外部 BGP を使用したサービス プロバイダーとの接続

このモジュールでは、Border Gateway Protocol（BGP; ボーダー ゲートウェイ プロトコル）ネットワークが、インターネット サービス プロバイダー（ISP）など外部ネットワークにあるピア デバイスへアクセスできるようにするための設定作業について説明します。BGP は、組織間にループのないルーティングを提供するために設計されたドメイン間ルーティング プロトコルです。異なる自律システムのピアとのルーティング アップデートの交換のために、External BGP（eBGP; 外部 BGP）ピアリング セッションが設定されます。トラフィックのフィルタリングのための BGP ポリシー設定作業など、インバウンドとアウトバウンドのトラフィックを管理するための作業について説明します。サービス プロバイダーへの接続に冗長性を持たせるためのマルチホーミングについても説明します。

機能情報の検索

ご使用のソフトウェア リリースが、このモジュールで説明している機能の一部をサポートしていない場合があります。最新の機能情報および警告については、ご使用のプラットフォームおよびソフトウェア リリースのリリースノートを参照してください。このモジュールに記載されている機能に関する情報を検索したり、各機能がサポートされているリリースに関するリストを参照したりするには、「外部 BGP を使用したサービス プロバイダーとの接続の機能情報」（P.81）を参照してください。

プラットフォームのサポートおよびソフトウェア イメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator には、http://www.cisco.com/go/cfn からアクセスしてください。Cisco.com のアカウントは必要ありません。

マニュアルの内容

外部 BGP を使用したサービス プロバイダーとの接続の前提条件

- サービス プロバイダーとの接続前に、BGP プロセスとピアの基本的な設定方法を理解しておく必要があります。詳しくは、「Cisco BGP Overview」および「Configuring a Basic BGP Network」モジュールを参照してください。
- ネットワークをサービス プロバイダーに接続する場合に BGP 機能を設定する際、この章の作業と概念が役立ちます。インターネットへの接続それぞれについて、Internet Assigned Numbers Authority（IANA; インターネット割り当て番号局）から割り当てられた自律システム番号を持っている必要があります。

サービス プロバイダーとの外部 BGP を使用した接続の制約事項

- Cisco IOS ソフトウェアを実行するルータは、1 つの BGP ルーティング プロセスだけを実行し、1 つの BGP 自律システムだけのメンバになるように設定できます。ただし、BGP ルーティング プロセスと自律システムでは、複数のアドレス ファミリ設定をサポートできます。
- ポリシー リストは、Cisco IOS Release 12.0(22)S および 12.2(15)T よりも前の Cisco IOS ソフトウェアではサポートされていません。古いバージョンの Cisco IOS ソフトウェアを実行中のルータをリロードすると、ルーティング ポリシーの設定の一部が失われることがあります。

外部 BGP を使用したサービス プロバイダーとの接続の概要

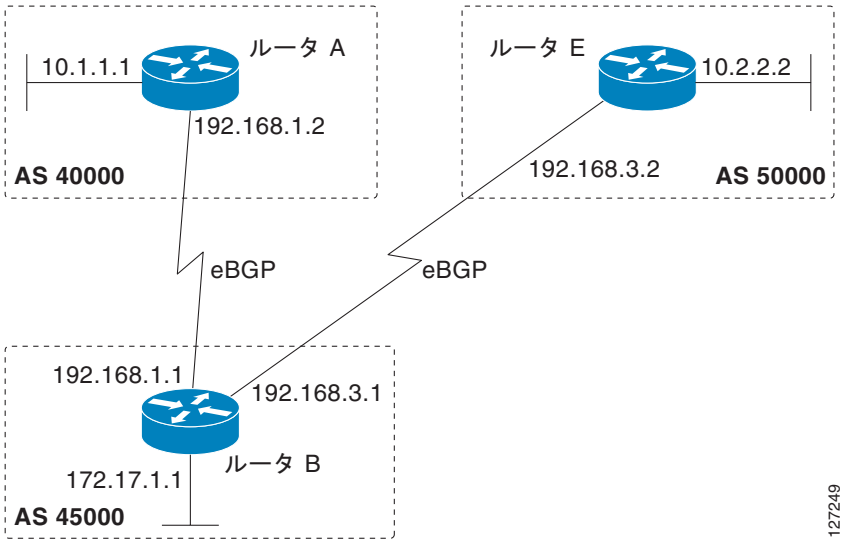
外部 BGP を使用して ISP への接続作業を行うには、次の概念を理解しておく必要があります。

外部 BGP ピアリング

BGP は、組織間にループが発生しないルーティング リンクを実現することを目的としたドメイン間ルーティング プロトコルです。BGP は、信頼できるトランスポート プロトコル上で運用するよう設計され、トランスポート プロトコルとして TCP（ポート 179）を使用します。宛先の TCP ポートは 179 が割り当てられ、ローカル ポートではランダムなポート番号が割り当てられます。Cisco IOS ソフトウェアは、ISP がインターネット構築に使用している BGP バージョン 4 をサポートしています。RFC 1771 では、プロトコルをインターネット規模での使用に合わせるため、新機能の BGP への追加や検討が多数行われました。

異なる自律システムの BGP ピアとのルーティング アップデートの交換のために、外部 BGP ピアリング セッションが設定されます。BGP ルーティング プロセスは、eBGP ピアが WAN 接続などによって直接接続されるものとして設計されています。しかし、実際の使用においてはこのルールではルーティングできないケースが多々あります。マルチホップ ネイバーのピアリング セッションは **neighbor ebgp-multihop** コマンドで設定します。図 1 に、3 つのルータ間のシンプルな eBGP ピアリングを示します。ルータ B は、ルータ A とルータ E にピアリングされています。非常にシンプルなネットワーク設計ですが、図 1 では、ルータ A とルータ E との間のピアリング確立に **neighbor ebgp-multihop** コマンドが使用できます。BGP はネットワーク内のネクストホップについての情報を NEXT_HOP アトリビュートを使用して転送します。デフォルトでは eBGP ピアリング セッション内のルートをアドバタイズするインターフェイスの IP アドレスに設定されています。発信元インターフェイスは、物理インターフェイスかループバック インターフェイスです。

図 1　　別の自律システム内の BGP ピア



eBGP ピアリング セッションの確立にはループバック インターフェイスが好まれます。ループバック インターフェイスの方がインターフェイス フラッピングの影響を受けにくいからです。ネットワーキング デバイスのインターフェイスは、障害が発生したり、メンテナンスのために運転を停止する場合があります。障害やメンテナンスのために管理上あるインターフェイスを起動や停止することを、フラップといいます。ループバック インターフェイスは安定した発信元インターフェイスを実現するもので、IP ルーティング プロトコルがループバック インターフェイスに割り当てられたサブネットをアドバタイズする限り、発信元インターフェイスに割り当てられた IP アドレスがいつでも到達可能になるようにします。ループバック インターフェイスにより、/32 ビット マスクのアドレス 1 つを設定することで、アドレス空間を節約できます。ループバック インターフェイスを eBGP ピアリング セッションのために設定する前に、**neighbor update-source** コマンドを設定してループバック インターフェイスを指定する必要があります。このように設定することで、ループバック インターフェイスが

発信元インターフェイスとなり、その IP アドレスがこのループバックを通してアドバタイズされるルートのネクストホップとしてアドバタイズされます。ループバック インターフェイスをシングルホップ eBGP ピアの接続に使用する場合、先に **neighbor disable-connected-check** コマンドを設定しなければ、eBGP ピアリング セッションは確立できません。

外部ネットワークとの接続により、使用するネットワークからのトラフィックを別のネットワークへ、インターネットを通じて転送することができるようになります。ネットワークに入ってくるトラフィックや、場合によっては通過して行くトラフィックもあるでしょう。BGP には、ネットワークへのトラフィックの出入りを変化させたり、インバウンドとアウトバウンドのトラフィックのフィルタリング用 BGP ポリシーを作成したりするための、さまざまな方法が含まれています。トラフィック フローを変化させるのに、BGP はアップデート メッセージに含まれる、または BGP ルーティング アルゴリズムで使用される BGP アトリビュートを使用します。トラフィックのフィルタリング用 BGP ポリシーでは、ルート マップ、AS-path アクセス リストなどのアクセス リスト、フィルタ リスト、ポリシー リスト、および配信リストを伴った BGP アトリビュートの一部も使用されます。バックアップやパフォーマンス向上のために 1 つの ISP への複数接続や複数の ISP への接続が存在する場合、外部接続の管理にマルチホーミング技術が関係してくることがあります。自律システムや物理的境界を超えてさまざまなコミュニティ アトリビュートによるタギングを BGP ルートに行うことで、個別に permit 文や deny 文を羅列した巨大なリストを扱わずにすみます。

BGP 自律システム番号の形式

2009 年 1 月まで、企業に割り当てられていた BGP 自律システム番号は、RFC 4271『*A Border Gateway Protocol 4 (BGP-4)*』に記述された、1 ～ 65535 の範囲の 2 オクテットの数値でした。自律システム番号の要求の増加に伴い、インターネット割り当て番号局（IANA）により割り当てられる自律システム番号は 2009 年 1 月から 65536 ～ 4294967295 の範囲の 4 オクテットの番号になります。RFC 5396『*Textual Representation of Autonomous System (AS) Numbers*』には、自律システム番号を表す 3 つの方式が記述されています。シスコでは、次の 2 つの方式を実装しています。

- asplain：10 進表記方式。2 バイトおよび 4 バイト自律システム番号をその 10 進数値で表します。たとえば、65526 は 2 バイト自律システム番号、234567 は 4 バイト自律システム番号になります。
- asdot：自律システム ドット付き表記。2 バイト自律システム番号は 10 進数で、4 バイト自律システム番号はドット付き表記で表されます。たとえば、65526 は 2 バイト自律システム番号、1.169031（10 進表記の 234567 をドット付き表記にしたもの）は 4 バイト自律システム番号になります。

自律システム番号を表す 3 つ目の方法については、RFC 5396 を参照してください。

asdot だけを使用する自律システム番号形式

Cisco IOS Release 12.0(32)S12、12.4(24)T、およびそれ以降のリリースでは、4 オクテット（4 バイト）の自律システム番号は asdot 表記法だけで入力および表示されます。たとえば、1.10 または 45000.64000 です。4 バイト自律システム番号のマッチングに正規表現を使用する場合、asdot 形式には正規表現で特殊文字となるピリオドが含まれていることに注意します。正規表現でのマッチングに失敗しないよう、1\.14 のようにピリオドの前にバックスラッシュを入力する必要があります。表 1 は、asdot 形式だけが使用できる Cisco IOS イメージで、2 バイトおよび 4 バイト自律システム番号の設定、正規表現とのマッチング、および **show** コマンド出力での表示に使用される形式をまとめたものです。

表 1　asdot だけを使用する 4 バイト自律システム番号形式

形式	設定形式	show コマンド出力および正規表現のマッチング形式
asdot	2 バイト：1 ～ 65535 4 バイト：1.0 ～ 65535.65535	2 バイト：1 ～ 65535 4 バイト：1.0 ～ 65535.65535

asplain をデフォルトとする自律システム番号形式

Cisco IOS Release 12.0(32)SY8、12.0(33)S3、12.2(33)SRE、12.2(33)XNE、12.2(33)SXI1、およびそれ以降のリリースでは、シスコ実装の 4 バイト自律システム番号で asplain がデフォルトの自律システム番号表示形式として使用されていますが、4 バイト自律システム番号は asplain および asdot 形式のどちらにも設定できます。また、正規表現で 4 バイト自律システム番号とマッチングするためのデフォルト形式は asplain であるため、4 バイト自律システム番号とマッチングする正規表現はすべて、asplain 形式で記述する必要があります。デフォルトの **show** コマンド出力で、4 バイト自律システム番号が asdot 形式で表示されるように変更する場合は、ルータ コンフィギュレーション モードで **bgp asnotation dot** コマンドを使用します。デフォルトで asdot 形式がイネーブルにされている場合、正規表現の 4 バイト自律システム番号のマッチングには、すべて asdot 形式を使用する必要があり、使用しない場合正規表現によるマッチングは失敗します。表 2 および 表 3 に示すように、4 バイト自律システム番号は asplain と asdot のどちらにも設定できるとはいえ、**show** コマンド出力と正規表現を用いた 4 バイト自律システム番号のマッチング制御には 1 つの形式だけが使用されます。デフォルトは asplain 形式です。**show** コマンド出力の表示と正規表現のマッチング制御で asdot 形式の 4 バイト自律システム番号を使用する場合、**bgp asnotation dot** コマンドを設定する必要があります。**bgp asnotation dot** コマンドをイネーブルにした後で、**clear ip bgp *** コマンドを入力し、すべての BGP セッションについて、ハード リセットを開始する必要があります。

（注） 4 バイト自律システム番号をサポートしているイメージにアップグレードしている場合でも、2 バイト自律システム番号を使用できます。4 バイト自律システム番号に設定された形式にかかわらず、2 バイト自律システムの **show** コマンド出力と正規表現のマッチングは変更されず、asplain（10 進数）形式のままになります。

表 2　asplain をデフォルトとする 4 バイト自律システム番号形式

形式	設定形式	show コマンド出力および正規表現のマッチング形式
asplain	2 バイト：1 ～ 65535 4 バイト：65536 ～ 4294967295	2 バイト：1 ～ 65535 4 バイト：65536 ～ 4294967295
asdot	2 バイト：1 ～ 65535 4 バイト：1.0 ～ 65535.65535	2 バイト：1 ～ 65535 4 バイト：65536 ～ 4294967295

表 3　asdot を使用する 4 バイト自律システム番号形式

形式	設定形式	show コマンド出力および正規表現のマッチング形式
asplain	2 バイト：1 ～ 65535 4 バイト：65536 ～ 4294967295	2 バイト：1 ～ 65535 4 バイト：1.0 ～ 65535.65535
asdot	2 バイト：1 ～ 65535 4 バイト：1.0 ～ 65535.65535	2 バイト：1 ～ 65535 4 バイト：1.0 ～ 65535.65535

予約済みおよびプライベートの自律システム番号

Cisco IOS Release 12.0(32)S12、12.0(32)SY8、12.0(33)S3、12.2(33)SRE、12.2(33)XNE、12.2(33)SXI1、12.4(24)T、およびそれ以降のリリースでは、RFC 4893 がシスコの BGP 実装でサポートされています。RFC 4893 は、2 バイト自律システム番号から 4 バイト自律システム番号への段階的移行を BGP がサポートできるように開発されました。RFC 4893 では新たに 23456 が予約済み（プライベート）自律システム番号に指定され、Cisco IOS CLI ではこの番号を自律システム番号として設定できなくなっています。

RFC 5398『*Autonomous System (AS) Number Reservation for Documentation Use*』では、文書化を目的として新たに予約された自律システム番号について説明されています。予約済み番号を使用することで、設定例を正確に文書化しつつ、その設定がそのままコピーされた場合でも製品ネットワークに競合が発生することを防止できます。予約済み番号は IANA 自律システム番号レジストリに記載されています。予約済み 2 バイト自律システム番号は 64496 ～ 64511 の連続したブロック、予約済み 4 バイト自律システム番号は 65536 ～ 65551 をその範囲としています。

64512 ～ 65534 を範囲とするプライベートの 2 バイト自律システム番号は依然有効で、65535 は特殊な目的のために予約されています。プライベート自律システム番号は内部ルーティング ドメインで使用できますが、インターネットにルーティングされるトラフィックについては変換が必要です。プライベート自律システム番号を外部ネットワークへアドバタイズするように BGP を設定しないでください。Cisco IOS ソフトウェアは、デフォルトではルーティング アップデートからプライベート自律システム番号を削除しません。ISP がプライベート自律システム番号をフィルタリングすることを推奨します。

（注） パブリック ネットワークおよびプライベート ネットワークに対する自律システム番号の割り当ては、IANA が管理しています。予約済み番号の割り当てや自律システム番号の登録申込など、自律システム番号についての情報については、http://www.iana.org/ を参照してください。

BGP アトリビュート

デフォルトでは、BGP は宛先ホストまたはネットワークへの最良パスとして 1 つのパスを選択します。どのルートを最良パスとして BGP ルーティング テーブルにインストールするかを決定するために、最良パス選択アルゴリズムはパスのアトリビュートを分析します。それぞれのパスには、BGP 最良パス分析で使用されるさまざまなアトリビュートがついています。Cisco IOS ソフトウェアは、Command-Line Interface（CLI; コマンドライン インターフェイス）を通してそのようなアトリビュートを変更することで、BGP パス選択に影響を与えられるようになっています。BGP パス選択はまた、標準 BGP ポリシー設定によっても変化させることができます。

BGP では、最良パス選択アルゴリズムを使用して、全体的に良好なルートのセットを検索します。このようなルートは、潜在的なマルチパスです。Cisco IOS Release 12.2(33)SRD 以降のリリースでは、許可される最大数よりも多くの全体的に良好なマルチパスが存在する場合、最も古いパスがマルチパスとして選択されます。

BGP は、アップデート メッセージにパス アトリビュート情報を含めることができます。BGP アトリビュートはルートの特徴を記述するもので、ソフトウェアはアトリビュートをどのルートをアドバタイズするか決定を下すのを助けるのに使用します。一部のアトリビュート情報は、BGP 対応のネットワーキング デバイスでも設定できます。アトリビュートには、アップデート メッセージに常に含まれる必須のものと、任意のものとがあります。次のような BGP アトリビュートが設定可能です。

- AS-path
- Community
- Local_Pref
- Multi_Exit_Discriminator（MED）
- Next_Hop
- Origin

AS-path

このアトリビュートは、ルーティング情報が通過してきた自律システム番号のセットまたはリストを含んでいます。BGP スピーカーは、アップデート メッセージを外部ピアへ転送する際に、自分の自律システム番号をリストに加えます。

Community

ネットワークや自律システム、または物理的境界にかかわりなく、共通のプロパティを持つネットワーキング デバイスをグループ化するには、BGP コミュニティを使用します。大規模ネットワークにおいて、共通のルーティング ポリシーをプレフィクス リストやアクセス リストで適用するには、ネットワーキング デバイスごとに個別のピア文が必要になります。BGP コミュニティ アトリビュートを使えば、共通のルーティング ポリシーを持つ BGP ネイバーに、コミュニティ タグに基づいてインバウンドやアウトバウンドのルート フィルタをインプリメントでき、個別に permit 文や deny 文を羅列した巨大なリストを扱わずにすみます。

Local_Pref

自律システム内で、Local_Pref アトリビュートは BGP ピア間のアップデート メッセージすべてに含まれます。同一の宛先に対し複数のパスがある場合、最も大きな値を持つローカル プリファレンス アトリビュートは、ローカルの自律システムからの優先アウトバウンド パスを示します。ランキングが最高のルートが内部のピアにアドバタイズされます。Local_Pref の値は外部ピアへは転送されません。

Multi_Exit_Discriminator

MED アトリビュートは、（外部ピアに）自律システムへの優先パスを示します。自律システムへのエントリ ポイントが複数ある場合、MED を使って別の自律システムに特定のエントリ ポイントを選択するようはたらきかけることができます。低い値の MED メトリックの方が高い値の MED メトリックより優先されるソフトウェアでは、メトリックが割り当てられます。MED メトリックは自律システムの間で交換されますが、MED が自律システムに転送された後、MED メトリックはデフォルト値である 0 にリセットされます。アップデートが内部 BGP（iBGP）ピアに送られると、MED はまったく変更を加えられずに受け渡されていくため、同一の自律システム内のすべてのピアが一貫したパス選択を行うことができます。

デフォルトでは、ルータは同じ自律システムにある BGP ピアからのパスの MED アトリビュートだけを比較します。**bgp always-compare-med** コマンドを設定することで、ルータに別の自律システムのピアからのメトリックを比較させることができます。

（注） BGP MED についての Internet Engineering Task Force（IETF; インターネット技術特別調査委員会）の決定では、欠落している MED には無限の値を割り当て、MED 変数の欠落したルートの優先度を最低にしています。Cisco IOS ソフトウェアが稼動する BGP ルータでは、MED アトリビュートのないルートを値 0 を持つ MED として扱い、MED 変数の欠落したルートが最優先とすることが、デフォルトの動作になっています。IETF 標準に準拠してルータを設定する場合、**bgp bestpath med missing-as-worst** ルータ コンフィギュレーション コマンドを使用します。

Next_Hop

Next_Hop アトリビュートは、宛先への BGP ネクストホップとして使用されるネクストホップ IP アドレスを示します。ルータは、再帰的ルックアップによってルーティング テーブルで BGP ネクストホップを検索します。外部 BGP（eBGP）では、ネクストホップはアップデートを送信したピアの IP アドレスです。内部 BGP（iBGP）は、内部で生成されたルートのプレフィクスをアドバタイズしたピアの IP アドレスを、ネクストホップのアドレスとして設定します。eBGP から学習した iBGP へのルートのいずれかがアドバタイズされた場合、Next_Hop アトリビュートは変更されません。

ルータが BGP ルートを使用するためには、BGP ネクストホップの IP アドレスが到達可能でなければなりません。到着可能性情報は通常 IGP によって提供され、IGP での変更はネットワーク バックボーンを介したネクストホップ アドレスの転送に影響を与える可能性があります。

Origin

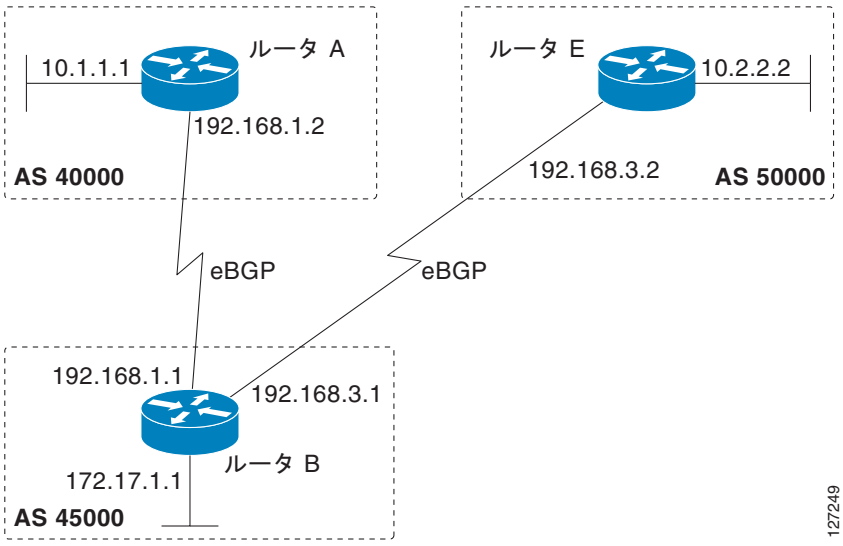
このアトリビュートは、ルートがどのように BGP ルーティング テーブルに含まれたかを示します。Cisco IOS ソフトウェアにおいて、BGP **network** コマンドを使用して定義されたルートには、Interior Gateway Protocol（IGP）の送信コードが与えられています。Exterior Gateway Protocol（EGP）から配信されたルートは、EGP の送信元を使用してコーディングされ、その他のプロトコルから再配布されたルートは「不完全」と定義されます。BGP の送信元決定ポリシーでは、「不完全」よりも EGP が、EGP よりも IGP が優先されます。

マルチホーミング

1 つの自律システムが複数のサービス プロバイダーに接続する場合に、マルチホーミングが定義されます。1 つのサービス プロバイダーの信頼性に何か問題が生じた場合、バックアップ接続を使用できます。パフォーマンスの問題もマルチホーミングで改善する場合があります。宛先ネットワークへのもっと適したパスを使用できることがあるからです。

自分がサービス プロバイダーでない場合、インターネットのトラフィックが自律システム内を通過して帯域幅を使いきってしまうことがないよう、ルーティング設定を注意深く検討する必要があります。図 2 では、自律システム 45000 が自律システム 40000 と自律システム 50000 とにマルチホーミングされています。自律システム 45000 がサービス プロバイダーでないと仮定すると、ロード バランシングや何らかのルーティング ポリシーを使用して、自律システム 45000 からのトラフィックが自律システム 40000 にも自律システム 50000 にも到達できるように、しかし同時に転送トラフィックはあったとしても少なく抑えるように設定する必要があります。

図 2　　マルチホーミング トポロジ



中継トラフィックと非中継トラフィック

自律システム内のほとんどのトラフィックは、その自律システム内にある発信元または宛先 IP アドレスを含んでおり、このトラフィックを非中継（またはローカル）トラフィックと呼びます。その他のトラフィックを中継トラフィックとして定義します。インターネットを介したトラフィックが増えるにつれて、中継トラフィックの制御がいっそう重要になります。

サービス プロバイダーは中継自律システムと考えることができ、他のすべての中継プロバイダーへの接続性を提供できなければなりません。現実には、ほとんどのサービス プロバイダーは中継トラフィックすべてを許容できるだけの帯域幅を持っていないため、それらのプロバイダーはそのような接続性を 1 次プロバイダーから購入する必要があります。

通常は中継トラフィックを許可しない自律システムはスタブ自律システムと呼ばれ、インターネットには 1 つのサービス プロバイダーを通してリンクします。

BGP ポリシー設定

BGP ポリシー設定は、BGP ルーティング プロセスによるプレフィクス処理を制御し、インバウンドおよびアウトバウンドのアドバタイズメントからルートをフィルタリングするために使われます。プレフィクス処理は、BGP タイマーの調整、BGP によるパス アトリビュートの扱いの変更、ルーティング プロセスが受け入れるプレフィクスの数の制限、および BGP プレフィクス ダンプニングの設定によって制御できます。インバウンドおよびアウトバウンドのアドバタイズメントは、ルート マップ、フィルタ リスト、IP プレフィクス リスト、自律システムパス アクセス リスト、IP ポリシー リスト、および配信リストを使用してフィルタリングされます。表 4 に、BGP ポリシー フィルタの処理順序を示します。

表 4　　BGP ポリシー処理順序

インバウンド	アウトバウンド
ルート マップ	配信リスト
フィルタリスト、AS パス アクセス リスト、または IP ポリシー	IP プレフィクス リスト
IP プレフィクス リスト	フィルタリスト、AS パス アクセス リスト、または IP ポリシー
配信リスト	ルート マップ

（注） Cisco IOS Release 12.0(22)S、12.2(15)T、12.2(18)S、およびそれ以降のリリースでは、**ip as-path access-list** コマンドを使用して設定できる自律システム アクセス リストの上限値が、199 から 500 に増加しました。

設定変更のためにルーティング ポリシーに変更が生じた場合は、必ず **clear ip bgp** コマンドを使用して、BGP ピアリング セッションをリセットする必要があります。Cisco IOS ソフトウェアは、BGP ピアリング セッションのリセットとして、次の 3 つのメカニズムをサポートしています。

- ハード リセット：ハード リセットでは、指定されたピアリング セッションは TCP 接続を含めて破棄され、指定されたピアからのルートが削除されます。
- ソフト リセット：ソフト リセットは、保存されたプレフィクス情報を使用し、既存のピアリング セッションを廃棄せずに BGP ルーティング テーブルの再構成とアクティブ化を行います。ソフト リセットは保存されたアップデート情報を使用するため、アップデート保存用のメモリを追加することで、ネットワークを中断することなく新しい BGP ポリシーを適用できます。ソフト リセットは、インバウンドとアウトバウンドのセッションに設定できます。
- ダイナミック インバウンド ソフト リセット：これは RFC 2918 に定義されているルート リフレッシュ機能で、サポートしているピアへのルート リフレッシュ要求を交換することにより、ローカル ルータがインバウンド ルーティング テーブルを動的にリセットできるようにするものです。ルート リフレッシュ機能は、中断を伴わないポリシー変更についてはアップデート情報をローカルに保存

しません。その代わり、サポートしているピアとの動的な交換に依存します。ルート リフレッシュは、最初にピア間の BGP 機能ネゴシエーションを通じてアドバタイズされる必要があります。すべての BGP ルータが、ルート リフレッシュ機能をサポートしていなければなりません。

BGP ルータがこの機能をサポートしているか確認するには、**show ip bgp neighbors** コマンドを使用します。ルータがルート リフレッシュ機能をサポートしている場合、次のメッセージが出力されます。

```
Received route refresh capability from peer.
```

BGP コミュニティ

BGP コミュニティは、ネットワーク、自律システム、または物理的境界にかかわりなく、共通のプロパティを持つルートをグループ化する（カラー ルートとも呼ばれる）のに使用されます。大規模ネットワークにおいて、共通のルーティング ポリシーをプレフィクス リストやアクセス リストで適用するには、ネットワーキング デバイスごとに個別のピア文が必要になります。BGP コミュニティ アトリビュートを使えば、共通のルーティング ポリシーを持つ BGP スピーカーに、コミュニティ タグに基づいてインバウンドやアウトバウンドのルート フィルタをインプリメントでき、個別に permit 文や deny 文を羅列した巨大なリストを扱わずにすみます。

標準コミュニティ リストは、よく知られたコミュニティと特定のコミュニティ番号を設定するために使用されます。拡張コミュニティ リストは、正規表現を使用してコミュニティをフィルタリングするために使用されます。正規表現は、コミュニティ アトリビュートのマッチング パターン設定に使用されます。

コミュニティ アトリビュートはオプションです。そのため、コミュニティを認識しないネットワーキング デバイスは通過できません。コミュニティを認識するネットワーキング デバイスでも、コミュニティを扱うよう設定しなければ、アトリビュートは無視されます。

4 つの定義済みコミュニティがあります。

- no-export：外部 BGP ピアへアドバタイズしない。
- no-advertise：このルートをどのピアにもアドバタイズしない。
- internet：このルートをインターネットにアドバタイズする。BGP 対応のネットワーキング デバイスはすべてその所属となります。
- local-as：ローカルの自律システムの外には送らない。

Cisco IOS Release 12.2(8)T では、BGP 名前付きコミュニティ リストが導入されました。BGP 名前付きコミュニティ リストによって、わかりやすい名前をコミュニティ リストに割り当てられるようになりました。設定可能なコミュニティ リスト数の制限はありません。名前付きコミュニティ リストは、正規表現や番号付きコミュニティ リストによって設定可能です。番号付きコミュニティのルールは、設定可能なコミュニティ リスト数の上限がないことを除き、すべて名前付きコミュニティ リストにも適用されます。

（注） 標準および拡張コミュニティ リストには、いずれも各タイプのリスト内で設定可能なコミュニティ グループ数に 100 という上限がありました。名前付きコミュニティ リストでは、この制限がありません。

拡張コミュニティ

拡張コミュニティ アトリビュートは、Virtual Routing and Forwarding（VRF; 仮想ルーティング/転送）インスタンスおよび Multiprotocol Label Switching（MPLS; マルチプロトコル ラベル スイッチング）Virtual Private Network（VPN; バーチャル プライベート ネットワーク）のルートの設定、フィルタリ

ング、識別に使用されます。アクセス リストの標準ルールすべてが、拡張コミュニティ リストの設定に適用されます。正規表現は、拡張コミュニティ リスト番号の拡張範囲によってサポートされています。正規表現の設定オプションはすべてサポートされます。Route Target（RT; ルート ターゲット）および Site of Origin（SoO）拡張コミュニティ アトリビュートは、拡張コミュニティ リストの標準範囲でサポートされます。

ルート ターゲット拡張コミュニティ アトリビュート

RT 拡張コミュニティ アトリビュートは、**ip extcommunity-list** コマンドの **rt** キーワードで設定されます。このアトリビュートは、configured route target とタグ付けされたルートを受け取る可能性があるサイトと VRF のセットとの識別に使用します。ルート付き route target 拡張コミュニティ アトリビュートにより、対応するサイトから受信したトラフィックのルーティングに使用するサイト別のフォワーディング テーブルにルートを置くことが可能になります。

Site of Origin 拡張コミュニティ アトリビュート

SoO 拡張コミュニティ アトリビュートは、**ip extcommunity-list** コマンドの **soo** キーワードで設定されます。このアトリビュートは、Provider Edge（PE; プロバイダー エッジ）ルータがルートを学習したサイトを一意に識別します。ある特定のサイトから学習したルートにはすべて、サイトが接続されている PE ルータの数にかかわらず、同一の SoO 拡張コミュニティ アトリビュートが割り当てられる必要があります。マルチホーミングされているサイトでは、このアトリビュートを設定することでルーティングにループが発生するのを防止できます。SoO 拡張コミュニティ アトリビュートはインターフェイス上で設定され、再配布によって BGP へ伝播されます。SoO 拡張コミュニティ アトリビュートは、VRF から学習したルートへ適用することができます。スタブ サイトやマルチホーミングされていないサイトには、SoO 拡張コミュニティ アトリビュートを設定しないでください。

IP 拡張コミュニティリスト コンフィギュレーション モード

名前付きおよび番号付きコミュニティ リストは、IP 拡張コミュニティリスト コンフィギュレーション モードで設定することができます。IP 拡張コミュニティリスト コンフィギュレーション モードは、グローバル コンフィギュレーション モードで使用できる機能すべてをサポートしています。さらに、次のような操作も行えます。

- 拡張コミュニティ リスト エントリにシーケンス番号を設定する。
- 既存の拡張コミュニティ リスト エントリのシーケンス番号を再設定する。
- デフォルト値を使用するよう、拡張コミュニティ リストを設定する。

デフォルトのシーケンス番号

シーケンス番号が指定されていない場合、デフォルト動作が設定されている場合、および拡張コミュニティリストのシーケンス番号が開始番号や後続エントリ用増分の指定なく再割り当てされた場合、拡張コミュニティ リスト エントリは 10 番から開始され、後続のエントリでは 1 エントリにつき 10 ずつ増えていきます。

拡張コミュニティ リストのシーケンス番号再割り当て

拡張コミュニティ リスト エントリは、拡張コミュニティ リスト単位を基本としてシーケンス番号の割り当てと再割り当てが行われます。**resequence** コマンドを引数なしで使用すると、リスト内のすべてのエントリにデフォルトのシーケンス番号割り当てを行えます。**resequence** コマンドでは、最初のエントリ用のシーケンス番号や後続のエントリごとの数値の増減範囲を設定することもできます。設定できるシーケンス番号の範囲は、1 ～ 2147483647 です。

管理ディスタンス

管理ディスタンスは、異なるルーティング プロトコルのプリファレンスを測定する方法です。BGP にある **distance bgp** コマンドで、外部、内部、ローカルという 3 つのルート タイプの管理ディスタンスを、それぞれ設定することができます。他のプロトコル同様、BGP も管理ディスタンスが最小となるルートを優先します。

BGP ルート マップ ポリシー リスト

BGP ルート マップ ポリシー リストにより、ネットワーク オペレータはルート マップ match 句をグループ化して、ポリシー リストと呼ばれる名前付きリストにすることができます。ポリシー リスト機能はマクロに似ています。ルート マップでポリシー リストが参照されると、match 句がすべて評価され、ルート マップで直接設定された場合と同様に処理されます。この機能強化により、中規模から大規模のネットワークでの BGP ルーティング ポリシーの BGP 設定が単純になりました。ネットワーク オペレータが match 句のグループを持つポリシー リストを事前に設定しておき、さまざまなルート マップ内でそれらのポリシー リストを参照できるからです。複数のルート マップのエントリに繰り返し現れる一群の match 句を、ネットワーク オペレータがそれぞれ手動で再設定する必要がなくなりました。

ルート マップで設定されるポリシー リスト機能はマクロに似ており、次のような機能や特長を持っています。

- ルート マップ内でポリシー リストが参照されると、ポリシー リスト内の match 文すべてが評価され、処理される。
- 1 つのルート マップに 2 つ以上のポリシー リストを設定できる。ポリシー リストはルート マップ内で AND や OR を使用して評価されるように設定可能です。
- ポリシーリストは、同じルート マップ内にあるがポリシー リストの外で設定されている他の既存の match および set 文とも共存可能。
- 1 つのルート マップ エントリ内で複数のポリシー リストがマッチングを行う場合、ポリシー リストすべては受信アトリビュートだけでマッチング。

ポリシー リストがサポートするのは match 句だけで、set 句はサポートしていません。ポリシー リストは、再配布を含めルート マップのアプリケーションすべてに設定でき、同一のルート マップ エントリ内でポリシー リストと別に設定される match および set 句と共存させることもできます。

（注） ポリシー リストは BGP だけでサポートされ、他の IP ルーティング プロトコルではサポートされません。

外部 BGP を使用したサービス プロバイダーとの接続方法

ここでは、次の作業について説明します。

- 「インバウンド パス選択の変更」（P.13）
- 「アウトバウンド パス選択への影響」（P.20）
- 「ISP との BGP ピアリングの設定」（P.27）
- 「BGP ポリシーの設定」（P.41）

インバウンド パス選択の変更

BGP を使用して、別の自律システムにあるパスの選択を変化させることができます。明らかに最適なルート以外のパスを BGP に選ばせたい場合もあります。たとえば、中継トラフィックの一部が自律システムを通過するのを避けたい場合や、非常に遅い、または輻輳しているリンクを避けたい場合です。BGP では、次の BGP アトリビュートのいずれかを使用して、インバウンド パスの選択を変化させることができます。

- AS-path
- MED

インバウンド パス選択を変化させる場合、次の作業のいずれかを実行します。

- 「AS-path アトリビュートの変更によるインバウンド パス選択の変化」（P.13）
- 「MED アトリビュートの設定によるインバウンド パス選択の変化」（P.17）

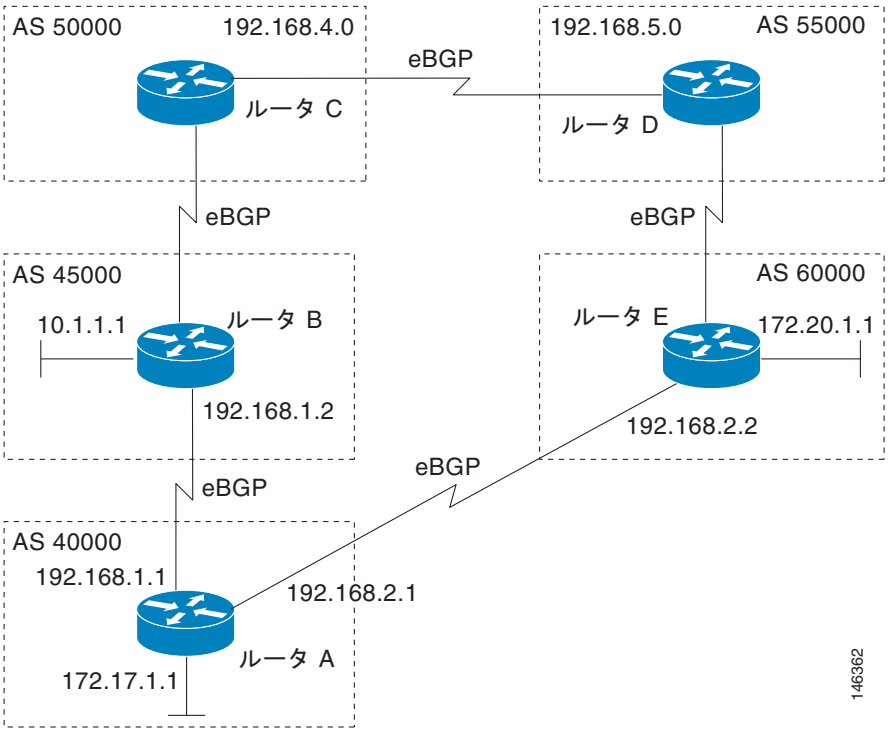
AS-path アトリビュートの変更によるインバウンド パス選択の変化

AS-path アトリビュートを変更して 172.17.1.0 ネットワークへ向かうトラフィックのインバウンド パス選択を変化させるには、次の作業を実行します。設定は、図 3 のルータ A で実行されます。asplain 形式の 4 バイト自律システム番号を使用した設定例については、「4 バイト自律システム番号を使用した AS-path アトリビュートの変更によるインバウンド パス選択の変更：例」（P.68）を参照してください。

AS-path アトリビュートの変更は、別の自律システムのパス選択を変化させるために BGP で使用可能な方法の 1 つです。たとえば、図 3 において、ルータ A は自身のネットワーク 172.17.1.0 を、自律システム 45000 および自律システム 60000 にある BGP ピアにアドバタイズします。ルーティング情報が自律システム 50000 に伝播されるとき、自律システム 50000 内のルータは、2 つの異なるルートからのネットワーク 172.17.1.0 の到達可能性情報を持つことになります。1 番目のルートは、45000 と 40000 で構成される AS-path を備えた自律システム 45000 によるもので、2 番目のルートは、55000、60000、40000 の AS-path を備えた自律システム 55000 によるものです。他の BGP アトリビュートがすべて同じだとすれば、自律システム 50000 内のルータ C はネットワーク 172.17.1.0 へのトラフィックのルートとして、自律システム 45000 を通るルートを選択します。通過した自律システムという点では最短ルートとなるからです。

自律システム 40000 は、自律システム 45000 を通して、自律システム 50000 から 172.17.1.0 ネットワークへのトラフィックすべてを受け取ることになります。しかし、自律システム 45000 と自律システム 40000 の間のリンクが非常に遅く輻輳している場合、**set as-path prepend** コマンドをルータ A で使用して、自律システム 45000 経由のルートが自律システム 60000 経由のパスよりも遠いように見せることで、172.17.1.0 ネットワークへのインバウンド パス選択を変化させることができます。図 3 のルータ A の設定は、アウトバウンド BGP アップデートをルータ B に適用することで完了します。**set as-path prepend** コマンドの使用により、ルータ A からルータ B へのアウトバウンド BGP アップデートはすべて、ローカル自律システム番号 40000 を 2 回追加するよう変更された AS-path アトリビュートを持つようになります。この設定の後、自律システム 50000 は 172.17.1.0 ネットワークについてのアップデートを、自律システム 45000 経由で受け取ることになります。新しい AS-path は 45000、40000、40000、40000 となり、これは自律システム 55000 からの AS-path（55000、60000、40000 で変更なし）よりも長くなります。自律システム 50000 内のネットワーキング デバイスは、172.17.1.0 ネットワーク内の宛先アドレスを持つパケットを転送するときに、自律システム 55000 経由のルートを優先するようになります。

図 3　AS-path アトリビュート変更のネットワーク トポロジ



手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
5. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
6. **network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*]
7. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
8. **neighbor** {*ip-address* | *peer-group-name*} **activate**
9. **exit-address-family**
10. **exit**
11. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
12. **set as-path** {**tag** | **prepend** *as-path-string*}
13. **end**
14. **show running-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	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 3	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 40000`	指定されたルーティング プロセスでルータ 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	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **例：** `Router(config-router)# neighbor 192.168.1.2 remote-as 45000`	指定された自律システム内のネイバーの IP アドレスまたはピア グループ名を、ローカル ルータの IPv4 マルチプロトコル BGP ネイバー テーブルに追加します。 • この例では、192.168.1.2 のルータ B の BGP ピアが IPv4 マルチプロトコル BGP ネイバー テーブルに追加され、BGP アップデートを受け取ることになります。
ステップ 5	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **例：** `Router(config-router)# address-family ipv4 unicast`	IPv4 アドレス ファミリを指定し、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。 • **unicast** キーワードは、IPv4 ユニキャスト アドレス ファミリを指定します。デフォルトでは、**address-family ipv4** コマンドに **unicast** キーワードが指定されていない場合、ルータは IPv4 ユニキャスト アドレス ファミリのアドレス ファミリ コンフィギュレーション モードになります。 • **multicast** キーワードは、IPv4 マルチキャスト アドレス プレフィクスを指定します。 • **vrf** キーワードと *vrf-name* 引数は、後続の IPv4 アドレス ファミリ コンフィギュレーション モード コマンドに関連付ける VRF インスタンスの名前を指定します。
ステップ 6	**network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*] **例：** `Router(config-router-af)# network 172.17.1.0 mask 255.255.255.0`	ネットワークを、この自律システムに対してローカルに設定し、BGP ルーティング テーブルに追加します。 • 外部プロトコルの場合、**network** コマンドはアドバタイズされるネットワークを制御します。内部プロトコルは **network** コマンドを使用して、アップデートの送信先を判断します。
ステップ 7	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**} **例：** `Router(config-router-af)# neighbor 192.168.1.2 route-map PREPEND out`	受信または発信ルートにルート マップを適用します。 • この例では、PREPEND という名前のルート マップが、ルータ B へのアウトバウンド ルートに適用されています。

	コマンドまたはアクション	目的
ステップ 8	**neighbor** {*ip-address* | *peer-group-name*} **activate** **例：** `Router(config-router-af)# neighbor 192.168.1.2 activate`	ルータ B 上の 192.168.1.2 にある BGP ネイバーのため、アドレス ファミリ IPv4 ユニキャスト用アドレス交換をイネーブルにします。
ステップ 9	**exit-address-family** **例：** `Router(config-router-af)# exit`	アドレス ファミリ コンフィギュレーション モードを終了して、ルータ コンフィギュレーション モードを開始します。
ステップ 10	**exit** **例：** `Router(config-router)# exit`	ルータ コンフィギュレーション モードを終了して、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 11	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **例：** `Router(config)# route-map PREPEND permit 10`	ルート マップを設定し、ルート マップ コンフィギュレーション モードを開始します。 • この例では、PREPEND という名前のルート マップが作成され、後続の条件一致があれば
ステップ 12	**set as-path** {**tag** | **prepend** *as-path-string*} **例：** `Router(config-route-map)# set as-path prepend 40000 40000`	BGP ルートの自律システム パスを変更します。 • 任意の自律システム パス ストリングを BGP ルートに「プリペンド」するには、**prepend** キーワードを使用します。通常、ローカルの自律システム番号は複数回プリペンドされ、自律システム パスの長さは増加します。 • この例では、2 つの自律システム エントリがルータ B へのアウトバウンド ルートの自律システム パスに追加されます。
ステップ 13	**end** **例：** `Router(config-route-map)# end`	ルート マップ コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 14	**show running-config** **例：** `Router# show running-config`	実行中のコンフィギュレーション ファイルを表示します。

例

次の **show running-config** コマンドからの出力の一部は、この作業で行った設定を示します。

ルータ A

```
Router# show running-config
.
.
.
router bgp 40000
 neighbor 192.168.1.2 remote-as 45000
 !
 address-family ipv4
 neighbor 192.168.1.2 activate
 neighbor 192.168.1.2 route-map PREPEND out
```

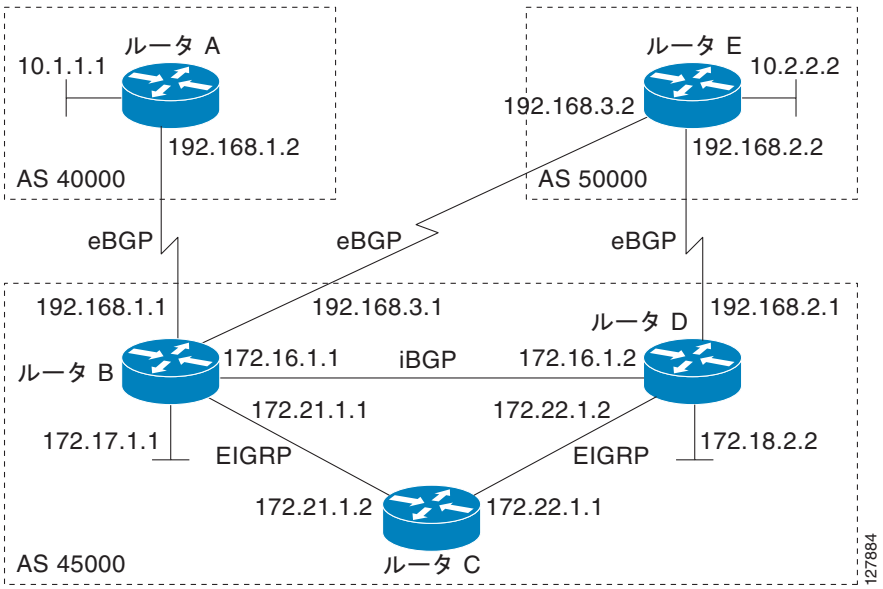
```
 no auto-summary
 no synchronization
 network 172.17.1.0 mask 255.255.255.0
 exit-address-family
!
route-map PREPEND permit 10
 set as-path prepend 40000 40000
.
.
.
```

MED アトリビュートの設定によるインバウンド パス選択の変化

MED アトリビュートの設定は、別の自律システムへのパス選択を変化させるために BGP で使用可能な方法の 1 つです。MED アトリビュートは、（外部ピアに）自律システムへの優先パスを示します。自律システムへのエントリ ポイントが複数ある場合、MED を使って別の自律システムに特定のエントリ ポイントを選択するようはたらきかけることができます。低い値の MED メトリックの方が高い値の MED メトリックより優先されるソフトウェアでは、ルート マップを使用してメトリックが割り当てられます。

MED メトリック アトリビュートの設定によってインバウンド パス選択を変化させるには、次の作業を行います。図 4 では、ルータ B とルータ D で設定を実行します。ルータ B はネットワーク 172.16.1.0 を自身の BGP ピアにアドバタイズし、ルータ E は自律システム 50000 にあります。シンプルなルート マップを使用して、ルータ B はアウトバウンド アップデートの MED メトリックを 50 に設定します。この作業がルータ D でも繰り返されますが、MED メトリックは 120 に設定されます。ルータ E がルータ B とルータ D の両方からアップデートを受け取ったとき、MED メトリックは BGP ルーティング テーブルに保存されます。ネットワーク 172.16.1.0 へパケットを転送する前に、ルータ E は同じ自律システム内の複数のピアから受信したアトリビュートを比較します（ルータ B とルータ D はどちらも自律システム 45000 にあります）。ルータ B の MED メトリックはルータ D の MED より小さいため、ルータ E はパケットをルータ B 経由で転送します。

図 4　　MED アトリビュート設定のネットワーク トポロジ



別の自律システムのピアからの MED アトリビュートを比較するには、**bgp always-compare-med** コマンドを使用します。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
5. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
6. **network** *network-number* **[***mask* **network-mask***] [*route-map route-map-name]
7. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
8. **exit**
9. **exit**
10. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
11. **set metric** *value*
12. **end**
13. ステップ 1 から ステップ 12 をルータ D で繰り返します。
14. **show ip bgp** [*network*] [*network-mask*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	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 3	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 45000`	指定されたルーティング プロセスでルータ 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	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **例：** `Router(config-router)# neighbor 192.168.3.2 remote-as 50000`	指定された自律システム内のネイバーの IP アドレスまたはピア グループ名を、ローカル ルータの IPv4 マルチプロトコル BGP ネイバー テーブルに追加します。

	コマンドまたはアクション	目的
ステップ 5	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **例：** `Router(config-router)# address-family ipv4 unicast`	IPv4 アドレス ファミリを指定し、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。 • **unicast** キーワードは、IPv4 ユニキャスト アドレス ファミリを指定します。デフォルトでは、**address-family ipv4** コマンドに **unicast** キーワードが指定されていない場合、ルータは IPv4 ユニキャスト アドレス ファミリのアドレス ファミリ コンフィギュレーション モードになります。 • **multicast** キーワードは、IPv4 マルチキャスト アドレス プレフィクスを指定します。 • **vrf** キーワードと *vrf-name* 引数は、後続の IPv4 アドレス ファミリ コンフィギュレーション モード コマンドに関連付ける VRF インスタンスの名前を指定します。
ステップ 6	**network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*] **例：** `Router(config-router-af)# network 172.16.1.0 mask 255.255.255.0`	ネットワークを、この自律システムに対してローカルに設定し、BGP ルーティング テーブルに追加します。 • 外部プロトコルの場合、**network** コマンドはアドバタイズされるネットワークを制御します。内部プロトコルは **network** コマンドを使用して、アップデートの送信先を判断します。
ステップ 7	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**} **例：** `Router(config-router-af)# neighbor 192.168.3.2 route-map MED out`	受信または発信ルートにルート マップを適用します。 • この例では、MED という名前のルート マップが、ルータ E にある BGP ピアへのアウトバウンド ルートに適用されます。
ステップ 8	**exit** **例：** `Router(config-router-af)# exit`	アドレス ファミリ コンフィギュレーション モードを終了して、ルータ コンフィギュレーション モードを開始します。
ステップ 9	**exit** **例：** `Router(config-router)# exit`	ルータ コンフィギュレーション モードを終了して、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 10	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **例：** `Router(config)# route-map MED permit 10`	ルート マップを設定し、ルート マップ コンフィギュレーション モードを開始します。 • この例では、MED という名前のルート マップが作成されます。
ステップ 11	**set metric** *value* **例：** `Router(config-route-map)# set metric 50`	MED メトリックの値を設定します。
ステップ 12	**end** **例：** `Router(config-route-map)# end`	ルート マップ コンフィギュレーション モードを終了して、特権 EXEC 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 13	ステップ 1 から ステップ 12 をルータ D で繰り返します。	—
ステップ 14	**show ip bgp** [*network*] [*network-mask*] **例：** `Router# show ip bgp 172.17.1.0 255.255.255.0`	（任意）BGP ルーティング テーブル内のエントリを表示します。 • 図 4 で、ルータ B とルータ D の両方が MED アトリビュートを設定しているとき、このコマンドをルータ E で実行します。 • この例では、この作業に適用される構文だけが使用されます。詳細については、『*Cisco IOS IP Routing: BGP Command Reference*』を参照してください。

例

次の出力は、この作業が図 4 のルータ B とルータ D の両方で実行された後の、ルータ E からのものです。ネットワーク 172.16.1.0 への 2 つのルートのメトリック（MED）値に注目してください。ルータ D にあるピア 192.168.2.1 は、ネットワーク 172.16.1.0 へのパスとしてメトリック 120 を持ち、ルータ B の 192.168.3.1 はメトリック 50 になっています。ルータ B のピア 192.168.3.1 のエントリでは、ルータ E がネットワーク 172.16.1.0 を宛先とするパケットを送るのに、MED メトリックが低いことからルータ B 経由での送信を選ぶことを示すため、エントリの最後に best という語が付いています。

```
Router# show ip bgp 172.16.1.0

BGP routing table entry for 172.16.1.0/24, version 10
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to update-groups:
     1
  45000
    192.168.2.1 from 192.168.2.1 (192.168.2.1)
      Origin IGP, metric 120, localpref 100, valid, external
  45000
    192.168.3.1 from 192.168.3.1 (172.17.1.99)
      Origin IGP, metric 50, localpref 100, valid, external, best
```

アウトバウンド パス選択への影響

BGP を使用して、ローカルの自律システムからのアウトバウンド トラフィックに対するパス選択を変化させることができます。このセクションでは、アウトバウンド パスの選択を変化させるのに BGP が使用可能な 2 つの方法を説明します。

- Local_Pref アトリビュートの使用
- BGP アウトバウンド ルート フィルタ（ORF）機能の使用

アウトバウンド パス選択を変化させる場合、次の作業のいずれかを実行します。

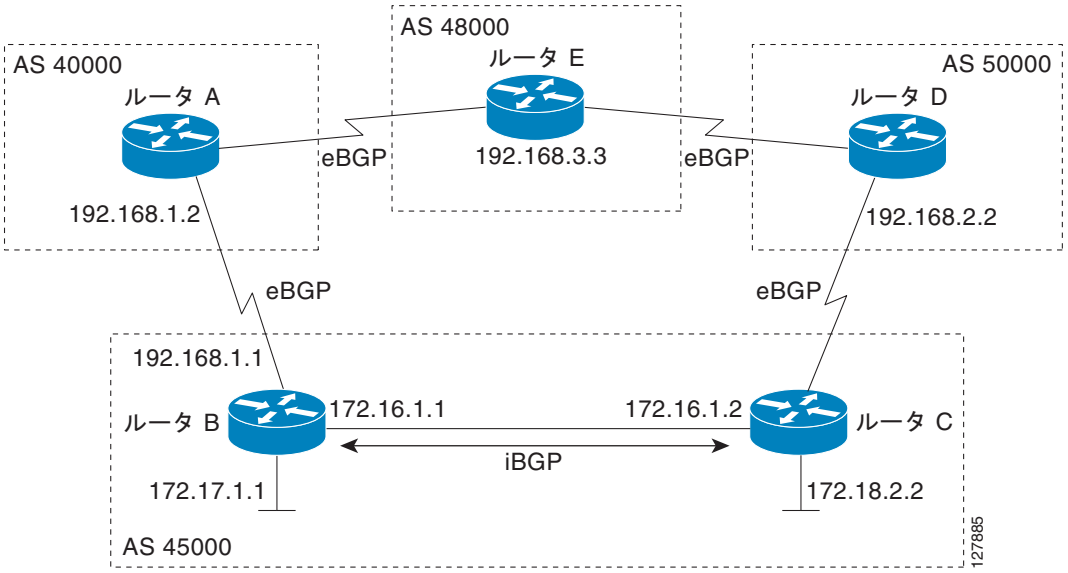
- 「Local_Pref アトリビュートを使用したアウトバウンド パス選択の変更」（P.21）
- 「アウトバウンド BGP ルート プレフィクスのフィルタリング」（P.23）

Local_Pref アトリビュートを使用したアウトバウンド パス選択の変更

アウトバウンド パス選択を変化させる方法の 1 つが、BGP Local-Pref アトリビュートの使用です。アウトバウンド パス選択を変化させるには、ローカル プリファレンス アトリビュートを使用してこの作業を実行します。同じ宛先への複数のパスがある場合、ローカル プリファレンス アトリビュートの値が最大であるものが、優先パスになります。

この作業で使用するネットワーク トポロジについては、図 5 を参照してください。ルータ B とルータ C の両方が設定されています。自律システム 45000 は、ネットワーク 192.168.3.0 のアップデートを自律システム 40000 と自律システム 50000 から受信します。ルータ B は、自律システム 40000 へのアップデートすべてに対し、ローカル プリファレンスの値を 150 にするよう設定されています。ルータ C は、自律システム 50000 へのアップデートすべてに対し、ローカル プリファレンスの値を 200 にするよう設定されています。設定の後、ローカル プリファレンス情報が自律システム 45000 との間で交換されます。ルータ B とルータ C は、ネットワーク 192.168.3.0 のアップデートで自律システム 50000 からの方が高いプリファレンス値を持つことがわかるため、自律システム 45000 内で宛先ネットワークが 192.168.3.0 のトラフィックは、すべてルータ C 経由で送られます。

図 5　　アウトバウンド パス選択のネットワーク トポロジ



手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
5. **bgp default local-preference** *value*
6. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
7. **network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*]
8. **neighbor** {*ip-address* | *peer-group-name*} **activate**
9. **end**

10. ステップ 1 から ステップ 9 をルータ C で繰り返します。ただし、ピアの IP アドレスと自律システム番号は変更し、ローカル プリファレンスの値を 200 に設定します。
11. **show ip bgp** [*network*] [*network-mask*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	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 3	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 45000`	指定されたルーティング プロセスでルータ 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	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **例：** `Router(config-router)# neighbor 192.168.1.2 remote-as 40000`	指定された自律システム内のネイバーの IP アドレスまたはピア グループ名を、ローカル ルータの IPv4 マルチプロトコル BGP ネイバー テーブルに追加します。
ステップ 5	**bgp default local-preference** *value* **例：** `Router(config-router)# bgp default local-preference 150`	ローカル プリファレンスのデフォルト値を変更します。 • この例では、自律システム 40000 から自律システム 45000 へのアップデートすべてのローカル プリファレンスが 150 に変更されます。 • ローカル プリファレンスの値は、デフォルトでは 100 です。
ステップ 6	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **例：** `Router(config-router)# address-family ipv4 unicast`	IPv4 アドレス ファミリを指定し、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。 • **unicast** キーワードは、IPv4 ユニキャスト アドレス ファミリを指定します。デフォルトでは、**address-family ipv4** コマンドに **unicast** キーワードが指定されていない場合、ルータは IPv4 ユニキャスト アドレス ファミリのアドレス ファミリ コンフィギュレーション モードになります。 • **multicast** キーワードは、IPv4 マルチキャスト アドレス プレフィクスを指定します。 • **vrf** キーワードと *vrf-name* 引数は、後続の IPv4 アドレス ファミリ コンフィギュレーション モード コマンドに関連付ける VRF インスタンスの名前を指定します。

	コマンドまたはアクション	目的
ステップ 7	**network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*] **例：** `Router(config-router-af)# network 172.17.1.0 mask 255.255.255.0`	ネットワークを、この自律システムに対してローカルに設定し、BGP ルーティング テーブルに追加します。 • 外部プロトコルの場合、**network** コマンドはアドバタイズされるネットワークを制御します。内部プロトコルは **network** コマンドを使用して、アップデートの送信先を判断します。
ステップ 8	**neighbor** {*ip-address* | *peer-group-name*} **activate** **例：** `Router(config-router-af)# neighbor 192.168.1.2 activate`	指定された自律システム内のネイバーの IP アドレスまたはピア グループ名を、ローカル ルータの IPv4 マルチプロトコル BGP ネイバー テーブルに追加します。
ステップ 9	**end** **例：** `Router(config-router-af)# end`	ルート マップ コンフィギュレーション モードを終了して、特権 EXEC モードを開始します。
ステップ 10	ステップ 1 から ステップ 9 をルータ C で繰り返します。ただし、ピアの IP アドレスと自律システム番号は変更し、ローカル プリファレンスの値を 200 に設定します。	—
ステップ 11	**show ip bgp** [*network*] [*network-mask*] **例：** `Router# show ip bgp 192.168.3.0 255.255.255.0`	BGP ルーティング テーブル内のエントリを表示します。 • ルータ B とルータ C の両方でこのコマンドを入力し、Local_Pref の値を記録します。最大のプリファレンス値を持つルートが、ネットワーク 192.168.3.0 への優先ルートになります。 **（注）** この例では、この作業に適用される構文だけが使用されます。詳細については、『*Cisco IOS IP Routing: BGP Command Reference*』を参照してください。

アウトバウンド BGP ルート プレフィクスのフィルタリング

BGP プレフィクスベース アウトバウンド ルート フィルタリングを使用してアウトバウンド パス選択を変化させるには、次の作業を行います。

BGP プレフィクスベース アウトバウンド ルート フィルタリング

BGP プレフィクスベース アウトバウンド ルート フィルタリングは、BGP ORF 送受信機能を使用して、BGP ピアの間で送られる BGP アップデートの数を最小化します。BGP ORF を設定すると、不要なルーティング アップデートをソースでフィルタリングできるため、ルーティング アップデートの生成や処理に必要なシステム リソースの量を減らす助けになります。たとえば、BGP ORF を使用して、サービス プロバイダー ネットワークからのルート全体を受け付けるのではないルータで、ルータに要求される処理の量を減らすことができます。

BGP プレフィクスベース アウトバウンド ルート フィルタリングはピア ルータへの ORF 機能のアドバタイズメントを通してイネーブルになります。ORF 機能のアドバタイズメントは、ある BGP ピアがネイバーからのプレフィクス リストを受け付け、そのプレフィクス リストをローカルで設定された ORF

に適用する（存在する場合）ことを示します。この機能がイネーブルの場合、BGP スピーカーはインバウンド プレフィクス リスト フィルタをアウトバウンド フィルタとしてリモート ピアにインストールでき、これにより不要なルーティング アップデートを減少させることができます。

BGP プレフィクスベース アウトバウンド ルート フィルタリングは、ORF 送受信機能を使用して設定できます。ローカル ピアは ORF 機能を send モードでアドバタイズします。リモート ピアは ORF 機能を受信モードで受信し、そのフィルタをアウトバウンド ポリシーとして適用します。ローカルとリモートのピアは、それぞれのルータの ORF を維持するために、アップデートを交換します。アップデートは、アドバタイズされた ORF プレフィクス リスト機能に依存するアドレス ファミリによってピア ルータの間で交換されます。リモート ピアは、**clear ip bgp in prefix-filter** コマンドで要求されたルート リフレッシュの後か、immediate ステータスの ORF プレフィクス リストが処理された後に、ローカル ピアにアップデートを送信し始めます。BGP ピアは、ローカル ピアがインバウンド プレフィクス リストをリモート ピアにプッシュした後、インバウンド プレフィクス リストを受信したアップデートに適用し続けます。

前提条件

プレフィクスベース ORF BGP の配信を受信できるようになる前に、ピアリング セッションが確立され、BGP ORF 機能が各参加ルータでイネーブルになっている必要があります。

制約事項

- BGP プレフィクスベース アウトバウンド ルート フィルタリングはマルチキャストをサポートしていません。
- アウトバウンド ルート フィルタリングに使用する IP アドレスは IP プレフィクス リストで定義されている必要があります。BGP 配信リストおよび IP アクセス リストはサポートしていません。
- アウトバウンド ルート フィルタリングはアドレス ファミリ単位ベースだけで設定され、ジェネラル セッションや BGP ルーティング プロセス下では設定できません。
- アウトバウンド ルート フィルタリングは、外部ピアリング セッションだけに設定できます。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**ge** *ge-value*] [**le** *le-value*]
4. **router bgp** *autonomous-system-number*
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
6. **neighbor** *ip-address* **ebgp-multihop** [*hop-count*]
7. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
8. **neighbor** *ip-address* **capability orf prefix-list** [**send** | **receive** | **both**]
9. **neighbor** {*ip-address* | *peer-group-name*} **prefix-list** *prefix-list-name* {**in** | **out**}
10. **end**
11. **clear ip bgp** {*ip-address* | *****} **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	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	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 3	**ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network*/*length* | **permit** *network*/*length*} [**ge** *ge-value*] [**le** *le-value*] **例：** `Router(config)# ip prefix-list FILTER seq 10 permit 192.168.1.0/24`	プレフィクスベース アウトバウンド ルート フィルタリング用にプレフィクス リストを作成します。 • アウトバウンド ルート フィルタリングは、プレフィクス長のマッチング、ワイルドカードベースのプレフィクス マッチング、アドレスファミリ単位ベースのアドレス プレフィクス マッチングをサポートします。 • アウトバウンド ルート フィルタを定義するためにプレフィクス リストが作成されます。アウトバウンド ルート フィルタリング機能が send モードまたは both モードでアドバタイズされるよう設定されているときは、フィルタの作成が必要です。ピアが receive モードだけでアドバタイズされるよう設定されている場合は不要です。 • この例では、アウトバウンド ルート フィルタリングのためにサブネット 192.168.1.0/24 を定義する、FILTER という名前のプレフィクス リストを作成します。
ステップ 4	**router bgp** autonomous-system-*number* **例：** `Router(config)# router bgp 100`	ルータ コンフィギュレーション モードを開始して、BGP ルーティング プロセスを作成します。
ステップ 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **例：** `Router(config-router)# neighbor 10.1.1.1 remote-as 200`	指定されたネイバーまたはピア グループとのピアリングを確立します。ORF 機能が交換できるようになるには、BGP ピアリングが確立されている必要があります。 • この例では、ネイバー 10.1.1.1 とのピアリングを確立します。
ステップ 6	**neighbor** *ip-address* **ebgp-multihop** [*hop-count*] **例：** `Router(config-router)# neighbor 10.1.1.1 ebgp-multihop`	直接接続されていないネットワークに存在する外部ピアへの BGP 接続を受け入れるか、または開始します。

	コマンドまたはアクション	目的
ステップ 7	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **例：** `Router(config-router)# address-family ipv4 unicast`	IPv4 アドレス ファミリを指定し、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。 • **unicast** キーワードは、IPv4 ユニキャスト アドレス ファミリを指定します。デフォルトでは、**address-family ipv4** コマンドに **unicast** キーワードが指定されていない場合、ルータは IPv4 ユニキャスト アドレス ファミリのアドレス ファミリ コンフィギュレーション モードになります。 • **multicast** キーワードは、IPv4 マルチキャスト アドレス プレフィクスを指定します。 • **vrf** キーワードと *vrf-name* 引数は、後続の IPv4 アドレス ファミリ コンフィギュレーション モード コマンドに関連付ける VRF インスタンスの名前を指定します。 **（注）** アウトバウンド ルート フィルタリングは、アドレス ファミリ単位ベースで設定されます。
ステップ 8	**neighbor** *ip-address* **capability orf prefix-list** [**send** | **receive** | **both**] **例：** `Router(config-router-af)# neighbor 10.1.1.1 capability orf prefix-list both`	ローカル ルータで ORF 機能をイネーブルにし、*ip-address* 引数で指定された BGP ピアへの ORF 機能アドバタイズメントをイネーブルにします。 • **send** キーワードは、ORF 送信機能をアドバタイズするようルータを設定します。 • **receive** キーワードは、ORF 受信機能をアドバタイズするようルータを設定します。 • **both** キーワードは、送受信機能をアドバタイズするようルータを設定します。 • アウトバウンド ルート フィルタリングがイネーブルにされる前に、リモート ピアで送信と受信いずれかの ORF 機能が設定されている必要があります。 • この例では、ネイバー 10.1.1.1 への送信と受信機能をアドバタイズするようルータを設定します。
ステップ 9	**neighbor** {*ip-address* | *peer-group-name*} **prefix-list** *prefix-list-name* {**in** | **out**} **例：** `Router(config-router-af)# neighbor 10.1.1.1 prefix-list FILTER in`	インバウンド プレフィクスリスト フィルタを適用し、BGP ネイバー情報を配信しないようにします。 • この例では、FILTER という名前のプレフィクス リストがネイバー 10.1.1.1 からの受信アドバタイズメントに適用され、サブネット 192.168.1.0/24 を配信しないようにしています。

	コマンドまたはアクション	目的
ステップ 10	**end** **例：** `Router(config-router-af)# end`	アドレス ファミリ コンフィギュレーション モードを終了して、特権 EXEC モードを開始します。
ステップ 11	**clear ip bgp** {*ip-address* | ***}** **in prefix-filter** **例：** `Router# clear ip bgp 10.1.1.1 in prefix-filter`	BGP アウトバウンド ルート フィルタをクリアし、インバウンド ソフト リセットを開始します。 • 単一のネイバーまたはすべてのネイバーを指定できます。 **（注）** この機能が正しく動作するために、**clear ip bgp** コマンドでインバウンド ソフト リセットを開始する必要があります。

ISP との BGP ピアリングの設定

BGP はドメイン間ルーティング プロトコルとして開発されたもので、ISP への接続は BGP の主要機能の 1 つです。使用するネットワークのサイズやビジネスの目的により、ISP への接続にはさまざまな方法があります。1 つ以上の ISP へのマルチホーミングは、ISP への外部リンクの 1 つに障害が発生した場合のための冗長性を提供します。このセクションでは、プロバイダーへのマルチホーミングの手法を使用した接続に応用可能なオプション作業の一部を紹介します。規模の小さい企業では 1 つの ISP との接続だけを使用することがありますが、ISP へのバックアップ ルートが必要になります。規模の大きい企業では、2 つの ISP へのアクセスを確保して 1 つをバックアップとして使用したり、中継用自律システムを設定する必要が生じたりすることがあります。

1 つ以上の ISP へ接続するには、次のオプション作業のいずれかを行います。

- 「2 つの ISP とのマルチホーミングの設定」（P.27）
- 「単一 ISP とのマルチホーミング」（P.31）
- 「マルチホーミングのフル インターネット ルーティング テーブル受信設定」（P.37）

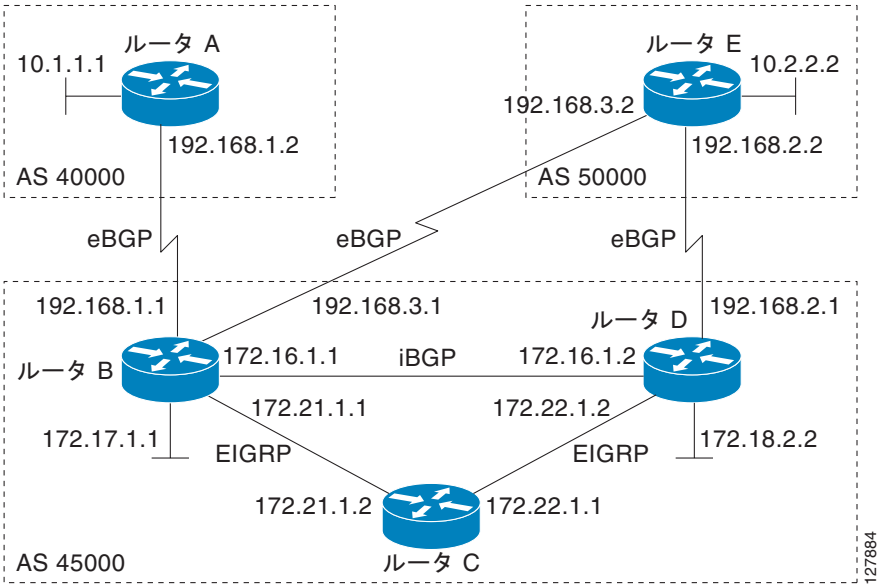
2 つの ISP とのマルチホーミングの設定

ネットワークを 2 つの ISP にアクセスさせるには、次の作業を行います。1 番目の ISP を優先ルート、2 番目の ISP はバックアップ ルートとします。図 6 において、自律システム 45000 のルータ B は、自律システム 40000 と自律システム 50000 の 2 つの ISP に BGP ピアを持っています。この作業を行うことで、ルータ B は自律システム 40000 内にあるルータ A の BGP ピアへのルートを優先するよう設定されます。

このネイバーから学習したすべてのルートに、重みが割り当てられます。特定のネットワークへのルートが複数ある場合、重みが最大のルートが優先ルートとして選ばれます。

（注） **set weight** ルート マップ コンフィギュレーション コマンドで割り当てられた重みは、**neighbor weight** コマンドで割り当てられた重みを上書きします。

図 6　　2 つの ISP とのマルチホーミング



手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
5. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
6. **network** *network-number* [**mask** *network-mask*]
7. **neighbor** {*ip-address* | *peer-group-name*} **weight** *number*
8. **exit**
9. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
10. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
11. **neighbor** {*ip-address* | *peer-group-name*} **weight** *number*
12. **end**
13. **clear ip bgp** {* | *ip-address* | *peer-group-name*} [**soft** [**in** | **out**]]
14. **show ip bgp** [*network-address*] [*network-mask*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	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 3	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 45000`	ルータ コンフィギュレーション モードを開始して、BGP ルーティング プロセスを作成します。
ステップ 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **例：** `Router(config-router)# neighbor 192.168.1.2 remote-as 40000`	指定された自律システム内のネイバーの IP アドレスまたはピア グループ名を、ローカル ルータの IPv4 マルチプロトコル BGP ネイバー テーブルに追加します。
ステップ 5	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **例：** `Router(config-router)# address-family ipv4 unicast`	IPv4 アドレス ファミリを指定し、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。 • **unicast** キーワードは、IPv4 ユニキャスト アドレス ファミリを指定します。デフォルトでは、**address-family ipv4** コマンドに **unicast** キーワードが指定されていない場合、ルータは IPv4 ユニキャスト アドレス ファミリのコンフィギュレーション モードになります。 • **multicast** キーワードは、IPv4 マルチキャスト アドレス プレフィクスを指定します。 • **vrf** キーワードと *vrf-name* 引数は、後続の IPv4 アドレス ファミリ コンフィギュレーション モード コマンドに関連付ける VRF インスタンスの名前を指定します。
ステップ 6	**network** *network-number* [**mask** *network-mask*] **例：** `Router(config-router-af)# network 172.17.1.0 mask 255.255.255.0`	ネットワークを、この自律システムに対してローカルに設定し、BGP ルーティング テーブルに追加します。 • 外部プロトコルの場合、**network** コマンドはアドバタイズされるネットワークを制御します。内部プロトコルは **network** コマンドを使用して、アップデートの送信先を判断します。
ステップ 7	**neighbor** {*ip-address* | *peer-group-name*} **weight** *number* **例：** `Router(config-router-af)# neighbor 192.168.1.2 weight 150`	BGP ピア接続に重みを割り当てます。 • この例では、ルートの weight アトリビュートが BGP ピア 192.168.1.2 から受け取る値は 150 に設定されています。
ステップ 8	**exit** **例：** `Router(config-router-af)# exit`	アドレス ファミリ コンフィギュレーション モードを終了して、ルータ 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 9	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **例：** `Router(config-router)# neighbor 192.168.3.2 remote-as 50000`	指定された自律システム内のネイバーの IP アドレスまたはピア グループ名を、ローカル ルータの IPv4 マルチプロトコル BGP ネイバー テーブルに追加します。
ステップ 10	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **例：** `Router(config-router)# address-family ipv4 unicast`	IPv4 アドレス ファミリを指定し、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。 • **unicast** キーワードは、IPv4 ユニキャスト アドレス ファミリを指定します。デフォルトでは、**address-family ipv4** コマンドに **unicast** キーワードが指定されていない場合、ルータは IPv4 ユニキャスト アドレス ファミリのコンフィギュレーション モードになります。 • **multicast** キーワードは、IPv4 マルチキャスト アドレス プレフィクスを指定します。 **vrf** キーワードと *vrf-name* 引数は、後続の IPv4 アドレス ファミリ コンフィギュレーション モード コマンドに関連付ける VRF インスタンスの名前を指定します。
ステップ 11	**neighbor** {*ip-address* | *peer-group-name*} **weight** *number* **例：** `Router(config-router-af)# neighbor 192.168.3.2 weight 100`	BGP ピア接続に重みを割り当てます。 • この例では、ルートの weight アトリビュートが BGP ピア 192.168.3.2 から受け取る値は 100 に設定されています。
ステップ 12	**end** **例：** `Router(config-router-af)# end`	アドレス ファミリ コンフィギュレーション モードを終了して、特権 EXEC モードを開始します。
ステップ 13	**clear ip bgp** {***** | *ip-address* | *peer-group-name*} [**soft** [**in** | **out**]] **例：** `Router# clear ip bgp *`	（任意）BGP アウトバウンドルート フィルタをクリアし、アウトバウンド ソフト リセットを開始します。単一のネイバーまたはすべてのネイバーを指定できます。
ステップ 14	**show ip bgp** [*network*] [*network-mask*] **例：** `Router# show ip bgp`	BGP ルーティング テーブル内のエントリを表示します。 • BGP ピアへのそれぞれのルートの weight アトリビュートを見るには、このコマンドをルータ B に入力します。weight アトリビュートが最大のルートが、ネットワーク 172.17.1.0 への優先ルートになります。 **（注）** この例では、この作業に適用される構文だけが使用されます。詳細については、『*Cisco IOS IP Routing: BGP Command Reference*』を参照してください。

例

次の例は、ルートに weight アトリビュートが割り当てられた、ルータ B の BGP ルーティング テーブルを示しています。192.168.3.2 を通るルート（図 6 のルータ E）は最大の weight アトリビュートを持っているため、ネットワーク 172.17.1.0 への優先ルートとなります。

```
BGP table version is 8, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0/24      192.168.1.2              0           100 40000 i
*> 10.2.2.0/24      192.168.3.2              0           150 50000 i
*> 172.17.1.0/24    0.0.0.0                  0         32768 i
```

単一 ISP とのマルチホーミング

ネットワークを単一の ISP との 2 つの接続のうち 1 つにアクセスさせるには、次の作業を行います。1 番目の接続を優先ルート、2 番目の接続をバックアップ ルートとします。図 6 において、自律システム 50000 のルータ E には、単一自律システムである自律システム 45000 内に 2 つの BGP ピアがあります。この作業を行うことで、自律システム 50000 は自律システム 45000 からどのルートも学習せず、BGP を使用して自身のルートを送信するようになります。この作業は、図 6 のルータ E で設定し、単一 ISP へのマルチホーミングに関する 3 つの機能をカバーします。

- アウトバウンド トラフィック：ルータ E は、ルータ B をプライマリ リンク、ルータ D をバックアップ リンクとして、デフォルト ルートとトラフィックを自律システム 45000 に転送します。ルータ B とルータ D にはスタティック ルートが設定され、ルータ B へのリンクのディスタンスの方が低く設定されています。
- インバウンド トラフィック：自律システム 45000 からのインバウンド トラフィックは、リンクに障害が生じたためにトラフィックをルータ D からバックアップ ルートで送る場合を除き、ルータ B から送信されるよう設定されます。この状態にするため、MED メトリックを使用したアウトバウンド フィルタが設定されています。
- 中継トラフィックの防止：自律システム 50000 のルータ E には、受信 BGP ルーティング アップデートをすべてブロックし、自律システム 50000 が自律システム 45000 の ISP からの中継トラフィックを受信しないよう、ルート マップが設定されます。

MED アトリビュート

MED アトリビュートの設定は、別の自律システムへのパス選択を変化させるために BGP が使用できるもう 1 つの方法です。MED アトリビュートは、（外部ピアに）自律システムへの優先パスを示します。自律システムへのエントリ ポイントが複数ある場合、MED を使って別の自律システムに特定のエントリ ポイントを選択するようはたらきかけることができます。低い値の MED メトリックの方が高い値の MED メトリックより優先されるソフトウェアでは、ルート マップを使用してメトリックが割り当てられます。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
5. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
6. **network** *network-number* **[***mask* **network-mask***]* *[*route-map route-map-name]
7. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
8. ステップ 7 で指定されたネイバーに別のルート マップを適用するには、ステップ 7 を繰り返します。

9. **exit**
10. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
11. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
12. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
13. ステップ 10 で指定されたネイバーに別のルート マップを適用するには、ステップ 10 を繰り返します。
14. **exit**
15. **exit**
16. **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent** | **track** *number*] [**tag** *tag*]
17. 別のルート マップを設定するには、ステップ 14 を繰り返します。
18. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
19. **set metric** *value*
20. **exit**
21. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
22. **set metric** *value*
23. **exit**
24. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
25. **end**
26. **show ip route** [*ip-address*] [*mask*] [**longer-prefixes**]
27. **show ip bgp** [*network*] [*network-mask*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	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 3	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 45000`	指定されたルーティング プロセスでルータ 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	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **例：** `Router(config-router)# neighbor 192.168.2.1 remote-as 45000`	指定された自律システム内のネイバーの IP アドレスまたはピア グループ名を、ローカル ルータの IPv4 マルチプロトコル BGP ネイバー テーブルに追加します。 • この例では、ルータ D にある BGP ピアが BGP ルーティング テーブルに追加されます。

	コマンドまたはアクション	目的
ステップ 5	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **例：** `Router(config-router)# address-family ipv4 unicast`	IPv4 アドレス ファミリを指定し、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。 • **unicast** キーワードは、IPv4 ユニキャスト アドレス ファミリを指定します。デフォルトでは、**address-family ipv4** コマンドに **unicast** キーワードが指定されていない場合、ルータは IPv4 ユニキャスト アドレス ファミリのアドレス ファミリ コンフィギュレーション モードになります。 • **multicast** キーワードは、IPv4 マルチキャスト アドレス プレフィクスを指定します。 • **vrf** キーワードと *vrf-name* 引数は、後続の IPv4 アドレス ファミリ コンフィギュレーション モード コマンドに関連付ける VRF インスタンスの名前を指定します。
ステップ 6	**network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*] **例：** `Router(config-router-af)# network 10.2.2.0 mask 255.255.255.0`	ネットワークを、この自律システムに対してローカルに設定し、BGP ルーティング テーブルに追加します。 • 外部プロトコルの場合、**network** コマンドはアドバタイズされるネットワークを制御します。内部プロトコルは **network** コマンドを使用して、アップデートの送信先を判断します。

ステップ 7	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
	
	例：
	`Router(config-router-af)# neighbor 192.168.2.1 route-map BLOCK in`
	
	例：
	`Router(config-router-af)# neighbor 192.168.2.1 route-map SETMETRIC1 out`
		受信または発信ルートにルート マップを適用します。
		• 1 番目の例では、BLOCK という名前のルート マップがルータ E のインバウンド ルートに適用されます。
		• 2 番目の例では、SETMETRIC1 という名前のルート マップがルータ D のアウトバウンド ルートに適用されます。
		（注） 作業例ではこれらの文の双方を設定する必要があるため、2 つの例を示しています。

ステップ 8	ステップ 7 で指定されたネイバーに別のルート マップを適用するには、ステップ 7 を繰り返します。	—
ステップ 9	**exit** **例：** `Router(config-router-af)# exit`	アドレス ファミリ コンフィギュレーション モードを終了して、ルータ コンフィギュレーション モードを開始します。
ステップ 10	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **例：** `Router(config-router)# neighbor 192.168.3.1 remote-as 45000`	指定された自律システム内のネイバーの IP アドレスまたはピア グループ名を、ローカル ルータの IPv4 マルチプロトコル BGP ネイバー テーブルに追加します。 • この例では、ルータ D にある BGP ピアが BGP ルーティング テーブルに追加されます。

	コマンドまたはアクション	目的
ステップ 11	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **例：** `Router(config-router)# address-family ipv4 unicast`	IPv4 アドレス ファミリを指定し、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。 • **unicast** キーワードは、IPv4 ユニキャスト アドレス ファミリを指定します。デフォルトでは、**address-family ipv4** コマンドに **unicast** キーワードが指定されていない場合、ルータは IPv4 ユニキャスト アドレス ファミリのアドレス ファミリ コンフィギュレーション モードになります。 • **multicast** キーワードは、IPv4 マルチキャスト アドレス プレフィクスを指定します。 **vrf** キーワードと *vrf-name* 引数は、後続の IPv4 アドレス ファミリ コンフィギュレーション モード コマンドに関連付ける VRF インスタンスの名前を指定します。

ステップ 12	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
	
	例：
	`Router(config-router-af)# neighbor 192.168.3.1 route-map BLOCK in`
	
	および
	
	例：
	`Router(config-router-af)# neighbor 192.168.3.1 route-map SETMETRIC2 out`
		受信または発信ルートにルート マップを適用します。
		• 1 番目の例では、BLOCK という名前のルート マップがルータ E のインバウンド ルートに適用されます。
		• 2 番目の例では、SETMETRIC2 という名前のルート マップがルータ D のアウトバウンド ルートに適用されます。
		（注） 作業例ではこれらの文の双方を設定する必要があるため、2 つの例を示しています。

ステップ 13	ステップ 10 で指定されたネイバーに別のルート マップを適用するには、ステップ 10 を繰り返します。	—
ステップ 14	**exit** **例：** `Router(config-router-af)# exit`	アドレス ファミリ コンフィギュレーション モードを終了して、ルータ 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	**exit** **例：** `Router(config-router)# exit`	ルータ コンフィギュレーション モードを終了して、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。

	コマンドまたはアクション	目的

ステップ 16	**ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent** | **track** *number*] [**tag** *tag*]
	
	例：
	`Router(config)# ip route 0.0.0.0 0.0.0.0 192.168.2.1 50`
	
	例：
	`Router(config)# ip route 0.0.0.0 0.0.0.0 192.168.2.1 50`
	
	および
	
	例：
	`Router(config)# ip route 0.0.0.0 0.0.0.0 192.168.3.1 40`
		スタティック ルートを確立します。
		• 1 番目の例では、BGP ピア 192.168.2.1 へのスタティック ルートが確立され、管理ディスタンスとして 50 が設定されます。
		• 2 番目の例では、BGP ピア 192.168.3.1 へのスタティック ルートが確立され、管理ディスタンスとして 40 が設定されます。管理ディスタンスが小さいことで、ルータ B を経由するこのルートが優先ルートになります。
		（注） 作業例ではこれらの文の双方を設定する必要があるため、2 つの例を示しています。

ステップ 17	別のスタティック ルートを確立するには、ステップ 14 を繰り返します。	—
ステップ 18	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **例：** `Router(config)# route-map SETMETRIC1 permit 10`	ルート マップを設定し、ルート マップ コンフィギュレーション モードを開始します。 • この例では、SETMETRIC1 という名前のルート マップが作成されます。
ステップ 19	**set metric** *value* **例：** `Router(config-route-map)# set metric 100`	MED メトリックの値を設定します。
ステップ 20	**exit** **例：** `Router(config-route-map)# exit`	ルート マップ コンフィギュレーション モードを終了して、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 21	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **例：** `Router(config)# route-map SETMETRIC2 permit 10`	ルート マップを設定し、ルート マップ コンフィギュレーション モードを開始します。 • この例では、SETMETRIC2 という名前のルート マップが作成されます。
ステップ 22	**set metric** *value* **例：** `Router(config-route-map)# set metric 50`	MED メトリックの値を設定します。
ステップ 23	**exit** **例：** `Router(config-route-map)# exit`	ルート マップ コンフィギュレーション モードを終了して、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。

	コマンドまたはアクション	目的
ステップ 24	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **例：** `Router(config)# route-map BLOCK deny 10`	ルート マップを設定し、ルート マップ コンフィギュレーション モードを開始します。 • この例では、自律システム 45000 からの受信ルートをすべてブロックするために、BLOCK という名前のルート マップが作成されます。
ステップ 25	**end** **例：** `Router(config-route-map)# end`	ルート マップ コンフィギュレーション モードを終了して、特権 EXEC モードを開始します。
ステップ 26	**show ip route** [*ip-address*] [*mask*] [**longer-prefixes**] **例：** `Router# show ip route`	（任意）ルーティング テーブルからのルート情報を表示します。 • ルータ B とルータ D がルータ E から MED メトリックを含んだアップデート情報を受信した後に、このコマンドを 図 6 のルータ E で使用します。 • この例では、この作業に適用される構文だけが使用されます。詳細については、『*Cisco IOS IP Routing: BGP Command Reference*』を参照してください。
ステップ 27	**show ip bgp** [*network*] [*network-mask*] **例：** `Router# show ip bgp 172.17.1.0 255.255.255.0`	（任意）BGP ルーティング テーブル内のエントリを表示します。 • ルータ B とルータ D がルータ E から MED メトリックを含んだアップデート情報を受信した後に、このコマンドを 図 6 のルータ E で使用します。 • この例では、この作業に適用される構文だけが使用されます。詳細については、『*Cisco IOS IP Routing: BGP Command Reference*』を参照してください。

例

次の例は、この設定作業が完了し、ルータ B とルータ D が MED メトリックを含んだアップデート情報を受信した後に、ルータ E で **show ip route** コマンドを入力したときの出力を示します。ラスト リゾート ゲートウェイがルータ B へのルートである 192.168.3.1 に設定されていることに注意してください。

```
Router# show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 192.168.3.1 to network 0.0.0.0

     10.0.0.0/24 is subnetted, 1 subnets
C       10.2.2.0 is directly connected, Ethernet0/0
C    192.168.2.0/24 is directly connected, Serial3/0
C    192.168.3.0/24 is directly connected, Serial2/0
S*   0.0.0.0/0 [40/0] via 192.168.3.1
```

次の例は、この設定作業が完了し、ルータ B とルータ D がルーティング アップデートを受信した後に、ルータ E で **show ip bgp** コマンドを入力したときの出力を示します。ルート マップ BLOCK は自律システム 45000 から入ってくるルートをすべて拒否しているため、表示される唯一のネットワークはローカル ネットワークです。

```
Router# show ip bgp

BGP table version is 2, local router ID is 10.2.2.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.2.2.0/24      0.0.0.0                  0         32768 i
```

次の例は、ルータ E でこの設定作業が完了し、ルータ B がルーティング アップデートを受信した後に、ルータ B で **show ip bgp** コマンドを入力したときの出力を示します。ネットワーク 10.2.2.0 のメトリックが 50 であることに注意してください。

```
Router# show ip bgp

BGP table version is 7, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0/24      192.168.1.2              0             0 40000 i
*> 10.2.2.0/24      192.168.3.2             50             0 50000 i
*> 172.16.1.0/24    0.0.0.0                  0         32768 i
*> 172.17.1.0/24    0.0.0.0                  0         32768 i
```

次の例は、ルータ E でこの設定作業が完了し、ルータ D がルーティング アップデートを受信した後に、ルータ D で **show ip bgp** コマンドを入力したときの出力を示します。ネットワーク 10.2.2.0 のメトリックが 100 であることに注意してください。

```
Router# show ip bgp

BGP table version is 3, local router ID is 192.168.2.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.2.2.0/24      192.168.2.2            100             0 50000 i
*> 172.16.1.0/24    0.0.0.0                  0         32768 i
```

マルチホーミングのフル インターネット ルーティング テーブル受信設定

アウトバウンド ルートをフィルタリングしながら、他の自律システム内の他のルータとのネイバー関係を作成するようネットワークを設定するには、次の作業を実行します。この作業では、フル インターネット ルーティング テーブルはネイバー自律システム内のサービス プロバイダーから受信しますが、ローカルで生成されたルートだけがサービス プロバイダーにアドバタイズされることになります。この作業は、図 6 のルータ B で設定され、ローカルで生成されたルートだけを許可するアクセス リストと、ローカルで生成されたルートだけが他の自律システムへアウトバウンドでアドバタイズされるようにしたルート マップを使用します。

（注） 2 つの ISP からのフル インターネット ルーティング テーブルを受信すると、小さいルータの場合メモリを使いきってしまう可能性があることに注意が必要です。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
5. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
6. **network** *network-number* [**mask** *network-mask*]
7. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
8. **exit**
9. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
10. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
11. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
12. **exit**
13. **exit**
14. **ip as-path access-list** *access-list-number* {**deny** | **permit**} *as-regular-expression*
15. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
16. **match as-path** *path-list-number*
17. **end**
18. **show ip bgp** [*network*] [*network-mask*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	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 3	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 45000`	指定されたルーティング プロセスでルータ 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	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **例：** `Router(config-router)# neighbor 192.168.1.2 remote-as 40000`	指定された自律システム内のネイバーの IP アドレスまたはピア グループ名を、ローカル ルータの IPv4 マルチプロトコル BGP ネイバー テーブルに追加します。
ステップ 5	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **例：** `Router(config-router)# address-family ipv4 unicast`	IPv4 アドレス ファミリを指定し、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。 • **unicast** キーワードは、IPv4 ユニキャスト アドレス ファミリを指定します。デフォルトでは、**address-family ipv4** コマンドに **unicast** キーワードが指定されていない場合、ルータは IPv4 ユニキャスト アドレス ファミリのアドレス ファミリ コンフィギュレーション モードになります。 • **multicast** キーワードは、IPv4 マルチキャスト アドレス プレフィクスを指定します。 • **vrf** キーワードと *vrf-name* 引数は、後続の IPv4 アドレス ファミリ コンフィギュレーション モード コマンドに関連付ける VRF インスタンスの名前を指定します。
ステップ 6	**network** *network-number* [**mask** *network-mask*] **例：** `Router(config-router-af)# network 172.17.1.0 mask 255.255.255.0`	ネットワークを、この自律システムに対してローカルに設定し、BGP ルーティング テーブルに追加します。 • 外部プロトコルの場合、**network** コマンドはアドバタイズされるネットワークを制御します。内部プロトコルは **network** コマンドを使用して、アップデートの送信先を判断します。
ステップ 7	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**} **例：** `Router(config-router-af)# neighbor 192.168.1.2 route-map localonly out`	受信または発信ルートにルート マップを適用します。 • この例では、localonly という名前のルート マップが、ルータ A へのアウトバウンド ルートに適用されています。
ステップ 8	**exit** **例：** `Router(config-router-af)# exit`	アドレス ファミリ コンフィギュレーション モードを終了して、ルータ コンフィギュレーション モードを開始します。
ステップ 9	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **例：** `Router(config-router)# neighbor 192.168.3.2 remote-as 50000`	指定された自律システム内のネイバーの IP アドレスまたはピア グループ名を、ローカル ルータの IPv4 マルチプロトコル BGP ネイバー テーブルに追加します。

	コマンドまたはアクション	目的
ステップ 10	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **例：** `Router(config-router)# address-family ipv4 unicast`	IPv4 アドレス ファミリを指定し、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。 • **unicast** キーワードは、IPv4 ユニキャスト アドレス ファミリを指定します。デフォルトでは、**address-family ipv4** コマンドに **unicast** キーワードが指定されていない場合、ルータは IPv4 ユニキャスト アドレス ファミリのアドレス ファミリ コンフィギュレーション モードになります。 • **multicast** キーワードは、IPv4 マルチキャスト アドレス プレフィクスを指定します。 **vrf** キーワードと *vrf-name* 引数は、後続の IPv4 アドレス ファミリ コンフィギュレーション モード コマンドに関連付ける VRF インスタンスの名前を指定します。
ステップ 11	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**} **例：** `Router(config-router-af)# neighbor 192.168.3.2 route-map localonly out`	受信または発信ルートにルート マップを適用します。 • この例では、localonly という名前のルート マップが、ルータ E へのアウトバウンド ルートに適用されています。
ステップ 12	**exit** **例：** `Router(config-router-af)# exit`	アドレス ファミリ コンフィギュレーション モードを終了して、ルータ コンフィギュレーション モードを開始します。
ステップ 13	**exit** **例：** `Router(config-router)# exit`	ルータ コンフィギュレーション モードを終了して、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 14	**ip as-path access-list** *access-list-number* {**deny** | **permit**} *as-regular-expression* **例：** `Router(config)# ip as-path access-list 10 permit ^$`	BGP-related アクセス リストを定義します。 • この例では、アクセス リスト番号 10 が、ローカルで生成された BGP ルートだけを許可するよう定義されています。
ステップ 15	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **例：** `Router(config)# route-map localonly permit 10`	ルート マップを設定し、ルート マップ コンフィギュレーション モードを開始します。 • この例では、localonly という名前のルート マップが作成されます。
ステップ 16	**match as-path** *path-list-number* **例：** `Router(config-route-map)# match as-path 10`	BGP 自律システム パス アクセス リストのマッチングを行います。 • この例では、match 句にステップ 12 で作成された BGP 自律システム パス アクセス リストが使用されます。

	コマンドまたはアクション	目的
ステップ 17	**end** **例：** `Router(config-route-map)# end`	ルート マップ コンフィギュレーション モードを終了して、特権 EXEC モードを開始します。
ステップ 18	**show ip bgp** [*network*] [*network-mask*] **例：** `Router# show ip bgp`	BGP ルーティング テーブル内のエントリを表示します。 **（注）** この例では、この作業に適用される構文だけが使用されます。詳細については、『*Cisco IOS IP Routing: BGP Command Reference*』を参照してください。

例

次に、この作業設定が完了した後の、図 6 のルータ B の BGP ルーティング テーブルの例を示します。ルーティング テーブルには、自律システム 40000 と 50000 のネットワークについての情報が含まれることに注意してください。

```
BGP table version is 5, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0/24      192.168.1.2              0             0 40000 i
*> 10.2.2.0/24      192.168.3.2              0             0 50000 i
*> 172.17.1.0/24    0.0.0.0                  0         32768 i
```

BGP ポリシーの設定

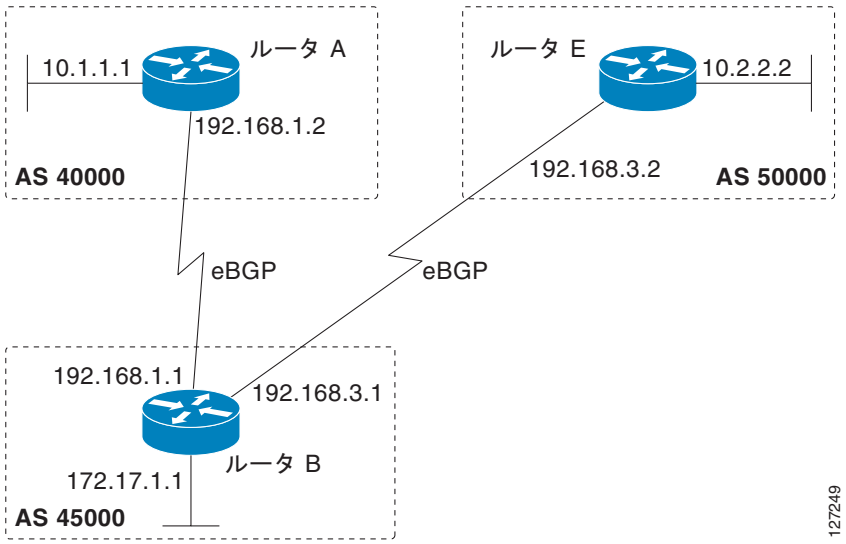
このセクションの作業は、BGP ネットワーク内でトラフィックをフィルタリングする BGP ポリシーの設定に役立ちます。次に示すオプション作業は、BGP ネットワークでトラフィックをフィルタリングするさまざまな方法の一部を示すものです。

- 「プレフィクス リストによる BGP プレフィクスのフィルタリング」（P.41）
- 「AS-path フィルタを使用した BGP プレフィクスのフィルタリング」（P.45）
- 「4 バイト自律システム番号を使用した AS-path フィルタによる BGP プレフィクスのフィルタリング」（P.48）
- 「コミュニティ リストを使用したトラフィック フィルタリング」（P.52）
- 「拡張コミュニティ リストを使用したトラフィック フィルタリング」（P.55）
- 「BGP ルート マップ ポリシー リストを使用したトラフィック フィルタリング」（P.59）
- 「BGP ルート マップでの continue 句の使用によるトラフィック フィルタリング」（P.62）

プレフィクス リストによる BGP プレフィクスのフィルタリング

プレフィクス リストを使用して BGP ルート情報をフィルタリングするには、次の作業を実行します。この設定作業は、図 7 においてルータ A とルータ E の両方が BGP ピアとしてセットアップされた状況で、ルータ B で実行します。アウトバウンドにするため、プレフィクス リストをネットワーク 10.2.2.0/24 からのルートだけを許可するよう設定します。実際には、ルータ A への転送のためルータ E から受信した情報がこれにより制限されます。プレフィクス リスト情報を表示し、ヒット カウントをリセットするためのオプション ステップが含まれます。

図 7　　BGP ポリシー設定作業の BGP トポロジ



制約事項

neighbor prefix-list コマンドおよび **neighbor distribute-list** コマンドは、BGP ピアに同時に使用できません。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. すべての BGP ピアにステップ 5 を繰り返します。
6. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
7. **network** *network-number* [**mask** *network-mask*]
8. **aggregate-address** *address mask* [**as-set**]
9. **neighbor** *ip-address* **prefix-list** *list-name* {**in** | **out**}
10. **exit**
11. **exit**
12. **ip prefix-list** *list-name* [**seq** *seq-number*] {**deny** *network/length* | **permit** *network/length*} [**ge** *ge-value*] [**le** *le-value*] [**eq** *eq-value*]
13. **end**
14. **show ip prefix-list** [**detail** | **summary**] [*prefix-list-name* [**seq** *seq-number* | *network/length* [**longer** | **first-match**]]]
15. **clear ip prefix-list** {* | *ip-address* | *peer-group-name*} **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	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	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 3	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 45000`	指定されたルーティング プロセスでルータ 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	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **例：** `Router(config-router)# neighbor 192.168.1.2 remote-as 40000`	指定された自律システム内のネイバーの IP アドレスをローカル ルータの BGP ネイバー テーブルに追加します。
ステップ 5	すべての BGP ピアにステップ 5 を繰り返します。	—
ステップ 6	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **例：** `Router(config-router)# address-family ipv4 unicast`	IPv4 アドレス ファミリを指定し、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。 • **unicast** キーワードは、IPv4 ユニキャスト アドレス ファミリを指定します。デフォルトでは、**address-family ipv4** コマンドに **unicast** キーワードが指定されていない場合、ルータは IPv4 ユニキャスト アドレス ファミリのアドレス ファミリ コンフィギュレーション モードになります。 • **multicast** キーワードは、IPv4 マルチキャスト アドレス プレフィクスを指定します。 • **vrf** キーワードと *vrf-name* 引数は、後続の IPv4 アドレス ファミリ コンフィギュレーション モード コマンドに関連付ける VRF インスタンスの名前を指定します。
ステップ 7	**network** *network-number* [**mask** *network-mask*] **例：** `Router(config-router-af)# network 172.17.1.0 mask 255.255.255.0`	（任意）ネットワークを、この自律システムに対してローカルに設定し、BGP ルーティング テーブルに追加します。 • 外部プロトコルの場合、**network** コマンドはアドバタイズされるネットワークを制御します。内部プロトコルは **network** コマンドを使用して、アップデートの送信先を判断します。

	コマンドまたはアクション	目的
ステップ 8	**aggregate-address** *address mask* [**as-set**] **例：** `Router(config-router-af)# aggregate-address 172.0.0.0 255.0.0.0`	BGP ルーティング テーブルに集約エントリを作成します。 • 指定されたルートは、BGP テーブル内に存在する必要があります。 • 指定された範囲に含まれる、より詳しい BGP ルートがある場合は、キーワードを指定せずに **aggregate-address** コマンドを使用して、集約エントリを作成します。 **（注）** この例では、一部の構文だけが使用されています。詳細については、『*Cisco IOS IP Routing: BGP Command Reference*』を参照してください。
ステップ 9	**neighbor** *ip-address* **prefix-list** *list-name* {**in** | **out**} **例：** `Router(config-router-af)# neighbor 192.168.1.2 prefix-list super172 out`	プレフィクス リストで指定された BGP ネイバー情報を配布します。 • この例では、super172 と呼ばれるプレフィクス リストがルータ A の発信ルートに設定されます。
ステップ 10	**exit** **例：** `Router(config-router-af)# exit`	アドレス ファミリ コンフィギュレーション モードを終了して、ルータ コンフィギュレーション モードを開始します。
ステップ 11	**exit** **例：** `Router(config-router) exit`	ルータ コンフィギュレーション モードを終了して、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 12	**ip prefix-list** *list-name* [**seq** *seq-number*] {**deny** *network*/*length* | **permit** *network*/*length*} [**ge** *ge-value*] [**le** *le-value*] [**eq** *eq-value*] **例：** `Router(config)# ip prefix-list super172 permit 172.0.0.0/8`	BGP 関連のプレフィクス リストを定義し、アクセス リスト コンフィギュレーション モードを開始します。 • この例では、転送されるルートとして 172.0.0.0/8 だけを許可する、super172 と呼ばれるプレフィクス リストが定義されます。 • すべてのプレフィクス リストの末尾には明示的な拒否があるため、他のルートはすべて拒否されます。
ステップ 13	**end** **例：** `Router(config-access-list)# end`	アクセス リスト コンフィギュレーション モードを終了し、特権 EXEC モードを開始します。
ステップ 14	**show ip prefix-list** [**detail** | **summary**] [*prefix-list-name* [**seq** *seq-number* | *network*/*length* [**longer** | **first-match**]]] **例：** `Router# show ip prefix-list detail super172`	プレフィクス リストについての情報を表示します。 • この例では、super172 という名前のプレフィクス リストの詳細が、ヒット カウントを含めて表示されます。ヒット カウントとは、エントリがルートに一致した回数のことです。
ステップ 15	**clear ip prefix-list** {***** | *ip-address* | *peer-group-name*} **out** **例：** `Router# clear ip prefix-list super172 out`	プレフィクス リスト エントリのヒット カウントをリセットします。 • この例では、super172 と呼ばれるプレフィクス リストのヒット カウントがリセットされます。

例

次に示す **show ip prefix-list** コマンドからの出力では、super172 という名前のプレフィクス リストの詳細が、ヒット カウントを含めて表示されます。**clear ip prefix-list** コマンドが入力されてヒット カウントがリセットされ、さらに再度 **show ip prefix-list** コマンドが入力されて、0 にリセットされたヒット カウントが表示されます。

```
Router# show ip prefix-list detail super172

ip prefix-list super172:
   count: 1, range entries: 0, sequences: 5 - 5, refcount: 4
   seq 5 permit 172.0.0.0/8 (hit count: 1, refcount: 1)

Router# clear ip prefix-list super172

Router# show ip prefix-list detail super172

ip prefix-list super172:
   count: 1, range entries: 0, sequences: 5 - 5, refcount: 4
   seq 5 permit 172.0.0.0/8 (hit count: 0, refcount: 1)
```

AS-path フィルタを使用した BGP プレフィクスのフィルタリング

フィルタ ルート情報への AS-path アトリビュートの値をベースにしたアクセス リスト付きの AS-path フィルタを使用して BGP プレフィクスをフィルタリングするには、次の作業を実行します。図 7 では、AS-path アクセス リストがルータ B で設定されます。アクセス リストの 1 行目では、AS-path 50000 に一致するものがすべて拒否され、2 行目では他のパスすべてが許可されています。ルータは **neighbor filter-list** コマンドを使用して、AS-path アクセス リストをアウトバウンド フィルタとして指定します。フィルタリングがイネーブルにされた後、トラフィックはルータ A とルータ C の両方で受信されますが、自律システム 50000（ルータ C）で生成されたアップデートがルータ B によりルータ A に転送されることはありません。ルータ C からのアップデートのうち、別の自律システムで生成されたものが何かあった場合、その中には自律システム 50000 だけでなく別の自律システム番号も含まれていることから、アップデートは転送されることになり、AS-path アクセス リストとは一致しないことになります。

（注） Cisco IOS Release 12.0(22)S、12.2(15)T、12.2(18)S、およびそれ以降のリリースでは、**ip as-path access-list** コマンドを使用して設定できる自律システム アクセス リストの上限値が、199 から 500 に増加しました。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
5. すべての BGP ピアにステップ 5 を繰り返します。
6. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
7. **network** *network-number* [**mask** *network-mask*]
8. **neighbor** {*ip-address* | *peer-group-name*} **filter-list** *access-list-number* {**in** | **out**}
9. **exit**

10. **exit**
11. **ip as-path access-list** *access-list-number* {**deny** | **permit**} *as-regular-expression*
12. AS-path アクセス リストで要求されているすべてのエントリについて、ステップ 9 を繰り返します。
13. **end**
14. **show ip bgp regexp** *as-regular-expression*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	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 3	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 45000`	指定されたルーティング プロセスでルータ 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	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **例：** `Router(config-router)# neighbor 192.168.1.2 remote-as 40000`	指定した自律システム内のネイバーの IP アドレスまたはピア グループ名を、ローカル ルータの BGP ネイバー テーブルに追加します。
ステップ 5	すべての BGP ピアについて ステップ 4 を繰り返します。	—
ステップ 6	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **例：** `Router(config-router)# address-family ipv4 unicast`	IPv4 アドレス ファミリを指定し、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。 • **unicast** キーワードは、IPv4 ユニキャスト アドレス ファミリを指定します。デフォルトでは、**address-family ipv4** コマンドに **unicast** キーワードが指定されていない場合、ルータは IPv4 ユニキャスト アドレス ファミリのアドレス ファミリ コンフィギュレーション モードになります。 • **multicast** キーワードは、IPv4 マルチキャスト アドレス プレフィクスを指定します。 • **vrf** キーワードと *vrf-name* 引数は、後続の IPv4 アドレス ファミリ コンフィギュレーション モード コマンドに関連付ける VRF インスタンスの名前を指定します。

	コマンドまたはアクション	目的
ステップ 7	**network** *network-number* [**mask** *network-mask*] **例：** `Router(config-router-af)# network 172.17.1.0 mask 255.255.255.0`	（任意）ネットワークを、この自律システムに対してローカルに設定し、BGP ルーティング テーブルに追加します。 • 外部プロトコルの場合、**network** コマンドはアドバタイズされるネットワークを制御します。内部プロトコルは **network** コマンドを使用して、アップデートの送信先を判断します。 **（注）** この例では、一部の構文だけが使用されています。詳細については、『*Cisco IOS IP Routing: BGP Command Reference*』を参照してください。
ステップ 8	**neighbor** {*ip-address* | *peer-group-name*} **filter-list** *access-list-number* {**in** | **out**} **例：** `Router(config-router-af)# neighbor 192.168.1.2 filter-list 100 out`	プレフィクス リストで指定された BGP ネイバー情報を配布します。 • この例では、アクセス リスト番号 100 が、ルータ A への発信ルートに設定されます。
ステップ 9	**exit** **例：** `Router(config-router-af)# exit`	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、ルータ コンフィギュレーション モードに戻ります。
ステップ 10	**exit** **例：** `Router(config-router)# exit`	ルータ 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 モードに戻ります。

ステップ 11	**ip as-path access-list** *access-list-number* {**deny** | **permit**} *as-regular-expression*
	
	例：
	`Router(config)# ip as-path access-list 100 deny ^50000$`
	
	および
	
	例：
	`Router(config)# ip as-path access-list 100 permit .*`
		BGP 関連のアクセス リストを定義し、アクセス リスト コンフィギュレーション モードを開始します。
		• 1 番目の例では、アクセス リスト番号 100 は 50000 で始まり 50000 で終わる AS-path はすべて拒否するように定義されています。
		• 2 番目の例では、AS-path アクセス リストの 1 番目の例での基準に一致しないルートは、すべて許可されます。ピリオドとアスタリスク記号は AS-path 内のすべての文字が一致することを示しているため、ルータ B はそれらのアップデートをルータ A に転送することになります。
		（注） 作業例ではこれらの文の双方を設定する必要があるため、2 つの例を示しています。

ステップ 12	AS-path アクセス リストで要求されているすべてのエントリについて、ステップ 11 を繰り返します。	—
ステップ 13	**end** **例：** `Router(config-access-list)# end`	アクセス リスト コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 14	**show ip bgp regexp** *as-regular-expression* **例：** `Router# show ip bgp regexp ^50000$`	正規表現に一致するルートを表示します。 • 正規表現の確認にこのコマンドを使用できます。 • この例では、「50000 で始まり 50000 で終わる」表現に一致するパスすべてが表示されます。

例

次の、**show ip bgp regexp** コマンドからの出力は、AS-path が 50000 で始まり 50000 で終わるという正規表現に一致する自律システム パスを表示します。

```
Router# show ip bgp regexp ^50000$

BGP table version is 9, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.2.2.0/24      192.168.3.2              0             150 50000 i
```

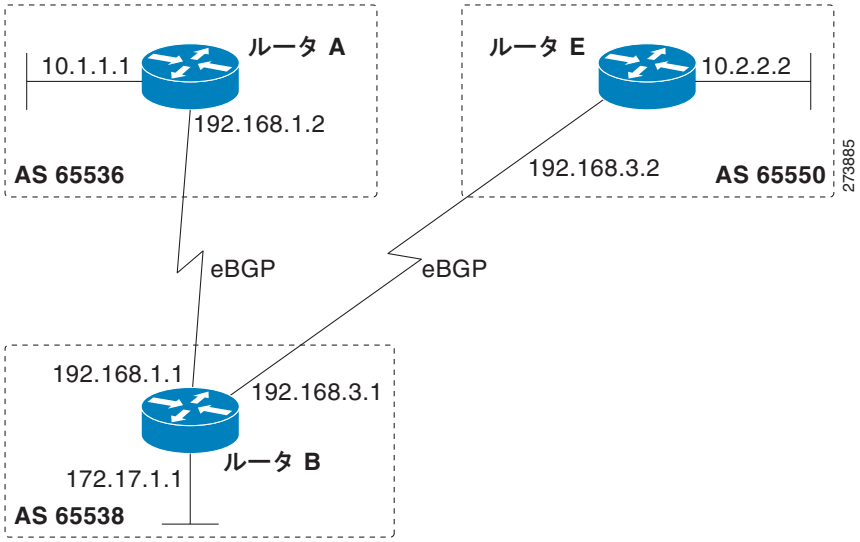
4 バイト自律システム番号を使用した AS-path フィルタによる BGP プレフィクスのフィルタリング

Cisco IOS Release 12.0(32)SY8、12.0(33)S3、12.2(33)SRE、12.2(33)SXI1、およびそれ以降のリリースで、BGP は 4 オクテット（4 バイト）自律システム番号をサポートするようになりました。この作業の 4 バイト自律システム番号は、デフォルトの asplain（10 進数）形式です。たとえば、図 8（P.49）において、ルータ B は自律システム番号 65538 にあります。4 バイト自律システム番号の詳細については、「BGP 自律システム番号の形式」（P.4）を参照してください。

4 バイト自律システム番号とルート情報フィルタ用の AS-path アトリビュートの値に基づくアクセス リストを使用して AS-path フィルタで BGP プレフィクスをフィルタリングするには、次の作業を実行します。図 8 では、AS-path アクセス リストがルータ B で設定されます。アクセス リストの 1 行目では、AS パス 65550 に一致するものがすべて拒否され、2 行目では他のパスすべてが許可されています。ルータは **neighbor filter-list** コマンドを使用して、AS-path アクセス リストをアウトバウンド フィルタとして指定します。フィルタリングがイネーブルにされた後、トラフィックはルータ A とルータ E の両方で受信されますが、自律システム 65550（ルータ E）で生成されたアップデートがルータ B によりルータ A に転送されることはありません。ルータ E からのアップデートのうち、別の自律システムで生成されたものが何かあった場合、その中には自律システム 65550 だけでなく別の自律システム番号も含まれていることから、アップデートは転送されることになり、AS-path アクセス リストとは一致しないことになります。

（注） Cisco IOS Release 12.0(22)S、12.2(15)T、12.2(18)S、およびそれ以降のリリースでは、**ip as-path access-list** コマンドを使用して設定できる自律システム アクセス リストの上限値が、199 から 500 に増加しました。

図 8　4 バイト自律システム番号を使用した AS-path フィルタによる BGP プレフィクス フィルタリングの BGP トポロジ



手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
5. すべての BGP ピアにステップ 5 を繰り返します。
6. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
7. **network** *network-number* [**mask** *network-mask*]
8. **neighbor** {*ip-address* | *peer-group-name*} **filter-list** *access-list-number* {**in** | **out**}
9. **exit**
10. **exit**
11. **ip as-path access-list** *access-list-number* {**deny** | **permit**} *as-regular-expression*
12. AS-path アクセス リストで要求されているすべてのエントリについて、ステップ 10 を繰り返します。
13. **end**
14. **show ip bgp regexp** *as-regular-expression*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	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 3	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 65538`	指定されたルーティング プロセスでルータ 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	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **例：** `Router(config-router-af)# neighbor 192.168.1.2 remote-as 65536`	指定した自律システム内のネイバーの IP アドレスまたはピア グループ名を、ローカル ルータの BGP ネイバー テーブルに追加します。 • この例では、ルータ A でのネイバーの IP アドレスが追加されます。
ステップ 5	すべての BGP ピアについて ステップ 4 を繰り返します。	—
ステップ 6	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **例：** `Router(config-router)# address-family ipv4 unicast`	IPv4 アドレス ファミリを指定し、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。 • **unicast** キーワードは、IPv4 ユニキャスト アドレス ファミリを指定します。デフォルトでは、**address-family ipv4** コマンドに **unicast** キーワードが指定されていない場合、ルータは IPv4 ユニキャスト アドレス ファミリのアドレス ファミリ コンフィギュレーション モードになります。 • **multicast** キーワードは、IPv4 マルチキャスト アドレス プレフィクスを指定します。 • **vrf** キーワードと *vrf-name* 引数は、後続の IPv4 アドレス ファミリ コンフィギュレーション モード コマンドに関連付ける VRF インスタンスの名前を指定します。
ステップ 7	**network** *network-number* [**mask** *network-mask*] **例：** `Router(config-router-af)# network 172.17.1.0 mask 255.255.255.0`	（任意）ネットワークを、この自律システムに対してローカルに設定し、BGP ルーティング テーブルに追加します。 • 外部プロトコルの場合、**network** コマンドはアドバタイズされるネットワークを制御します。内部プロトコルは **network** コマンドを使用して、アップデートの送信先を判断します。 **（注）** この例では、一部の構文だけが使用されています。詳細については、『*Cisco IOS IP Routing: BGP Command Reference*』を参照してください。

	コマンドまたはアクション	目的
ステップ 8	**neighbor** {*ip-address* | *peer-group-name*} **filter-list** *access-list-number* {**in** | **out**} **例：** `Router(config-router-af)# neighbor 192.168.1.2 filter-list 99 out`	プレフィクス リストで指定された BGP ネイバー情報を配布します。 • この例では、アクセス リスト番号 99 が、ルータ A への発信ルートに設定されます。
ステップ 9	**exit** **例：** `Router(config-router-af)# exit`	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、ルータ コンフィギュレーション モードに戻ります。
ステップ 10	**exit** **例：** `Router(config-router)# exit`	ルータ 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 モードに戻ります。

ステップ 11	**ip as-path access-list** *access-list-number* {**deny** | **permit**} *as-regular-expression*
	
	例：
	`Router(config)# ip as-path access-list 99 deny ^65550$`
	
	および
	
	例：
	`Router(config)# ip as-path access-list 99 permit .*`
		BGP 関連のアクセス リストを定義し、アクセス リスト コンフィギュレーション モードを開始します。
		• 1 番目の例では、アクセス リスト番号 99 は 65550 で始まり 65550 で終わる AS-path はすべて拒否するように定義されています。
		• 2 番目の例では、AS-path アクセス リストの 1 番目の例での基準に一致しないルートは、すべて許可されます。ピリオドとアスタリスク記号は AS-path 内のすべての文字が一致することを示しているため、ルータ B はそれらアップデートをルータ A に転送することになります。
		（注） 作業例ではこれらの文の双方を設定する必要があるため、2 つの例を示しています。

ステップ 12	AS-path アクセス リストで要求されているすべてのエントリについて、ステップ 11 を繰り返します。	—
ステップ 13	**end** **例：** `Router(config-access-list)# end`	アクセス リスト コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 14	**show ip bgp regexp** *as-regular-expression* **例：** `Router# show ip bgp regexp ^65550$`	正規表現に一致するルートを表示します。 • 正規表現の確認にこのコマンドを使用できます。 • この例では、「65550 で始まり 65550 で終わる」表現に一致するパスすべてが表示されます。

例

次の、**show ip bgp regexp** コマンドからの出力は、AS-path が 65550 で始まり 65550 で終わるという正規表現に一致する自律システム パスを表示します。

```
RouterB# show ip bgp regexp ^65550$

BGP table version is 4, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
```

```
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.2.2.0/24      192.168.3.2              0             0 65550 i
```

コミュニティ リストを使用したトラフィック フィルタリング

BGP コミュニティ リストを作成し、受信ルートの制御のためにルート マップ内で参照することによりトラフィックをフィルタリングするには、次の作業を実行します。BGP コミュニティは、複雑で規模の大きなネットワークでインバウンドとアウトバウンドのルートをフィルタリングする手段を提供します。個別のピアについて長大なアクセス リストやプレフィクス リストを編集する代わりに、BGP では同一のルーティングポリシーを持つピアを、たとえそれらが異なる自律システムやネットワークに分散していたとしても、グループ化することができます。

この作業では、受信ルートを制御するために、図 7 のルータ B を、いくつかのルート マップとコミュニティ リストを使用して設定します。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
5. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
6. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *route-map-name* {**in** | **out**}
7. **exit**
8. **exit**
9. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
10. **match community** {*standard-list-number* | *expanded-list-number* | *community-list-name* [**exact**]}
11. **set weight** *weight*
12. **exit**
13. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
14. **match community** {*standard-list-number* | *expanded-list-number* | *community-list-name* [**exact**]}
15. **set community** *community-number*
16. **exit**
17. **ip community-list** {*standard-list-number* | **standard** *list-name* {**deny** | **permit**} [*community-number*] [*AA*:*NN*] **[internet]** [**local-AS**] [**no-advertise**] [**no-export**]} | {*expanded-list-number* | **expanded** *list-name* {**deny** | **permit**} *regular-expression*}
18. ステップ 15 を繰り返して、必要なコミュニティ リストすべてを作成します。
19. **end**
20. **show ip community-list** [*standard-list-number* | *expanded-list-number* | *community-list-name*] [**exact-match**]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	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 3	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 45000`	指定されたルーティング プロセスでルータ 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	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **例：** `Router(config-router)# neighbor 192.168.3.2 remote-as 50000`	指定した自律システム内のネイバーの IP アドレスまたはピア グループ名を、ローカル ルータの BGP ネイバー テーブルに追加します。
ステップ 5	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **例：** `Router(config-router)# address-family ipv4 unicast`	IPv4 アドレス ファミリを指定し、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。 • **unicast** キーワードは、IPv4 ユニキャスト アドレス ファミリを指定します。デフォルトでは、**address-family ipv4** コマンドに **unicast** キーワードが指定されていない場合、ルータは IPv4 ユニキャスト アドレス ファミリのアドレス ファミリ コンフィギュレーション モードになります。 • **multicast** キーワードは、IPv4 マルチキャスト アドレス プレフィクスを指定します。 • **vrf** キーワードと *vrf-name* 引数は、後続の IPv4 アドレス ファミリ コンフィギュレーション モード コマンドに関連付ける VRF インスタンスの名前を指定します。
ステップ 6	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *route-map-name* {**in** | **out**} **例：** `Router(config-router-af)# neighbor 192.168.3.2 route-map 2000 in`	インバウンドまたはアウトバウンドのルートにルート マップを適用します。 • この例では、2000 と呼ばれるルート マップが、192.168.3.2 の BGP ピアからのインバウンド ルートに適用されます。
ステップ 7	**exit** **例：** `Router(config-router-af)# exit`	アドレス ファミリ コンフィギュレーション モードを終了して、ルータ 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：** `Router(config-router)# exit`	ルータ コンフィギュレーション モードを終了して、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。

	コマンドまたはアクション	目的
ステップ 9	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **例：** `Router(config)# route-map 2000 permit 10`	ルート マップを作成し、ルート マップ コンフィギュレーション モードを開始します。 • この例では、2000 と呼ばれるルート マップが定義されます。
ステップ 10	**match community** {*standard-list-number* | *expanded-list-number* | *community-list-name* [**exact**]} **例：** `Router(config-route-map)# match community 1`	BGP コミュニティ リストのマッチングを行います。 • この例では、コミュニティ アトリビュートがコミュニティ リスト 1 と一致しています。
ステップ 11	**set weight** *weight* **例：** `Router(config-route-map)# set weight 30`	ルーティング テーブルの BGP weight を指定します。 • この例では、コミュニティ リスト 1 に一致するすべてのルートが、30 に設定された BGP weight を持つことになります。
ステップ 12	**exit** **例：** `Router(config-route-map)# exit`	ルート マップ コンフィギュレーション モードを終了して、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 13	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **例：** `Router(config)# route-map 3000 permit 10`	ルート マップを作成し、ルート マップ コンフィギュレーション モードを開始します。 • この例では、3000 と呼ばれるルート マップが定義されます。
ステップ 14	**match community** {*standard-list-number* | *expanded-list-number* | *community-list-name* [**exact**]} **例：** `Router(config-route-map)# match community 2`	BGP コミュニティ リストのマッチングを行います。 • この例では、コミュニティ アトリビュートがコミュニティ リスト 2 と一致しています。
ステップ 15	**set community** *community-number* **例：** `Router(config-route-map)# set community 99`	BGP コミュニティ アトリビュートを設定します。 • この例では、コミュニティ リスト 2 に一致するすべてのルートが、99 に設定された BGP コミュニティ アトリビュートを持つことになります。
ステップ 16	**exit** **例：** `Router(config-route-map)# exit`	ルート マップ コンフィギュレーション モードを終了して、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。

	コマンドまたはアクション	目的

ステップ 17	**ip community-list** {*standard-list-number* | **standard** *list-name* {**deny** | **permit**} [*community-number*] [*AA:NN*] [**internet**] [**local-AS**] [**no-advertise**] [**no-export**]} | {*expanded-list-number* | **expanded** *list-name* {**deny** | **permit**} *regular-expression*}
	
	例：
	`Router(config)# ip community-list 1 permit 100`
	
	および
	
	例：
	`Router(config)# ip community-list 2 permit internet`
		BGP のコミュニティ リストを作成してアクセスを制御します。
		• 1 番目の例では、コミュニティ リスト 1 はコミュニティ アトリビュートが 100 のルートを許可しています。ルータ C のルートはすべてコミュニティ アトリビュートが 100 であるため、weight は 30 に設定されます。
		• 2 番目の例では、コミュニティ リスト 2 は **internet** キーワードを使用することで、効果的にすべてのルートを許可しています。コミュニティ リスト 1 に一致しなかったルートはどれも、コミュニティ リスト 2 でチェックされます。すべてのルートが許可されますが、route アトリビュートには変化が加えられません。
		（注） 作業例ではこれらの文の双方を設定する必要があるため、2 つの例を示しています。

ステップ 18	ステップ 15 を繰り返して、必要なコミュニティ リストすべてを作成します。	—
ステップ 19	**exit** **例：** `Router(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードを開始します。
ステップ 20	**show ip community-list** [*standard-list-number* | *expanded-list-number* | *community-list-name*] [**exact-match**] **例：** `Router# show ip community-list 1`	設定された BGP コミュニティ リスト エントリを表示します。

例

次の出力例は、コミュニティ リスト 1 が作成されたことを確認し、コミュニティ アトリビュートが 100 のルートがコミュニティ リスト 1 で許可されていることを示しています。

```
Router# show ip community-list 1

Community standard list 1
     permit 100
```

次の出力例は、コミュニティ リスト 2 が作成されたことを確認し、**internet** キーワードの使用により効果的にすべてのルートがコミュニティ リスト 2 で許可されたことを示しています。

```
Router# show ip community-list 2

Community standard list 2
     permit internet
```

拡張コミュニティ リストを使用したトラフィック フィルタリング

拡張 BGP コミュニティ リストを作成してアウトバウンド ルートを制御することによりトラフィックをフィルタリングするには、次の作業を実行します。BGP コミュニティは、複雑で規模の大きなネットワークでインバウンドとアウトバウンドのルートをフィルタリングする手段を提供します。個別のピア

について長大なアクセス リストやプレフィクス リストを編集する代わりに、BGP では同一のルーティングポリシーを持つピアを、たとえそれらが異なる自律システムやネットワークに分散していたとしても、グループ化することができます。

この作業において 図 7 のルータ B は、拡張名前付きコミュニティ リストを使用して設定され、192.168.1.2 の BGP ピアが自律システム 50000 からの、または 50000 経由のパスについてのアドバタイズメントも送られないよう指定されます。IP 拡張コミュニティリスト コンフィギュレーション モードが使用され、エントリのシーケンス番号再割り当て機能が示されます。

拡張コミュニティ リスト

拡張コミュニティ リストは、VRF インスタンスと MPLS VPN のルートを設定し、フィルタリングし、識別するために使用されます。名前付きまたは番号付きコミュニティ リストの設定には、**ip extcommunity-list** コマンドを使用します。アクセス リストの標準ルールすべてが、拡張コミュニティ リストの設定に適用されます。正規表現は、拡張コミュニティ リスト番号の拡張範囲によってサポートされています。

制約事項

拡張コミュニティ リストのエントリにはすべて、コンフィギュレーション モードにかかわらずデフォルトでシーケンス番号が適用されます。拡張コミュニティ リスト エントリのシーケンス番号の明示的な割り当てと再割り当ては、IP 拡張コミュニティリスト コンフィギュレーション モードだけで設定でき、グローバル コンフィギュレーション モードでは設定できません。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip extcommunity-list** {*expanded-list-number* | **expanded** *list-name standard-list-number* | **standard** *list-name*}
4. [*sequence-number*] {**deny** [*regular-expression*] | **exit** | **permit** [*regular-expression*]}
5. 拡張コミュニティ リスト内のすべての必要な許可や拒否エントリについて、ステップ 4 を繰り返します。
6. **resequence** [*starting-sequence*] [*sequence-increment*]
7. **exit**
8. **router bgp** *autonomous-system-number*
9. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
10. 必要な BGP ピアすべてについて、ステップ 10 を繰り返します。
11. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
12. **network** *network-number* [**mask** *network-mask*]
13. **end**
14. **show ip extcommunity-list** [*list-number* | *list-name*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	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 3	**ip extcommunity-list** {*expanded-list-number* | **expanded** *list-name* | *standard-list-number* | **standard** *list-name*} **例：** `Router(config)# ip extcommunity-list expanded DENY50000`	IP 拡張コミュニティリスト コンフィギュレーション モードを開始し、拡張コミュニティ リストの作成や設定を行います。 • この例では、拡張コミュニティ リスト DENY50000 が作成されます。

ステップ 4	[*sequence-number*] {**deny** [*regular-expression*] | **exit** | **permit** [*regular-expression*]}
	
	例：
	`Router(config-extcomm-list)# 10 deny _50000_`
	
	および
	
	例：
	`Router(config-extcomm-list)# 20 deny ^50000 .*`
		拡張コミュニティ リスト エントリを設定します。
		• 1 番目の例では、自律システム 50000 からのパスについてのアドバタイズメントを拒否するよう、シーケンス番号 10 の拡張コミュニティ リスト エントリが設定されます。
		• 2 番目の例では、自律システム 50000 を経由するパスについてのアドバタイズメントを拒否するよう、シーケンス番号 20 の拡張コミュニティ リスト エントリが設定されます。
		（注） 作業例ではこれらの文の双方を設定する必要があるため、2 つの例を示しています。
		
		（注） この例では、この作業に適用される構文だけが使用されます。詳細については、『*Cisco IOS IP Routing: BGP Command Reference*』を参照してください。

ステップ 5	拡張コミュニティ リスト内のすべての必要な許可や拒否エントリについて、ステップ 4 を繰り返します。	—
ステップ 6	**resequence** [*starting-sequence*] [*sequence-increment*] **例：** `Router(config-extcomm-list)# resequence 50 100`	拡張コミュニティ リスト エントリのシーケンス番号を再割り当てします。 • この例では、最初の拡張コミュニティ リスト エントリを 50 に、続くエントリは 100 ずつ増えるように設定されます。そのため、2 番目の拡張コミュニティ リスト エントリは 150 になります。 **（注）** この例では、この作業に適用される構文だけが使用されます。詳細については、『*Cisco IOS IP Routing: BGP Command Reference*』を参照してください。

	コマンドまたはアクション	目的
ステップ 7	**exit** **例：** `Router(config-extcomm-list)# exit`	拡張コミュニティリスト 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 8	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 45000`	指定されたルーティング プロセスでルータ コンフィギュレーション モードを開始します。
ステップ 9	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **例：** `Router(config-router)# neighbor 192.168.3.2 remote-as 50000`	指定した自律システム内のネイバーの IP アドレスまたはピア グループ名を、ローカル ルータの BGP ネイバー テーブルに追加します。
ステップ 10	必要な BGP ピアすべてについて、ステップ 10 を繰り返します。	—
ステップ 11	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **例：** `Router(config-router)# address-family ipv4 unicast`	IPv4 アドレス ファミリを指定し、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。 • **unicast** キーワードは、IPv4 ユニキャスト アドレス ファミリを指定します。デフォルトでは、**address-family ipv4** コマンドに **unicast** キーワードが指定されていない場合、ルータは IPv4 ユニキャスト アドレス ファミリのアドレス ファミリ コンフィギュレーション モードになります。 • **multicast** キーワードは、IPv4 マルチキャスト アドレス プレフィクスを指定します。 **（注）** **vrf** キーワードと *vrf-name* 引数は、後続の IPv4 アドレス ファミリ コンフィギュレーション モード コマンドに関連付ける VRF インスタンスの名前を指定します。
ステップ 12	**network** *network-number* [**mask** *network-mask*] **例：** `Router(config-router-af)# network 172.17.1.0 mask 255.255.255.0`	（任意）ネットワークを、この自律システムに対してローカルに設定し、BGP ルーティング テーブルに追加します。 • 外部プロトコルの場合、**network** コマンドはアドバタイズされるネットワークを制御します。内部プロトコルは **network** コマンドを使用して、アップデートの送信先を判断します。 **（注）** この例では、この作業に適用される構文だけが使用されます。詳細については、『*Cisco IOS IP Routing: BGP Command Reference*』を参照してください。
ステップ 13	**end** **例：** `Router(config-router-af)# end`	アドレス ファミリ コンフィギュレーション モードを終了して、特権 EXEC モードを開始します。
ステップ 14	**show ip extcommunity-list** [*list-name*] **例：** `Router# show ip extcommunity-list DENY50000`	設定された拡張 BGP コミュニティ リスト エントリを表示します。

例

次の出力例は、BGP 拡張コミュニティ リスト DENY50000 が作成されたことを確認するもので、出力は自律システム 50000 についてのアドバタイズメントを拒否するエントリのシーケンス番号が、10 と 20 から再割り当てによって 50 と 150 になったことを示しています。

```
Router# show ip extcommunity-list DENY50000

Expanded extended community-list DENY50000
     50 deny _50000_
     150 deny ^50000 .*
```

BGP ルート マップ ポリシー リストを使用したトラフィック フィルタリング

BGP ポリシー リストを作成してルート マップ内で参照するには、次の作業を実行します。

ポリシー リストは、match 句だけを含んだルート マップのようなものです。ポリシー リストに伴う match 句セマンティックやルート マップ機能の変更はありません。match 句はポリシー リスト内で permit と deny 文により設定されます。ルート マップはこれを評価して各 match 句を処理し、設定に基づいてルートの許可や拒否を行います。ルート マップ機能での AND および OR セマンティックは、match 句の扱いについてポリシー リストと同様です。

ポリシー リストにより、中規模以上のネットワークでの BGP ルーティング ポリシー設定を簡素化できます。ネットワーク オペレータは、ルート マップ内で一群の match 句を持つ事前に設定されたポリシー リストを参照することで、BGP ルーティング ポリシーへの一般的な変更を簡単に適用することができます。複数のルート マップのエントリに繰り返し現れる一群の match 句を、ネットワーク オペレータがそれぞれ手動で再設定する必要がなくなりました。

自律システム パスとルータの MED が一致するトラフィックをフィルタリングする BGP ポリシー リストを作成し、それからポリシー リストを参照するルート マップを作成するには、次の作業を実行します。

前提条件

ネットワークで BGP ルーティングが設定され、BGP ネイバーが確立されている必要があります。

制約事項

- BGP ルート マップ ポリシー リストは、ポリシー リスト内での IP バージョン 6（IPv6）の match 句の設定をサポートしていません。
- ポリシー リストは、Cisco IOS Release 12.0(22)S および 12.2(15)T よりも前の Cisco IOS ソフトウェアではサポートされていません。古いバージョンの Cisco IOS ソフトウェアを実行中のルータをリロードすると、ルーティング ポリシーの設定の一部が失われることがあります。
- ポリシー リストがサポートするのは match 句だけで、set 句はサポートしていません。ただし、ポリシー リストは、ポリシー リストとは別に設定された match および set 句と、同一のルート マップ エントリ内で共存することができます。
- ポリシー リストは BGP だけでサポートされます。他の IP ルーティング プロトコルではサポートされません。この制限が再配布を含めたルート マップの通常動作を妨げることはありません。ポリシー リスト機能は BGP の中で透過的に動作し、他の IP ルーティング プロトコルからは見ることができないからです。
- ポリシー リストがサポートするのは match 句だけで、set 句はサポートしていません。ただし、ポリシー リストは、ポリシー リストとは別に設定された match および set 句と、同一のルート マップ エントリ内で共存することができます。1 番目のルート マップの例では AND セマンティックを

設定し、2 番目のルート マップ設定例はセマンティックを設定しています。このセクションの例はいずれも、ポリシー リストと個別の match および set 句サンプル ルート マップ設定とを、同じ設定の中で参照するルート マップのサンプルとなっています。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip policy-list** *policy-list-name* {**permit** | **deny**}
4. **match as-path** *as-number*
5. **match metric** *metric*
6. **exit**
7. **route-map** *map-name* [**permit | deny**] [*sequence-number*]
8. **match ip-address** {*access-list-number* | *access-list-name*} [*... access-list-number* | *... access-list-name*]
9. **match policy-list** *policy-list-name*
10. **set community** {*community-number* [**additive**] [*well-known-community*] | **none**}
11. **set local-preference** *preference-value*
12. **end**
13. **show ip policy-list** [*policy-list-name*]
14. **show route-map** [*route-map-name*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	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 3	**ip policy-list** *policy-list-name* {**permit**	**deny**} **例：** `Router(config)# ip policy-list POLICY-LIST-NAME-1 permit`	ポリシー リスト コンフィギュレーション モードを開始し、続く match 句で許容されるルートを許可する BGP ポリシー リストを作成します。
ステップ 4	**match as-path** *as-number* **例：** `Router(config-policy-list)# match as-path 500`	指定した自律システム パスからのルートを許可する match 句を作成します。

	コマンドまたはアクション	目的
ステップ 5	**match metric** *metric* **例：** `Router(config-policy-list)# match metric 10`	指定したメトリックのルートを許可する match 句を作成します。
ステップ 6	**exit** **例：** `Router(config-policy-list)# exit`	ポリシー リスト 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 7	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **例：** `Router(config)# route-map MAP-NAME-1 permit 10`	ルート マップを作成し、ルート マップ コンフィギュレーション モードを開始します。
ステップ 8	**match ip address** {*access-list-number* | *access-list-name*} [... *access-list-number* | ... *access-list-name*] **例：** `Router(config-route-map)# match ip address 1`	指定した *access-list-number* または *access-list-name* 引数に一致するルートを許可する match 句を作成します。
ステップ 9	**match policy-list** *policy-list-name* **例：** `Router(config-route-map)# match policy-list POLICY-LIST-NAME-1`	指定したポリシー リストに一致する句を作成します。 • ポリシー リスト内の match 句すべてが評価され、処理されます。このコマンドで、複数のポリシー リストを参照できます。 • このコマンドはまた、標準の match 句と同様に AND や OR セマンティックをサポートします。
ステップ 10	**set community** *community-number* [**additive**] [*well-known-community*] | **none**} **例：** `Router(config-route-map)# set community 10:1`	指定したコミュニティを設定または削除する句を作成します。
ステップ 11	**set local-preference** *preference-value* **例：** `Router(config-route-map)# set local-preference 140`	指定したローカル プリファレンス値を設定する句を作成します。
ステップ 12	**end** **例：** `Router(config-route-map)# end`	ルート マップ コンフィギュレーション モードを終了して、特権 EXEC モードを開始します。
ステップ 13	**show ip policy-list** [*policy-list-name*] **例：** `Router# show ip policy-list POLICY-LIST-NAME-1`	設定されたポリシー リストとポリシー リスト エントリについての情報を表示します。
ステップ 14	**show route-map** [*route-map-name*] **例：** `Router# show route-map`	ローカルで設定されたルート マップとルート マップ エントリを表示します。

例

次の出力例は、ポリシー リストが作成されたことを確認し、ポリシー リスト名と設定された match 句を表示しています。

```
Router# show ip policy-list POLICY-LIST-NAME-1

policy-list POLICY-LIST-NAME-1 permit
  Match clauses:
    metric 20
    as-path (as-path filter): 1
```

（注） ポリシー リスト名は、**show ip policy-list** コマンドが入力されたときに指定できます。このオプションは、このコマンドの出力をフィルタリングして、1 つのポリシー リストを確認するときに便利です。

次の **show route-map** コマンドの出力例は、ルート マップが作成され、ポリシー リストが参照されたことを確認します。このコマンドの出力は、ルート マップ名と、設定されたルート マップで参照されたポリシー リストとを表示します。

```
Router# show route-map

route-map ROUTE-MAP-NAME-1, deny, sequence 10
  Match clauses:
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes
route-map ROUTE-MAP-NAME-1, permit, sequence 10
  Match clauses:
    IP Policy lists:
        POLICY-LIST-NAME-1
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes
```

BGP ルート マップでの continue 句の使用によるトラフィック フィルタリング

BGP ルート マップで continue 句を使用してトラフィックのフィルタリングを行うには、次の作業を実行します。Cisco IOS Release 12.3(2)T、12.0(24)S、12.2(33)SRB、およびそれ以降のリリースでは、BGP ルート マップ設定に continue 句が導入されています。continue 句によって、ポリシー設定とルート フィルタリングのプログラム性は高まり、正常な match および set 句によってエントリが実行された後に追加のエントリを実行する機能が導入されました。continue 句によって、ネットワーク オペレータはポリシー定義をさらにモジュール化して設定できるようになり、特定のポリシー設定を同じルート マップ内で繰り返す必要がなくなりました。continue 句の導入以前は、ルート マップの設定はリニア的であり、ルート マップのフローを制御することがまったくできませんでした。

Cisco IOS Release 12.0(31)S、12.2(33)SB、12.2(33)SRB、12.2(33)SXI、12.4(4)T、およびそれ以降のリリースでは、アウトバウンド ルート マップで continue 句がサポートされるようになりました。

continue 句を使用しないルート マップの動作

ルート マップは一致が出現するまで match 句を評価します。一致が出現すると、ルート マップは match 句の評価を停止し、設定された順序で set 句の実行を開始します。一致が出現しない場合、ルート マップはマッチングに「失敗」し、ルート マップの次のシーケンス番号を評価します。これをすべての設定されたルート マップ エントリが評価されるか、一致が出現するまで続けます。各ルート マップは、エントリを識別するシーケンス番号でタグ付けされています。ルート マップ エントリは、シーケンス番号が最小のものから評価が始まり、最大のシーケンス番号を持つもので終わります。ルート マップに set 句だけが含まれる場合、set 句は自動的に実行され、ルート マップは他のルート マップ エントリを評価しません。

continue 句を使用したルート マップの動作

continue 句を設定すると、ルート マップは一致が出現した後も、指定されたルート マップ エントリで match 句の評価と実行を続けます。continue 句は、シーケンス番号を指定することで特定のルート マップ エントリに移動する（またはジャンプする）よう設定できます。シーケンス番号が指定されていない場合、continue 句は次のシーケンス番号へ移動します。この動作は「黙示的継続」と呼ばれます。match 句がある場合、continue 句は一致が出現した場合にだけ実行されます。一致が出現しなかった場合、continue 句は無視されます。

continue 句を使用した match 動作

match 句がルート マップ エントリに存在しないのに continue 句が存在する場合、continue 句は自動的に実行され、指定されたルート マップ エントリへ移動します。ルート マップ エントリに match 句が存在する場合、continue 句は一致が出現した場合にだけ実行されます。一致が出現し、かつ continue 句が存在する場合、ルート マップは set 句を実行し、それから指定されたルート マップ エントリへ移動します。その次のルート マップ エントリに continue が含まれている場合、ルート マップは一致が出現すればその continue 句を実行します。continue 句がその次のルート マップ エントリに存在しない場合、ルート マップは通常どおり評価されます。continue 句がその次のルート マップ エントリに存在するが一致が出現しない場合、ルート マップは継続せずに「失敗」し、その次のシーケンス番号が存在すればそこへ移動します。

continue 句を使用した Set 動作

set 句は、match 句の評価中は残しておかれ、ルート マップ評価が完了した後に実行されます。set 句は、設定された順番に評価され、処理されます。ルート マップに match 句が存在しない場合を除き、set 句は一致が出現した後にだけ実行されます。continue 文は、設定された set アクションが実行された後にだけ、指定のルート マップ エントリへと進みます。set アクションが最初のルート マップで発生し、それから後続のルート マップ エントリにおいて再び同じ set アクションが異なる値で発生した場合、同じ **set** コマンドで設定された set アクションは、**set** コマンドが複数の値を許可する場合を除き、最後の set アクションによってそれ以前のものが上書きされます。たとえば、**set as-path prepend** コマンドは複数の自律システム番号の設定を許可しています。

（注） ルート マップ エントリに match 句が含まれない場合、continue 句は一致の出現なしで実行できます。

（注） ルート マップはリニア動作であり、入れ子動作ではありません。あるルートがいったん continue コマンド句を伴ったルート マップ許可エントリで一致すると、ルート マップ末尾の黙示的拒否により処理されません。例として、「BGP ルート マップでの continue 句の使用によるトラフィック フィルタリング：例」（P.77）を参照してください。

制約事項

- アウトバウンド ルート マップの continue 句は、Cisco IOS Release 12.0(31)S、12.2(33)SB、12.2(33)SRB、12.2(33)SXI、12.4(4)T、およびそれ以降のリリースだけでサポートされています。
- continue 句ではより大きな値のエントリ（シーケンス番号が自身より大きいルート マップ エントリ）にだけ移動できます。小さな値のルート マップ エントリには移動できません。

手順の概要

1. **enable**
2. **configure terminal**

3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
5. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
6. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
7. **exit**
8. **exit**
9. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
10. **match ip-address** {*access-list-number* | *access-list-name*} [*... access-list-number* | *... access-list-name*]
11. **set community** {*community-number* [**additive**] [*well-known-community*] | **none**}
12. **continue** [*sequence-number*]
13. **end**
14. **show route-map** [*map-name*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードなどの上位の特権レベルをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** **例：** `Router# configure terminal`	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 3	**router bgp** *autonomous-system-number* **例：** `Router(config)# router bgp 50000`	ルータ コンフィギュレーション モードを開始して、BGP ルーティング プロセスを作成します。
ステップ 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **例：** `Router(config-router)# neighbor 10.0.0.1 remote-as 50000`	指定された自律システム内のネイバーの IP アドレスまたはピア グループ名を、ローカル ルータの IPv4 マルチプロトコル BGP ネイバー テーブルに追加します。

	コマンドまたはアクション	目的
ステップ 5	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **例：** `Router(config-router)# address-family ipv4 unicast`	IPv4 アドレス ファミリを指定し、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。 • **unicast** キーワードは、IPv4 ユニキャスト アドレス ファミリを指定します。デフォルトでは、**address-family ipv4** コマンドに **unicast** キーワードが指定されていない場合、ルータは IPv4 ユニキャスト アドレス ファミリのアドレス ファミリ コンフィギュレーション モードになります。 • **multicast** キーワードは、IPv4 マルチキャスト アドレス プレフィクスを指定します。 **vrf** キーワードと *vrf-name* 引数は、後続の IPv4 アドレス ファミリ コンフィギュレーション モード コマンドに関連付ける VRF インスタンスの名前を指定します。
ステップ 6	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**} **例：** `Router(config-router-af)# neighbor 10.0.0.1 route-map ROUTE-MAP-NAME in`	インバウンド ルート マップを指定されたネイバーから受信したルートに適用します。もしくは、アウトバウンド ルート マップを指定されたネイバーへアドバタイズされたルートへ適用します。
ステップ 7	**exit** **例：** `Router(config-router-af)# exit`	アドレス ファミリ コンフィギュレーション モードを終了して、ルータ 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：** `Router(config-router)# exit`	ルータ コンフィギュレーション モードを終了して、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 9	**route-map** *map-name* {**permit** | **deny**} [*sequence-number*] **例：** `Router(config)# route-map ROUTE-MAP-NAME permit 10`	ルート マップ コンフィギュレーション モードを開始し、ルート マップを作成または設定します。
ステップ 10	**match ip address** {*access-list-number* | *access-list-name*} [... *access-list-number* | ... *access-list-name*] **例：** `Router(config-route-map)# match ip address 1`	ポリシー ルーティングとルート フィルタリングが発生する条件を指定する **match** コマンドを設定します。 • 複数の **match** コマンドを設定できます。**match** コマンドが設定された場合、continue 文が実行されるには一致の発生が必要になります。**match** コマンドが設定されない場合、set および continue 句は実行されます。 **（注）** この作業で使用する **match** コマンドおよび **set** コマンドは、**continue** コマンドの動作を記述するための例です。具体的な **match** コマンドおよび **set** コマンドのリストについては、『*Cisco IOS IP Routing: BGP Command Reference*』の **continue** コマンドを参照してください。

	コマンドまたはアクション	目的
ステップ 11	**set community** *community-number* [**additive**] [*well-known-community*] | **none**} **例：** `Router(config-route-map)# set community 10:1`	**set** コマンドを設定して、**match** コマンドで適用された条件が満たされた場合のルーティング アクションを指定します。 • 複数の **set** コマンドを設定できます。 • この例では、指定したコミュニティをセットする句が作成されます。
ステップ 12	**continue** [*sequence-number*] **例：** `Router(config-route-map)# continue`	一致が出現した後も match 文の評価と実行を継続するよう、ルート マップを設定します。 • シーケンス番号が指定された場合、continue 句は指定されたシーケンス番号のルート マップへ移動します。 • シーケンス番号が指定されない場合、continue 句はその次のシーケンス番号のルート マップへ移動します。この動作は、「黙示的継続」と呼ばれます。 **（注）** アウトバウンド ルート マップの continue 句は、Cisco IOS Release 12.0(31)S、12.2(33)SB、12.2(33)SRB、12.2(33)SXI、12.4(4)T、およびそれ以降のリリースだけでサポートされています。
ステップ 13	**end** **例：** `Router(config-route-map)# end`	ルート マップ コンフィギュレーション モードを終了して、特権 EXEC モードを開始します。
ステップ 14	**show route-map** [*map-name*] **例：** `Router# show route-map`	（任意）ローカルで設定されたルート マップを表示します。出力をフィルタリングするためのルート マップ名は、このコマンドの構文内で指定できます。

例

次に、**show route-map** コマンドを使用して continue 句の設定を確認する方法の出力例を示します。設定されたルート マップが、match、set、および continue 句を含め、出力に表示されます。

```
Router# show route-map

route-map MARKETING, permit, sequence 10
  Match clauses:
    ip address (access-lists): 1
    metric 10
  Continue: sequence 40
  Set clauses:
    as-path prepend 10
  Policy routing matches: 0 packets, 0 bytes
route-map MARKETING, permit, sequence 20
  Match clauses:
    ip address (access-lists): 2
    metric 20
  Set clauses:
    as-path prepend 10 10
  Policy routing matches: 0 packets, 0 bytes
route-map MARKETING, permit, sequence 30
  Match clauses:
  Continue: to next entry 40
  Set clauses:
    as-path prepend 10 10 10
  Policy routing matches: 0 packets, 0 bytes
```

```
route-map MARKETING, permit, sequence 40
  Match clauses:
    community (community-list filter): 10:1
  Set clauses:
    local-preference 104
  Policy routing matches: 0 packets, 0 bytes
route-map MKTG-POLICY-MAP, permit, sequence 10
  Match clauses:
  Set clauses:
    community 655370
  Policy routing matches: 0 packets, 0 bytes
```

外部 BGP を使用したサービス プロバイダーとの接続の設定例

ここでは、次の例について説明します。

インバウンド パス選択の変更：例

次に、ルート マップを使用してネイバーからの受信データを変更する方法の例を示します。10.222.1.1 から受信した、自律システム アクセス リスト 200 で設定されたフィルタ パラメータに一致するルートはどれも、その weight は 200 に、ローカル プリファレンスは 250 に設定され、それが受け入れられることになります。

```
router bgp 100
!
 neighbor 10.222.1.1 route-map FIX-WEIGHT in
 neighbor 10.222.1.1 remote-as 1
!
ip as-path access-list 200 permit ^690$
ip as-path access-list 200 permit ^1800
!
route-map FIX-WEIGHT permit 10
 match as-path 200
 set local-preference 250
 set weight 200
```

次の例では、finance という名前のルート マップが、自律システム 690 で生成されたパスすべてを、MED メトリック アトリビュート 127 でマークしています。2 番目の permit 句は、自律システム パス リスト 1 に一致しないルートを引き続きネイバー 10.1.1.1 へ送るために必要です。

```
router bgp 65000
 neighbor 10.1.1.1 route-map finance out
!
ip as-path access-list 1 permit ^690_
ip as-path access-list 2 permit .*
!
route-map finance permit 10
 match as-path 1
 set metric 127
!
route-map finance permit 20
 match as-path 2
```

インバウンド ルート マップはプレフィクスベースのマッチングを行って、アップデートのさまざまなパラメータを設定できます。自律システム パスとコミュニティ リスト マッチングに加え、インバウンド プレフィクス マッチングが利用できます。次に、**set local-preference** ルート マップ コンフィギュレーション コマンドでどのようにインバウンド プレフィクス 172.20.0.0/16 のローカル プリファレンスを 120 に設定するかを例に示します。

```
!
router bgp 65100
 network 10.108.0.0
 neighbor 10.108.1.1 remote-as 65200
 neighbor 10.108.1.1 route-map set-local-pref in
!
route-map set-local-pref permit 10
 match ip address 2
 set local preference 120
!
route-map set-local-pref permit 20
!
access-list 2 permit 172.20.0.0 0.0.255.255
access-list 2 deny any
```

4 バイト自律システム番号を使用した AS-path アトリビュートの変更によるインバウンド パス選択の変更：例

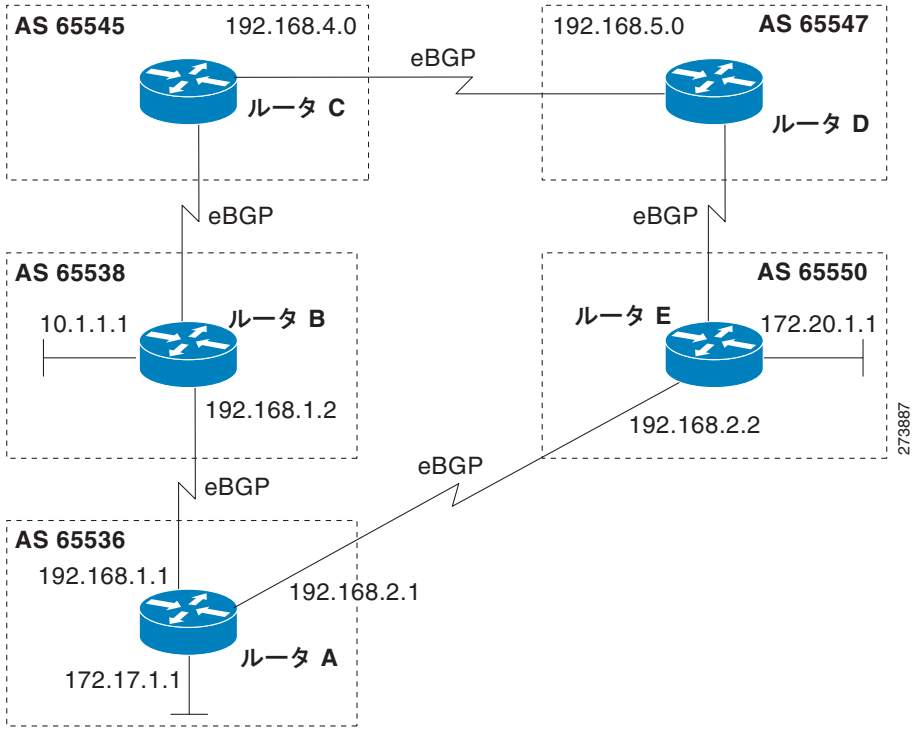
この例は、AS-path アトリビュートの変更によって 172.17.1.0 宛てトラフィックのインバウンド パス選択を変化させるために BGP を設定する方法を示します。Cisco IOS Release 12.0(32)SY8、12.0(33)S3、12.2(33)SXI1、およびそれ以降のリリースで、BGP は 4 オクテット（4 バイト）自律システム番号をサポートするようになりました。この例の 4 バイト自律システム番号は、デフォルトの asplain（10 進数）形式です。たとえば、図 8（P.49）において、ルータ B は自律システム番号 65538 にあります。4 バイト自律システム番号についてのさらに詳しい紹介は、「BGP 自律システム番号の形式」（P.4）を参照してください。

AS-path アトリビュートの変更は、別の自律システムのパス選択を変化させるために BGP で使用可能な方法の 1 つです。たとえば、図 9 において、ルータ A は自身のネットワーク 172.17.1.0 を、自律システム 65538 および自律システム 65550 にある BGP ピアにアドバタイズします。ルーティング情報が自律システム 65545 に伝播されるとき、自律システム 65545 内のルータは、2 つの異なるルートからのネットワーク 172.17.1.0 の到達可能性情報を持つことになります。1 番目のルートは、65538 と 65536 で構成される AS-path を備えた自律システム 65538 によるものです。2 番目のルートは自律システム 65547 を経由するもので、AS-path は 65547、65550、65536 です。他の BGP アトリビュートが

すべて同じだとすれば、自律システム 65545 内のルータ C はネットワーク 172.17.1.0 へのトラフィックのルートとして、自律システム 65538 を通るルートを選択します。通過した自律システムという点では最短ルートとなるからです。

自律システム 65536 は自律システム 65545 のネットワーク 172.17.1.0 へのトラフィックすべてを自律システム 65538 のルータ B 経由で受信するようになります。しかし、自律システム 65538 と自律システム 65536 の間のリンクが非常に遅く輻輳している場合、**set as-path prepend** コマンドをルータ A で使用して、自律システム 65538 経由のルートが自律システム 65550 経由のパスよりも遠いように見せることで、172.17.1.0 ネットワークへのインバウンド パス選択を変化させることができます。図 9 のルータ A の設定は、アウトバウンド BGP アップデートをルータ B に適用することで完了します。**set as-path prepend** コマンドの使用により、ルータ A からルータ B へのアウトバウンド BGP アップデートはすべて、ローカル自律システム番号 65536 を 2 回追加するよう変更された AS-path アトリビュートを持つようになります。この設定の後、自律システム 65545 は 172.17.1.0 ネットワークについてのアップデートを、自律システム 65538 経由で受け取ることになります。新しい AS-path は 65538、65536、65536、65536 となり、これは自律システム 65547 からの AS-path（65547、65550、65536 で変更なし）よりも長くなります。自律システム 65545 内のネットワーキング デバイスは、172.17.1.0 ネットワーク内の宛先アドレスを持つパケットを転送するときに、自律システム 65547 経由のルートを優先するようになります。

図 9　　AS-path アトリビュート変更のネットワーク トポロジ



この例の設定は、図 9 のルータ A で実行されます。

```
router bgp 65536
 address-family ipv4 unicast
  network 172.17.1.0 mask 255.255.255.0
  neighbor 192.168.1.2 remote-as 65538
  neighbor 192.168.1.2 activate
  neighbor 192.168.1.2 route-map PREPEND out
  exit-address-family
 exit
route-map PREPEND permit 10
```

```
set as-path prepend 65536 65536
```

アウトバウンド パス選択の変更：例

次に、アウトバウンド ルート フィルタを作成し、ルータ A（10.1.1.1）がルータ B（172.16.1.2）へフィルタをアドバタイズするよう設定する例を示します。FILTER という名前の IP プレフィクスが作成され、サブネット 192.168.1.0/24 をアウトバウンド ルート フィルタリングに指定します。ルータ A がアウトバウンド ルート フィルタをルータ B へアドバタイズできるよう、ORF 送信機能がルータ A で設定されます。

ルータ A 設定（送信側）

```
ip prefix-list FILTER seq 10 permit 192.168.1.0/24
!
router bgp 65100
 address-family ipv4 unicast
 neighbor 172.16.1.2 remote-as 65200
 neighbor 172.16.1.2 ebgp-multihop
 neighbor 172.16.1.2 capability orf prefix-list send
 neighbor 172.16.1.2 prefix-list FILTER in
 end
```

ルータ B 設定（受信側）

次に、ORF 受信機能をルータ A へアドバタイズするようにルータ B を設定する例を示します。ORF 機能が交換された後、ルータ B は FILTER プレフィクス リストで定義されたアウトバウンド ルート フィルタをインストールします。アウトバウンド ルート フィルタをアクティブ化するため、この設定の最後にルータ B でインバウンド ソフト リセットが開始されます。

```
router bgp 65200
 address-family ipv4 unicast
 neighbor 10.1.1.1 remote-as 65100
 neighbor 10.1.1.1 ebgp-multihop 255
 neighbor 10.1.1.1 capability orf prefix-list receive
 end
clear ip bgp 10.1.1.1 in prefix-filter
```

次の例は、set-as-path という名前のルート マップがどのようにネイバー 10.69.232.70 へのアウトバウンド アップデートに適用されるかを示します。ルート マップは自律システム パス「65100 65100」を、アクセス リスト 1 を渡すルートにプリペンドします。ルート マップの 2 番目の部分は、他のルータへのアドバタイズを許可するためのものです。

```
router bgp 65100
 network 172.16.0.0
 network 172.17.0.0
 neighbor 10.69.232.70 remote-as 65200
 neighbor 10.69.232.70 route-map set-as-path out
!
route-map set-as-path 10 permit
 match address 1
 set as-path prepend 65100 65100
!
route-map set-as-path 20 permit
 match address 2
!
access-list 1 permit 172.16.0.0 0.0.255.255
access-list 1 permit 172.17.0.0 0.0.255.255
!
access-list 2 permit 0.0.0.0 255.255.255.255
```

プレフィクス リストによる BGP プレフィクスのフィルタリング：例

ここでは、次の例について説明します。

シングル プレフィクス リストを使用した BGP プレフィクスのフィルタリング

次に、プレフィクス リストでデフォルト ルート 0.0.0.0/0 を拒否する例を示します。

```
ip prefix-list abc deny 0.0.0.0/0
```

次に、プレフィクス リストでプレフィクス 10.0.0.0/8 に一致するルートを許可する例を示します。

```
ip prefix-list abc permit 10.0.0.0/8
```

次の例に、プレフィクス長が /8 ～ /24 のプレフィクスだけを受け入れるように BGP プロセスを設定する方法を示します。

```
router bgp 40000
 network 10.20.20.0
 distribute-list prefix max24 in
!
ip prefix-list max24 seq 5 permit 0.0.0.0/0 ge 8 le 24
```

次に、プレフィクス 10.1.1.0/24 がルーティング テーブルに存在する場合に、条件付きでデフォルト ルート（0.0.0.0/0）を Routing Information Protocol（RIP）に生成する設定例を示します。

```
ip prefix-list cond permit 10.1.1.0/24
!
route-map default-condition permit 10
 match ip address prefix-list cond
!
router rip
 default-information originate route-map default-condition
```

次の例に、プレフィクスの長さによるフィルタリングに加え、192.168.1.1 からのルーティング アップデートだけを受け入れるよう BGP を設定する方法を示します。

```
router bgp 40000
 distribute-list prefix max24 gateway allowlist in
!
ip prefix-list allowlist seq 5 permit 192.168.1.1/32
!
```

次に、*name1* を使用してプレフィクスへの受信アップデートをフィルタリングし、アップデートされているプレフィクスのゲートウェイ（ネクストホップ）をプレフィクス リスト *name2* へマッチングするよう、イーサネット インターフェイス 0 上で BGP プロセスに指示する例を示します。

```
router bgp 103
 distribute-list prefix name1 gateway name2 in ethernet 0
```

プレフィクスのグループを使用した BGP プレフィクスのフィルタリング

次に、ネットワーク 192/8 でプレフィクス長が 24 以下のルートを許可するよう BGP を設定する例を示します。

```
ip prefix-list abc permit 192.0.0.0/8 le 24
```

次に、192/8 でプレフィクス長が 25 より大きいルートを拒否するよう BGP を設定する例を示します。

```
ip prefix-list abc deny 192.0.0.0/8 ge 25
```

次に、すべてのアドレス空間でプレフィクス長が 8 より大きく 24 より小さいルートを許可するよう BGP を設定する例を示します。

```
ip prefix-list abc permit 0.0.0.0/0 ge 8 le 24
```

次に、すべてのアドレス空間でプレフィクス長が 25 より大きいルートを拒否するよう BGP を設定する例を示します。

```
ip prefix-list abc deny 0.0.0.0/0 ge 25
```

次に、ネットワーク 10/8 のルートをすべて拒否するよう BGP を設定する例を示します。これは、クラス A ネットワーク 10.0.0.0/8 内のルートのマスクが 32 ビット以下である場合、そのルートが拒否されるためです。

```
ip prefix-list abc deny 10.0.0.0/8 le 32
```

次に、192.168.1.0/24 でマスクが 25 より大きいルートを拒否するよう BGP を設定する例を示します。

```
ip prefix-list abc deny 192.168.1.0/24 ge 25
```

次に、すべてのルートを許可するよう BGP を設定する例を示します。

```
ip prefix-list abc permit 0.0.0.0/0 le 32
```

プレフィクス リスト エントリの追加と削除

プレフィクス リストの初期設定が次のようになっている場合、プレフィクス リスト内のエントリを個別に追加、削除できます。

```
ip prefix-list abc deny 0.0.0.0/0 le 7
ip prefix-list abc deny 0.0.0.0/0 ge 25
ip prefix-list abc permit 192.168.0.0/15
```

次に、プレフィクス リストからエントリを削除して 192.168.0.0 を許可しないようにし、10.0.0.0/8 を許可する新しいエントリを追加する例を示します。

```
no ip prefix-list abc permit 192.168.0.0/15
ip prefix-list abc permit 10.0.0.0/8
```

新しい設定は次のようになります。

```
ip prefix-list abc deny 0.0.0.0/0 le 7
ip prefix-list abc deny 0.0.0.0/0 ge 25
ip prefix-list abc permit 10.0.0.0/8
```

コミュニティ リストを使用したトラフィック フィルタリング：例

このセクションでは、BGP コミュニティをルート マップと使用した 2 つの例を示します。

1 番目の例は、set-community というルート マップがネイバー 172.16.232.50 のアウトバウンド アップデートにどのように適用されるかを示します。アクセス リスト 1 を渡すルートは、特別なコミュニティ アトリビュート値 no-export を持っています。残りのルートは通常どおりアドバタイズされます。この特別なコミュニティ値は、自律システム 200 内の BGP スピーカーがそれらのルートのアドバタイズメントを行うのを自動的に防止します。

```
router bgp 100
 neighbor 172.16.232.50 remote-as 200
 neighbor 172.16.232.50 send-community
 neighbor 172.16.232.50 route-map set-community out
!
route-map set-community permit 10
 match address 1
 set community no-export
!
route-map set-community permit 20
 match address 2
```

2 番目の例は、*set-community* というルート マップがネイバー 172.16.232.90 のアウトバウンド アップデートにどのように適用されるかを示します。自律システム 70 で生成されるルートはすべて、コミュニティ値 200 200 を自身の既存の値に追加します。他のルートはすべて、通常と同じようにアドバタイズされます。

```
route-map bgp 200
 neighbor 172.16.232.90 remote-as 100
 neighbor 172.16.232.90 send-community
 neighbor 172.16.232.90 route-map set-community out
!
route-map set-community permit 10
 match as-path 1
 set community 200 200 additive
!
route-map set-community permit 20
!
ip as-path access-list 1 permit 70$
ip as-path access-list 2 permit .*
```

AS-path フィルタを使用したトラフィック フィルタリング：例

次に、ネイバーによる BGP パス フィルタリングの例を示します。自律システム パス access list 2 を通過するルートだけが 192.168.12.10 に送られます。同様に、access list 3 を通過するルートだけが 192.168.12.10 から受け入れられます。

```
router bgp 200
 neighbor 192.168.12.10 remote-as 100
 neighbor 192.168.12.10 filter-list 1 out
 neighbor 192.168.12.10 filter-list 2 in
 exit
ip as-path access-list 1 permit _109_
ip as-path access-list 2 permit _200$
ip as-path access-list 2 permit ^100$
ip as-path access-list 3 deny _690$
ip as-path access-list 3 permit .*
```

4 バイト自律システム番号を使用した AS-path フィルタによるトラフィック フィルタリング：例

Cisco IOS Release 12.0(32)SY8、12.0(33)S3、12.2(33)SRE、12.2(33)SXI1、およびそれ以降のリリースにおける asplain デフォルト形式

次の例は Cisco IOS Release 12.0(32)SY8、12.0(33)S3、12.2(33)SRE、12.2(33)XNE、12.2(33)SXI1、およびそれ以降のリリースで使用できるもので、4 バイト自律システム番号を asplain 形式で使用し、ネイバーによる BGP パス フィルタリングを行います。自律システム パス access list 2 を通過するルートだけが 192.168.3.2 に送られます。

```
ip as-path access-list 2 permit ^65536$
router bgp 65538
 address-family ipv4 unicast
  neighbor 192.168.3.2 remote-as 65550
  neighbor 192.168.3.2 activate
  neighbor 192.168.3.2 filter-list 2 in
  end
```

Cisco IOS Release 12.0(32)S12 および 12.4(24)T における asdot デフォルト形式

次の例は Cisco IOS Release 12.0(32)S12、12.4(24)T、およびそれ以降のリリースで使用できるもので、4 バイト自律システム番号を asdot 形式で使用し、ネイバーによる BGP パス フィルタリングを行います。自律システム パス access list 2 を通過するルートだけが 192.168.3.2 に送られます。

（注） Cisco IOS Release 12.0(32)SY8、12.0(33)S3、12.2(33)SRE、12.2(33)XNE、12.2(33)SXI1、およびそれ以降のリリースでは、この例が正常に動作するのは、**bgp asnotation dot** コマンドを使用して、asdot をデフォルトの表示形式として設定した場合だけです。

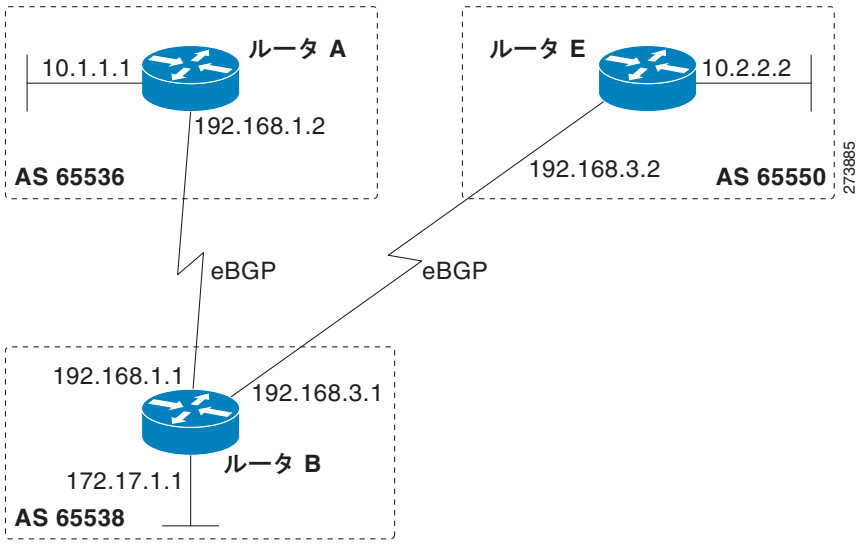
```
ip as-path access-list 2 permit ^1\.0$
router bgp 1.2
 address-family ipv4 unicast
  neighbor 192.168.3.2 remote-as 1.14
  neighbor 192.168.3.2 filter-list 2 in
  end
```

4 バイト自律システム番号と拡張コミュニティ リストを使用したトラフィック フィルタリング：例

Cisco IOS Release 12.0(32)SY8、12.0(33)S3、12.2(33)SRE、12.2(33)SXI1、およびそれ以降のリリースにおける asplain デフォルト形式

次に、アウトバウンド ルートを制御するために拡張 BGP コミュニティ リストを作成することによるトラフィック フィルタリングの例を示します。Cisco IOS Release 12.0(32)SY8、12.0(33)S3、12.2(33)SRE、12.2(33)XNE、12.2(33)SXI1、およびそれ以降のリリースでは、拡張 BGP コミュニティはデフォルトで asplain の正規表現中の 4 バイト自律システム番号をサポートしています。拡張コミュニティ リストは、VRF インスタンスと MPLS VPN のルートを設定し、フィルタリングし、識別するために使用されます。名前付きまたは番号付きコミュニティ リストの設定には、**ip extcommunity-list** コマンドを使用します。アクセス リストの標準ルールすべてが、拡張コミュニティ リストの設定に適用されます。正規表現は、拡張コミュニティ リスト番号の拡張範囲によってサポートされています。

図 10　asplain 形式の 4 バイト自律システム番号と拡張コミュニティ リストを使用したトラフィック フィルタリングの BGP トポロジ



（注）　拡張コミュニティ リストのエントリにはすべて、コンフィギュレーション モードにかかわらずデフォルトでシーケンス番号が適用されます。拡張コミュニティ リスト エントリのシーケンス番号の明示的な割り当てと再割り当ては、IP 拡張コミュニティリスト コンフィギュレーション モードだけで設定でき、グローバル コンフィギュレーション モードでは設定できません。

この例では、図 10 のルータ B は、拡張名前付きコミュニティ リストを使用して設定され、192.168.1.2 の BGP ピアが 4 バイト自律システム 65550 からの、または 65550 経由のパスについてのアドバタイズメントも送られないよう指定されます。IP 拡張コミュニティリスト コンフィギュレーション モードが使用され、エントリのシーケンス番号再割り当て機能が示されます。

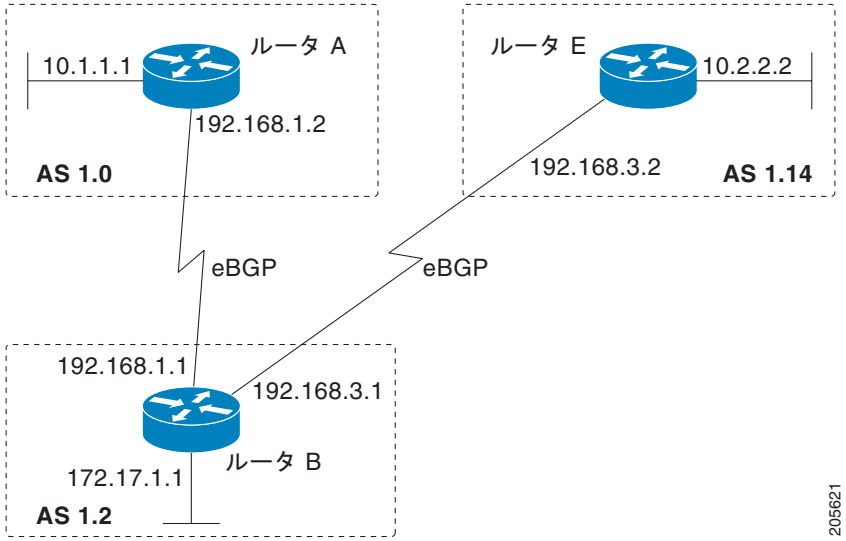
```
ip extcommunity-list expanded DENY65550
 10 deny _65550_
 20 deny ^65550 .*
 resequence 50 100
 exit
router bgp 65538
 network 172.17.1.0 mask 255.255.255.0
 address-family ipv4 unicast
  neighbor 192.168.3.2 remote-as 65550
  neighbor 192.168.1.2 remote-as 65536
  neighbor 192.168.3.2 activate
  neighbor 192.168.1.2 activate
  end
show ip extcommunity-list DENY65550
```

Cisco IOS Release 12.0(32)S12 および 12.4(24)T における asdot デフォルト形式

次に、アウトバウンド ルートを制御するために拡張 BGP コミュニティ リストを作成することによるトラフィック フィルタリングの例を示します。Cisco IOS Release 12.0(32)S12、12.4(24)T、およびそれ以降のリリースでは、拡張 BGP コミュニティは正規表現中の 4 バイト自律システム番号を asdot 形式だけでサポートします。拡張コミュニティ リストは、VRF インスタンスと MPLS VPN のルートを設定し、フィルタリングし、識別するために使用されます。名前付きまたは番号付きコミュニティ リストの設定には、**ip extcommunity-list** コマンドを使用します。アクセス リストの標準ルールすべてが、拡張コミュニティ リストの設定に適用されます。正規表現は、拡張コミュニティ リスト番号の拡張範囲によってサポートされています。

（注） Cisco IOS Release 12.0(32)SY8、12.0(33)S3、12.2(33)SXI1、およびそれ以降のリリースでは、この例が正常に動作するのは、**bgp asnotation dot** コマンドを使用して、asdot をデフォルトの表示形式として設定した場合だけです。

図 11　asdot 形式の 4 バイト自律システム番号と拡張コミュニティ リストを使用したトラフィック フィルタリングの BGP トポロジ



（注） 拡張コミュニティ リストのエントリにはすべて、コンフィギュレーション モードにかかわらずデフォルトでシーケンス番号が適用されます。拡張コミュニティ リスト エントリのシーケンス番号の明示的な割り当てと再割り当ては、IP 拡張コミュニティリスト コンフィギュレーション モードだけで設定でき、グローバル コンフィギュレーション モードでは設定できません。

この例では、図 11 のルータ B は、拡張名前付きコミュニティ リストを使用して設定され、192.168.1.2 の BGP ピアが 4 バイト自律システム 65550 からの、または 65550 経由のパスについてのアドバタイズメントも送られないよう指定されます。IP 拡張コミュニティリスト コンフィギュレーション モードが使用され、エントリのシーケンス番号再割り当て機能が示されます。

```
ip extcommunity-list expanded DENY114
 10 deny _1\.14_
 20 deny ^1\.14 .*
 resequence 50 100
 exit
router bgp 1.2
 network 172.17.1.0 mask 255.255.255.0
 address-family ipv4 unicast
  neighbor 192.168.3.2 remote-as 1.14
  neighbor 192.168.1.2 remote-as 1.0
  neighbor 192.168.3.2 activate
  neighbor 192.168.1.2 activate
  end
show ip extcommunity-list DENY114
```

BGP ルート マップを使用したトラフィック フィルタリング：例

次に、アクセス リスト 1 に一致している場合、ネイバー 10.1.1.1 からのユニキャストおよびマルチキャスト ルートを受け入れるように、アドレス ファミリを使用して BGP を設定する例を示します。

```
route-map filter-some-multicast
 match ip address 1
 exit
router bgp 65538
 neighbor 10.1.1.1 remote-as 65537
 address-family ipv4 unicast
  neighbor 10.1.1.1 activate
  neighbor 10.1.1.1 route-map filter-some-multicast in
  exit
 exit
router bgp 65538
 neighbor 10.1.1.1 remote-as 65537
 address-family ipv4 multicast
  neighbor 10.1.1.1 activate
  neighbor 10.1.1.1 route-map filter-some-multicast in
  end
```

BGP ルート マップでの continue 句の使用によるトラフィック フィルタリング：例

次に、ルート マップ シーケンスでの continue 句設定の例を示します。

（注） アウトバウンド ルート マップの continue 句は、Cisco IOS Release 12.0(31)S、12.2(33)SB、12.2(33)SRB、12.2(33)SXI、12.4(4)T、およびそれ以降のリリースだけでサポートされています。

ルート マップ エントリ 10 にある 1 番目の continue 句は、一致が出現した場合にルート マップがエントリ 30 に移動することを示します。一致が出現しなければ、ルート マップは「失敗」してエントリ 20 へ移動します。ルート マップ エントリ 20 で一致が出現すると、set アクションが実行され、ルート マップはそれ以上どのルート マップ エントリも評価しません。最初に一致した IP アドレスだけをサポートします。

ルート マップ エントリ 20 で一致が出現しない場合、ルート マップはマッチングに「失敗」してルート マップ エントリ 30 へ移動します。このシーケンスには match 句が含まれていないため、set 句は自動的に実行され、continue 句にはシーケンス番号が指定されていないため、その次のルート マップ エントリへ移動することになります。

一致が出現しない場合、ルート マップはマッチングに「失敗」してエントリ 30 へ移動し、set 句を実行します。continue 句にはシーケンス番号が指定されていないため、ルート マップ エントリ 40 が評価されることになります。

後続の continue 句エントリで、同じ **set** コマンドが繰り返される場合、2 種類の動作が考えられます。値の加算や累積を設定する **set** コマンド（**set community additive**、**set extended community additive**、**set as-path prepend** など）では、後続のエントリによって後続の値が加算されます。次に、この動作の例を示します。match 句の各セットの後に、as-path に自律システム番号を追加するため **set as-path prepend** コマンドが設定されています。一致が出現すると、ルート マップは match 句の評価を停止し、設定された順序で set 句の実行を開始します。一致が何度出現するかに応じて、as-path には 1 つ、2 つ、または 3 つの自律システム番号がプリペンドされます。

```
route-map ROUTE-MAP-NAME permit 10
 match ip address 1
 match metric 10
```

```
 set as-path prepend 10
 continue 30
 !
route-map ROUTE-MAP-NAME permit 20
 match ip address 2
 match metric 20
 set as-path prepend 10 10
 !
route-map ROUTE-MAP-NAME permit 30
 set as-path prepend 10 10 10
 continue
 !
route-map ROUTE-MAP-NAME permit 40
 match community 10:1
 set local-preference 104
```

この例では、同じ **set** コマンドが後続の continue 句エントリで繰り返されますが、動作は 1 番目の例と異なります。絶対値を設定する **set** コマンドの場合、最後のインスタンスの値がそれ以前の値を上書きします。次に、この動作の例を示します。シーケンス 20 の set 句の値が、シーケンス 10 の set 句の値を上書きします。ネットワーク 172.16/16 からのプレフィクスのネクストホップは 10.2.2.2 に設定され、10.1.1.1 にはなりません。

```
ip prefix-list 1 permit 172.16.0.0/16
ip prefix-list 2 permit 192.168.1.0/24
route-map RED permit 10
 match ip address prefix-list 1
 set ip next hop 10.1.1.1
 continue 20
 exit
route-map RED permit 20
 match ip address prefix-list 2
 set ip next hop 10.2.2.2
 end
```

（注） ルート マップはリニア動作であり、入れ子動作ではありません。あるルートがいったん continue コマンド句を伴ったルート マップ許可エントリで一致すると、ルート マップ末尾の黙示的拒否により処理されません。次に、この場合の例を示します。

次の例では、ルートの as-path が 10、20、または 30 に一致する場合、ルートは許可され、continue 句は明示的 deny 句をジャンプして IP アドレス プレフィクス リストのマッチング処理へ移動します。一致が出現すると、ルート メトリックが 100 に設定されます。as-path が 10、20、または 30 に一致せず、かつコミュニティ番号が 30 に一致するルートだけが拒否されます。他のルータを拒否するには、明示的 deny 文を設定する必要があります。

```
route-map test permit 10
 match as-path 10 20 30
 continue 30
 exit
route-map test deny 20
 match community 30
 exit
route-map test permit 30
 match ip address prefix-list 1
 set metric 100
 exit
```

次の作業

- BGP の拡張機能の設定を行う場合、「Configuring Advanced BGP Features」モジュールに進みます。
- BGP ネイバー セッションのオプションを設定するには、「Configuring BGP Neighbor Session Options」モジュールに進みます。
- 内部 BGP の設定を行う場合、「Configuring Internal BGP Features」モジュールに進みます。

参考資料

次のセクションでは、外部 BGP を使用したサービス プロバイダーとの接続に関連した参考資料を紹介します。

関連資料

関連項目	参照先
Cisco IOS コマンド	『*Cisco IOS Master Commands List, All Releases*』
BGP コマンド：コマンド構文の詳細、コマンド モード、デフォルト、コマンド履歴、使用上の注意事項、および例	『*Cisco IOS IP Routing: BGP Command Reference*』
BGP の概要	「Cisco BGP Overview」モジュール
BGP 基本作業の設定	「Configuring a Basic BGP Network」モジュール
BGP の基礎と説明	『*Large-Scale IP Network Solutions*』Khalid Raza、Mark Turner（Cisco Press, 2000）
拡張可能なネットワークへの BGP の実装と制御	『*Building Scalable Cisco Networks*』Catherine Paquet、Diane Teare（Cisco Press, 2001）
ドメイン間ルーティングの基本	『*Internet Routing Architectures*』Bassam Halabi（Cisco Press, 1997）

規格

規格	タイトル
MDT SAFI	MDT SAFI

MIB

MIB	MIB リンク
CISCO-BGP4-MIB	選択したプラットフォーム、Cisco IOS リリース、および機能セットの MIB の場所を検索しダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

RFC

RFC	タイトル
RFC 1772	『*Application of the Border Gateway Protocol in the Internet*』
RFC 1773	『*Experience with the BGP Protocol*』
RFC 1774	『*BGP-4 Protocol Analysis*』
RFC 1930	『*Guidelines for Creation, Selection, and Registration of an Autonomous System (AS)*』
RFC 2519	『*A Framework for Inter-Domain Route Aggregation*』
RFC 2858	『*Multiprotocol Extensions for BGP-4*』
RFC 2918	『*Route Refresh Capability for BGP-4*』
RFC 3392	『*Capabilities Advertisement with BGP-4*』
RFC 4271	『*A Border Gateway Protocol 4 (BGP-4)*』
RFC 4893	『*BGP Support for Four-Octet AS Number Space*』
RFC 5396	『*Textual Representation of Autonomous system (AS) Numbers*』
RFC 5398	『*Autonomous System (AS) Number Reservation for Documentation Use*』

シスコのテクニカル サポート

説明	リンク
右の URL にアクセスして、シスコのテクニカル サポートを最大限に活用してください。 以下を含むさまざまな作業にこの Web サイトが役立ちます。 ・テクニカル サポートを受ける ・ソフトウェアをダウンロードする ・セキュリティの脆弱性を報告する、またはシスコ製品のセキュリティ問題に対する支援を受ける ・ツールおよびリソースへアクセスする - Product Alert の受信登録 - Field Notice の受信登録 - Bug Toolkit を使用した既知の問題の検索 ・Networking Professionals（NetPro）コミュニティで、技術関連のディスカッションに参加する ・トレーニング リソースへアクセスする ・TAC Case Collection ツールを使用して、ハードウェアや設定、パフォーマンスに関する一般的な問題をインタラクティブに特定および解決する この Web サイト上のツールにアクセスする際は、Cisco.com のログイン ID およびパスワードが必要です。	http://www.cisco.com/cisco/web/support/index.html

外部 BGP を使用したサービス プロバイダーとの接続の機能情報

表 5 に、このモジュールで説明した機能をリストし、特定の設定情報へのリンクを示します。

このテクノロジーに含まれる、ここで記述されていない機能の情報については、『Cisco BGP Implementation Roadmap』を参照してください。

プラットフォームのサポートおよびソフトウェア イメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator を使用すると、ソフトウェア イメージがサポートする特定のソフトウェア リリース、機能セット、またはプラットフォームを確認できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn からアクセスしてください。Cisco.com のアカウントは必要ありません。

（注） 表 5 には、一連のソフトウェア リリースのうち、特定の機能が初めて導入されたソフトウェア リリースだけが記載されています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

表 5　外部 BGP を使用したサービス プロバイダーとの接続の機能情報

機能名	リリース	機能の設定情報
BGP がサポートする番号付き AS-path アクセス リストの数が 500 に増加	12.0(22)S 12.2(15)T 12.2(18)S 12.2(18)SXD 12.2(27)SBC 15.0(1)S	BGP がサポートする番号付き AS-path アクセス リストの数が 500 に増加したことにより、**ip as-path access-list** コマンドを使用して設定できる自律システム アクセス リストの最大数が 199 から 500 に増加しました。 次のセクションで、この機能に関する情報を参照できます。 • 「BGP ポリシー設定」（P.9） • 「AS-path フィルタを使用した BGP プレフィクスのフィルタリング」（P.45）
BGP 名前付きコミュニティ リスト	12.2(8)T 12.2(14)S 15.0(1)S	BGP 名前付きコミュニティ リスト機能により、名前付きコミュニティ リストと呼ばれる新しいタイプのコミュニティ リストが導入されます。BGP 名前付きコミュニティ リスト機能により、ネットワーク オペレータはコミュニティ リストに意味がわかりやすい名前を割り当てることができるようになり、設定可能なコミュニティ リストの数も増加しました。名前付きコミュニティ リストは、正規表現や番号付きコミュニティ リストによって設定可能です。番号付きコミュニティのルールは、名前付きコミュニティリストに設定可能なコミュニティ アトリビュート数の上限がないことを除き、すべて名前付きコミュニティ リストにも適用されます。 次のセクションで、この機能に関する情報を参照できます。 • 「BGP コミュニティ」（P.10） • 「コミュニティ リストを使用したトラフィック フィルタリング」（P.52）

表 5　外部 BGP を使用したサービス プロバイダーとの接続の機能情報 （続き）

機能名	リリース	機能の設定情報
BGP プレフィクスベース アウトバウンド ルート フィルタリング	12.0(22)S 12.2(4)T 12.2(14)S 15.0(1)S	BGP プレフィクスベース アウトバウンド ルート フィルタリング機能は、BGP ORF 送受信機能を使用して、BGP ピアの間で送られる BGP アップデートの数を最小化します。この機能を設定すると、不要なルーティング アップデートをソースでフィルタリングできるため、ルーティング アップデートの生成や処理に必要なシステム リソースの量を減らす助けになります。たとえば、この機能を使用して、サービス プロバイダー ネットワークからのルート全体を受け付けるのではないルータで、ルータに要求される処理の量を減らすことができます。 次のセクションで、この機能に関する情報を参照できます。 • 「アウトバウンド BGP ルート プレフィクスのフィルタリング」（P.23） • 「アウトバウンド パス選択の変更：例」（P.70）
BGP ルート マップ継続	12.0(24)S 12.2(18)S 12.2(18)SXD 12.2(27)SBC 12.3(2)T 15.0(1)S Cisco IOS XE 3.1.0SG	BGP ルート マップ継続機能により、continue 句が BGP ルート マップ設定に導入されます。continue 句によって、ポリシー設定とルート フィルタリングのプログラム性は高まり、正常な match および set 句によってエントリが実行された後に追加のエントリを実行する機能が導入されます。continue 句によって、ネットワーク オペレータはポリシー定義をさらにモジュール化して設定できるようになり、特定のポリシー設定を同じルート マップ内で繰り返す必要がなくなりました。 次のセクションで、この機能に関する情報を参照できます。 • 「BGP ルート マップでの continue 句の使用によるトラフィック フィルタリング」（P.62） • 「BGP ルート マップでの continue 句の使用によるトラフィック フィルタリング：例」（P.77）
アウトバウンド ポリシーに対する BGP ルート マップ継続のサポート	12.0(31)S 12.2(33)SB 12.2(33)SRB 12.2(33)SXI 12.4(4)T 15.0(1)S Cisco IOS XE 3.1.0SG	アウトバウンド ポリシーに対する BGP ルート マップ継続のサポート機能により、continue 句のアウトバウンド ルート マップへの適用がサポートされます。 次のセクションで、この機能に関する情報を参照できます。 • 「BGP ルート マップでの continue 句の使用によるトラフィック フィルタリング」（P.62） • 「BGP ルート マップでの continue 句の使用によるトラフィック フィルタリング：例」（P.77）

表 5　外部 BGP を使用したサービス プロバイダーとの接続の機能情報 （続き）

機能名	リリース	機能の設定情報
BGP ルート マップ ポリシー リスト サポート	12.0(22)S 12.2(15)T 12.2(18)S 12.2(18)SXD 12.2(27)SBC 15.0(1)S	BGP ルート マップ ポリシー リスト サポート機能により、BGP ルート マップに新しい機能性が追加されます。ネットワーク オペレータはこの機能を使用して、ルート マップの match 句をポリシー リストと呼ばれる名前付きリストにグループ化できます。ポリシー リスト機能はマクロに似ています。ルート マップでポリシー リストが参照されると、match 句がすべて評価され、ルート マップで直接設定された場合と同様に処理されます。この機能強化により、中規模から大規模のネットワークでの BGP ルーティング ポリシーの BGP 設定が単純になりました。ネットワーク オペレータが match 句のグループを持つポリシー リストを事前に設定しておき、さまざまなルート マップ内でそれらのポリシー リストを参照できるからです。複数のルート マップのエントリに繰り返し現れる一群の match 句を、ネットワーク オペレータがそれぞれ手動で再設定する必要がなくなりました。 次のセクションで、この機能に関する情報を参照できます。 • 「BGP ルート マップ ポリシー リスト」（P.12） • 「BGP ルート マップ ポリシー リストを使用したトラフィック フィルタリング」（P.59）

表 5　外部 BGP を使用したサービス プロバイダーとの接続の機能情報 （続き）

機能名	リリース	機能の設定情報
4 バイト ASN に対する BGP サポート	12.0(32)S12 12.0(32)SY8 12.0(33)S3 12.2(33)SRE 12.2(33)XNE 12.2(33)SXI1 12.4(24)T 15.0(1)S Cisco IOS XE 3.1.0SG	4 バイト ASN に対する BGP サポート機能により、4 バイト自律システム番号がサポートされるようになりました。自律システム番号の要求の増加に伴い、IANA は 2009 年 1 月から 65536 ～ 4294967295 の範囲の 4 バイト自律システム番号の割り当てを開始します。 Cisco IOS Release 12.0(32)SY8、12.0(33)S3、12.2(33)SRE、12.2(33)XNE、および 12.2(33)SXI1 では、シスコは 4 バイト自律システム番号の実装時に、asplain 形式を正規表現マッチングのデフォルト、また自律システム番号の出力表示形式として使用しています。しかし、RFC 5396 が記述する asplain と asdot 形式のどちらでも、4 バイト自律システム番号を設定できます。4 バイト自律システム番号の正規表現マッチングと出力表示のデフォルトを asdot 形式に変更するには、**bgp asnotation dot** コマンドを使用します。 Cisco IOS Release 12.0(32)S12 および 12.4(24)T では、4 バイト自律システム番号の設定形式、正規表現マッチング、出力表示の実装として、シスコは asdot だけを使用しており、asplain はサポートされていません。 次のセクションで、この機能に関する情報を参照できます。 • 「BGP 自律システム番号の形式」（P.4） • 「4 バイト自律システム番号を使用した AS-path フィルタによる BGP プレフィクスのフィルタリング」（P.48） • 「4 バイト自律システム番号を使用した AS-path アトリビュートの変更によるインバウンド パス選択の変更：例」（P.68） • 「4 バイト自律システム番号を使用した AS-path フィルタによるトラフィック フィルタリング：例」（P.74） • 「4 バイト自律システム番号と拡張コミュニティ リストを使用したトラフィック フィルタリング：例」（P.74） この機能により、次の各コマンドが追加または変更されています。**bgp asnotation dot**、**bgp confederation identifier**、**bgp confederation peers**、自律システム番号を設定するすべての **clear ip bgp** コマンド、**ip as-path access-list**、**ip extcommunity-list**、**match source-protocol**、**neighbor local-as**、**neighbor remote-as**、**neighbor soo**、**redistribute** (IP)、**router bgp**、**route-target**、**set as-path**、**set extcommunity**、**set origin**、**soo**、自律システム番号を表示するすべての **show ip bgp** コマンド、および **show ip extcommunity-list**。

表 5　外部 BGP を使用したサービス プロバイダーとの接続の機能情報 （続き）

機能名	リリース	機能の設定情報
名前付き拡張コミュニティ リストに対する BGP サポート	12.2(25)S 12.2(27)SBC 12.2(33)SRA 12.2(33)SXH 12.3(11)T 15.0(1)S	名前付き拡張コミュニティ リストに対する BGP サポート機能により、既存の数字形式に加え、名前を使用しても拡張コミュニティ リストを設定できるようになりました。 次のセクションで、この機能に関する情報を参照できます。 • 「BGP コミュニティ」（P.10） • 「拡張コミュニティ リストを使用したトラフィック フィルタリング」（P.55）
拡張コミュニティ リスト内のシーケンスされたエントリに対する BGP サポート	12.2(25)S 12.2(27)SBC 12.2(33)SRA 12.2(33)SXH 12.3(11)T 15.0(1)S	拡張コミュニティ リスト内のシーケンスされたエントリに対する BGP サポート機能により、BGP 拡張コミュニティ リスト内の個別のエントリに自動シーケンスが導入されます。この機能により、既存の拡張コミュニティ リスト全体を削除することなく、拡張コミュニティ リスト エントリの削除やシーケンス再割り当てを行うことも可能になりました。 次のセクションで、この機能に関する情報を参照できます。 • 「BGP コミュニティ」（P.10） • 「拡張コミュニティ リストを使用したトラフィック フィルタリング」（P.55）
BGP 4 プレフィクス フィルタおよびインバウンド ルート マップ	Cisco IOS XE 3.1.0SG	この機能については、次の項に説明があります。 • 「BGP ポリシー設定」（P.9） • 「インバウンド パス選択の変更：例」（P.67）