがあります。また、**show bgp <vrf> <afi> <safi> summary** コマンドの出力のネイバーの「InQ」と「OutQ」は、BGP メッセージングのレベルを示しています。コンバージェンス後は、InQ と OutQ の両方が 0 になる必要があります。ネイバーはトラフィックの送信を停止させる必要があります。ただし、代替パスがない場合、トラフィックの送信が停止されることはありません。この場合、トラフィック損失を防ぐことはできません。

BGP ルータ（すべてのネイバー）でのグレースフルメンテナンスのアクティブ化

グレースフルメンテナンスを BGP ルータでアクティブにすることで、すべてのネイバーに対して **graceful-maintenance** に **activate** が設定されることになります。この 1 つの設定で、**graceful-maintenance** が設定されているすべてのネイバーに移動し、そこに **activate** を追加するのと同じ結果が得られます。キーワードの **all-neighbors** を追加し、それにより **graceful-maintenance activate all-neighbors** となった場合、ルータは、すべてのネイバーに **graceful-maintenance activate** を設定したかのように動作します。

（注） すべてのネイバーのすべてのルートに GSHUT コミュニティの送信ができる場合にのみ、BGP ルータインスタンスでグレースフルメンテナンスをアクティブにすることをお勧めします。すべてのネイバーへのすべてのルートを再送信すると、大規模なルータでは著しい時間がかかることがあります。代替ルートを持たないネイバーへの GSHUT の送信は無意味です。ルータにこのようなネイバーが多数ある場合は、それらのネイバーでグレースフルメンテナンスをアクティブにしないことによって、多くの時間を節約できます。

BGP グレースフルメンテナンス機能を使用すると、単一のネイバー、BGP セッション全体のネイバーグループ、またはすべてのネイバーで、グレースフルメンテナンスを有効にすることができます。ネイバーサブモードでグレースフルメンテナンスを有効にするには、次の 2 つの点を考慮します。

1. グレースフルシャットダウン属性を持つこのネイバーにアドバタイズされたすべてのルートは、GSHUT コミュニティを使用してそのネイバーにアドバタイズされます。
2. グレースフル メンテナンス コンフィギュレーション モードを開始して、さらに設定ができるようにします。

グレースフルメンテナンスで **activate** キーワードを使用すると、次のようになります。

1. このネイバーから受信したすべてのルートがグレースフルシャットダウン属性を取得します。
2. このネイバーにアドバタイズされたすべてのルートは、GSHUT コミュニティを使用してそのネイバーに再アドバタイズされます。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **graceful-maintenance activate** ［ **all-neighbors** | **retain-routes** ］
4. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	BGP AS 番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**graceful-maintenance activate** ［ **all-neighbors**	**retain-routes** ］ 例：	ネイバーで設定されているように、g-shut コミュニティとその他の属性を持つルートをアナウンスします。これにより、ネイバーはこのルータからのルートを拒否し、代替を選択します。これにより、ルー

	コマンドまたはアクション	目的
	`RP/0/RSP0/cpu 0: router(config-bgp)# graceful-maintenance activate all-neighhbors`	タをグレースフルにするか、または非稼働状態にすることができます。 **all-neighbors** キーワードを使用した場合、グレースフルメンテナンスはアクティブ化されていないネイバーに対してもアクティブになります。**retain-routes** を選択すると、BGP プロセスが停止したときに、RIB が BGP ルートを保持するようになります。 ルータ全体ではなくBGPのみをダウンさせる必要がある場合や、ローカルBGPのメンテナンス時に隣接するルータが動作し続けることがわかっている場合は、**retain-routes** オプションを使用します。別のプロトコルまたはデフォルトルートによって提供される代替ルートがRIBにある場合は、BGPプロセスが停止した後にBGPルートを保持しないことを推奨します。
ステップ 4	**commit**	

次のタスク

グレースフルメンテナンスをアクティブにした後は、すべてのルートが送信されるのを待ってから、隣接しているネイバーが、ルータまたはメンテナンス中のリンクからトラフィックをリダイレクトするようにする必要があります。トラフィックがリダイレクトされた後は、ルータまたはリンクをサービスに復帰させても差し支えありません。すべてのルートがいつ送信されたのかを明確に知る方法はありませんが、 **show bgp summary** コマンドを使用してネイバーの OutQ を確認できます。OutQ が値 0 に達すれば、送信されるアップデートはありません。

単一のネイバーでのグレースフルメンテナンスのアクティブ化

単一のネイバーに対してグレースフルメンテナンスをアクティブにするには、次の手順を実行します。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **neighbor** *ip-address*
4. **graceful-maintenance activate**
5. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	

	コマンドまたはアクション	目的
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	BGP AS 番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**neighbor** *ip-address* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 172.168.40.24`	BGP ルーティングのためにルータをネイバー コンフィギュレーション モードにして、ネイバーの IP アドレスを BGP ピアとして設定します。
ステップ 4	**graceful-maintenance activate** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# graceful-maintenance activate`	グレースフルメンテナンス属性を持つルートをアナウンスします。
ステップ 5	**commit**	

ネイバーグループのグレースフルメンテナンスのアクティブ化

ネイバーのグループでグレースフルメンテナンスをアクティブにするには、次の手順を実行します。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **neighbor-group** *Neighbor-group name*
4. **graceful-maintenance activate**
5. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	BGP AS 番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**neighbor-group** *Neighbor-group name* 例：	ルータをネイバー グループ コンフィギュレーション モードにします。

	コマンドまたはアクション	目的
	`RP/0/RSP0/cpu 0: router(config-bgp)# neighbor-group AS_1`	
ステップ 4	**graceful-maintenance activate** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbrgrp)# graceful-maintenance activate`	グレースフルメンテナンス属性を持つルートをアナウンスします。
ステップ 5	**commit**	

次のタスク

GSHUT コミュニティを追加するには、このルータの eBGP ネイバーのネイバーアドレスファミリに、 **send-community-gshut-ebgp** コマンドを設定する必要があります。

（注） GSHUT コミュニティの送信は、eBGP ネイバーのすべてのアドレスファミリでも望ましいとは限りません。特定のアドレスファミリセットを GSHUT コミュニティの対象にするには、 **send community-gshut-ebgp** コマンドを使用します。

ルートの優先順位を下げるためのルータへの指示

BGP グレースフルメンテナンス機能は、代替パスの可用性がある場合にのみ動作します。代替ルートがリンクまたはルータを停止する前に引き継ぐことができるように、より低い優先順位のルートをアドバタイズする必要があります。ルートの優先順位を変更するには、次の手順を実行します。

（注） グレースフルメンテナンスの属性は、アウトバウンドポリシーが適用された後に、ルート更新メッセージに追加されます。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **neighbor** *ip-address*
4. **remote-as** *as-number*
5. **graceful-maintenance** **as-prepends** *value*| **local-preference** *value*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	BGP AS 番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**neighbor** *ip-address* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 172.168.40.24`	BGP ルーティングのためにルータをネイバー コンフィギュレーション モードにして、ネイバーの IP アドレスを BGP ピアとして設定します。
ステップ 4	**remote-as** *as-number* 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# remote-as 2002`	ネイバーを作成し、リモート自律システム番号を割り当てます。
ステップ 5	**graceful-maintenance as-prepends** *value*| **local-preference** *value* 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# graceful-maintenance local-preference 4`	ローカル AS 番号がルートの AS パスの先頭に追加され、ルートに指定されたローカルの優先順位の値を使用して GSHUT コミュニティをアドバタイズする回数を指定します。ルータが GSHUT コミュニティをアドバタイズするときにルートに追加するときに、LOCAL_PREF 属性も変更して、コマンドに指定されているローカル AS 番号を先頭に追加します。GSHUT を送信することで、ネイバールータが低い優先順位を処理する方法に柔軟性がもたらされます。そのため、ルートポリシーで照合したうえで、最適な処理を行うことができます。一方、単純なネットワークでは、他の場所でルートポリシーを作成するよりも、ローカルの優先順位を 0 に設定する方が簡単です。 （注） LOCAL_PREF は実際の eBGP ネイバーには送信されませんが、コンフェデレーション AS eBGP ネイバーに送信されます。eBGP ネイバーの優先順位を下げるには、as-prepends の値を入力する必要があります。

例：ルートポリシーと一致する GSHUT コミュニティを設定してルートの優先順位を下げる

```
route-policy gshut
  if community matches-any gshut then
    set local-preference 0
  endif
  pass
end-policy


neighbor 666.0.0.3
    address-family ipv4 unicast
      route-policy gshut in
```

（注） GSHUT ネイバーから受信したルートは、GSHUT 属性でマークされ、GSHUT コミュニティを使用して受信したルートと区別されます。ネイバーがメンテナンスから除外されると、そのパスの属性は削除されますが、コミュニティでは削除されません。この属性は内部的なものであり、BGP メッセージでは送信されません。パス選択時にルートを拒否するために使用されます。

ルータまたはリンクの動作の再開

ルータまたはリンクの動作を再開させる前に、グレースフルメンテナンスを最初にアクティブにしてから、**activate** 設定を削除する必要があります。

BGP グレースフルメンテナンスを確認するための show コマンドの出力

この項では、BGP グレースフルメンテナンスがアクティブになっていることを確認し、関連する属性を確認するために使用できる show コマンドを示します。

BGP グレースフルメンテナンスがアクティブになっている場合にグレースフルシャットダウンコミュニティとグレースフルシャットパスの属性を表示するには、**show bgp <IP address>** コマンドを使用します。

```
RP/0/0/CPU0:R4#show bgp 5.5.5.5
...
10.10.10.1 from 10.10.10.1 (192.168.0.5)
Received Label 24000
Origin incomplete, metric 0, localpref 100, valid, internal, best, group-best,
import-candidate
Received Path ID 0, Local Path ID 1, version 4
Community: graceful-shutdown
Originator: 192.168.0.5, Cluster list: 192.168.0.1
```

次の **show bgp community graceful-shutdown** コマンドの出力例には、グレースフル メンテナンス機能が表示されています。

```
RP/0/0/CPU0:R4#show bgp community graceful-shutdown
BGP router identifier 192.168.0.4, local AS number 4
BGP generic scan interval 60 secs
BGP table state: Active
```

```
Table ID: 0xe0000000 RD version: 18
BGP main routing table version 18
BGP scan interval 60 secs
Status codes: s suppressed, d damped, h history, * valid, > best
i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
Network Next Hop Metric LocPrf Weight Path
* 5.5.5.5/32 10.10.10.1 88 0 1 ?
Processed 1 prefixes, 1 paths
```

次に、グレースフルメンテナンス機能の属性を表示するために、IPアドレスと設定引数およびキーワードを指定した **show bgp neighbors** コマンドの出力例を示します

```
RP/0/0/CPU0:R1#show bgp neighbor 12.12.12.5
...
Graceful Maintenance locally active, Local Pref=45, AS prepends=3
...
For Address Family: IPv4 Unicast
...
GSHUT Community attribute sent to this neighbor
...
********************************************************************************
RP/0/0/CPU0:R1#show bgp neighbor 12.12.12.5 configuration
neighbor 12.12.12.5
remote-as 1 []
graceful-maintenance 1 []
gr-maint local-preference 45 []
gr-maint as-prepends 3 []
gr-maint activate []
```

次に、グレースフルメンテナンス機能の属性が表示される **show rpl community-set** コマンドの出力例を示します。

```
RP/0/0/CPU0:R5#show rpl community-set
Listing for all Community Set objects
community-set gshut
graceful-shutdown
end-set
```

次に、グレースフルメンテナンスがアクティブになっている BGP ネイバーが起動したときに発行されるsyslogの例を示します。これは、コンバージェンス後にグレースフルメンテナンスを非アクティブ化するように通知する警告テキストです。

```
RP/0/0/CPU0:Jan 28 22:01:36.356 : bgp[1056]: %ROUTING-BGP-5-ADJCHANGE : neighbor 10.10.10.4
 Up (VRF: default) (AS: 4)
WARNING: Graceful Maintenance is Active
```

L3VPN iBGP PE-CE

L3VPN iBGP PE-CE 機能は、プロバイダー エッジ（PE）デバイスとカスタマー エッジ（CE）デバイス間でBGPルーティング情報を交換するiBGP（内部Border Gateway Protocol）セッションの確立に役立ちます。2 つの BGP ピア間の BGP セッションは、それらの BGP ピアが同じ自律システム内に存在する場合に、iBGP セッションと呼ばれます。

L3VPN iBGP PE-CE の概要

プロバイダーエッジ（PE）またはカスタマーエッジ（CE）のルーティングプロトコルとしてBGPを使用すると、VPNプロバイダー自律システム（AS）とカスタマー ネットワーク自律シ

ステム間の外部ピアリングとしてピエアリングセッションが設定されます。L3VPN iBGP PE-CE 機能では、PE デバイスと CE デバイスが、PE と CE 間で広く使用されている外部 BGP ピアリングの代わりに内部 ボーダー ゲートウェイ プロトコル（iBGP）としてピアリングを行って Border Gateway Protocol（BGP）ルーティング情報を交換できます。このメカニズムは、VRF ベースの CE が iBGP として設定されている各 PE デバイスで適用されます。これにより、サービス プロバイダー（SP）は、CE に自律システムのオーバーライドを設定する必要がなくなります。この機能を有効にした場合は、異なる自律システムを使用した仮想プライベート ネットワーク（VPN）サイトの設定は不要です。

neighbor internal-vpn-client コマンドを使用すると、PE デバイスが VPN クラウド全体を CE デバイスに対して内部 VPN クライアントとして動作させることができます。これらの CE デバイスは、VRF 内部の iBGP PE-CE 接続を通じて VPN クラウドに内部的に接続されます。この接続が確立されると、PE デバイスは CE-learned パスを ATTR_SET という属性内にカプセル化し、それを VPN コアからリモートの PE デバイスまで iBGP-sourced パスで伝送します。リモートの PE デバイスでは、この属性に個別の属性が割り当てられ、送信元 CE パスが抽出されてリモート CE デバイスに送信されます。

ATTR_SET はオプションの遷移属性で、受け取った CE パス属性を伝送します。ATTR_SET 属性は、次のように BGP 更新メッセージ内にエンコードされます。

```
+-------------------------------+
| Attr Flags (O|T) Code = 128  |
+-------------------------------+
| Attr. Length (1 or 2 octets) |
+-------------------------------+
| Origin AS (4 octets)         |
+-------------------------------+
| Path attributes (variable)   |
+-------------------------------+
```

Origin AS は、ATTR_SET が生成される VPN カスタマーの AS です。ATTR_SET の最小長は 4 バイト、最大長は BGP 更新メッセージの必須フィールドと属性を考慮した後のパス属性でサポートされる最大値です。最大長は 3,500 バイトまでにすることをお勧めします。ATTR_SET には、属性の MP_REACH、MP_UNREACH、NEW_AS_PATH、NEW_AGGR、NEXT_HOP、および ATTR_SET 自体を含めること（ATTR_SET 内に ATTR_SET）はできません。ATTR_SET の中にこれらの属性が見つかった場合、ATTR_SET は無効と見なされ、対応するエラー処理メカニズムが呼び出されます。

L3VPN iBGP PE-CE の制限

次に、L3VPN iBGP PE-CE の設定に適用される制限を示します。

- IBGP PE CE 機能を切り替えてネイバーが route-refresh または soft-reconfiguration inbound をサポートしなくなった場合は、手動のセッション フラップを実行して変更を確認する必要があります。これが発生した場合は、次のメッセージが表示されます。

```
RP/0/0/CPU0: %ROUTING-BGP-5-CFG_CHG_RESET: Internal VPN client configuration change
 on neighbor 10.10.10.1 requires HARD reset
(clear bgp 10.10.10.1) to take effect.
```

- iBGP PE CE CLI 設定は、ネイバー/セッション グループを除き、デフォルト VRF のピアには使用できません。

- この機能は、通常の VPN クライアント（eBGP VPN クライアント）上では動作しません。
- ATTR_SET 内にパックされた属性は、iBGP CE 上の inbound route-policy で加えられた変更を反映し、指定した VRF の export route-policy で加えられた変更は反映しません。
- iBGP PE-CE ピアリング セッションで設定された同じ VPN の異なる VRF（つまり、異なる PE ルータ内）は、それぞれの VRF で異なるルート識別子（RD）を使用する必要があります。iBGP PE CE 機能は、RD 値が入力 VRF と出力 VRFで同じである場合は機能しません。

L3VPN iBGP PE-CE の設定

L3VPN iBGP PE-CE は、ネイバー、ネイバー グループ、またはセッション グループで有効にすることができます。L3VPN iBGP PE-CE を設定するには、次のステップを実行します。

始める前に

CE は、内部 BGP ピアである必要があります。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **vrf** *vrf-name*
4. **neighbor** *ip-address* **internal-vpn-client**
5. **commit**
6. **show bgp vrf** *vrf-name* **neighbors** *ip-address*
7. **show bgp** {**vpnv4** | **vpnv6** } **unicast rd**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**vrf** *vrf-name* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# vrf blue`	VRF インスタンスを設定します。
ステップ 4	**neighbor** *ip-address* **internal-vpn-client** 例：	ルーティング情報を交換する相手の CE ネイバー デバイスを設定します。**neighbor internal-vpn-client** コ

	コマンドまたはアクション	目的
	`RP/0/RSP0/cpu 0: router(config-bgp-vrf)# neighbor 10.0.0.0 internal-vpn-client`	マンドは VPN 属性セット内の iBGP-CE ネイバー パスをスタックします。
ステップ 5	**commit**	
ステップ 6	**show bgp vrf** *vrf-name* **neighbors** *ip-address*	VRF CE ピアの iBGP PE-CE 機能が有効かどうかが表示されます。
ステップ 7	**show bgp** {**vpnv4** | **vpnv6** } **unicast rd**	L3VPN iBGP PE-CE が CE 上で有効になっている場合は、コマンドの出力に ATTR_SET 属性が表示されます。

例

例：L3VPN iBGP PE-CE の設定

次の例は、L3VPN iBGP PE-CE の設定方法を示しています。

```
R1(config-bgp-vrf-nbr)#neighbor 10.10.10.1 ?
. . .
 internal-vpn-client     Preserve iBGP CE neighbor path in ATTR_SET across VPN core
. . .
R1(config-bgp-vrf-nbr)#neighbor 10.10.10.1 internal-vpn-client
router bgp 65001
 bgp router-id 100.100.100.2
 address-family ipv4 unicast
address-family vpnv4 unicast
 !
 vrf ce-ibgp
  rd 65001:100
  address-family ipv4 unicast
  !
  neighbor 10.10.10.1
   remote-as 65001
   internal-vpn-client
```

次に、L3VPN iBGP PE-CE が CE ピアで有効になっている場合の **show bgp vrf** *vrf-name* **neighbors** *ip-address* コマンドの出力例を示します。

```
R1#show bgp vrf ce-ibgp neighbors 10.10.10.1
BGP neighbor is 10.10.10.1, vrf ce-ibgp
 Remote AS 65001, local AS 65001, internal link
 Remote router ID 100.100.100.1
  BGP state = Established, up for 00:00:19
  . . .
 Multi-protocol capability received
  Neighbor capabilities:
    Route refresh: advertised (old + new) and received (old + new)
    4-byte AS: advertised and received
    Address family IPv4 Unicast: advertised and received
CE attributes will be preserved across the core
  Received 2 messages, 0 notifications, 0 in queue
  Sent 2 messages, 0 notifications, 0 in queue
  . . .
```

次に、L3VPN iBGP PE-CE が CE ピアで有効になっている場合の **show bgp vpn4/vpn6 unicast rd** コマンドの出力例を示します。

```
BGP routing table entry for 1.1.1.0/24, Route Distinguisher: 200:300
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                 10          10
Last Modified: Aug 28 13:11:17.000 for 00:01:00
Paths: (1 available, best #1)
  Advertised to update-groups (with more than one peer):
    0.2
Path #1: Received by speaker 0
  Advertised to update-groups (with more than one peer):
    0.2
  Local, (Received from a RR-client)
    20.20.20.2 from 20.20.20.2 (100.100.100.2)
      Received Label 24000
      Origin IGP, localpref 100, valid, internal, best, group-best, import-candidate,
      not-in-vrf Received Path ID 0, Local Path ID 1, version 10
      Extended community: RT:228:237
     ATTR-SET [
         Origin-AS: 200
         AS-Path: 51320 52325 59744 12947 21969 50346 18204 36304 41213
 23906 33646
         Origin: incomplete
         Metric: 204
         Local-Pref: 234
         Aggregator: 304 34.3.3.3
         Atomic Aggregator
        Community: 1:60042 2:41661 3:47008 4:9280 5:39778 6:1069 7:15918
 8:8994 9:52701
10:10268 11:26276 12:8506 13:7131 14:65464 15:14304 16:33615 17:54991
18:40149 19:19401
         Extended community: RT:100:1 RT:1.1.1.1:1]
```

フロー タグの伝達

フロー タグ伝達機能では、ルート ポリシーとユーザ ポリシー間に相関関係を構築できます。BGP を使用したフロー タグ伝達では、AS 番号、プレフィックス リスト、コミュニティ文字列、および拡張コミュニティなどのルーティング属性に基づいてユーザ側でトラフィックをステアリングできます。フロー タグは論理数値識別子で、FIB ルックアップ テーブル内の FIB エントリのルーティング属性の 1 つとして RIB を通じて配布されます。フロー タグは、RPL からの「set」操作を使用してインスタンス化され、フロー タグ値に対してアクション（ポリシールール）が関連付けられている C3PL PBR ポリシーで参照されます。

フロー タグの伝達は次の場合に使用できます。

- 宛先 IP アドレス（コミュニティ番号を使用）またはプレフィックス（コミュニティ番号または AS 番号を使用）に基づいてトラフィックを分類する。
- カスタマー サイトのサービス レベル契約（SLA）に基づくサービス エッジに到達するパスのコストに合致する TE グループを選択する。

- SLA とそのクライアントに基づいて、特定のカスタマーにトラフィック ポリシー（TE グループの選択）を適用する。
- アプリケーション サーバまたはキャッシュ サーバにトラフィックを迂回させる。

フロータグ伝達のコマンドの詳細については、 *Routing Command Reference for Cisco ASR 9000 Series Routers*の「BGP Commands」のモジュールを参照してください。

フロー タグ伝達の制限

Border Gateway Protocol を使用した QoS ポリシー伝達（QPPB）とフロー タグ機能の ASR9K プラットフォームでの併用については、いくつかの制約事項があります。次の作業を行います。

- ルート ポリシーには、「set qos-group」または「set flow-tag」のいずれかを使用できますが、prefix-set に両方は使用できません。
- qos-group と route policy flow-tag のルート ポリシーに重複するルートは使用できません。QPPB とフロー タグの機能は、それらが使用するルート ポリシーに重複するルートがない場合に限り、（同じインターフェイス上でも、異なるインターフェイス上でも）共存できます。
- ルート ポリシーとポリシー マップに qos-group と flow-tag を混在させて使用することはお勧めしません。

ソース ベースと宛先ベースのフロー タグ

ソースベースのフローのタグ機能では、着信パケットの発信元アドレスに割り当てられているフロータグに基づいてパケットを照合できます。一致した場合は、このポリシーでサポートされている PBR アクションを適用できます。

送信元と送信先ベースのフロー タグの設定

指定したインターフェイスにフロータグを適用するには、このタスクを実行します。パケットは、着信パケットの発信元アドレスに割り当てられているフロー タグに基づいて照合されます。

（注） インターフェイスでQPPBとフロータグ機能の両方を同時にイネーブルにすることはできません。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **ipv4 | ipv6 bgp policy propagation input flow-tag**{**destination** | **source**}
4. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**interface** *type interface-path-id* 例： `RP/0/RSP0/cpu 0: router(config-if)# interface GigabitEthernet 0/0/0/0`	インターフェイス コンフィギュレーション モードを開始して、1 つ以上のインターフェイスを VRF に関連付けます。
ステップ 3	**ipv4 | ipv6 bgp policy propagation input flow-tag{destination | source}** 例： `RP/0/RSP0/cpu 0: router(config-if)# ipv4 bgp policy propagation input flow-tag source`	送信元または送信先の IP アドレスのフロー タグ ポリシーの伝達をインターフェイスで有効にします。
ステップ 4	**commit**	

例

次の show コマンドは、ルータに適用された RBP ポリシーを使用して出力を表示します。

```
show running-config interface gigabitEthernet 0/0/0/12
Thu Feb 12 01:51:37.820 UTC
interface GigabitEthernet0/0/0/12
 service-policy type pbr input flowMatchPolicy
 ipv4 bgp policy propagation input flow-tag source
 ipv4 address 192.5.1.2 255.255.255.0
!

RP/0/RSP0/CPU0:ASR9K-0#show running-config policy-map type pbr flowMatchPolicy
Thu Feb 12 01:51:45.776 UTC
policy-map type pbr flowMatchPolicy
 class type traffic flowMatch36
  transmit
 !
 class type traffic flowMatch38
  transmit
 !
 class type traffic class-default
 !
 end-policy-map
!

RP/0/RSP0/CPU0:ASR9K-0#show running-config class-map type traffic flowMatch36
Thu Feb 12 01:52:04.838 UTC
class-map type traffic match-any flowMatch36
 match flow-tag 36
 end-class-map
!
```

BGP での VPN ルーティングおよび転送インスタンスの設定

機能を設定するライン カード スロットに使用可能なレイヤ 3 VPN ライセンスがある場合に限り、レイヤ 3（仮想プライベート ネットワーク）を設定できます。拡張 IP ライセンスが有効になっている場合、インターフェイスで 4096 レイヤ 3 VPN ルーティングおよび転送インスタンス（VRF）を設定できます。インフラストラクチャ VRF のライセンスが有効な場合は、8 つのレイヤ 3 VRF をラインカードに設定できます。

拡張 IP ライセンスの詳細については、*System Management Configuration Guide for Cisco ASR 9000 Series Routers*の「Software Entitlement on Cisco IOS XR Software」のモジュールを参照してください。

適切なライセンスが有効になっていないと、次のエラー メッセージが表示されます。

```
RP/0/RSP0/cpu 0: router#LC/0/0/CPU0:Dec 15 17:57:53.653 : rsi_agent[247]:
%LICENSE-ASR9K_LICENSE-2-INFRA_VRF_NEEDED : 5 VRF(s) are configured without license
A9K-iVRF-LIC in violation of the Software Right To Use Agreement.
This feature may be disabled by the system without the appropriate license.
Contact Cisco to purchase the license immediately to avoid potential service interruption.
```

（注） L2VPN サービスの設定に AIP ライセンスは必要ありません。

次の作業は、BGP に VPN ルーティングおよび転送（VRF）インスタンスを設定する場合に実行します。

プロバイダーエッジ ルータでの仮想ルーティングおよび転送テーブルの定義

プロバイダー エッジ（PE）ルータに VPN ルーティングおよび転送（VRF）テーブルを定義するには、次の作業を実行します。

手順の概要

1. **configure**
2. **vrf** *vrf-name*
3. **address-family** { **ipv4** | **ipv6** } **unicast**
4. **maximum prefix** *maximum* [*threshold*]
5. **import route-policy** *policy-name*
6. **import route-target** [*as-number* **:** *nn* | *ip-address* **:** *nn*]
7. **export route-policy** *policy-name*
8. **export route-target** [*as-number* **:** *nn* | *ip-address* **:** *nn*]
9. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	

	コマンドまたはアクション	目的
ステップ 2	**vrf** *vrf-name* 例： `RP/0/RSP0/cpu 0: router(config)# vrf vrf_pe`	VRF インスタンスを設定します。
ステップ 3	**address-family** { **ipv4** | **ipv6** } **unicast** 例： `RP/0/RSP0/cpu 0: router(config-vrf)# address-family ipv4 unicast`	IPv4 または IPv6 のいずれかのアドレス ファミリを指定し、アドレスファミリのコンフィギュレーション サブモードを開始します。 このコマンドのすべてのキーワードと引数のリストを参照するには、CLI ヘルプ（?）を使用します。
ステップ 4	**maximum prefix** *maximum* [*threshold*] 例： `RP/0/RSP0/cpu 0: router(config-vrf-af)# maximum prefix 2300`	VRF テーブルで許可するプレフィックスの数に制限を設定します。 ルートの最大数はダイナミック ルーティング プロトコルと、スタティックまたは接続されたルートに適用されます。 *mid-threshold* 引数を使用して、プレフィックスを制限するしきい値のパーセンテージを指定できます。
ステップ 5	**import route-policy** *policy-name* 例： `RP/0/RSP0/cpu 0: router(config-vrf-af)# import route-policy policy_a`	（任意）VRF にインポートする内容をより細かく制御します。このインポート フィルタでは、指定された *policy-name* 引数に一致しないプレフィックスは破棄されます。
ステップ 6	**import route-target** [*as-number* **:** *nn* | *ip-address* **:** *nn*] 例： `RP/0/RSP0/cpu 0: router(config-vrf-af)# import route-target 234:222`	ルート ターゲット（RT）拡張コミュニティのリストを指定します。指定されたインポート ルート ターゲット拡張コミュニティと関連付けられているプレフィックスだけが VRF にインポートされます。
ステップ 7	**export route-policy** *policy-name* 例： `RP/0/RSP0/cpu 0: router(config-vrf-af)# export route-policy policy_b`	（任意）VRF にエクスポートする内容をより細かく制御します。このエクスポート フィルタでは、指定された *policy-name* 引数に一致しないプレフィックスは破棄されます。
ステップ 8	**export route-target** [*as-number* **:** *nn* | *ip-address* **:** *nn*] 例： `RP/0/RSP0/cpu 0: routerr(config-vrf-af)# export route-target 123;234`	ルート ターゲット拡張コミュニティのリストを指定します。エクスポート ルート ターゲット コミュニティは、リモート PE にアドバタイズされる際にプレフィックスと関連付けられます。リモート PE は、これらのエクスポート ルート ターゲット コミュニティと一致するインポート RT を持つ VRF に、これらのプレフィックスをインポートします。

	コマンドまたはアクション	目的
ステップ 9	**commit**	

ルート識別子の設定

ルート識別子（RD）により、複数の VPN ルーティングおよび転送（VRF）インスタンスにおいてプレフィックスが固有になります。

L3VPN マルチパス同一ルート識別子（RD）環境では、プレフィックスを RIB にインストールするかどうかは、プレフィックスの最適パスに基づいて決まります。稀に設定が誤っている場合（最適パスが RIB にインストールできる有効なパスではない場合）、BGP はプレフィックスをドロップし、その他のパスを考慮しません。この動作は RD のセットアップによって異なります。最適マルチパスが RIB にインストールするパスとして無効な場合には、非最適マルチパスがインストールされます。

RD を設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **bgp router-id** *ip-address*
4. **vrf** *vrf-name*
5. **rd** { *as-number* **:** *nn* | *ip-address* **:** *nn* | **auto** }
6. 次のいずれかを実行します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	BGP コンフィギュレーション モードを開始します。このモードでは BGP ルーティング プロセスを設定できます。
ステップ 3	**bgp router-id** *ip-address* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# bgp router-id 10.0.0.0`	BGP スピーキング ルータの固定ルータ ID を設定します。
ステップ 4	**vrf** *vrf-name* 例：	VRF インスタンスを設定します。

	コマンドまたはアクション	目的
	`RP/0/RSP0/cpu 0: router(config-bgp)# vrf vrf_pe`	
ステップ 5	**rd** { *as-number* **:** *nn* | *ip-address* **:** *nn* | **auto** } **例**： `RP/0/RSP0/cpu 0: router(config-bgp-vrf)# rd 345:567`	ルート識別子を設定します。 ルータが自動的に一意の RD を VRF に割り当てるようにする場合は、 **auto** キーワードを使用します。 ルータ コンフィギュレーション モードで **bgp router-id** コマンドを使用してルータ ID が設定されている場合にのみ、RD を自動で割り当てることができます。これにより、自動 RD 生成に使用できるグローバルで固有のルータ ID を設定できます。VRF のルータ ID はグローバルで固有である必要はありません。また、自動 RD 生成で VRF ルータ ID を使用することは正しくありません。ルータ ID を 1 つにすると、いつ再起動してもルータ ID が固定であるため、BGP グレースフル リスタートで RD 情報のチェックポイントも行いやすくなります。

ステップ 6	次のいずれかを実行します。
	• **end**
	• **commit**
	例：
	`RP/0/RSP0/cpu 0: router(config-bgp-vrf)# end`
	または
	`RP/0/RSP0/cpu 0: router(config-bgp-vrf)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:`
		• **yes** を入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC コンフィギュレーション モードに戻ります。
		• **no** を入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC コンフィギュレーション モードに戻ります。変更はコミットされません。
		• **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

PE-PE または PE-RR 内部 BGP セッションの設定

BGP がプロバイダー エッジ（PE）ルータ間で VPN 到着達可能性情報を送信できるようにするには、PE-PE 内部 BGP（iBGP）セッションを設定する必要があります。PE はリモート PE ルータから送信される VPN 情報を使用して VPN 接続と使用するラベル値を判別します。これにより、リモート（出力）ルータはパケット転送で正しい VPN へのパケットを逆多重化できます。

PE ルータで設定されている VPN に接続するすべての PE および RR ルータに対して PE-PE、PE ルート リフレクタ（RR）iBGP セッションが定義されます。

PE-PE iBGP セッションを設定し、PE でグローバル VPN オプションを設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **address-family vpnv4 unicast**
4. **exit**
5. **neighbor** *ip-address*
6. **remote-as** *as-number*
7. **description** *text*
8. **password** { **clear** | **encrypted** } *password*
9. **shutdown**
10. **timers** *keepalive hold-time*
11. **update-source** *type interface-id*
12. **address-family vpnv4 unicast**
13. **route-policy** *route-policy-name* **in**
14. **route-policy** *route-policy-name* **out**
15. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	自律システム番号を指定し、BGP コンフィギュレーションモードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**address-family vpnv4 unicast** 例： `RP/0/RSP0/cpu 0: router(config-bgp)# address-family vpvn4 unicast`	VPN アドレス ファミリ コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 4	**exit** 例： `RP/0/RSP0/cpu 0: router(config-bgp-af)# exit`	現在のコンフィギュレーション モードを終了します。
ステップ 5	**neighbor** *ip-address* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 172.16.1.1`	PE の iBGP ネイバーを設定します。
ステップ 6	**remote-as** *as-number* 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# remote-as 1`	ネイバーをリモート自律システム番号に割り当てます。
ステップ 7	**description** *text* 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# description neighbor 172.16.1.1`	（任意）ネイバーの説明を指定します。description は、コメントを保存するために使用されます。ソフトウェアの機能には影響しません。
ステップ 8	**password** { **clear** | **encrypted** } *password* 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# password encrypted 123abc`	2 つの BGP ネイバーの間の TCP 接続上で Message Digest 5（MD5）認証をイネーブルにします。
ステップ 9	**shutdown** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# shutdown`	指定されたネイバーのあらゆるアクティブセッションを終了し、すべての関連するルーティング情報を削除します。
ステップ 10	**timers** *keepalive hold-time* 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# timers 12000 200`	BGP ネイバーのタイマーを設定します。
ステップ 11	**update-source** *type interface-id* 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# update-source gigabitEthernet 0/1/5/0`	ネイバーとの iBGP セッションを形成するときに、iBGP セッションが特定のインターフェイスのプライマリ IP アドレスをローカル アドレスとして使用できるようにします。

	コマンドまたはアクション	目的
ステップ 12	**address-family vpnv4 unicast** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# address-family vpvn4 unicast`	VPN ネイバー アドレス ファミリ設定モードを開始します。
ステップ 13	**route-policy** *route-policy-name* **in** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# route-policy pe-pe-vpn-in in`	着信ルートのルーティング ポリシーを指定します。ポリシーを使用すると、ルートのフィルタリングやルート属性の変更ができます。
ステップ 14	**route-policy** *route-policy-name* **out** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# route-policy pe-pe-vpn-out out`	発信ルートのルーティング ポリシーを指定します。ポリシーを使用すると、ルートのフィルタリングやルート属性の変更ができます。
ステップ 15	**commit**	

RTコミュニティの定義済みセットがあるルートを保持するためのルートリフレクタの設定

プロバイダーエッジ（PE）は、設定されている VPN のインポート ルート ターゲット（RT）に一致するルートを保持している必要があります。PE ルータは、他の VPNv4ルートをすべて破棄できます。ただし、ルートリフレクタ（RR）はすべての VPNv4 ルートを維持する必要があります。これは、RR は PE ルータとピアになる可能性があり、別の PE が別の RT タグ付き VPNv4 ルートを要求する（RR をスケーラブルにしない）場合があるためです。RR は RT コミュニティの定義済みのセットを持つルートだけを保持するように設定できます。また、一部の RR は、別の VPN セットを提供するように設定することもできます（これによりスケーラビリティが高まります）。PE で設定された VRF にサービスを提供するすべての RR とピアになるように PE を設定します。PE がまだルートを保持していない RT を使用して、新しい VRF を設定すると、この PE は RR に対してルート リフレッシュ要求を発行し、関連する VPN ルートを取得します。

（注） PE-RR のセッションで拡張コミュニティの Outbound Route Filter（ORF）をサポートしている場合には、このプロセスの効率が高まる場合があることに注意してください。

特定の RT でタグ付けされたルートを保持するようにリフレクタを設定するには、次のタスクを実行します。

手順の概要

1. **configure**

2. **router bgp** *as-number*
3. **address-family vpnv4 unicast**
4. **retain route-target** { **all** | **route-policy** *route-policy-name* }
5. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	自律システム番号を指定し、BGP コンフィギュレーションモードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**address-family vpnv4 unicast** 例： `RP/0/RSP0/cpu 0: router(config-bgp)# address-family vpvn4 unicast`	VPN アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 4	**retain route-target** { **all** | **route-policy** *route-policy-name* } 例： `RP/0/RSP0/cpu 0: router(config-bgp-af)# retain route-target route-policy rr_ext-comm`	特定の RT でタグ付けされたルートを保持するようにリフレクタを設定します。 *route-policy-name* 引数には、RR がパスを保持するためにそのパスに含まれている必要がある拡張コミュニティをリストするポリシー名を指定します。 （注） これがルートリフレクタのデフォルトの動作であるため、 **all** キーワードは不要です。
ステップ 5	**commit**	

PE-CE プロトコルとしての BGP の設定

PE で BGP を設定し、BGP を使用した PE-CE 通信を確立するには、次のタスクを実行します。このタスクは、VRF と VRF 以外の両方の設定で実行できます。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **vrf** *vrf-name*
4. **bgp router-id** *ip-address*
5. **label mode per-ce**
6. **address-family** { **ipv4** | **ipv6** } **unicast**
7. **network** { *ip-address* / *prefix-length* | *ip-address mask* }

8. **aggregate-address** *address* / *mask-length*
9. **exit**
10. **neighbor** *ip-address*
11. **remote-as** *as-number*
12. **password** { **clear** | **encrypted** } *password*
13. **ebgp-multihop** [*ttl-value*]
14. 次のいずれかを実行します。
    - **address-family** { **ipv4** | **ipv6** } **unicast**
    - **address-family {ipv4 {unicast | labeled-unicast} | ipv6 unicast}**
15. **site-of-origin** [*as-number* **:** *nn* | *ip-address* **:** *nn*]
16. **as-override**
17. **allowas-in** [*as-occurrence-number*]
18. **route-policy** *route-policy-name* **in**
19. **route-policy** *route-policy-name* **out**
20. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**vrf** *vrf-name* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# vrf vrf_pe_2`	PE ルータで特定の VRF の BGP ルーティングをイネーブルにします。
ステップ 4	**bgp router-id** *ip-address* 例： `RP/0/RSP0/cpu 0: router(config-bgp-vrf)# bgp router-id 172.16.9.9`	BGP スピーキング ルータの固定ルータ ID を設定します。
ステップ 5	**label mode per-ce** 例： `RP/0/RSP0/cpu 0: router(config-bgp-vrf)# label mode per-ce`	• CE 単位のラベルモードを設定して PE ルータでの追加ルックアップを回避し、ラベル スペースを節約します（デフォルトのラベルモードはプレフィックス単位です）。このモードでは、PE ルータは、すべての即時ネクスト ホップ（ほとんどの場合、これは CE ルータ）に 1 個のラベルを割り当てます。このラベルはネクスト ホップに直接マップされるため、データ転送中に VRF ルート ルックアップが実行される

	コマンドまたはアクション	目的
		ことはありません。ただし、割り当てられるラベルの数は、各 VRF に 1 つではなく、各 CE に 1 個です。BGP はすべてのネクスト ホップを認識するため、各ネクスト ホップにラベルを割り当てます（各 PE-CE インターフェイスではありません）。発信インターフェイスがマルチアクセス インターフェイスで、ネイバーのメディア アクセス コントロール（MAC）アドレスが不明な場合は、アドレス解決プロトコル（ARP）がパケット転送の間にトリガーされます。 • **per-vrf** キーワードは、一意の VRF からアドバタイズされたすべてのルートに同じラベルを使用するように設定します。
ステップ 6	**address-family** { **ipv4** | **ipv6** } **unicast** 例： `RP/0/RSP0/cpu 0: router(config-vrf)# address-family ipv4 unicast`	IPv4 または IPv6 のいずれかのアドレス ファミリ ユニキャストを指定し、アドレス ファミリのコンフィギュレーション サブモードを開始します。 このコマンドのすべてのキーワードと引数のリストを参照するには、CLI ヘルプ（?）を使用します。
ステップ 7	**network** { *ip-address* / *prefix-length* | *ip-address mask* } 例： `RP/0/RSP0/cpu 0: router(config-bgp-vrf-af)# network 172.16.5.5/24`	VRF のコンテキストでアドレス ファミリのテーブルのネットワーク プレフィックスを発信します。
ステップ 8	**aggregate-address** *address* / *mask-length* 例： `RP/0/RSP0/cpu 0: router(config-bgp-vrf-af)# aggregate-address 10.0.0.0/24`	コアに保持されている状態を削減するためルーティング情報を集約するように VRF アドレス ファミリ コンテキストで集約を設定します。この集約により、PE エッジでの効率がいくらか低下します。これはパケットの最終ネクスト ホップを決定するために、さらにルックアップが必要になるためです。設定すると、一連のコンポーネント プレフィックスの代わりにサマリー プレフィックスがアドバタイズされます。これはより詳細な集約です。PE は集約のラベルを 1 つだけアドバタイズします。コンポーネント プレフィックスでは CE へのネクスト ホップが異なることがあるため、データ転送時に追加のルックアップを実行する必要があります。

	コマンドまたはアクション	目的
ステップ 9	**exit** 例： `RP/0/RSP0/cpu 0: router(config-bgp-vrf-af)# exit`	現在のコンフィギュレーション モードを終了します。
ステップ 10	**neighbor** *ip-address* 例： `RP/0/RSP0/cpu 0: router(config-bgp-vrf)# neighbor 10.0.0.0`	CE ネイバーを設定します。 *ip-address* 引数は、プライベート アドレスにする必要があります。
ステップ 11	**remote-as** *as-number* 例： `RP/0/RSP0/cpu 0: router(config-bgp-vrf-nbr)# remote-as 2`	CE ネイバーのリモート AS を設定します。
ステップ 12	**password** { **clear** | **encrypted** } *password* 例： `RP/0/RSP0/cpu 0: router(config-bgp-vrf-nbr)# password encrypted 234xyz`	2 つの BGP ネイバー間の TCP 接続で Message Digest 5（MD5）認証をイネーブルにします。
ステップ 13	**ebgp-multihop** [*ttl-value*] 例： `RP/0/RSP0/cpu 0: router(config-bgp-vrf-nbr)# ebgp-multihop 55`	直接接続していないネットワーク上の外部ピアへの BGP 接続を受け入れて試行するように CE ネイバーを設定します。
ステップ 14	次のいずれかを実行します。 • **address-family** { **ipv4** | **ipv6** } **unicast** • **address-family {ipv4 {unicast | labeled-unicast} | ipv6 unicast}** 例： `RP/0/RSP0/cpu 0: router(config-vrf)# address-family ipv4 unicast`	IPv4 または IPv6 のいずれかのアドレス ファミリ ユニキャストを指定し、アドレス ファミリのコンフィギュレーション サブモードを開始します。 このコマンドのすべてのキーワードと引数のリストを参照するには、CLI ヘルプ（?）を使用します。
ステップ 15	**site-of-origin** [*as-number* **:** *nn* | *ip-address* **:** *nn*] 例： `RP/0/RSP0/cpu 0: router(config-bgp-vrf-nbr-af)# site-of-origin 234:111`	site-of-origin（SoO）拡張コミュニティを設定します。この CE ネイバーから学習されたルートは、その他の PE にアドバタイズされる前に SoO 拡張コミュニティのタグが付けられます。PE ルータで as-override が設定されている場合にループを検出する目的で SoO が使用されることがよくあります。プレフィックスが同じサイトにループする場合、

	コマンドまたはアクション	目的
		PE はこのことを検出して CE に更新を送信しません。
ステップ 16	**as-override** 例： `RP/0/RSP0/cpu 0: router(config-bgp-vrf-nbr-af)# as-override`	PE ルータで AS オーバーライドを設定します。これにより PE ルータは CE の ASN をそれ自体の（PE）ASN に置き換えます。 （注） この情報が失われることが原因でルーティング ループが発生することがあります。as-override によって引き起こされるループを防ぐには、as-override と site-of-origin を組み合わせて使用します。
ステップ 17	**allowas-in** [*as-occurrence-number*] 例： `RP/0/RSP0/cpu 0: router(config-bgp-vrf-nbr-af)# allowas-in 5`	PE 自律システム番号（ASN）を持つ AS パスを指定された回数だけ許可します。 ハブアンドスポーク型VPNネットワークは、HUB CEを通じて、HUB PEへのルーティング情報のループバックを必要とします。この場合、PE ASNが存在するために HUB PE によってループバック情報がドロップされます。これを回避するため、PE ASNが指定された回数に達していても **allowas-in** コマンドを使用してプレフィックスを許可します。
ステップ 18	**route-policy** *route-policy-name* **in** 例： `RP/0/RSP0/cpu 0: router(config-bgp-vrf-nbr-af)# route-policy pe_ce_in_policy in`	着信ルートのルーティングポリシーを指定します。ポリシーを使用すると、ルートのフィルタリングやルート属性の変更ができます。
ステップ 19	**route-policy** *route-policy-name* **out** 例： `RP/0/RSP0/cpu 0: router(config-bgp-vrf-nbr-af)# route-policy pe_ce_out_policy out`	発信ルートのルーティングポリシーを指定します。ポリシーを使用すると、ルートのフィルタリングやルート属性の変更ができます。
ステップ 20	**commit**	

IGP の BGP への再配布

VRF アドレス ファミリへのプロトコルの再配布を設定するには、次の作業を実行します。

内部ゲートウェイ プロトコル（IGP）がPE-CEプロトコルとして使用されている場合でも、インポート ロジックは BGP を経由して実行されます。したがって、すべての IGP ルートを BGP VRF テーブルにインポートする必要があります。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **vrf** *vrf-name*
4. **address-family** { **ipv4** | **ipv6** } **unicast**
5. 次のいずれかを実行します。
   - **redistribute connected** [**metric** *metric-value*] [**route-policy** *route-policy-name*]
   - **redistribute eigrp** *process-id* [**match** { **external** | **internal** }] [**metric** *metric-value*] [**route-policy** *route-policy-name*]
   - **redistribute isis** *process-id* [**level** { **1** | **1-inter-area** | **2** }] [**metric** *metric-value*] [**route-policy** *route-policy-name*]
   - **redistribute ospf** *process-id* [**match** { **external** [**1** | **2**] | **internal** | **nssa-external** [**1** | **2**] }] [**metric** *metric-value*] [**route-policy** *route-policy-name*]
   - **redistribute ospfv3** *process-id* [**match** { **external** [**1** | **2**] | **internal** | **nssa-external** [**1** | **2**] }] [**metric** *metric-value*] [**route-policy** *route-policy-name*]
   - **redistribute rip** [**metric** *metric-value*] [**route-policy** *route-policy-name*]
   - **redistribute static** [**metric** *metric-value*] [**route-policy** *route-policy-name*]
6. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	自律システム番号を指定し、BGP コンフィギュレーションモードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**vrf** *vrf-name* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# vrf vrf_a`	PE ルータで特定の VRF の BGP ルーティングをイネーブルにします。
ステップ 4	**address-family** { **ipv4**	**ipv6** } **unicast** 例： `RP/0/RSP0/cpu 0: router(config-vrf)# address-family ipv4 unicast`	IPv4 または IPv6 のいずれかのアドレス ファミリ ユニキャストを指定し、アドレスファミリのコンフィギュレーション サブモードを開始します。 このコマンドのすべてのキーワードと引数のリストを参照するには、CLI ヘルプ（?）を使用します。
ステップ 5	次のいずれかを実行します。 • **redistribute connected** [**metric** *metric-value*] [**route-policy** *route-policy-name*]	VRF アドレス ファミリ コンテキストでプロトコルの再配布を設定します。 redistribute コマンドは、PE-CE ルータ間で BGP が使用されていない場合に使用します。PE-CE ルータ間

	コマンドまたはアクション	目的
	• **redistribute eigrp** *process-id* [**match** { **external** | **internal** }] [**metric** *metric-value*] [**route-policy** *route-policy-name*] • **redistribute isis** *process-id* [**level** { **1** | **1-inter-area** | **2** }] [**metric** *metric-value*] [**route-policy** *route-policy-name*] • **redistribute ospf** *process-id* [**match** { **external** [**1** | **2**] | **internal** | **nssa-external** [**1** | **2**]}] [**metric** *metric-value*] [**route-policy** *route-policy-name*] • **redistribute ospfv3** *process-id* [**match** { **external** [**1** | **2**] | **internal** | **nssa-external** [**1** | **2**]}] [**metric** *metric-value*] [**route-policy** *route-policy-name*] • **redistribute rip** [**metric** *metric-value*] [**route-policy** *route-policy-name*] • **redistribute static** [**metric** *metric-value*] [**route-policy** *route-policy-name*] 例： `RP/0/RSP0/cpu 0: router(config-bgp-vrf-af)# redistribute eigrp 23`	で BGP が使用されている場合は、使用されている IGP を BGP に再配布して、他方の PE サイトとの VPN接続を確立する必要があります。テーブル間でのインポートおよびエクスポートにも再配布が必要です。
ステップ 6	**commit**	

BGP のキーチェーンの設定

キーチェーンは、さまざまな MAC 認証アルゴリズムをサポートして安全な認証を実現し、円滑なキー ロールオーバーを実装します。BGP のキーチェーンを設定するには、次の作業を実行します。このタスクはオプションです。

（注） ネイバー グループまたはセッション グループのキーチェーンが設定されている場合、そのグループを使用するネイバーはキーチェーンを継承します。あるネイバーのために特別に設定されたコマンドの値は、継承された値を上書きします。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **neighbor** *ip-address*
4. **remote-as** *as-number*
5. **keychain** *name*
6. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**neighbor** *ip-address* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 172.168.40.24`	BGP ルーティングのためにルータをネイバー コンフィギュレーション モードにして、ネイバーの IP アドレスを BGP ピアとして設定します。
ステップ 4	**remote-as** *as-number* 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# remote-as 2002`	ネイバーを作成し、リモート自律システム番号を割り当てます。
ステップ 5	**keychain** *name* 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# keychain kych_a`	キーチェーンに基づく認証を設定します。
ステップ 6	**commit**	

BGP ネイバーの無効化

設定を削除せずにネイバーを管理シャットダウンするには、次の作業を実行します。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **neighbor** *ip-address*
4. **shutdown**
5. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	

	コマンドまたはアクション	目的
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 127`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**neighbor** *ip-address* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 172.168.40.24`	BGP ルーティングのためにルータをネイバー コンフィギュレーション モードにして、ネイバーの IP アドレスを BGP ピアとして設定します。
ステップ 4	**shutdown** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# shutdown`	指定されたネイバーのすべてのアクティブ セッションをディセーブルにします。
ステップ 5	**commit**	

BGP インバウンド ソフト リセットを使用したネイバーのリセット

指定されたグループまたはネイバーの指定アドレス ファミリに対してインバウンド ソフト リセットをトリガーするには、次の作業を実行します。グループは、 ***** 、 *ip-address* 、 *as-number* 、または **external** キーワードおよび引数によって指定されます。

ネイバーのインバウンドポリシーまたはアウトバウンドポリシーを変更する場合、またはルーティングアップデートの送信または受信に影響を与えるその他の設定を変更する場合には、ネイバーのリセットが便利です。インバウンド ソフト リセットがトリガーされた場合、ネイバーが ROUTE_REFRESH 機能をアドバタイズしていれば、BGP はデフォルトでこのネイバーに REFRESH 要求を送信します。ネイバーが ROUTE_REFRESH 機能をアドバタイズしているかどうかを判別するには、**show bgp neighbors** コマンドを使用します。

手順の概要

1. **show bgp neighbors**
2. **clear bgp** { **ipv4** { **unicast** | **multicast** | **all** | **tunnel** } | **ipv6** **unicast** | **all** { **unicast** | **multicast** | **all** | **tunnel** } | **vpnv4 unicast** | **vrf** { *vrf-name* | **all** } { **ipv4** **unicast** | **ipv6** **unicast** } { ***** | *ip-address* | **as** *as-number* | **external** } **soft** [**in** [**prefix-filter**] | **out**]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**show bgp neighbors** 例： `RP/0/RSP0/cpu 0: router# show bgp neighbors`	ネイバーから受信したルート リフレッシュ機能がイネーブルであることを確認します。

	コマンドまたはアクション	目的
ステップ 2	**clear bgp** { **ipv4** { **unicast** | **multicast** | **all** | **tunnel** } | **ipv6 unicast** | **all** { **unicast** | **multicast** | **all** | **tunnel** } | **vpnv4 unicast** | **vrf** { *vrf-name* | **all** } { **ipv4 unicast** | **ipv6 unicast** } { ***** | *ip-address* | **as** *as-number* | **external** } **soft** [**in** [**prefix-filter**] | **out**] 例： `RP/0/RSP0/cpu 0: router# clear bgp ipv4 unicast 10.0.0.1 soft in`	BGP ネイバーをソフト リセットします。 • ***** キーワードを指定すると、すべての BGP ネイバーがリセットされます。 • *ip-address* 引数では、リセットするネイバーのアドレスを指定します。 • *as-number* 引数では、自律システム番号に一致するすべてのネイバーがリセットされることを指定します。 • **external** キーワードは、すべての外部ネイバーがリセットされることを指定します。

BGP アウトバウンド ソフト リセットを使用したネイバーのリセット

指定されたグループまたはネイバーの指定アドレス ファミリに対してアウトバウンド ソフト リセットをトリガーするには、次の作業を実行します。グループは、*****、*ip-address*、*as-number*、または **external** キーワードおよび引数によって指定されます。

ネイバーのアウトバウンド ポリシーまたはアウトバウンド ポリシーを変更する場合、またはルーティング アップデートの送信または受信に影響を与えるその他の設定を変更する場合には、ネイバーのリセットが便利です。

アウトバウンド ソフト リセットがトリガーされると、BGP は、このアドレス ファミリに対するルートをすべて、指定されたネイバーに再送信します。

ネイバーが ROUTE_REFRESH 機能をアドバタイズしているかどうかを判別するには、**show bgp neighbors** コマンドを使用します。

手順の概要

1. **show bgp neighbors**
2. **clear bgp** { **ipv4** { **unicast** \| **multicast** \| **all** \| **tunnel** } \| **ipv6 unicast** \| **all** { **unicast** \| **multicast** \| **all** \| **tunnel** } \| **vpnv4 unicast** \| **vrf** { *vrf-name* \| **all** } { **ipv4 unicast** \| **ipv6 unicast** } { ***** \| *ip-address* \| **as** *as-number* \| **external** } **clear bgp** { **ipv4** \| **ipv6**} { **unicast** \| **labeled-unicast** } **soft** [**in** [**prefix-filter**] \|]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**show bgp neighbors** 例： `RP/0/RSP0/cpu 0: router# show bgp neighbors`	ネイバーから受信したルート リフレッシュ機能がイネーブルであることを確認します。

	コマンドまたはアクション	目的
ステップ 2	**clear bgp** { **ipv4** { **unicast** | **multicast** | **all** | **tunnel** } | **ipv6** **unicast** | **all** { **unicast** | **multicast** | **all** | **tunnel** } | **vpnv4 unicast** | **vrf** { *vrf-name* | **all** } { **ipv4** **unicast** | **ipv6** **unicast** } { ***** | *ip-address* | **as** *as-number* | **external** } **clear bgp** { **ipv4 | ipv6**} { **unicast** | **labeled-unicast** } **soft** [**in** [**prefix-filter**] |] 例： `RP/0/RSP0/cpu 0: router# clear bgp ipv4 unicast 10.0.0.2 soft out`	BGP ネイバーをソフト リセットします。 • ***** キーワードを指定すると、すべての BGP ネイバーがリセットされます。 • *ip-address* 引数では、リセットするネイバーのアドレスを指定します。 • *as-number* 引数では、自律システム番号に一致するすべてのネイバーがリセットされることを指定します。 • **external** キーワードは、すべての外部ネイバーがリセットされることを指定します。

BGP ハード リセットを使用したネイバーのリセット

ハード リセットを使用してネイバーをリセットするには、次の作業を実行します。ハード リセットにより、ネイバーへの TCP 接続が削除され、ネイバーから受信したすべてのルートが BGP テーブルから削除され、その後このネイバーとのセッションが再確立されます。**graceful** キーワードを指定すると、ネイバーからのルートは BGP テーブルから即座に削除されず、古い（stale）ルートとしてマークされます。セッションの再確立後、ネイバーから再受信されなかった古いルートはすべて削除されます。

手順の概要

1. **clear bgp** { **ipv4** { **unicast** \| **multicast** \| **all** \| **tunnel** } \| **ipv6** **unicast** \| **all** { **unicast** \| **multicast** \| **all** \| **tunnel** } \| **vpnv4 unicast** \| **vrf** { *vrf-name* \| **all** } { **ipv4** **unicast** \| **ipv6 unicast** } \| { ***** \| *ip-address* \| **as** *as-number* \| **external** } [**graceful**] **soft** [**in** [**prefix-filter**] \| **out**] **clear bgp** { **ipv4 \| ipv6**} { **unicast** \| **labeled-unicast** }

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**clear bgp** { **ipv4** { **unicast** | **multicast** | **all** | **tunnel** } | **ipv6** **unicast** | **all** { **unicast** | **multicast** | **all** | **tunnel** } | **vpnv4 unicast** | **vrf** { *vrf-name* | **all** } { **ipv4** **unicast** | **ipv6 unicast** } | { ***** | *ip-address* | **as** *as-number* | **external** } [**graceful**] **soft** [**in** [**prefix-filter**] | **out**] **clear bgp** { **ipv4 | ipv6**} { **unicast** | **labeled-unicast** } 例： `RP/0/RSP0/cpu 0: router# clear bgp ipv4 unicast 10.0.0.3 graceful soft out`	BGP ネイバーをクリアします。 • ***** キーワードを指定すると、すべての BGP ネイバーがリセットされます。 • *ip-address* 引数では、リセットするネイバーのアドレスを指定します。 • *as-number* 引数では、自律システム番号に一致するすべてのネイバーがリセットされることを指定します。

	コマンドまたはアクション	目的
		• **external** キーワードは、すべての外部ネイバーがリセットされることを指定します。 **graceful** キーワードはグレースフル リスタートを指定します。

キャッシュ、テーブル、およびデータベースのクリア

特定のキャッシュ、テーブル、またはデータベースのすべての内容を削除するには、次のタスクを実行します。**clear bgp** コマンドは、指定されたネイバーグループのセッションをリセット（ハードリセット）します。これにより、ネイバーへの TCP 接続が削除され、ネイバーから受信したすべてのルートが BGP テーブルから削除され、その後このネイバーとのセッションが再確立されます。キャッシュ、テーブル、またはデータベースは、特定の構造が無効になったり、無効になるおそれのあるときに、クリアすることが必要になります。

手順の概要

1. **clear bgp** { **ipv4** { **unicast** | **multicast** | **all** | **tunnel** } | **ipv6 unicast** | **all** { **unicast** | **multicast** | **all** | **tunnel** } | **vpnv4 unicast** | **vrf** { *vrf-name* | **all** } { **ipv4 unicast** | **ipv6 unicast** } *ip-address*
2. **clear bgp external**
3. **clear bgp ***

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**clear bgp** { **ipv4** { **unicast**	**multicast**	**all**	**tunnel** }	**ipv6 unicast**	**all** { **unicast**	**multicast**	**all**	**tunnel** }	**vpnv4 unicast**	**vrf** { *vrf-name*	**all** } { **ipv4 unicast**	**ipv6 unicast** } *ip-address* 例： `RP/0/RSP0/cpu 0: router# clear bgp ipv4 172.20.1.1`	指定されたネイバーをクリアします。
ステップ 2	**clear bgp external** 例： `RP/0/RSP0/cpu 0: router# clear bgp external`	すべての外部ピアをクリアします。
ステップ 3	**clear bgp *** 例： `RP/0/RSP0/cpu 0: router# clear bgp *`	すべての BGP ネイバーをクリアします。

システムおよびネットワーク統計情報の表示

特定の統計情報（BGPルーティングテーブル、キャッシュ、およびデータベースの内容など）を表示するには、次のタスクを実行します。提供される情報は、リソースの使用状況を判定してネットワークの問題を解決するために使用されます。さらに、ノードの到達可能性に関する情報を表示し、そのパケットが経由するネットワーク内のルーティングパスを検出することもできます。

手順の概要

1. **show bgp cidr-only**
2. **show bgp community** *community-list* [**exact-match**]
3. **show bgp regexp** *regular-expression*
4. **show bgp**
5. **show bgp neighbors** *ip-address* [**advertised-routes** | **dampened-routes** | **flap-statistics** | **performance-statistics** | **received** *prefix-filter* | **routes**]
6. **show bgp paths**
7. **show bgp neighbor-group** *group-name* **configuration**
8. **show bgp summary**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**show bgp cidr-only** 例： `RP/0/RSP0/cpu 0: router# show bgp cidr-only`	不自然なネットワーク マスク（クラスレス ドメイン間ルーティング（CIDR））を持つルートを表示します。
ステップ 2	**show bgp community** *community-list* [**exact-match**] 例： `RP/0/RSP0/cpu 0: router# show bgp community 1081:5 exact-match`	指定されたBGPコミュニティに一致するルートを表示します。
ステップ 3	**show bgp regexp** *regular-expression* 例： `RP/0/RSP0/cpu 0: router# show bgp regexp "^3 "`	指定した自律システム パスの正規表現と一致するルートを表示します。
ステップ 4	**show bgp** 例： `RP/0/RSP0/cpu 0: router# show bgp`	BGPルーティングテーブル内のエントリを表示します。

	コマンドまたはアクション	目的
ステップ 5	**show bgp neighbors** *ip-address* [**advertised-routes** | **dampened-routes** | **flap-statistics** | **performance-statistics** | **received** *prefix-filter* | **routes**] 例： `RP/0/RSP0/cpu 0: router# show bgp neighbors 10.0.101.1`	指定したネイバーへのBGP接続に関する情報を表示します。 • **advertised-routes** キーワードを指定すると、ルータがネイバーにアドバタイズするすべてのルートが表示されます。 • **dampened-routes** キーワードを指定すると、ネイバーから学習したダンプ済みのルートが表示されます。 • **flap-statistics** キーワードを指定すると、ネイバーから学習したルートのフラップ統計情報が表示されます。 • **performance-statistics** キーワードを指定すると、このネイバーのBGPプロセスによって実行された作業に関連するパフォーマンス統計情報が表示されます。 • **received** *prefix-filter* キーワードと引数を指定すると、プレフィックス リスト フィルタが表示されます。 • **routes** キーワードを指定すると、ネイバーから学習したルートが表示されます。
ステップ 6	**show bgp paths** 例： `RP/0/RSP0/cpu 0: router# show bgp paths`	データベース内のすべてのBGPパスを表示します。
ステップ 7	**show bgp neighbor-group** *group-name* **configuration** 例： `RP/0/RSP0/cpu 0: router# show bgp neighbor-group group_1 configuration`	指定したネイバーグループによって継承された設定を含む、ネイバーグループの有効な設定を表示します。
ステップ 8	**show bgp summary** 例： `RP/0/RSP0/cpu 0: router# show bgp summary`	BGP 接続すべての状況を表示します。

BGP プロセス情報の表示

特定の BGP プロセス情報を表示するには、次のタスクを実行します。

手順の概要

1. **show bgp process**
2. **show bgp ipv4 unicast summary**
3. **show bgp vpnv4 unicast summary**
4. **show bgp vrf** (*vrf-name* | **all** }
5. **show bgp process detail**
6. **show bgp summary**
7. **show placement program bgp**
8. **show placement program brib**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**show bgp process** 例： `RP/0/RSP0/cpu 0: router# show bgp process`	BGP プロセスのステータスと要約情報を表示します。出力には、さまざまなグローバルおよびアドレス ファミリ固有の BGP 設定が表示されます。プロセスによって送受信されたネイバー、アップデート メッセージ、および通知メッセージの数の要約も表示されます。
ステップ 2	**show bgp ipv4 unicast summary** 例： `RP/0/RSP0/cpu 0: router# show bgp ipv4 unicast summary`	IPv4 ユニキャスト アドレス ファミリのネイバーの要約を表示します。
ステップ 3	**show bgp vpnv4 unicast summary** 例： `RP/0/RSP0/cpu 0: router# show bgp vpnv4 unicast summary`	VPNv4 ユニキャスト アドレス ファミリのネイバーの要約を表示します。
ステップ 4	**show bgp vrf** (*vrf-name* | **all** } 例： `RP/0/RSP0/cpu 0: router# show bgp vrf vrf_A`	BGP VPN 仮想ルーティングおよび転送（VRF）情報を表示します。
ステップ 5	**show bgp process detail** 例： `RP/0/RSP0/cpu 0: router# show bgp processes detail`	さまざまな内部構造タイプによって使用されているメモリなど、詳細なプロセス情報を表示します。
ステップ 6	**show bgp summary** 例： `RP/0/RSP0/cpu 0: router# show bgp summary`	BGP 接続すべての状況を表示します。

	コマンドまたはアクション	目的
ステップ 7	**show placement program bgp** 例： `RP/0/RSP0/cpu 0: router# show placement program bgp`	BGP プログラムの情報を表示します。 • 「拒否された場所」としてプログラムが表示される場合（プログラムの場所を特定できないなど）、**show placement program bgp** コマンドを使用して、その場所を表示できます。 • プログラムが配置されても起動されない場合、プログラムが配置されてから経過した時間の長さが [Waiting to start] 列に表示されます。
ステップ 8	**show placement program brib** 例： `RP/0/RSP0/cpu 0: router# show placement program brib`	bRIB プログラムの情報を表示します。 • 「拒否された場所」としてプログラムが表示される場合（プログラムの場所を特定できないなど）、**show placement program bgp** コマンドを使用して、その場所を表示できます。 • プログラムが配置されても起動されない場合、プログラムが配置されてから経過した時間の長さが [Waiting to start] 列に表示されます。

BGP アップデート グループのモニタリング

この作業では、BGP アップデート グループの処理に関する情報を表示します。

手順の概要

1. **show bgp** [**ipv4** { **unicast** | **multicast** | **all** | **tunnel** } | **ipv6** { **unicast** | **all** } | **all** { **unicast** | **multicast** | **all** | **tunnel** } | **vpnv4 unicast** | **vrf** { *vrf-name* | **all** } [**ipv4 unicast**] **update-group** [**neighbor** *ip-address* | *process-id.index* [**summary** | **performance-statistics**]]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**show bgp** [**ipv4** { **unicast**	**multicast**	**all**	**tunnel** }	**ipv6** { **unicast**	**all** }	**all** { **unicast**	**multicast**	**all**	**tunnel** }	**vpnv4 unicast**	**vrf** { *vrf-name*	**all** } [**ipv4 unicast**] **update-group** [**neighbor** *ip-address*	*process-id.index* [**summary**	**performance-statistics**]] 例：	BGP アップデート グループの情報を表示します。 • *ip-address* 引数を指定すると、そのネイバーが属するアップデート グループが表示されます。 • *process-id.index* 引数では、表示する特定のアップデート グループを選択します。この引数は「プロセスID（ドット）インデックス」の形式で指定します。プロセス ID の範囲は 0 ～ 254

	コマンドまたはアクション	目的
	`RP/0/RSP0/cpu 0: router# show bgp update-group 0.0`	です。インデックスの範囲は 0 ～ 4294967295 です。 • **summary** キーワードを指定すると、特定のアップデート グループに含まれているネイバーに関する要約情報が表示されます。 • このコマンドに引数を指定しないと、（指定したアドレス ファミリの）すべてのアップデート グループの情報が表示されます。 • **performance-statistics** キーワードを指定すると、アップデート グループのパフォーマンス統計情報が表示されます。

BGP ノンストップルーティングの設定

BGP ノンストップルーティング（BGP NSR）はデフォルトで有効になっています。また、無効になっている BGP NSR を有効に戻すには、**no nsr disable** コマンドを使用します。

BGP ノンストップ ルーティングの無効化

BGP ノンストップ ルーティング（NSR）を無効にするには、次のタスクを実行します。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **nsr disable**
4. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	BGP ルーティング プロセスを設定するため、BGP AS 番号を指定して BGP コンフィギュレーション モードを開始します。
ステップ 3	**nsr disable** 例：	BGP ノンストップ ルーティングを無効にします。

	コマンドまたはアクション	目的
	`RP/0/RSP0/cpu 0: router(config-bgp)# nsr disable`	
ステップ 4	**commit**	

BGP ノンストップ ルーティングの再有効化

BGP ノンストップ ルーティング（NSR）が無効になっている場合、次のステップを使用して BGP NSR を有効にします。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **no nsr disable**
4. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	BGP ルーティング プロセスを設定するため、BGP AS 番号を指定して BGP コンフィギュレーション モードを開始します。
ステップ 3	**no nsr disable** 例： `RP/0/RSP0/cpu 0: router(config-bgp)# nsr disable`	BGP ノンストップ ルーティングを有効にします。
ステップ 4	**commit**	

Prefix Independent Convergence（PIC）のプライマリバックアップパスのインストール

転送テーブルにバックアップパスをインストールし、PE-CE リンク障害が発生した場合に Prefix Independent Convergence（PIC）を提供するには、次のタスクを実行します。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. 次のいずれかを実行します。

- **address-family** {**vpnv4 unicast** | **vpnv6 unicast**}
- **vrf** *vrf-name* {**ipv4 unicast** | **ipv6 unicast**}

4. **additional-paths selection route-policy** *route-policy-name*
5. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 100`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	次のいずれかを実行します。 • **address-family** {**vpnv4 unicast**	**vpnv6 unicast**} • **vrf** *vrf-name* {**ipv4 unicast**	**ipv6 unicast**} 例： `RP/0/RSP0/cpu 0: router(config-bgp)# address-family vpnv4 unicast`	アドレスファミリまたはVRFアドレスファミリを指定して、アドレスファミリまたはVRFアドレスファミリのコンフィギュレーションサブモードを開始します。

ステップ 4	**additional-paths selection route-policy** *route-policy-name*
	例：
	`RP/0/RSP0/cpu 0: router(config-bgp-af)# additional-paths selection route-policy ap1`
		プレフィックスの追加パス選択モードを設定します。
		（注） **additional-paths selection** コマンドを適切なルートポリシーとともに使用して、バックアップパスを計算し、プレフィックス独立コンバージェンス（PIC）機能を有効にします。
		ルートポリシーの設定は、プレフィックスの追加パス選択モードを設定するための前提条件です。追加選択コマンドで使用するルートポリシー設定の例を次に示します。
		`route-policy ap1`
		`  set path-selection backup 1 install`
		`end-policy`

ステップ 5	**commit**	

プライマリ パスのローカル ラベル割り当ての保持

プライマリ PE で以前にプライマリ パスに割り当てられたローカル ラベルを、再コンバージェンス後に設定期間にわたって保持するには、次の作業を実行します。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **address-family { vpnv4 unicast | vpnv6 unicast }**
4. **retain local-label** *minutes*
5. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 100`	自律システム番号を指定し、BGP コンフィギュレーションモードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**address-family { vpnv4 unicast | vpnv6 unicast }** 例： `RP/0/RSP0/cpu 0: router(config-bgp)# address-family vpnv4 unicast`	アドレス ファミリを指定し、アドレス ファミリのコンフィギュレーション サブモードを開始します。
ステップ 4	**retain local-label** *minutes* 例： `RP/0/RSP0/cpu 0: router(config-bgp-af)# retain local-label 10`	プライマリ PE で以前にプライマリ パスに割り当てられたローカル ラベルを、再コンバージェンス後 10 分間保持します。
ステップ 5	**commit**	

BGP 追加パスの設定

BGP 追加パス機能を設定するには、次の作業を行います。

手順の概要

1. **configure**
2. **route-policy** *route-policy-name*
3. **if** *conditional-expression* **then** *action-statement* **else**
4. **pass endif**
5. **end-policy**
6. **router bgp** *as-number*
7. **address-family {ipv4 {unicast | multicast} | ipv6 {unicast | multicast | l2vpn vpls-vpws | vpnv4 unicast | vpnv6 unicast }**
8. **additional-paths receive**
9. **additional-paths send**

10. **additional-paths selection route-policy** *route-policy-name*
11. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**route-policy** *route-policy-name* 例： `RP/0/RSP0/cpu 0: router (config)#route-policy add_path_policy`	ルート ポリシーを定義して、ルート ポリシー コンフィギュレーション モードを開始します。

ステップ 3	**if** *conditional-expression* **then** *action-statement* **else**
	例：
	`RP/0/RSP0/cpu 0: router (config-rpl)#if community matches-any (*) then`
	`set path-selection all advertise`
	`else`
		特定のルートのアクションとディスポジションを決定します。

ステップ 4	**pass endif**
	例：
	`RP/0/RSP0/cpu 0: router(config-rpl-else)#pass`
	`RP/0/RSP0/cpu 0: router(config-rpl-else)#endif`
		処理のためにルートを渡し、ifステートメントを終了します。

ステップ 5	**end-policy** 例： `RP/0/RSP0/cpu 0: router(config-rpl)#end-policy`	ルートポリシーの定義を終了して、ルートポリシー コンフィギュレーション モードを終了します。
ステップ 6	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)#router bgp 100`	自律システム番号を指定し、BGPコンフィギュレーションモードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 7	**address-family** {**ipv4** {**unicast** | **multicast**} | **ipv6** {**unicast** | **multicast** | **l2vpn vpls-vpws** | **vpnv4 unicast** | **vpnv6 unicast** } 例： `RP/0/RSP0/cpu 0: router(config-bgp)#address-family ipv4 unicast`	アドレス ファミリを指定し、アドレス ファミリのコンフィギュレーションサブモードを開始します。
ステップ 8	**additional-paths receive** 例： `RP/0/RSP0/cpu 0: router(config-bgp-af)#additional-paths receive`	対応ピアのプレフィックスのマルチパス受信機能を設定します。
ステップ 9	**additional-paths send** 例：	対応ピアのプレフィックスのマルチパス送信機能を設定します。

	コマンドまたはアクション	目的
	`RP/0/RSP0/cpu 0:` `router(config-bgp-af)#additional-paths send`	

ステップ 10	**additional-paths selection route-policy** *route-policy-name*
	例：
	`RP/0/RSP0/cpu 0:`
	`router(config-bgp-af)#additional-paths selection route-policy add_path_policy`
		プレフィックスの追加パス選択機能を設定します。

ステップ 11	**commit**	

iBGP マルチパス ロード シェアリングの設定

iBGP マルチパス ロード シェアリングを設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **address-family** {**ipv4**|**ipv6**} {**unicast**|**multicast**}
4. **maximum-paths ibgp** *number*
5. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 100`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。

ステップ 3	**address-family** {**ipv4**|**ipv6**} {**unicast**|**multicast**}
	例：
	`RP/0/RSP0/cpu 0:`
	`router(config-bgp)# address-family ipv4 multicast`
		IPv4 または IPv6 のいずれかのアドレス ファミリを指定し、アドレス ファミリのコンフィギュレーション サブモードを開始します。

ステップ 4	**maximum-paths ibgp** *number*
	例：
	`RP/0/RSP0/cpu 0:`
	`router(config-bgp-af)# maximum-paths ibgp 30`
		ロード シェアリング用の iBGP パスの最大数を設定します。

ステップ 5	**commit**	

AiGP によるプレフィックスの生成

AiGP メトリックを使用したルートの生成を設定するには、次の作業を実行します。

始める前に

Accumulated Interior Gateway Protocol（AiGP）メトリックを使用したルートの生成は設定により制御されます。次の条件を満たす再配布ルートに AiGP 属性が付加されます。

- AiGP でルートを再配布するプロトコルがイネーブルに設定されている。
- このルートは、ボーダー ゲートウェイ プロトコル（BGP）に再配布された Interior Gateway Protocol（iGP）ルートです。AiGP 属性に割り当てられた値はルートの iGP ネクスト ホップの値か、または route-policy によって設定された値です。
- このルートは BGP に再配布されたスタティック ルートです。割り当てられた値はルートのネクスト ホップの値か、route-policy によって設定された値です。
- このルートはネットワーク ステートメントによって BGP にインポートされます。割り当てられた値はルートのネクスト ホップの値か、route-policy によって設定された値です。

手順の概要

1. **configure**
2. **route-policy** *aigp_policy*
3. **set aigp-metricigp-cost**
4. **exit**
5. **router bgp** *as-number*
6. **address-family** {**ipv4** | **ipv6**} **unicast**
7. **redistribute ospf** *osp* **route-policy** *plcy_name***metric** *value*
8. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**route-policy** *aigp_policy* 例： `RP/0/RSP0/cpu 0: router(config)# route-policy aip_policy`	ルート ポリシー コンフィギュレーション モードを開始してルート ポリシーを設定します。
ステップ 3	**set aigp-metricigp-cost** 例： `RP/0/RSP0/cpu 0: router(config-rpl)# set aigp-metric igp-cost`	内部ルーティング プロトコル コストを aigp メトリックとして設定します。

	コマンドまたはアクション	目的
ステップ 4	**exit** 例： `RP/0/RSP0/cpu 0: router(config-rpl)# exit`	ルートポリシー コンフィギュレーション モードを終了します。
ステップ 5	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 100`	BGP AS 番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。

ステップ 6	**address-family** {**ipv4** | **ipv6**} **unicast**
	例：
	`RP/0/RSP0/cpu 0:`
	`router(config-bgp)# address-family ipv4 unicast`
		IPv4 または IPv6 のいずれかのアドレス ファミリを指定し、アドレス ファミリのコンフィギュレーション サブモードを開始します。

ステップ 7	**redistribute ospf** *osp* **route-policy** *plcy_name***metric** *value*
	例：
	`RP/0/RSP0/cpu 0:`
	`router(config-bgp-af)#redistribute ospf osp`
	`route-policy aigp_policy metric 1`
		OSPF への AiBGP メトリックの再配布を許可します。

ステップ 8	**commit**	

BGP Accept Own の設定

BGP Accept Own を設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **neighbor** *ip-address*
4. **remote-as** *as-number*
5. **update-source** *type interface-path-id*
6. **address-family** {*vpnv4 unicast* | *vpnv6 unicast*}
7. **accept-own** [**inheritance-disable**]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `Router(config)#router bgp 100`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。

	コマンドまたはアクション	目的
ステップ 3	**neighbor** *ip-address* 例： `Router(config-bgp)#neighbor 10.1.2.3`	BGP ルーティングのためにルータをネイバー コンフィギュレーション モードにして、ネイバーの IP アドレスを BGP ピアとして設定します。
ステップ 4	**remote-as** *as-number* 例： `Router(config-bgp-nbr)#remote-as 100`	ネイバーにリモート自律システム番号を割り当てます。
ステップ 5	**update-source** *type interface-path-id* 例： `Router(config-bgp-nbr)#update-source Loopback0`	ネイバーでセッションを形成するとき、特定のインターフェイスからのプライマリ IP アドレスをローカル アドレスとしてセッションで使用できます。
ステップ 6	**address-family** {*vpnv4 unicast* | *vpnv6 unicast*} 例： `Router(config-bgp-nbr)#address-family vpnv6 unicast`	アドレス ファミリを VPNv4 または IPv6 として指定し、ネイバーアドレスファミリのコンフィギュレーション モードを開始します。
ステップ 7	**accept-own** [**inheritance-disable**] 例： `Router(config-bgp-nbr-af)#accept-own`	Accept_Own コミュニティが含まれる自動送信 VPN ルートの処理をイネーブルにします。 「Accept Own」設定をディセーブルにし、親コンフィギュレーションから「Accept Own」が継承されないようにするには、 **inheritance-disable** キーワードを使用します。

BGP リンク状態の設定

BGP リンク状態の設定

BGP リンクステート（LS）情報を BGP ネイバーと交換するには、次のステップを実行します。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **neighbor** *ip-address*
4. **remote-as** *as-number*
5. **address-family link-state link-state**
6. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 100`	BGP AS 番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**neighbor** *ip-address* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 10.0.0.2`	CE ネイバーを設定します。ip-address 引数は、プライベート アドレスである必要があります。
ステップ 4	**remote-as** *as-number* 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# remote-as 1`	CE ネイバーのリモート AS を設定します。
ステップ 5	**address-family link-state link-state** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# address-family link-state link-state`	BGP リンクステート情報を指定されたネイバーに配布します。
ステップ 6	**commit**	

ドメイン識別子の設定

固有識別子 4 オクテット ASN を設定するには、次のステップを実行します。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **address-family link-state link-state**
4. **domain-distinguisher** *unique-id*
5. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 100`	BGP AS 番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**address-family link-state link-state** 例： `RP/0/RSP0/cpu 0: router(config-bgp)# address-family link-state link-state`	アドレスファミリ リンクステート コンフィギュレーション モードを開始します。
ステップ 4	**domain-distinguisher** *unique-id* 例： `RP/0/RSP0/cpu 0: router(config-bgp-af)# domain-distinguisher 1234`	固有識別子 4 オクテット ASN を設定します。範囲は 1 ～ 4294967295 です。
ステップ 5	**commit**	

BGP パーマネントネットワークの設定

BGP パーマネントネットワークの設定

BGP パーマネント ネットワークを設定するには、次のタスクを実行します。パーマネント ネットワーク（パス）が設定されるプレフィックス（ネットワーク）のセットを識別するには、少なくとも 1 つのルート ポリシーを設定する必要があります。

手順の概要

1. **configure**
2. **prefix-set** *prefix-set-name*
3. **exit**
4. **route-policy** *route-policy-name*
5. **end-policy**
6. **router bgp** *as-number*
7. **address-family { ipv4 | ipv6 } unicast**
8. **permanent-network route-policy** *route-policy-name*
9. **commit**
10. **show bgp {ipv4 | ipv6} unicast** *prefix-set*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	

ステップ 2	**prefix-set** *prefix-set-name*
	例：
	`RP/0/RSP0/cpu 0: router(config)# prefix-set PERMANENT-NETWORK-IPv4`
	`RP/0/RSP0/cpu 0: router(config-pfx)# 1.1.1.1/32,`
	`RP/0/RSP0/cpu 0: router(config-pfx)# 2.2.2.2/32,`
	`RP/0/RSP0/cpu 0: router(config-pfx)# 3.3.3.3/32`
	`RP/0/RSP0/cpu 0: router(config-pfx)# end-set`
		プレフィックス セット コンフィギュレーション モードを開始し、連続したビット セットと非連続のビット セットに対しプレフィックス セットを定義します。

ステップ 3	**exit** 例： `RP/0/RSP0/cpu 0: router(config-pfx)# exit`	プレフィックス セット コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードを開始します。

ステップ 4	**route-policy** *route-policy-name*
	例：
	`RP/0/RSP0/cpu 0: router(config)# route-policy POLICY-PERMANENT-NETWORK-IPv4`
	`RP/0/RSP0/cpu 0: router(config-rpl)# if destination in PERMANENT-NETWORK-IPv4 then`
	`RP/0/RSP0/cpu 0: router(config-rpl)# pass`
	`RP/0/RSP0/cpu 0: router(config-rpl)# endif`
		ルート ポリシーを作成し、ルート ポリシー コンフィギュレーション モードを開始します。このモードではルート ポリシーを定義できます。

ステップ 5	**end-policy** 例： `RP/0/RSP0/cpu 0: router(config-rpl)# end-policy`	ルート ポリシーの定義を終了して、ルート ポリシー コンフィギュレーション モードを終了します。
ステップ 6	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 100`	自律システム番号を指定して、BGP コンフィギュレーション モードを開始します。
ステップ 7	**address-family { ipv4 | ipv6 } unicast** 例： `RP/0/RSP0/cpu 0: router(config-bgp)# address-family ipv4 unicast`	IPv4 または IPv6 のいずれかのアドレス ファミリ ユニキャストを指定し、アドレス ファミリのコンフィギュレーション サブモードを開始します。

	コマンドまたはアクション	目的
ステップ 8	**permanent-network route-policy** *route-policy-name* 例： `RP/0/RSP0/cpu 0: router(config-bgp-af)# permanent-network route-policy POLICY-PERMANENT-NETWORK-IPv4`	ルート ポリシーで定義されているプレフィックスのセットに対しパーマネントネットワーク（パス）を設定します。
ステップ 9	**commit**	
ステップ 10	**show bgp {ipv4 | ipv6} unicast** *prefix-set* 例： `RP/0/RSP0/cpu 0: routershow bgp ipv4 unicast`	（オプション）プレフィックス セットが BGP でパーマネント ネットワークであるかどうかを表示します。

パーマネントネットワークのアドバタイズ方法

固定パスがアドバタイズされる必要があるピアを識別するには、このタスクを実行します。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **neighbor** *ip-address*
4. **remote-as** *as-number*
5. **address-family { ipv4 | ipv6 } unicast**
6. **advertise permanent-network**
7. **commit**
8. **show bgp {ipv4 | ipv6} unicast neighbor** *ip-address*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 100`	自律システム番号を指定して、BGP コンフィギュレーション モードを開始します。
ステップ 3	**neighbor** *ip-address* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# neighbor`	BGP ルーティングのためにルータをネイバー コンフィギュレーション モードにして、ネイバーの IP アドレスを BGP ピアとして設定します。

	コマンドまたはアクション	目的
	`10.255.255.254`	
ステップ 4	**remote-as** *as-number* 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# remote-as 4713`	ネイバーをリモート自律システム番号に割り当てます。
ステップ 5	**address-family { ipv4 | ipv6 } unicast** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# address-family ipv4 unicast`	IPv4 または IPv6 のいずれかのアドレス ファミリ ユニキャストを指定し、アドレスファミリのコンフィギュレーション サブモードを開始します。
ステップ 6	**advertise permanent-network** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# advertise permanent-network`	パーマネント ネットワーク（パス）がアドバタイズされるピアを指定します。
ステップ 7	**commit**	
ステップ 8	**show bgp {ipv4 | ipv6} unicast neighbor** *ip-address* 例： `RP/0/RSP0/cpu 0: routershow bgp ipv4 unicast neighbor 10.255.255.254`	（オプション）ネイバーが BGP パーマネント ネットワークを受信できるかどうかを表示します。

BGP 不等コストの連続ロードバランシングの有効化

外部 BGP（eBGP）、内部 BGP（iBGP）、および eiBGP の不等コストの連続ロードバランシングを有効にし、BGP が非武装地帯（DMZ）リンクのリンク帯域幅属性を送信できるようにするには、次のタスクを実行します。

マルチプロトコル内部 BGP（MP-iBGP）セッション（IPv4 または VPNv4）を介した、リモート PE への PE ルータのアップデートにリンク帯域幅拡張コミュニティが含まれている場合、**maximum-paths** コマンドが有効になっていれば、リモート PE が自動的にロードバランシングを実行します。

不等コストの連続ロード バランシングは、最大で 8 つのパスに対してのみ行われます。

（注） BGP不等コスト連続ロードバランシング機能の有効化は、CPPベースのカードではサポートされていません。

手順の概要

1. **configure**
2. **router bgp** *as-number*
3. **address-family** { **ipv4** | **ipv6** } **unicast**
4. **maximum-paths** { **ebgp** | **ibgp** | **eibgp** } *maximum* [**unequal-cost**]
5. **exit**
6. **neighbor** *ip-address*
7. **dmz-link-bandwidth**
8. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**router bgp** *as-number* 例： `RP/0/RSP0/cpu 0: router(config)# router bgp 120`	自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。
ステップ 3	**address-family** { **ipv4**	**ipv6** } **unicast** 例： `RP/0/RSP0/cpu 0: router(config-bgp)# address-family ipv4 unicast`	IPv4 または IPv6 のいずれかのアドレス ファミリ ユニキャストを指定し、アドレス ファミリのコンフィギュレーション サブモードを開始します。 このコマンドのすべてのキーワードと引数のリストを参照するには、CLI ヘルプ（?）を使用します。
ステップ 4	**maximum-paths** { **ebgp**	**ibgp**	**eibgp** } *maximum* [**unequal-cost**] 例： `RP/0/RSP0/cpu 0: router(config-bgp-af)# maximum-paths ebgp 3`	BGPによりルーティングテーブルにインストールされるパラレル ルートの最大数を設定します。 （注） • 最大パスの有効な値は、ASR 9000イーサネットラインカードの場合は 8、ASR 9000拡張イーサネットラインカードの場合は 32 です。 • ASR 9000 イーサネットラインカードは、設定されている最大パス値が 8 を超える場合でも、転送ハードウェアにインストールするルートの数を 8 に制限します。

	コマンドまたはアクション	目的
		• **ebgp** *maximum* ：マルチパスに eBGP パスのみを考慮します。 • **ibgp** *maximum* [**unequal-cost**] ：iBGP 学習パス間でのロード バランシングを考慮します。 • **eibgp** *maximum* ：eBGP および iBGP 学習パスの両方のロードバランシングを考慮します。eiBGP は常に不等コスト ロード バランシングを実行します。 eiBGP が適用されると eBGP ロード バランシングまたは iBGP ロード バランシングは設定できませんが、eBGP ロード バランシングと iBGP ロード バランシングは共存できます。
ステップ 5	**exit** 例： `RP/0/RSP0/cpu 0: router(config-bgp-af)# exit`	現在のコンフィギュレーション モードを終了します。
ステップ 6	**neighbor** *ip-address* 例： `RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 10.0.0.0`	CE ネイバーを設定します。 *ip-address* 引数は、プライベート アドレスにする必要があります。
ステップ 7	**dmz-link-bandwidth** 例： `RP/0/RSP0/cpu 0: router(config-bgp-nbr)# dmz-link-bandwidth`	eBGP および iBGP ネイバーへのリンクのために、非武装地帯（DMZ）リンク帯域幅拡張コミュニティを開始します。
ステップ 8	**commit**	

VRF ダイナミックルートのリークの設定

次のステップを実行して、デフォルト VRF から非デフォルト VRF にルートをインポートするか、または非デフォルト VRF からデフォルト VRF にルートをインポートします。

始める前に

ダイナミック ルート リークを設定するには、ルート ポリシーが必要です。ルート ポリシーを設定するには、グローバル コンフィギュレーション モードで**route-policy** *route-policy-name* コマンドを使用します。

手順の概要

1. **configure**
2. **vrf** *vrf_name*
3. **address-family** {**ipv4** | **ipv6**} **unicast**
4. 次のいずれかのオプションを使用します。
   - **import from default-vrf route-policy** *route-policy-name* [**advertise-as-vpn**]
   - **export to default-vrf route-policy** *route-policy-name*
5. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure**	
ステップ 2	**vrf** *vrf_name* 例： `RP/0/RSP0/CPU0:PE51_ASR-9010(config)#vrf vrf_1`	VRF コンフィギュレーション モードを開始します。
ステップ 3	**address-family** {**ipv4**	**ipv6**} **unicast** 例： `RP/0/RSP0/cpu 0: router(config-vrf)#address-family ipv6 unicast`	VRF アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 4	次のいずれかのオプションを使用します。
	• **import from default-vrf route-policy** *route-policy-name* [**advertise-as-vpn**]
	• **export to default-vrf route-policy** *route-policy-name*
	例：
	`RP/0/RSP0/cpu 0: router(config-vrf-af)#import from default-vrf route-policy rpl_dynamic_route_import`
	または
	`RP/0/RSP0/cpu 0: router(config-vrf-af)#export to default-vrf route-policy rpl_dynamic_route_export`
		デフォルト VRF から非デフォルト VRF にルートをインポートするか、または非デフォルト VRF からデフォルト VRF にルートをインポートします。
		• **import from default-vrf**：デフォルト VRF から非デフォルト VRF へのインポートを設定します。
		advertise-as-vpn オプションが設定されている場合、デフォルト VRF から非デフォルト VRF にインポートしたパスは、PE と CE にアドバタイズされます。**advertise-as-vpn** オプションが設定されていない場合、デフォルト VRF から非デフォルト VRF にインポートされたパスは PE にアドバタイズされません。ただし、この場合も CE にはパスがアドバタイズされます。
		• **export to default-vrf**：非デフォルト VRF からデフォルト VRF へのインポートを設定します。デフォルト VRF からインポートされたパスが他の PE にアドバタイズされます。

ステップ 5	**commit**	

次のタスク

次の **show bgp** コマンドの出力には、ダイナミック ルート リーク設定の情報が表示されます。

- **show bgp prefix** コマンドを使用すると、インポートしたパスの送信元 RD と送信元 VRF が表示されます。これには、IPv4 または IPv6 ユニキャスト プレフィックスにインポートしたパスがある場合も含まれます。
- **show bgp imported-routes** コマンドを使用すると、デフォルト VRF の IPv4 ユニキャストおよび IPv6 ユニキャストのアドレスファミリが表示されます。

選択的 VRF ダウンロードの有効化

選択的 VRF ダウンロードを有効にするには、**svd platform enable** コマンドの後にルータのリロードを設定します。

（注） デフォルトでは、選択的 VRF ダウンロードは無効になっています。

手順の概要

1. **admin**
2. **configure**
3. **svd platform enable**
4. **commit**
5. **show svd state**
6. **admin**
7. **reload location all**
8. **exit**
9. **show svd role**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**admin** 例： `RP/0/RSP0/cpu 0: router# admin`	管理 EXEC モードを開始します。
ステップ 2	**configure** 例： `RP/0/RSP0/cpu 0: router(admin)#configure`	管理コンフィギュレーション モードを開始します。
ステップ 3	**svd platform enable** 例：	選択的 VRF ダウンロードを有効にします。

	コマンドまたはアクション	目的
	RP/0/RSP0/cpu 0: router(admin-config)#svd platform enable	
ステップ 4	**commit**	
ステップ 5	**show svd state** 例： RP/0/RSP0/cpu 0: router#show svd state Selective VRF Download (SVD) Feature State: SVD Configuration State Enabled SVD Operational State Enabled	選択的VRFダウンロード機能の状態情報を表示します。
ステップ 6	**admin** 例： RP/0/RSP0/cpu 0: router#admin	管理者モードを開始します。
ステップ 7	**reload location all** 例： RP/0/RSP0/cpu 0: router(admin)#reload loc all Tue Feb 12 07:51:25.279 UTC Preparing system for backup. This may take a few minutes especially for large configurations. Status report: node0_RSP0_CPU0: START TO BACKUP Status report: node0_RSP0_CPU0: BACKUP HAS COMPLETED SUCCESSFULLY [Done] Proceed with reload? [confirm]RP/0/RSP0/CPU0::This node received reload	シャーシをリロードします。
ステップ 8	**exit** 例： RP/0/RSP0/cpu 0: router(admin)#exit	管理者 EXEC モードを終了し、EXEC モードを開始します。
ステップ 9	**show svd role** 例： RP/0/RSP0/cpu 0: router#show svd role Tue Feb 12 07:50:26.908 UTC Codes: (C) : user Configured role Node Name IPv4 Role IPv6 Role ——————————————————————— 0/RSP0/CPU0 Standard Standard 0/0/CPU0 Customer Facing Not Interested 0/1/CPU0 Customer Facing Not Interested	VRF インターフェイスがあるラインカードに SVD ロールが「カスタマー向け」であることを確認することで、選択的 VRF ダウンロードがアクティブになっているかどうかを確認します。

次のタスク

svd platform enable コマンドを使用して SVD を有効にした後に **selective-vrf-download disable** を使用して SVD をオフにしないでください。

選択的 VRF ダウンロードの無効化

デフォルトでは、選択的VRFダウンロードは無効になっています。ただし、SVDが有効になっている場合は、次のタスクを実行して機能を無効にします。

手順の概要

1. **admin**
2. **configure**
3. **no svd platform enable**
4. **commit**
5. **show svd state**
6. **admin**
7. **reload location all**
8. **exit**
9. **show svd role**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**admin** 例： `RP/0/RSP0/cpu 0: router# admin`	管理 EXEC モードを開始します。
ステップ 2	**configure** 例： `RP/0/RSP0/cpu 0: router(admin)#configure`	管理コンフィギュレーションモードを開始します。
ステップ 3	**no svd platform enable** 例： `RP/0/RSP0/CPU0:PE51_ASR-9010(admin-config)#no svd platform enable`	選択的 VRF ダウンロードを無効にします。
ステップ 4	**commit**	

ステップ 5	**show svd state**
	例：
	`RP/0/RSP0/cpu 0: router#show svd state`
	`Selective VRF Download (SVD) Feature State:`
	`SVD Configuration State Unsupported`
	`SVD Operational State Unsupported`
		選択的VRFダウンロード機能の状態情報を表示します。

	コマンドまたはアクション	目的
ステップ 6	**admin** 例： `RP/0/RSP0/cpu 0: router#admin`	管理者モードを開始します。

ステップ 7	**reload location all**
	例：
	`RP/0/RSP0/cpu 0: router(admin)#reload loc all`
	`Tue Feb 12 07:51:25.279 UTC`
	
	`Preparing system for backup. This may take a few minutes especially for large configurations.`
	`Status report: node0_RSP0_CPU0: START TO BACKUP`
	`Status report: node0_RSP0_CPU0: BACKUP HAS COMPLETED SUCCESSFULLY`
	`[Done]`
	`Proceed with reload? [confirm]RP/0/RSP0/CPU0::This node received reload`
		シャーシをリロードします。

ステップ 8	**exit** 例： `RP/0/RSP0/cpu 0: router(admin)#exit`	管理者 EXEC モードを終了し、EXEC モードを開始します。

ステップ 9	**show svd role**
	例：
	`RP/0/RSP0/cpu 0: router#show svd role`
	
	`Codes: (C) : user Configured role`
	`Node Name IPv4 Role IPv6 Role`
	`0/RSP0/CPU0 Standard Standard`
	`0/0/CPU0 Standard Standard`
	`0/1/CPU0 Standard Standard`
		VRF インターフェイスがあるラインカードに SVD ロールが「標準」であることを確認することで、選択的VRF ダウンロードが非アクティブになっているかどうかを確認します。

復元力のある CE 単位のラベルモードの設定

VRF アドレスファミリでの復元力のある CE 単位のラベルモードの設定

VRF アドレスファミリに復元力のある CE 単位のラベルモードを設定するには、次のタスクを実行します。

（注） 復元力のある CE 単位の 6PE ラベル割り当ては、CRS-1 ルータと CRS-3 ルータではサポートされていません。ASR 9000 ルータでのみサポートされています。

手順の概要

1. **configure**
2. **router bgp***as-number*
3. **vrf***vrf-instance*
4. **address-family {ipv4 | ipv6} unicast**
5. **label mode per-ce**
6. 次のいずれかを実行します。
   - **end**
   - **commit**

手順の詳細

ステップ 1 **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
RP/0/RSP0/cpu 0: router(config)#
```

グローバル コンフィギュレーション モードを開始します。

ステップ 2 **router bgp***as-number*

例：

```
RP/0/RSP0/cpu 0: router(config)# router bgp 666
RP/0/RSP0/cpu 0: router(config-bgp)#
```

自律システム番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。

ステップ 3 **vrf***vrf-instance*

例：

```
RP/0/RSP0/cpu 0: router(config-bgp)# vrf vrf-pe
RP/0/RSP0/cpu 0: router(config-bgp-vrf)#
```

VRF インスタンスを設定します。

ステップ 4 **address-family {ipv4 | ipv6} unicast**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp-vrf)# address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-vrf-af)#
```

IPv4 または IPv6 のいずれかのアドレス ファミリ ユニキャストを指定し、アドレス ファミリのコンフィギュレーション サブモードを開始します。

ステップ 5 **label mode per-ce**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp-vrf-af)# label mode per-ce
RP/0/RSP0/cpu 0: router(config-bgp-vrf-af)#
```

復元力のある CE 単位のラベルモードを設定します。

ステップ 6 次のいずれかを実行します。

- **end**
- **commit**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp-vrf-af)# end
```

または

```
RP/0/RSP0/cpu 0: router(config-bgp-vrf-af)# commit
```

設定変更を保存します。

- **end** コマンドを実行すると、変更をコミットするように要求されます。

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:
```

  - **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
  - **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
  - **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。

- 実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

ルートポリシーを使用した復元力のある CE 単位のラベルモードの設定

ルートポリシーを使用して復元力のある CE 単位のラベルモードを設定するには、次のタスクを実行します。

（注） 復元力のある CE 単位の 6PE ラベル割り当ては、CRS-1 ルータと CRS-3 ルータではサポートされていません。ASR 9000 ルータでのみサポートされています。

手順の概要

1. **configure**
2. **route-policy** *policy-name*
3. **set label mode per-ce**
4. 次のいずれかを実行します。
   - **end**
   - **commit**

手順の詳細

ステップ 1 **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
RP/0/RSP0/cpu 0: router(config)#
```

グローバル コンフィギュレーション モードを開始します。

ステップ 2 **route-policy** *policy-name*

例：

```
RP/0/RSP0/cpu 0: router(config)# route-policy route1
RP/0/RSP0/cpu 0: router(config-rpl)#
```

ルート ポリシーを作成し、ルート ポリシー コンフィギュレーション モードを開始します。

ステップ 3 **set label mode per-ce**

例：

```
RP/0/RSP0/cpu 0: router(config-rpl)# set label mode per-ce
RP/0/RSP0/cpu 0: router(config-rpl)#
```

復元力のある CE 単位のラベルモードを設定します。

ステップ 4 次のいずれかを実行します。

- **end**
- **commit**

例：

```
RP/0/RSP0/cpu 0: router(config-rpl)# end
```

または

```
RP/0/RSP0/cpu 0: router(config-rpl)# commit
```

設定変更を保存します。

- **end** コマンドを実行すると、変更をコミットするように要求されます。

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:
```

- **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
- **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
- **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。

- 実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

BGP の実装の設定例

ここでは、次の設定例について説明します。

BGP のイネーブル化：例

次に、BGP をイネーブルにする例を示します。

```
prefix-set static
   2020::/64,
   2012::/64,
   10.10.0.0/16,
   10.2.0.0/24
end-set

route-policy pass-all
  pass
end-policy
route-policy set_next_hop_agg_v4
  set next-hop 10.0.0.1
end-policy

route-policy set_next_hop_static_v4
  if (destination in static) then
    set next-hop 10.1.0.1
  else
    drop
  endif
end-policy

route-policy set_next_hop_agg_v6
  set next-hop 2003::121
end-policy

route-policy set_next_hop_static_v6
  if (destination in static) then
     set next-hop 2011::121
```

```
  else
     drop
  endif
end-policy

router bgp 65000
  bgp fast-external-fallover disable
  bgp confederation peers
    65001
    65002
  bgp confederation identifier 1
  bgp router-id 1.1.1.1
  address-family ipv4 unicast
    aggregate-address 10.2.0.0/24 route-policy set_next_hop_agg_v4
    aggregate-address 10.3.0.0/24
    redistribute static route-policy set_next_hop_static_v4
  address-family ipv4 multicast
    aggregate-address 10.2.0.0/24 route-policy set_next_hop_agg_v4
    aggregate-address 10.3.0.0/24
    redistribute static route-policy set_next_hop_static_v4
  address-family ipv6 unicast
    aggregate-address 2012::/64 route-policy set_next_hop_agg_v6
    aggregate-address 2013::/64
    redistribute static route-policy set_next_hop_static_v6
  address-family ipv6 multicast
    aggregate-address 2012::/64 route-policy set_next_hop_agg_v6
    aggregate-address 2013::/64
    redistribute static route-policy set_next_hop_static_v6
  neighbor 10.0.101.60
    remote-as 65000
    address-family ipv4 unicast
    address-family ipv4 multicast
  neighbor 10.0.101.61
    remote-as 65000
    address-family ipv4 unicast
    address-family ipv4 multicast
  neighbor 10.0.101.62
    remote-as 3
    address-family ipv4 unicast
      route-policy pass-all in
      route-policy pass-all out
    address-family ipv4 multicast
      route-policy pass-all in
      route-policy pass-all out
  neighbor 10.0.101.64
    remote-as 5
    update-source Loopback0
    address-family ipv4 unicast
      route-policy pass-all in
      route-policy pass-all out
    address-family ipv4 multicast
      route-policy pass-all in
      route-policy pass-all out
```

BGP アップデート グループの表示：例

次に、EXEC コンフィギュレーション モードで実行された **show bgp update-group** コマンドの出力例を示します。

```
show bgp update-group

 Update group for IPv4 Unicast, index 0.1:
   Attributes:
     Outbound Route map:rm
     Minimum advertisement interval:30
   Messages formatted:2, replicated:2
   Neighbors in this update group:
     10.0.101.92

 Update group for IPv4 Unicast, index 0.2:
   Attributes:
     Minimum advertisement interval:30
   Messages formatted:2, replicated:2
   Neighbors in this update group:
     10.0.101.91
```

BGP ネイバー設定：例

情報を共有するように自律システムの BGP ネイバーを設定する例を次に示します。この例ではBGP ルータを自律システム 109 に割り当て、自律システムの送信元として 2 つのネットワークのリストが表示される例を示します。3 つのリモートルータ（とその自律システム）のアドレスのリストが表示されます。設定したルータは、ネットワーク 172 .16 .0.0 と 192.168 .7.0 と隣接ルータに関する情報を共有します。リストの 1 番目のルータは別の自律システムにあり、2 番目の **neighbor** および **remote-as** コマンドによってアドレス 172 .26 .234.2; の内部ネイバーが（同じ自律システム番号を使用して）指定され、3 番目の **neighbor** および **remote-as** コマンドによって別の自律システムのネイバーが指定されます。

```
route-policy pass-all
 pass
end-policy
router bgp 109
 address-family ipv4 unicast
  network 172.16.0.0 255.255.0.0
  network 192.168.7.0 255.255.0.0
  neighbor 172.16.200.1
   remote-as 167
   exit
 address-family ipv4 unicast
  route-policy pass-all in
  route-policy pass-out out
  neighbor 172.26.234.2
   remote-as 109
   exit
 address-family ipv4 unicast
  neighbor 172.26.64.19
   remote-as 99
   exit
 address-family ipv4 unicast
  route-policy pass-all in
  route-policy pass-all out
```

BGP コンフェデレーション：例

次に、コンフェデレーションのいくつかのピアを表示する設定の例を示します。このコンフェデレーションは、自律システム番号 6001、6002、および 6003 の 3 つの内部自律システムから構成されています。コンフェデレーション外の BGP スピーカーには、このコンフェデレーションは（**bgp confederation identifier** コマンドによって指定される）自律システム番号 666 を持つ通常の自律システムのように見えます。

自律システム 6001 の BGP スピーカーで、**bgp confederation peers** コマンドは、自律システム 6002 および 6003 からのピアを特別な eBGP ピアとしてマークします。したがって、ピア 171.16 .232.55 および 171。16 .232.56 は、この更新でローカルプリファレンス、ネクスト ホップ、および未変更の MED を取得します。171 のルータ。19 .69.1 のルータは通常の eBGP スピーカーであり、このピアからの更新は、自律システム 666 のピアから受け取る通常の eBGP 更新とまったく同じです。

```
router bgp 6001
 bgp confederation identifier 666
 bgp confederation peers
  6002
  6003
   exit
 address-family ipv4 unicast
  neighbor 171.16.232.55
  remote-as 6002
   exit
 address-family ipv4 unicast
  neighbor 171.16.232.56
  remote-as 6003
   exit
 address-family ipv4 unicast
  neighbor 171.19.69.1
  remote-as 777
```

自律システム 6002 の BGP スピーカーでは、自律システム 6001 および 6003 からのピアは特別な eBGP ピアとして設定されます。ピア 171 。17 .70.1 のピアは通常の iBGP ピアであり、ピア 199.99.99.2 は自律システム 700 の通常の eBGP ピアです。

```
router bgp 6002
 bgp confederation identifier 666
 bgp confederation peers
  6001
  6003
   exit
 address-family ipv4 unicast
  neighbor 171.17.70.1
   remote-as 6002
   exit
 address-family ipv4 unicast
  neighbor 171.19.232.57
   remote-as 6001
   exit
 address-family ipv4 unicast
  neighbor 171.19.232.56
   remote-as 6003
   exit
```

```
 address-family ipv4 unicast
  neighbor 171.19.99.2
   remote-as 700
   exit
 address-family ipv4 unicast
  route-policy pass-all in
  route-policy pass-all out
```

自律システム 6003 の BGP スピーカーでは、自律システム 6001 および 6002 からのピアは特別な eBGP ピアとして設定されます。ピア 192 。168 .200.200 のピアは、自律システム 701 の通常の eBGP ピアです。

```
router bgp 6003
 bgp confederation identifier 666
 bgp confederation peers
  6001
  6002
   exit
 address-family ipv4 unicast
  neighbor 171.19.232.57
   remote-as 6001
   exit
 address-family ipv4 unicast
  neighbor 171.19.232.55
   remote-as 6002
   exit
 address-family ipv4 unicast
  neighbor 192.168.200.200
   remote-as 701
   exit
 address-family ipv4 unicast
  route-policy pass-all in
  route-policy pass-all out
```

下記は、同じ例の自律システム 701 の BGP スピーカー 192 .168 .200.205 から受信する設定の一部を示します。ネイバー 171.16 .232.56 は自律システム 666 の通常の eBGP スピーカーとして設定されます。コンフェデレーション外部のピアは、この自律システムが複数の自律システムに内部分割されることを認識しません。

```
router bgp 701
 address-family ipv4 unicast
  neighbor 172.16.232.56
   remote-as 666
   exit
 address-family ipv4 unicast
  route-policy pass-all in
  route-policy pass-all out
   exit
 address-family ipv4 unicast
  neighbor 192.168.200.205
   remote-as 701
```

BGP ルートリフレクタ：例

次に、アドレスファミリを使用して、内部BGP ピア 10.1.1.1 をユニキャストプレフィックスとマルチキャスト プレフィックスの両方のルート リフレクタ クライアントとして設定する例を示します。

```
router bgp 140
 address-family ipv4 unicast
  neighbor 10.1.1.1
   remote-as 140
   address-family ipv4 unicast
    route-reflector-client
    exit
   address-family ipv4 multicast
    route-reflector-client
```

BGP ノンストップルーティング設定：例

次に、BGP NSR を有効にする例を示します。

```
configure
router bgp 120
nsr
end
```

次に、BGP NSR をディセーブルにする例を示します。

```
configure
router bgp 120
no nsr
end
```

プライマリバックアップパスのインストール：例

次に、プライマリバックアップパスのインストールを有効にする例を示します。

```
 router bgp 120
address-family ipv4 unicast
 additional-paths receive
 additional-paths send
 additional-paths selection route-policy bgp_add_path
!
!
end
```

ローカル ラベル割り当ての保持：例

次に、プライマリ PE のプライマリ パスに以前に割り当てたローカル ラベルを再コンバージェンス後 10 分にわたって維持する例を示します。

```
router bgp 100
address-family l2vpn vpls-vpws
  retain local-label 10
end
```

iBGP マルチパス負荷共有設定：例

次に、負荷共有に 30 のパスが使用されている設定の例を示します。

```
router bgp 100
 address-family ipv4 multicast
  maximum-paths ibgp 30
 !
!
end
```

過剰パスの破棄の設定：例

次に、IPv4 アドレス ファミリに対する過剰パスの破棄機能を設定する例を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# router bgp 10
RP/0/RSP0/CPU0:router(config-bgp)# neighbor 10.0.0.1
RP/0/RSP0/CPU0:router(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RSP0/CPU0:router(config-bgp-nbr-af)# maximum-prefix 1000 discard-extra-paths
RP/0/RSP0/CPU0:router(config-bgp-vrf-af)# commit
```

過剰パス情報の破棄の表示：例

次の画面出力では、過剰パスの破棄オプションの詳細を示しています。

```
RP/0/0/CPU0:ios# show bgp neighbor 10.0.0.1

BGP neighbor is 10.0.0.1
Remote AS 10, local AS 10, internal link
Remote router ID 0.0.0.0
BGP state = Idle (No best local address found)
Last read 00:00:00, Last read before reset 00:00:00
Hold time is 180, keepalive interval is 60 seconds
Configured hold time: 180, keepalive: 60, min acceptable hold time: 3
Last write 00:00:00, attempted 0, written 0
Second last write 00:00:00, attempted 0, written 0
Last write before reset 00:00:00, attempted 0, written 0
Second last write before reset 00:00:00, attempted 0, written 0
Last write pulse rcvd not set last full not set pulse count 0
```

```
Last write pulse rcvd before reset 00:00:00
Socket not armed for io, not armed for read, not armed for write
Last write thread event before reset 00:00:00, second last 00:00:00
Last KA expiry before reset 00:00:00, second last 00:00:00
Last KA error before reset 00:00:00, KA not sent 00:00:00
Last KA start before reset 00:00:00, second last 00:00:00
Precedence: internet
Multi-protocol capability not received
Received 0 messages, 0 notifications, 0 in queue
Sent 0 messages, 0 notifications, 0 in queue
Minimum time between advertisement runs is 0 secs

For Address Family: IPv4 Unicast
BGP neighbor version 0
Update group: 0.1 Filter-group: 0.0 No Refresh request being processed
Route refresh request: received 0, sent 0
0 accepted prefixes, 0 are bestpaths
Cumulative no. of prefixes denied: 0.
Prefix advertised 0, suppressed 0, withdrawn 0
Maximum prefixes allowed 10 (discard-extra-paths) <<<<<<<<<<<<<<<<<<<<<
Threshold for warning message 75%, restart interval 0 min
AIGP is enabled
An EoR was not received during read-only mode
Last ack version 1, Last synced ack version 0
Outstanding version objects: current 0, max 0
Additional-paths operation: None
Send Multicast Attributes

Connections established 0; dropped 0
Local host: 0.0.0.0, Local port: 0, IF Handle: 0x00000000
Foreign host: 10.0.0.1, Foreign port: 0
Last reset 00:00:00
```

ネイバー単位の TCP MSS の設定：例

次に、ネイバー単位の TCP MSS を設定する方法、ネイバー単位の TCP MSS を無効にする方法、および TCP MSS の設定を解除する方法の例を示します。

トポロジのシナリオ

次の図に、ネイバー単位の TCP MSS 設定の基本的なシナリオを示します。

R1 の設定は次のようになります。

```
router bgp 1
  bgp router-id 10.0.0.1
  address-family ipv4 unicast
  !
  neighbor-group n1
    tcp mss 100
    address-family ipv4 unicast
  !
 !
```

```
  neighbor 10.0.0.2
    remote-as 1
    use neighbor-group n1
    address-family ipv4 unicast
  !
 !
!
```

R2 の設定は次のようになります。

```
router bgp 1
  bgp router-id 10.0.0.2
  address-family ipv4 unicast
  !
  neighbor 10.0.0.1
    remote-as 1
    address-family ipv4 unicast
  !
 !
!
```

ネイバー単位の TCP MSS の設定: 例

次に、ネイバーグループにネイバー単位の TCP MSS を設定する例を示します。

```
router bgp 1
bgp router-id 10.0.0.1
address-family ipv4 unicast
!
neighbor-group n1
tcp mss 500
address-family ipv4 unicast
!
!
neighbor 10.0.0.2
remote-as 1
use neighbor-group n1
address-family ipv4 unicast
!
!
!
!
end
```

ネイバー単位の TCP MSS の無効化：例

次に、ネイバーグループに TCP MSS を設定し、TCP MSS 値を継承するネイバーのいずれかで継承の無効化を設定する例を示します。

```
router bgp 1
bgp router-id 10.0.0.1
address-family ipv4 unicast
!
neighbor-group n1
tcp mss 500
address-family ipv4 unicast
!
```

```
!
neighbor 10.0.0.2
remote-as 1
use neighbor-group n1
tcp mss inheritance-disable
address-family ipv4 unicast
!
!
!
!
end
```

TCP MSS の設定解除：例

次に、TCP MSS の設定を解除する例を示します。

```
RP/0/0/CPU0:ios(config)#router bgp 1
RP/0/0/CPU0:ios(config-bgp)#neighbor-group n1
RP/0/0/CPU0:ios(config-bgp-nbrgrp)#no tcp mss 500
RP/0/0/CPU0:ios(config-bgp-nbrgrp)#commit
```

ネイバー単位の TCP MSS の確認：例

次に、ルータのネイバー単位の TCP MSS 機能を確認する例を示します。

```
RP/0/0/CPU0:ios#show bgp neighbor 10.0.0.2

BGP neighbor is 10.0.0.2
Remote AS 1, local AS 1, internal link
Remote router ID 10.0.0.2
BGP state = Established, up for 00:09:17
Last read 00:00:16, Last read before reset 00:00:00
Hold time is 180, keepalive interval is 60 seconds
Configured hold time: 180, keepalive: 60, min acceptable hold time: 3
Last write 00:00:16, attempted 19, written 19
Second last write 00:01:16, attempted 19, written 19
Last write before reset 00:00:00, attempted 0, written 0
Second last write before reset 00:00:00, attempted 0, written 0
Last write pulse rcvd Dec 7 11:58:42.411 last full not set pulse count 23
Last write pulse rcvd before reset 00:00:00
Socket not armed for io, armed for read, armed for write
Last write thread event before reset 00:00:00, second last 00:00:00
Last KA expiry before reset 00:00:00, second last 00:00:00
Last KA error before reset 00:00:00, KA not sent 00:00:00
Last KA start before reset 00:00:00, second last 00:00:00
Precedence: internet
Multi-protocol capability received
Neighbor capabilities:
Route refresh: advertised (old + new) and received (old + new)
Graceful Restart (GR Awareness): advertised and received
4-byte AS: advertised and received
Address family IPv4 Unicast: advertised and received
Received 12 messages, 0 notifications, 0 in queue
Sent 12 messages, 0 notifications, 0 in queue
Minimum time between advertisement runs is 0 secs
TCP Maximum Segment Size 500

For Address Family: IPv4 Unicast
```

```
BGP neighbor version 4
Update group: 0.2 Filter-group: 0.1 No Refresh request being processed
Route refresh request: received 0, sent 0
0 accepted prefixes, 0 are bestpaths
Cumulative no. of prefixes denied: 0.
Prefix advertised 0, suppressed 0, withdrawn 0
Maximum prefixes allowed 1048576
Threshold for warning message 75%, restart interval 0 min
AIGP is enabled
An EoR was received during read-only mode
Last ack version 4, Last synced ack version 0
Outstanding version objects: current 0, max 0
Additional-paths operation: None
Send Multicast Attributes
```

次に、TCP MSS の設定を確認する例を示します。

```
RP/0/0/CPU0:ios#show bgp neighbor 10.0.0.2 configuration

neighbor 10.0.0.2
remote-as 1 []
tcp-mss 400 [n:n1]
address-family IPv4 Unicast []
```

次に、TCP 接続エンドポイントの情報を表示する例を示します。

```
RP/0/0/CPU0:ios#show tcp brief

  PCB      VRF-ID     Recv-Q Send-Q Local Address          Foreign Address         State
0x08789b28 0x60000000      0      0  :::179                  :::0
LISTEN
0x08786160 0x00000000      0      0  :::179                  :::0
LISTEN
0xecb0c9f8 0x60000000      0      0  10.0.0.1:12404          10.0.0.2:179            ESTAB
0x0878b168 0x60000000      0      0  11.0.0.1:179            11.0.0.2:61177          ESTAB
0xecb0c6b8 0x60000000      0      0  0.0.0.0:179             0.0.0.0:0
LISTEN
0x08781590 0x00000000      0      0  0.0.0.0:179             0.0.0.0:0
LISTEN
```

次に、特定の PCB 値について TCP 接続情報を表示する例を示します。

```
RP/0/0/CPU0:ios#show tcp pcb 0xecb0c9f8

Connection state is ESTAB, I/O status: 0, socket status: 0
Established at Sun Dec 7 11:49:39 2014

PCB 0xecb0c9f8, SO 0xecb01b68, TCPCB 0xecb01d78, vrfid 0x60000000,
Pak Prio: Medium, TOS: 192, TTL: 255, Hash index: 1322
Local host: 10.0.0.1, Local port: 12404 (Local App PID: 19840)
Foreign host: 10.0.0.2, Foreign port: 179

Current send queue size in bytes: 0 (max 24576)
Current receive queue size in bytes: 0 (max 32768) mis-ordered: 0 bytes
Current receive queue size in packets: 0 (max 0)

Timer Starts Wakeups Next(msec)
Retrans 17 2 0
SendWnd 0 0 0
```

```
TimeWait 0 0 0
AckHold 13 5 0
KeepAlive 1 0 0
PmtuAger 0 0 0
GiveUp 0 0 0
Throttle 0 0 0

iss: 1728179225 snduna: 1728179536 sndnxt: 1728179536
sndmax: 1728179536 sndwnd: 32517 sndcwnd: 1000
irs: 2055835995 rcvnxt: 2055836306 rcvwnd: 32536 rcvadv: 2055868842

SRTT: 206 ms, RTTO: 300 ms, RTV: 59 ms, KRTT: 0 ms
minRTT: 10 ms, maxRTT: 230 ms

ACK hold time: 200 ms, Keepalive time: 0 sec, SYN waittime: 30 sec
Giveup time: 0 ms, Retransmission retries: 0, Retransmit forever: FALSE
Connect retries remaining: 30, connect retry interval: 30 secs

State flags: none
Feature flags: Win Scale, Nagle
Request flags: Win Scale

Datagrams (in bytes): MSS 500, peer MSS 1460, min MSS 500, max MSS 1460

Window scales: rcv 0, snd 0, request rcv 0, request snd 0
Timestamp option: recent 0, recent age 0, last ACK sent 0
Sack blocks {start, end}: none
Sack holes {start, end, dups, rxmit}: none

Socket options: SO_REUSEADDR, SO_REUSEPORT, SO_NBIO
Socket states: SS_ISCONNECTED, SS_PRIV
Socket receive buffer states: SB_DEL_WAKEUP
Socket send buffer states: SB_DEL_WAKEUP
Socket receive buffer: Low/High watermark 1/32768
Socket send buffer : Low/High watermark 2048/24576, Notify threshold 0

PDU information:
#PDU's in buffer: 0
FIB Lookup Cache: IFH: 0x200 PD ctx: size: 0 data:
Num Labels: 0 Label Stack:
```

AiGP によるプレフィックスの生成：例

次に、AiGP メトリック属性を使用してプレフィックスを生成するための設定例を示します。

```
route-policy aigp-policy
  set aigp-metric 4
  set aigp-metric igp-cost
end-policy
!
router bgp 100
 address-family ipv4 unicast
  network 10.2.3.4/24 route-policy aigp-policy
  redistribute ospf osp1 metric 4 route-policy aigp-policy
 !
!
end
```

BGP Accept Own の設定：例

次に、BGP Accept Own を PE ルータに設定する例を示します。

```
router bgp 100
 neighbor 45.1.1.1
   remote-as 100
   update-source Loopback0
   address-family vpnv4 unicast
    route-policy pass-all in
    accept-own
    route-policy drop_111.x.x.x out
   !
   address-family vpnv6 unicast
    route-policy pass-all in
    accept-own
    route-policy drop_111.x.x.x out
   !
  !
```

次の例は、BGP Accept Own のための InterAS-RR の設定を示しています。

```
router bgp 100
 neighbor 45.1.1.1
  remote-as 100
  update-source Loopback0
  address-family vpnv4 unicast
   route-policy rt_stitch1 in
   route-reflector-client
   route-policy add_bgp_ao out
  !
  address-family vpnv6 unicast
   route-policy rt_stitch1 in
   route-reflector-client
   route-policy add_bgp_ao out
  !
 !
extcommunity-set rt cs_100:1
  100:1
end-set
!
extcommunity-set rt cs_1001:1
  1001:1
end-set
!
route-policy rt_stitch1
  if extcommunity rt matches-any cs_100:1 then
    set extcommunity rt cs_1000:1 additive
 endif
end-policy
!
route-policy add_bgp_ao
  set community (accept-own) additive
end-policy
!
```

BGP 不等コストの連続ロード バランシング：例

次に、不等コストの連続ロード バランシングの設定例を示します。

```
interface Loopback0
 ipv4 address 20.20.20.20 255.255.255.255
!
interface MgmtEth0/RSP0/CPU0/0
 ipv4 address 8.43.0.10 255.255.255.0
!
interface TenGigE0/3/0/0
 bandwidth 8000000
 ipv4 address 11.11.11.11 255.255.255.0
 ipv6 address 11:11:0:1::11/64
!
interface TenGigE0/3/0/1
 bandwidth 7000000
 ipv4 address 11.11.12.11 255.255.255.0
 ipv6 address 11:11:0:2::11/64
!
interface TenGigE0/3/0/2
 bandwidth 6000000
 ipv4 address 11.11.13.11 255.255.255.0
 ipv6 address 11:11:0:3::11/64
!
interface TenGigE0/3/0/3
 bandwidth 5000000
 ipv4 address 11.11.14.11 255.255.255.0
 ipv6 address 11:11:0:4::11/64
!
interface TenGigE0/3/0/4
 bandwidth 4000000
 ipv4 address 11.11.15.11 255.255.255.0
 ipv6 address 11:11:0:5::11/64
!
interface TenGigE0/3/0/5
 bandwidth 3000000
 ipv4 address 11.11.16.11 255.255.255.0
 ipv6 address 11:11:0:6::11/64
!
interface TenGigE0/3/0/6
 bandwidth 2000000
 ipv4 address 11.11.17.11 255.255.255.0
 ipv6 address 11:11:0:7::11/64
!
interface TenGigE0/3/0/7
 bandwidth 1000000
 ipv4 address 11.11.18.11 255.255.255.0
 ipv6 address 11:11:0:8::11/64
!
interface TenGigE0/4/0/0
 description CONNECTED TO IXIA 1/3
 transceiver permit pid all
!
interface TenGigE0/4/0/2
 ipv4 address 9.9.9.9 255.255.0.0
 ipv6 address 9:9::9/64
 ipv6 enable
!
route-policy pass-all
  pass
end-policy
!
router static
 address-family ipv4 unicast
  202.153.144.0/24 8.43.0.1
 !
!
```

```
router bgp 100
 bgp router-id 20.20.20.20
 address-family ipv4 unicast
  maximum-paths eibgp 8
  redistribute connected
 !
 neighbor 11.11.11.12
  remote-as 200
  dmz-link-bandwidth
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
 neighbor 11.11.12.12
  remote-as 200
  dmz-link-bandwidth
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
 neighbor 11.11.13.12
  remote-as 200
  dmz-link-bandwidth
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
 neighbor 11.11.14.12
  remote-as 200
  dmz-link-bandwidth
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
 neighbor 11.11.15.12
  remote-as 200
  dmz-link-bandwidth
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
 neighbor 11.11.16.12
  remote-as 200
  dmz-link-bandwidth
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
 neighbor 11.11.17.12
  remote-as 200
  dmz-link-bandwidth
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
 neighbor 11.11.18.12
  remote-as 200
```

```
  dmz-link-bandwidth
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
!
end
```

VRF ダイナミック ルートの設定：例

次に、VRF ダイナミックルートリークを設定する例を示します。

デフォルトの VRF からデフォルト以外の VRF へのルートのインポート

```
vrf vrf_1
 address-family ipv6 unicast
  import from default-vrf route-policy rpl_dynamic_route_import
 !
end
```

非デフォルト VRF からデフォルト VRF へのルートのインポート：

```
vrf vrf_1
 address-family ipv6 unicast
    export to default-vrf route-policy rpl_dynamic_route_export
 !
end
```

復元力のある CE 単位のラベルモードの設定：例

VRF アドレスファミリでの復元力のある CE 単位のラベルモードの設定：例

次に、VRF アドレス ファミリに復元力のある CE 単位のラベルモードを設定する例を示します。

```
RP/0/RSP0/cpu 0: router# configure
RP/0/RSP0/cpu 0: router(config)# router bgp 666
RP/0/RSP0/cpu 0: router(config-bgp)# vrf vrf-pe
RP/0/RSP0/cpu 0: router(config-bgp-vrf)# address-family ipv4 unicast
RP/0/RSP0/cpu 0: router(config-bgp-vrf-af)# label mode per-ce
RP/0/RSP0/cpu 0: router(config-bgp-vrf-af)# end
```

ルートポリシーを使用した復元力のある CE 単位のラベルモードの設定：例

次に、ルートポリシーを使用してふぐ元力のある CE 単位のラベルモードを設定する例を示します。

```
RP/0/RSP0/cpu 0: router# configure
RP/0/RSP0/cpu 0: router(config)# route-policy route1
RP/0/RSP0/cpu 0: router(config-rpl)# set label mode per-ce
RP/0/RSP0/cpu 0: router(config-rpl)# end
```

フロー タグの伝達

フロー タグ伝達機能では、ルート ポリシーとユーザ ポリシー間に相関関係を構築できます。BGP を使用したフロー タグ伝達では、AS 番号、プレフィックス リスト、コミュニティ文字列、および拡張コミュニティなどのルーティング属性に基づいてユーザ側でトラフィックをステアリングできます。フロー タグは論理数値識別子で、FIB ルックアップ テーブル内の FIB エントリのルーティング属性の 1 つとして RIB を通じて配布されます。フロー タグは、RPL からの「set」操作を使用してインスタンス化され、フロー タグ値に対してアクション（ポリシールール）が関連付けられている C3PL PBR ポリシーで参照されます。

フロー タグの伝達は次の場合に使用できます。

- 宛先 IP アドレス（コミュニティ番号を使用）またはプレフィックス（コミュニティ番号または AS 番号を使用）に基づいてトラフィックを分類する。
- カスタマー サイトのサービス レベル契約（SLA）に基づくサービス エッジに到達するパスのコストに合致する TE グループを選択する。
- SLA とそのクライアントに基づいて、特定のカスタマーにトラフィック ポリシー（TE グループの選択）を適用する。
- アプリケーション サーバまたはキャッシュ サーバにトラフィックを迂回させる。

フロー タグ伝達の制限

Border Gateway Protocol を使用した QoS ポリシー伝達（QPPB）とフロー タグ機能の併用については、いくつかの制限があります。次の作業を行います。

- ルート ポリシーには、「set qos-group」または「set flow-tag」のいずれかを使用できますが、prefix-set に両方は使用できません。
- qos-group と route policy flow-tag のルート ポリシーに重複するルートは使用できません。QPPB とフロー タグの機能は、それらが使用するルート ポリシーに重複するルートがない場合に限り、（同じインターフェイス上でも、異なるインターフェイス上でも）共存できます。
- ルート ポリシーとポリシー マップに qos-group と flow-tag を混在させて使用することはお勧めしません。

次の作業

BGP コマンドの詳細については、 *Routing Command Reference for Cisco ASR 9000 Series Routers* を参照してください。

その他の参考資料

ここでは、BGP の実装に関する関連資料について説明します。

関連資料

関連項目	マニュアル タイトル
BGP コマンド：コマンド構文の詳細、コマンド モード、コマンド履歴、デフォルト設定、使用上のガイドライン、および例	*Routing Command Reference for Cisco ASR 9000 Series Routers*
シスコ エクスプレス フォワーディング（CEF）コマンド：詳細なコマンド構文、コマンド モード、コマンド履歴、デフォルト、使用上のガイドライン、および例	*IP Addresses and Services Command Reference for Cisco ASR 9000 Series Routers*
MPLS VPN 設定情報。	*MPLS Configuration Guide for Cisco ASR 9000 Series RoutersMPLS Configuration Guide for Cisco NCS 560 Series Routers*
双方向フォワーディング検出（BFD）	*Interface and Hardware Component Configuration Guide for Cisco ASR 9000 Series Routers* および *Interface and Hardware Component Command Reference for Cisco ASR 9000 Series Routers*
タスク ID 情報。	*System Security Configuration Guide for Cisco ASR 9000 Series Routers*の「Configuring AAA Services on Cisco ASR 9000 Series Router」のモジュール

標準

標準	タイトル
draft-bonica-tcp-auth-05.txt	『*Authentication for TCP-based Routing and Management Protocols*』（R. Bonica、B. Weis、S. Viswanathan、A. Lange、O. Wheeler）
draft-ietf-idr-bgp4-26.txt	『*A Border Gateway Protocol 4*』（Y. Rekhter、T.Li、S. Hares）
draft-ietf-idr-bgp4-mib-15.txt	『*Definitions of Managed Objects for the Fourth Version of Border Gateway Protocol (BGP-4)*』（J. Hass、S. Hares）
draft-ietf-idr-cease-subcode-05.txt	『*Subcodes for BGP Cease Notification Message*』（Enke Chen、V. Gillet）
draft-ietf-idr-avoid-transition-00.txt	『*Avoid BGP Best Path Transitions from One External to Another*』（Enke Chen、Srihari Sangli）
draft-ietf-idr-as4bytes-12.txt	『*BGP Support for Four-octet AS Number Space*』（Quaizar Vohra、Enke Chen）

MIB

MIB	MIB のリンク
—	Cisco IOS XR ソフトウェアを使用して MIB の場所を特定してダウンロードするには、次の URL にある Cisco MIB Locator を使用して、[Cisco Access Products] メニューからプラットフォームを選択します。 https://mibs.cloudapps.cisco.com/ITDIT/MIBS/servlet/index

RFC

RFC	タイトル
RFC 1700	『Assigned Numbers』
RFC 1997	『BGP Communities Attribute』
RFC 2385	『Protection of BGP Sessions via the TCP MD5 Signature Option』
RFC 2439	『BGP Route Flap Damping』
RFC 2545	『Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing』
RFC 2796	『BGP Route Reflection - An Alternative to Full Mesh IBGP』
RFC 2858	『Multiprotocol Extensions for BGP-4』
RFC 2918	『Route Refresh Capability for BGP-4』
RFC 3065	『Autonomous System Confederations for BGP』
RFC 3392	『Capabilities Advertisement with BGP-4』
RFC 4271	『A Border Gateway Protocol 4 (BGP-4)』
RFC 4364	『BGP/MPLS IP Virtual Private Networks (VPNs)』
RFC 4724	『*Graceful Restart Mechanism for BGP*』

シスコのテクニカル サポート

説明	リンク
シスコのテクニカル サポート Web サイトでは、製品、テクノロジー、ソリューション、技術的なヒント、およびツールへのリンクなどの、数千ページに及ぶ技術情報が検索可能です。Cisco.com に登録済みのユーザは、このページから詳細情報にアクセスできます。	http://www.cisco.com/techsupport